



Informatica® Multidomain MDM
10.4

Guide des services d'entité d'entreprise

Ce logiciel et la documentation associée sont fournis uniquement sous un accord de licence séparé contenant des restrictions d'utilisation et de divulgation. Il est interdit de reproduire ou de transmettre sous quelle que forme et par quel que moyen que ce soit (électronique, photocopie, enregistrement ou autre) tout ou partie de ce document sans le consentement préalable d'Informatica LLC.

U.S. GOVERNMENT RIGHTS Les programmes, les logiciels, les bases de données et les documents connexes et les données techniques fournis aux clients du gouvernement américain sont des « logiciels commerciaux » ou des « données techniques commerciales », conformément au règlement fédéral sur les acquisitions et aux règlements supplémentaires propres à l'Agence. En tant que tel, l'utilisation, la duplication, la divulgation, la modification et l'adaptation sont assujetties aux restrictions et aux conditions de licence énoncées dans le contrat gouvernemental applicable et, dans la mesure applicable par les termes du contrat gouvernemental, les droits additionnels énoncés dans la réglementation FAR 52.227-19, licence de logiciel d'ordinateur commercial.

Informatica, le logo Informatica et ActiveVOS sont des marques ou des marques déposées d'Informatica LLC aux États-Unis et dans de nombreux autres pays. Une liste actuelle des marques déposées d'Informatica est disponible sur le site <https://www.informatica.com/trademarks.html>. Les autres noms de société ou de produit peuvent être des marques de commerce ou des marques déposées de leurs détenteurs respectifs.

Certaines parties de ce logiciel et/ou de cette documentation sont soumises à des droits d'auteur détenus par des tiers. Les notifications de tiers requises sont incluses avec le produit.

Les renseignements contenus dans cette documentation sont sujets à modification sans préavis. Si vous constatez des problèmes liés à la documentation, merci de les signaler par courriel à l'adresse infa_documentation@Informatica.com.

Les produits Informatica sont garantis conformément aux termes et conditions des accords en vertu desquels ils sont fournis. **INFORMATICA FOURNIT LES INFORMATIONS DE CE DOCUMENT « EN L'ÉTAT » SANS GARANTIE D'AUCUNE SORTE, EXPRESSE OU IMPLICITE, NOTAMMENT AUCUNE GARANTIE DE QUALITÉ MARCHANDE, D'ADAPTATION À UN USAGE PARTICULIER ET D'ABSENCE DE CONTREFAÇON**

Sommaire

Préface.....	9
Ressources Informatica.	9
Informatica Network.	9
Base de connaissances Informatica.	9
Documentation Informatica.	10
Matrices de disponibilité des produits Informatica.	10
Informatica Velocity.	10
Informatica Marketplace.	10
Support client international Informatica.	10
 Chapitre 1: Introduction aux services d'entité commerciale.....	 11
Présentation des services d'entité commerciale.	11
Services d'entité commerciale.	12
Service d'entité d'entreprise ReadBE.	12
Service d'entité d'entreprise WriteBE.	12
Service d'entité d'entreprise SearchBE.	13
Points d'extrémité des services d'entité commerciale.	13
Point d'extrémité Enterprise JavaBeans pour les services d'entité commerciale.	13
Point d'extrémité REST pour les services d'entité commerciale.	14
Appels de services d'entité commerciale REST et EJB.	14
Point d'extrémité SOAP pour les services d'entité commerciale.	14
Identification d'un enregistrement racine.	15
Filtres de sécurité et de données.	15
Authentification par certificat.	16
 Chapitre 2: Appels de services d'entité commerciale Enterprise Java Bean... 	 17
Présentation des appels de services d'entité commerciale Enterprise Java Bean.	17
Exemple de code Java contenant des classes SDO standard.	17
Exemple de code Java avec des classes SDO générées.	21
 Chapitre 3: Appels de services d'entité commerciale Representational State Transfer.....	 26
Présentation des API REST de services d'entité d'entreprise.	26
Méthodes REST prises en charge.	27
Méthode d'authentification.	27
Cookies d'authentification pour la connexion à partir d'applications tierces.	28
Fichier Web Application Description Language.	28
Uniform Resource Locator REST.	29
Configuration de l'en-tête et du corps.	30
En-tête de demande.	31

Corps de demande.	31
Paramètres de requête standard.	33
Date et heure au format UTC.	33
Configuration de WebLogic pour exécuter des appels REST de services d'entité commerciale.	34
Affichage des paramètres d'entrée et de sortie.	35
Modèle JavaScript.	35
Exemple de code JavaScript.	36
Référence d'API REST pour les services d'entité d'entreprise	38
Obtenir les métadonnées	38
Répertoire des métadonnées.	42
Répertoire les colonnes de correspondance.	46
Lire l'enregistrement.	48
Créer un enregistrement.	54
Mettre à jour l'enregistrement.	56
Supprimer un enregistrement.	60
Lister l'enregistrement.	61
Rechercher un enregistrement.	63
Générateur de suggestions.	68
SearchQuery.	69
SearchMatch.	73
Obtenir les métadonnées BPM.	78
Lister les tâches.	79
Lire la tâche.	85
Créer une tâche.	87
Mettre à jour la tâche.	90
Tâche terminée.	93
Exécuter l'action de tâche.	95
Lister les utilisateurs pouvant être attribués.	97
Réclamer des tâches en bloc.	98
Libérer des tâches en bloc.	100
Attribuer des tâches en bloc.	101
Modifier des tâches en bloc.	103
Action de tâches en bloc.	105
Obtenir les actions de tâches.	106
Répertoire les propriétaires potentiels des tâches.	108
Répertoire les propriétaires potentiels d'une tâche.	110
Lister les métadonnées des fichiers.	111
Créer les métadonnées du fichier.	112
Obtenir les métadonnées du fichier.	113
Mettre à jour les métadonnées du fichier.	114
Charger le contenu du fichier.	115
Obtenir le contenu du fichier.	116

Supprimer un fichier.	116
Prévisualiser les enregistrements remplacés.	117
Mettre à jour l'option Rechercher et remplacer.	121
Importer le nouveau fichier.	123
Importer le fichier correspondant	124
Obtenir les propriétés du fichier.	125
Enregistrer les propriétés du fichier.	127
Renvoyer les propriétés des fichiers.	128
Prévisualiser les fichiers analysés.	129
Obtenir les fichiers analysés.	131
Erreurs d'analyse de l'importation du fichier.	132
Erreurs de chargement de l'importation du fichier.	132
Prévisualiser la promotion.	133
Promouvoir.	135
Supprimer les modifications en attente.	136
Prévisualiser la fusion.	136
Mettre à jour la fusion en attente.	139
Fusion en attente.	142
PromoteMerge.	143
Transformation du fichier.	143
Suggérer le mappage.	145
Fusionner les enregistrements.	147
Annuler la fusion des enregistrements.	149
Créer un mappage.	150
Prévisualiser le mappage.	152
Rechercher le mappage.	154
Lire le mappage.	156
Mettre à jour le mappage.	157
Transformation du fichier.	159
Lire la relation.	161
Créer une relation.	163
Mettre à jour une relation.	165
Supprimer une relation.	166
Obtenir les enregistrements associés.	167
Exporter les entités d'entreprise liées.	171
Liste Hiérarchies.	171
Obtenir les métadonnées de la hiérarchie.	172
Obtenir le chemin de la hiérarchie.	177
Obtenir les parents.	180
Obtenir les enfants.	182
Exporter la hiérarchie.	189
Exporter les enfants et les parents directs.	190

Obtenir les modifications de la hiérarchie.	192
Modifications des relations en bloc.	198
Promouvoir en bloc.	200
Rejeter en bloc.	206
Démarrer la correspondance.	210
Lire les enregistrements correspondants.	211
Mettre à jour les enregistrements correspondants.	212
Supprimer les enregistrements correspondants.	213
Obtenir les données de correspondance dans CSV.	214
Obtenir les données de correspondance dans JSON.	215
Importer le fichier correspondant	216
Obtenir les métadonnées du système source	217
Obtenir les événements d'historique de l'enregistrement.	220
Obtenir les détails de l'événement.	223
Liste Rapports.	225
Obtenir la configuration et les données du rapport.	226
Obtenir la configuration du rapport et les rapports d'exploration.	228
Enregistrer le rapport.	230
Mettre à jour la configuration du rapport.	231
Ajouter ou mettre à jour les données du rapport.	232
Supprimer le rapport.	234
Exécuter le travail de mise à jour du rapport.	234
Obtenir le statut du travail de mise à jour du rapport.	235
Répertorier les groupes de travaux	236
Répertorier un groupe de travaux.	238
Obtenir les métadonnées DaaS.	239
Recherche DaaS	240
Lecture DaaS.	245
WriteMerge.	247
Importation DaaS.	248
Mise à jour DaaS.	251

Chapitre 4: Appels de services d'entité commerciale du protocole SOAP (Simple Object Access Protocol)..... 254

Appels du protocole SOAP (Simple Object Access Protocol) pour les services d'entité commerciale.	254
Méthode d'authentification.	255
Cookies d'authentification pour la connexion à partir d'applications tierces.	255
Fichier Web Services Description Language.	256
URL SOAP.	257
Demandes et réponses SOAP.	258
Affichage des paramètres d'entrée et de sortie.	259
Référence d'API SOAP.	260

Exemple de demande et de réponse SOAP.	261
--	-----

Chapitre 5: Services d'enregistrements de références croisées et de calculs de MVV..... 263

Aperçu des calculs de services d'enregistrements de références croisées et de la MVV.	263
Obtenir les calculs de données de références croisées et d'examen de la MVV.	263
Obtenir les enregistrements de références croisées.	264
Déterminer les contributeurs à l'enregistrement principal.	264
Obtenir les scores d'approbation des champs Enregistrements des références croisées contributifs.	265
Obtention des scores d'approbation de tous les champs Enregistrements de références croisées.	266
Obtenir les informations sur les systèmes source.	267
Exemple d'obtention d'informations sur les systèmes source.	267
Réponses de filtrage et de pagination.	268
Exemples de demandes de filtration.	268
Établir la meilleure version de la vérité.	268
Sélectionner le champ de contribution correct.	268
Sélectionner l'exemple de champ de contribution correct.	269
Écrire la valeur correcte dans l'enregistrement principal.	269
Exemple d'écriture de valeur correcte dans l'enregistrement principal.	271
Supprimer des données source sans correspondance.	272
Supprimer l'exemple de données source sans correspondance.	273
Réponse d'annulation de fusion	274

Chapitre 6: Prise en charge du service de liaison d'entreprise..... 275

Présentation.	275
Services d'entité d'entreprise d'importation et de mise à jour DaaS.	275
Configuration de la prise en charge de la liaison.	276
Application personnalisée pour le fractionnement des données de liaison.	276

Chapitre 7: Appels externes pour le nettoyage, l'analyse et la transformation de données..... 277

Présentation.	277
Événements pris en charge.	278
Configuration des appels externes.	278
Exemple : validation et logique personnalisées des services d'entité d'entreprise.	279
Prérequis.	279
Étape 1. Tester la validation personnalisée.	279
Étape 2. Tester la logique personnalisée.	280

Annexe A: Utilisation d'API REST pour ajouter des enregistrements..... 285

Présentation de l'utilisation d'API REST pour ajouter des enregistrements.	285
Structure de l'entité commerciale Person.	286

Étape 1. Obtenir des informations sur le schéma.	286
Obtenir la réponse des métadonnées.	287
Étape 2. Création d'un enregistrement.	292
Créer une réponse de l'enregistrement.	293
Étape 3. Lire l'enregistrement.	294
Lire la réponse de l'enregistrement.	295
 Annexe B: Utilisation d'API REST pour le chargement de fichiers.....	299
Présentation de l'utilisation d'API REST pour le chargement de fichiers.	299
API REST pour les fichiers.	300
Composants de fichier.	300
Types de stockage.	301
Joindre des fichiers à des enregistrements.	301
Joindre des fichiers à des tâches.	303
Chargement de fichiers de groupement de ressources.	306
 Annexe C: Utilisation des API REST pour gérer les rapports.....	307
Présentation de l'utilisation des API REST pour gérer les rapports.	307
API REST pour les rapports.	308
Configuration du rapport.	308
Données du rapport.	309
Rapports d'exploration.	310
Rapports prêts à l'emploi.	311
Gestion des rapports prêts à l'emploi.	312
Rapports personnalisés.	315
Gestion des rapports personnalisés.	315
Gestion des rapports personnalisés avec des rapports d'exploration.	317
Dépannage des API de rapports.	320
 Index.	321

Préface

Consultez le *Guide des services d'entité d'entreprise pour MDM Multidomain* d'Informatica® pour en savoir plus sur les services d'entité d'entreprise disponibles en tant que services Web. Utilisez ce guide pour apprendre à utiliser Enterprise Java Beans (EJB), Representational State Transfer (REST), Simple Object Access Protocol (SOAP), ainsi que les appels de service Web externes permettant d'effectuer des opérations sur les données d'entité d'entreprise. Vous pouvez également apprendre à configurer des interfaces utilisateur personnalisées pour effectuer des appels de service d'entité d'entreprise.

Ressources Informatica

Informatica vous fournit toute une gamme de ressources de produits via Informatica Network et autres portails en ligne. Utilisez ces ressources pour tirer le meilleur parti de vos produits et solutions Informatica, et pour apprendre d'autres utilisateurs et experts en la matière d'Informatica.

Informatica Network

Informatica Network est la passerelle à de nombreuses ressources, y compris la base de connaissances Informatica et le support client international Informatica. Pour accéder à Informatica Network, visitez le site <https://network.informatica.com>.

En tant que membre d'Informatica Network, vous disposez des options suivantes :

- Rechercher les ressources de produits dans la base de connaissances.
- Afficher les informations de disponibilité des produits.
- Créer et vérifier vos dossiers de support.
- Rechercher votre réseau de groupe d'utilisateurs local Informatica et collaborer avec vos pairs.

Base de connaissances Informatica

Utilisez la base de connaissances Informatica pour rechercher des ressources de produits telles que des articles pratiques, des meilleures pratiques, des didacticiels vidéo et des questions fréquemment posées.

Pour rechercher dans la base de connaissances, visitez le site <https://search.informatica.com>. N'hésitez pas à contacter l'équipe Base de connaissances Informatica à l'adresse KB_Feedback@informatica.com pour lui faire part de vos questions, commentaires ou suggestions concernant la base de connaissances.

Documentation Informatica

Utilisez le portail de documentation Informatica pour explorer une vaste bibliothèque de documentation pour les versions de produits actuelles et récentes. Pour explorer le portail de documentation, visitez le site <https://docs.informatica.com>.

N'hésitez pas à contacter l'équipe Documentation Informatica à l'adresse infa_documentation@informatica.com pour lui faire part de vos questions, commentaires ou suggestions concernant la documentation des produits.

Matrices de disponibilité des produits Informatica

Les matrices de disponibilité des produits (PAM) indiquent les versions des systèmes d'exploitation, les bases de données et les types de source et cible de données pris en charge par une version d'un produit. Vous pouvez parcourir les PAM Informatica à l'adresse <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity est un ensemble de conseils et de meilleures pratiques développés par les services professionnels d'Informatica et basés sur les expériences réelles de centaines de projets de gestion des données. Informatica Velocity représente le savoir collectif de consultants d'Informatica qui collaborent avec des organisations du monde entier pour planifier, développer, déployer et gérer des solutions performantes de gestion des données.

Vous trouverez les ressources d'Informatica Velocity à l'adresse <http://velocity.informatica.com>. Si vous avez des questions, des commentaires ou des suggestions sur Informatica Velocity, contactez les services professionnels d'Informatica à l'adresse ips@informatica.com.

Informatica Marketplace

Informatica Marketplace est un forum dans lequel vous pouvez trouver des solutions qui permettent d'augmenter et d'améliorer vos implémentations Informatica. Exploitez les centaines de solutions de développeurs et de partenaires Informatica sur Marketplace pour améliorer votre productivité et accélérer le délai d'implémentation de vos projets. Vous trouverez Informatica Marketplace à l'adresse <https://marketplace.informatica.com>.

Support client international Informatica

Vous pouvez contacter un centre de support international par téléphone ou via Informatica Network.

Pour rechercher le numéro de téléphone du support client international Informatica local, visitez le site Web Informatica à l'adresse <https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

Pour rechercher des ressources de support en ligne sur Informatica Network, visitez le site <https://network.informatica.com> et sélectionnez l'option eSupport.

CHAPITRE 1

Introduction aux services d'entité commerciale

Ce chapitre comprend les rubriques suivantes :

- [Présentation des services d'entité commerciale, 11](#)
- [Services d'entité commerciale, 12](#)
- [Points d'extrémité des services d'entité commerciale, 13](#)
- [Identification d'un enregistrement racine, 15](#)
- [Filtres de sécurité et de données, 15](#)
- [Authentification par certificat, 16](#)

Présentation des services d'entité commerciale

Un service d'entité commerciale est un ensemble d'opérations qui exécutent un code MDM Hub pour créer, mettre à jour, supprimer et rechercher des enregistrements d'objets de base dans une entité commerciale. Vous pouvez développer une interface utilisateur personnalisée capable d'exécuter un code Java ou JavaScript pour effectuer des appels de services d'entité commerciale.

Par exemple, vous pouvez créer un service d'entité commerciale pour étoffer un enregistrement fournisseur avec des données Dun and Bradstreet. Configurez le service d'entité commerciale de sorte à ce qu'il utilise l'enregistrement fournisseur comme entrée, récupère des informations de Dun and Bradstreet, mette à jour l'enregistrement et génère l'enregistrement fournisseur mis à jour.

Les objets de base d'une entité commerciale comportent les services d'entité commerciale suivants :

Lire

Chaque entité commerciale est dotée d'un service d'entité commerciale qui effectue une opération de lecture.

Écrire

Chaque entité commerciale est dotée d'un service d'entité commerciale qui effectue une opération d'écriture.

Rechercher

Toute entité commerciale comportant des champs de recherche est dotée d'un service d'entité commerciale qui effectue une opération de recherche.

Par exemple, une entité commerciale Person contient des champs de recherche. MDM Hub génère un service d'entité commerciale ReadPerson, WritePerson et SearchPerson. Les étapes de service d'entité commerciale

de lecture, d'écriture et de recherche vous permettent de lire, créer, mettre à jour, supprimer et rechercher des enregistrements dans une entité commerciale.

Services d'entité commerciale

Un service d'entité commerciale effectue une opération. Vous pouvez utiliser les services d'entité commerciale ReadBE, WriteBE et SearchBE.

Un service d'entité commerciale comprend des étapes de service. Les demandes entrantes suivent chaque étape de service. La sortie d'une étape constitue une entrée pour l'étape suivante. La sortie d'une étape peut transmettre des informations à l'entrée de l'étape suivante. Toutes les étapes de service d'entité commerciale sont exécutées comme un seul appel Enterprise Java Bean dans une transaction unique. MDM Hub gère les exceptions.

Remarque: Avant d'utiliser les services d'entité commerciale, validez le stockage de référence opérationnelle.

Service d'entité d'entreprise ReadBE

Le service d'entité d'entreprise ReadBE lit les données d'un enregistrement d'objet de base dans une entité d'entreprise.

Vous pouvez spécifier les paramètres de pagination avec l'étape ReadBE pour définir le nombre d'enregistrements à renvoyer et la page de résultats à afficher.

Les résultats du service ReadBE n'incluent pas les enregistrements supprimés de manière réversible.

Si vous ne transmettez pas le paramètre EffectiveDate dans la demande de service d'entité d'entreprise, MDM Hub suppose que la date effective est une valeur Null et le service d'entité d'entreprise lit les données de l'objet de base. Si vous transmettez le paramètre EffectiveDate, MDM Hub calcule la meilleure version de la vérité parmi les enregistrements de références croisées et le service d'entité d'entreprise de lecture renvoie la meilleure version de la vérité à jour.

Service d'entité d'entreprise WriteBE

Le service d'entité d'entreprise WriteBE peut mettre à jour les données d'un élément d'entité d'entreprise, créer un élément d'entité d'entreprise enfant ou supprimer des éléments d'entité d'entreprise enfants.

Remarque: Le service d'entité d'entreprise WriteBE utilise les paramètres d'approbation existants pour calculer l'approbation sur des objets de base. Il ne permet pas d'effectuer de remplacement d'approbation.

Paramètres facultatifs

Le tableau suivant décrit les paramètres facultatifs que vous pouvez utiliser avec le service d'entité d'entreprise WriteBE :

Paramètre	Description
recordState	Définissez l'état de l'enregistrement sur ACTIVE, PENDING ou DELETED. Remarque: Lorsque vous définissez <code>recordState=ACTIVE</code> et que vous exécutez le service sur un enregistrement supprimé de manière réversible, le service restaure l'enregistrement à l'état actif.
EffectivePeriod	Indiquez une période effective. Si vous ne transmettez pas le paramètre <code>EffectivePeriod</code> , MDM Hub suppose que la période est illimitée. MDM Hub ne vérifie pas l'alignement des périodes effectives entre les objets racines et les objets enfants. Lorsque vous créez ou mettez à jour des enregistrements, vérifiez que les périodes effectives de l'enregistrement parent et des enregistrements enfants sont alignées.

Service d'entité d'entreprise SearchBE

Utilisez le service d'entité d'entreprise SearchBE pour rechercher un enregistrement racine dans une entité d'entreprise.

Pour plus d'informations sur la configuration des entités d'entreprise pour la recherche, consultez le *Guide de configuration de MDM Multidomain*.

Points d'extrémité des services d'entité commerciale

Vous pouvez accéder aux services d'entité commerciale via les points d'extrémité suivants : Enterprise JavaBeans (EJB), Representational State Transfer (REST) ou Simple Object Access Protocol (SOAP).

Les points d'extrémité REST sont construits sur un point EJB. La configuration des services d'entité commerciale REST définit la manière dont les URL REST sont mappées aux appels de services d'entité commerciale EJB.

Point d'extrémité Enterprise JavaBeans pour les services d'entité commerciale

Le point d'extrémité Enterprise JavaBeans (EJB) est le point d'extrémité sous-jacent de tous les types d'appels de services d'entité commerciale. Tous les autres points d'extrémité sont mappés au point d'extrémité EJB.

Les services d'entité commerciale sont exposés en tant qu'EJB sans état. Un conteneur EJB sans état peut regrouper des instances, attribuer des instances et appliquer des stratégies d'équilibrage de charge pour répartir la charge entre différents serveurs dans le domaine.

Un point d'extrémité EJB accepte un nom d'utilisateur et un mot de passe pour l'authentification.

Point d'extrémité REST pour les services d'entité commerciale

Les appels de point d'extrémité REST (Representational State Transfer) rendent les services d'entité commerciale disponibles en tant que services Web.

Les fichiers Web Application Description Language (WADL) contiennent les descriptions XML des services Web REST, toutes les URL REST et tous les paramètres REST. MDM Hub génère un fichier WADL pour chaque stockage de référence opérationnelle.

Vous pouvez télécharger les fichiers WADL pour chaque stockage de référence opérationnelle dans l'emplacement suivant :

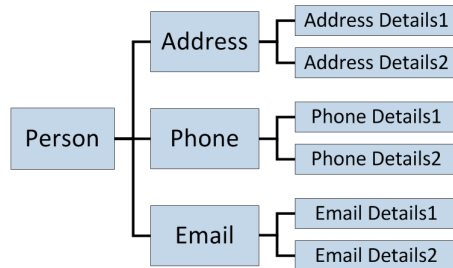
```
http://<hôte>:<port>/cmx/csfiles
```

Appels de services d'entité commerciale REST et EJB

Lorsque vous effectuez un appel de services d'entité commerciale, vous pouvez spécifier des branches enfants au lieu de demander l'entité commerciale toute entière.

Par exemple, vous voulez effectuer une opération de lecture sur une entité commerciale contenant un nœud racine Person et plusieurs branches enfants. L'objet de base Person contient les objets de base enfants Address, Phone et Email. Chaque objet de base enfant contient deux objets de base petits-enfants.

L'image suivante montre la structure d'une entité commerciale contenant plusieurs branches :



Vous pouvez lire plusieurs branches enfants à différents niveaux dans une seule demande. Par exemple, vous pouvez lire Person, Phone, Phone Details 1, Phone Details 2, Email, et Email Details 2 dans une seule demande.

L'exemple d'URL suivant montre comment effectuer une demande de lecture REST pour obtenir l'enregistrement Person avec l'ID de ligne 1 242, en plus des enregistrements enfants Address Details 1 et Email :

```
http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1242?children=Address/  
Address_Details_1,Email
```

Point d'extrémité SOAP pour les services d'entité commerciale

Les appels de point d'extrémité du protocole SOAP (Simple Object Access Protocol) rendent les services d'entité commerciale disponibles en tant que services Web.

Les fichiers Web Services Description Language (WSDL) contiennent les descriptions XML des services Web, les formats des demandes et des réponses SOAP et tous les paramètres. MDM Hub génère un fichier WSDL pour chaque stockage de référence opérationnelle.

Identification d'un enregistrement racine

Vous pouvez utiliser l'une des méthodes suivantes pour identifier un enregistrement racine :

- rowid. Valeur dans la colonne ROWID_OBJECT de l'enregistrement.
- systemName et sourceKey. Le paramètre systemName représente le nom du système auquel l'enregistrement appartient. Le paramètre sourceKey représente la valeur dans la colonne PKEY_SRC_OBJECT de l'enregistrement.
- Identificateur d'entreprise global (GBID) d'un objet. Un GBID peut être une valeur composée, auquel cas vous devez transférer toutes les valeurs.

Remarque: La méthode GBID fonctionne uniquement avec le service ReadBE.

L'exemple de code suivant utilise les paramètres systemName et sourceKey pour identifier un enregistrement :

```
String systemName = "SFA";

Properties config = new Properties();
config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL, EjbSiperianClient.PROTOCOL_NAME);
CompositeServiceClient client = CompositeServiceClient.newCompositeServiceClient(config);
CallContext callContext = new CallContext(orsId, user, pass);
helperContext = client.getHelperContext(callContext);
DataFactory dataFactory = helperContext.getDataFactory();

//String personRowId = "1097";
String pkeySrcObject = "CST1379";

//Set custom key pkey
pkey = (Key) dataFactory.create(Key.class);
pkey.setSystemName(systemName);
pkey.setSourceKey(val);
writePerson.setKey(pkey);
```

Filtres de sécurité et de données

Lorsque des objets de base et des ressources sont associés à des privilèges de rôle d'utilisateur, l'entité d'entreprise hérite ces privilèges. Pour accéder aux enregistrements de l'entité d'entreprise, le rôle d'utilisateur doit disposer des privilèges appropriés sur l'objet de base racine pour l'entité d'entreprise ainsi que sur d'autres ressources.

Les services de l'entité d'entreprise héritent également tous les filtres de données que vous avez définis sur les champs de l'entité d'entreprise.

Pour plus d'informations sur les filtres de sécurité et de données, consultez le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

Authentification par certificat

MDM Hub utilise un mécanisme d'authentification par certificat pour sécuriser l'authentification entre les composants de MDM Hub et les applications approuvées. Le mécanisme d'authentification est également pris en charge pour les API des services d'entité d'entreprise.

Les applications du client externe peuvent effectuer des demandes de services d'entité d'entreprise à MDM Hub. Vous devez enregistrer les utilisateurs de l'application du client externe en tant qu'utilisateurs de l'application approuvée dans MDM Hub. Vous devez également enregistrer le certificat public pour les utilisateurs associés à l'application du client externe.

Après l'enregistrement, les utilisateurs du client externe peuvent envoyer des demandes d'authentification chiffrées à MDM Hub à l'aide de la clé privée. MDM Hub déchiffre la demande d'authentification à l'aide de la clé publique et répond avec les données demandées en texte brut.

Les demandes d'authentification de l'utilisateur pour la charge utile des services d'entité d'entreprise sont chiffrées à l'aide de la clé privée et sont déchiffrées à l'aide de la clé publique.

Remarque: Pour enregistrer la clé publique et le certificat dans MDM Hub, contactez le support client international Informatica.

Pour plus d'informations sur la configuration des applications approuvées et de l'authentification par certificat, consultez le chapitre Authentification par certificat dans le Guide de sécurité d'Informatica Multidomain MDM 10.4.

CHAPITRE 2

Appels de services d'entité commerciale Enterprise Java Bean

Ce chapitre comprend les rubriques suivantes :

- [Présentation des appels de services d'entité commerciale Enterprise Java Bean, 17](#)
- [Exemple de code Java contenant des classes SDO standard, 17](#)
- [Exemple de code Java avec des classes SDO générées, 21](#)

Présentation des appels de services d'entité commerciale Enterprise Java Bean

Vous pouvez effectuer des appels de services d'entité commerciale Enterprise Java Bean (EJB) pour créer, mettre à jour, supprimer et rechercher des enregistrements d'objets de base dans une entité commerciale. Vous pouvez créer un code Java pour exécuter des appels de services d'entité commerciale EJB.

Vous pouvez créer un code Java basé sur des classes standard Service Data Objects (SDO) ou sur des classes Java générées par MDM Hub en fonction de la configuration de l'entité commerciale et des services d'entité commerciale.

Exemple de code Java contenant des classes SDO standard

L'exemple montre un code Java pour exécuter des appels Enterprise Java Bean (EJB), basé sur des classes standard Service Data Objects (SDO).

Cet exemple figure dans le fichier suivant du kit de ressources : `C:\<répertoire d'installation de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\DynamicSDO.java`

Le code Java suivant est basé sur les classes SDO standard et exécute des appels de services d'entité d'entreprise EJB pour créer un enregistrement d'objet de base Person, ajouter plusieurs enregistrements

enfants, supprimer un enregistrement enfant et supprimer l'enregistrement Person et tous les enregistrements enfants :

```
package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
import com.siperian.sif.client.EjbSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.Property;
import commonj.sdo.Type;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class DynamicSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {
            System.err.println("USAGE: DynamicSDO <ors> <user> <pass>");
            return;
        }

        new DynamicSDO(args[0], args[1], args[2]).execute();

    }

    private String orsId;
    private String user;
    private String pass;
    private HelperContext helperContext;
    private PrintStream out = System.out;

    public DynamicSDO(String orsId, String user, String pass) {
        this.orsId = orsId;
        this.user = user;
        this.pass = pass;
    }

    public void execute() throws CompositeServiceException {

        String systemName = "Admin";

        Properties config = new Properties();
        config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
            EjbSiperianClient.PROTOCOL_NAME);
        CompositeServiceClient client =
            CompositeServiceClient.newCompositeServiceClient(config);

        CallContext callContext = new CallContext(orsId, user, pass);

        helperContext = client.getHelperContext(callContext);

        DataFactory dataFactory = helperContext.getDataFactory();

        // types for Read requests
        Type coFilterType = helperContext.getTypeHelper().getType("urn:cs-
            base.informatica.mdm", "CoFilter");
        Type coFilterNodeType = helperContext.getTypeHelper().getType("urn:cs-
            base.informatica.mdm", "CoFilterNode");
        Type keyType = helperContext.getTypeHelper().getType("urn:cs-
            base.informatica.mdm", "Key");

        // ReadCO & WriteCO request types
        Type readPersonType = helperContext.getTypeHelper().getType("urn:cs-
```

```

ors.informatica.mdm", "ReadPerson");
    Type writePersonType = helperContext.getTypeHelper().getType("urn:cs-
ors.informatica.mdm", "WritePerson");

    // 1. Create new person
    DataObject createPerson = dataFactory.create(writePersonType);
    DataObject createPersonParameters = createPerson.createDataObject("parameters");
    createPersonParameters.setString("systemName", systemName);
    DataObject person = createPerson.createDataObject("object");

    person.getChangeSummary().beginLogging();

    DataObject personRoot = person.createDataObject("Person");
    personRoot.setString("firstName", "John");
    personRoot.setString("lastName", "Smith");

    person.getChangeSummary().endLogging();

    dump("*** CREATE NEW PERSON ...", createPerson);

    DataObject createPersonResponse = client.process(callContext, createPerson);

    dump("*** PERSON CREATED:", createPersonResponse);

    String personRowId = createPersonResponse.getString("object/Person/rowidObject");

    DataObject readPerson = dataFactory.create(readPersonType);
    DataObject readPersonParameters = readPerson.createDataObject("parameters");
    DataObject coFilter = readPersonParameters.createDataObject("coFilter");
    DataObject coFilterNode = coFilter.createDataObject("object");
    coFilterNode.set("name", "Person");
    DataObject key = coFilterNode.createDataObject("key");
    key.set("rowid", personRowId);

    dump("*** READ CREATED PERSON...", readPerson);

    DataObject readPersonResponse = client.process(callContext, readPerson);

    dump("*** READ RESULT:", readPersonResponse);

    person = readPersonResponse.getDataObject("object");
    person.detach();

    person.getChangeSummary().beginLogging();

    personRoot = person.getDataObject("Person");
    // add new 'one' child
    DataObject genderCd = personRoot.createDataObject("genderCd");
    genderCd.setString("genderCode", "M");

    // add two 'many' children
    DataObject phonePager = personRoot.createDataObject("TelephoneNumbers");
    Property item = phonePager.getInstanceProperty("item");
    Type phoneType = item.getType();

    DataObject phone1 = dataFactory.create(phoneType);
    phone1.setString("phoneNumber", "111-11-11");
    DataObject phone2 = dataFactory.create(phoneType);
    phone2.setString("phoneNumber", "222-22-22");

    phonePager.setList(item, Arrays.asList(phone1, phone2));

    person.getChangeSummary().endLogging();

    DataObject updatePerson = dataFactory.create(writePersonType);
    updatePerson.setDataObject("object", person);
    DataObject updatePersonParameters = updatePerson.createDataObject("parameters");
    updatePersonParameters.setString("systemName", systemName);
    updatePersonParameters.setString("interactionId", "");

    dump("*** UPDATE PERSON...", updatePerson);

```

```

DataObject updatePersonResponse = client.process(callContext, updatePerson);

dump("*** PERSON UPDATED:", updatePersonResponse);

coFilterNode.set("depth", 3);

readPersonParameters.setBoolean("readSystemFields", true);

dump("*** READ UPDATED PERSON WITH CHILDREN...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

genderCd = person.getDataObject("Person").createDataObject("genderCd");
genderCd.setString("genderCode", "F");

// delete one phone
DataObject phoneItem = person.getDataObject("Person/TelephoneNumbers/item[1]");
phoneItem.delete();

person.getChangeSummary().endLogging();

DataObject deletePhone = dataFactory.create(writePersonType);
deletePhone.setDataObject("object", person);
DataObject deletePhoneParameters = deletePhone.createDataObject("parameters");
deletePhoneParameters.setString("systemName", systemName);

dump("*** DELETE CHILD...", deletePhone);

DataObject deletePhoneResponse = client.process(callContext, deletePhone);

dump("*** CHILD DELETED:", deletePhoneResponse);

readPersonParameters.setBoolean("readSystemFields", false);

dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

person.getDataObject("Person").detach();

person.getChangeSummary().endLogging();

DataObject deletePerson = dataFactory.create(writePersonType);
deletePerson.setDataObject("object", person);
DataObject deletePersonParameters = deletePerson.createDataObject("parameters");
deletePersonParameters.setString("systemName", systemName);

dump("*** DELETE PERSON...", deletePerson);

DataObject deletePersonResponse = client.process(callContext, deletePerson);

dump("*** PERSON DELETED:", deletePersonResponse);

dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

try {

```

```

        readPersonResponse = client.process(callContext, readPerson);

        dump("*** READ RESULT:", readPersonResponse);
    } catch (CompositeServiceException e) {
        out.println("*** READ RESULT: " + e.getLocalizedMessage());
    }

}

private void dump(String title, DataObject dataObject) {
    String xml = helperContext.getXMLHelper().save(
        dataObject,
        dataObject.getType().getURI(),
        dataObject.getType().getName());
    out.println(title);
    out.println(xml);
    out.println();
}
}

```

Exemple de code Java avec des classes SDO générées

L'exemple montre un code Java pour exécuter des appels Enterprise Java Bean (EJB), basé sur des classes Java générées par MDM Hub en fonction de la configuration de l'entité commerciale et des services d'entité commerciale.

Cet exemple figure dans le fichier suivant du kit de ressources : C:\<répertoire d'installation de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\GeneratedSDO.java

Le code Java suivant est basé sur les classes générées et exécute des appels de services d'entité commerciale EJB pour créer un enregistrement d'objet de base Person, ajouter plusieurs enregistrements enfants, supprimer un enregistrement enfant et supprimer l'enregistrement Person et tous les enregistrements enfants :

```

package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
import com.informatica.mdm.sdo.cs.base.CoFilter;
import com.informatica.mdm.sdo.cs.base.CoFilterNode;
import com.informatica.mdm.sdo.cs.base.Key;
import com.siperian.sif.client.EjbSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;
import mdm.informatica.co_ors.*;
import mdm.informatica.cs_ors.*;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class GeneratedSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {

```

```

        System.err.println("USAGE: GeneratedSDO <ors> <user> <pass>");
        return;
    }

    new GeneratedSDO(args[0], args[1], args[2]).execute();
}

private String orsId;
private String user;
private String pass;
private HelperContext helperContext;
private PrintStream out = System.out;

public GeneratedSDO(String orsId, String user, String pass) {
    this.orsId = orsId;
    this.user = user;
    this.pass = pass;
}

public void execute() throws CompositeServiceException {

    String systemName = "Admin";

    Properties config = new Properties();
    config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
EjbsSiperianClient.PROTOCOL_NAME);
    CompositeServiceClient client =
CompositeServiceClient.newCompositeServiceClient(config);

    CallContext callContext = new CallContext(orsId, user, pass);

    helperContext = client.getHelperContext(callContext);

    DataFactory dataFactory = helperContext.getDataFactory();

    // 1. Create new person
    WritePerson createPerson = (WritePerson)dataFactory.create(WritePerson.class);
    WritePersonParameters createPersonParameters =
(WritePersonParameters)dataFactory.create(WritePersonParameters.class);
    createPersonParameters.setSystemName(systemName);
    createPerson.setParameters(createPersonParameters);

    Person person = (Person)dataFactory.create(Person.class);
    createPerson.setObject(person);

    person.getChangeSummary().beginLogging();

    PersonRoot personRoot = (PersonRoot)dataFactory.create(PersonRoot.class);
    personRoot.setFirstName("John");
    personRoot.setLastName("Smith");
    person.setPerson(personRoot);

    person.getChangeSummary().endLogging();

    dump("*** CREATE NEW PERSON ...", createPerson);

    WritePersonReturn createPersonResponse =
(WritePersonReturn)client.process(callContext, (DataObject)createPerson);

    dump("*** PERSON CREATED:", createPersonResponse);

    String personRowId =
createPersonResponse.getObject().getPerson().getRowidObject();

    Key key = (Key)dataFactory.create(Key.class);
    key.setRowid(personRowId);
    CoFilterNode coFilterNode = (CoFilterNode)dataFactory.create(CoFilterNode.class);
    coFilterNode.setName(Person.class.getSimpleName());
    coFilterNode.setKey(key);
    CoFilter coFilter = (CoFilter)dataFactory.create(CoFilter.class);

```

```

        coFilter.setObject(coFilterNode);
        ReadPersonParameters readPersonParameters =
(ReadPersonParameters) dataFactory.create(ReadPersonParameters.class);
        readPersonParameters.setCoFilter(coFilter);

        ReadPerson readPerson = (ReadPerson) dataFactory.create(ReadPerson.class);
        readPerson.setParameters(readPersonParameters);

        dump("*** READ CREATED PERSON...", readPerson);

        ReadPersonReturn readPersonResponse =
(ReadPersonReturn) client.process(callContext, (DataObject) readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject) person).detach();

        person.getChangeSummary().beginLogging();

        personRoot = person.getPerson();
        // add new 'one' child
        LUGenderLookup genderCd =
(LUGenderLookup) dataFactory.create(LUGenderLookup.class);
        genderCd.setGenderCode("M");
        personRoot.setGenderCd(genderCd);

        // add two 'many' children
        PersonTelephoneNumbersPager phonePager =
(PersonTelephoneNumbersPager) dataFactory.create(PersonTelephoneNumbersPager.class);

        PersonTelephoneNumbers phone1 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
        phone1.setPhoneNumber("111-11-11");
        PersonTelephoneNumbers phone2 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
        phone2.setPhoneNumber("222-22-22");

        phonePager.setItem(Arrays.asList(phone1, phone2));
        personRoot.setTelephoneNumbers(phonePager);

        person.getChangeSummary().endLogging();

        WritePerson updatePerson = (WritePerson) dataFactory.create(WritePerson.class);
        updatePerson.setObject(person);
        WritePersonParameters updatePersonParameters =
(WritePersonParameters) dataFactory.create(WritePersonParameters.class);
        updatePersonParameters.setSystemName(systemName);
        updatePersonParameters.setInteractionId("");
        updatePerson.setParameters(updatePersonParameters);

        dump("*** UPDATE PERSON...", updatePerson);

        WritePersonReturn updatePersonResponse =
(WritePersonReturn) client.process(callContext, (DataObject) updatePerson);

        dump("*** PERSON UPDATED:", updatePersonResponse);

        coFilterNode.setDepth(3);

        readPersonParameters.setReadSystemFields(true);

        dump("*** READ UPDATED PERSON WITH CHILDREN (with system fields)...",
readPerson);

        readPersonResponse = (ReadPersonReturn) client.process(callContext,
(DataObject) readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();

```

```

        ((DataObject)person).detach();

        person.getChangeSummary().beginLogging();

        // delete one phone
        person.getPerson().getTelephoneNumbers().getItem().remove(0);

        // change gender
        genderCd = (LUGenderLookup)dataFactory.create(LUGenderLookup.class);
        genderCd.setGenderCode("F");
        personRoot.setGenderCd(genderCd);

        person.getChangeSummary().endLogging();

        WritePerson deletePhone = (WritePerson)dataFactory.create(WritePerson.class);
        deletePhone.setObject(person);
        WritePersonParameters deletePhoneParameters =
        (WritePersonParameters)dataFactory.create(WritePersonParameters.class);
        deletePhoneParameters.setSystemName(systemName);
        deletePhone.setParameters(deletePhoneParameters);

        dump("*** DELETE CHILD...", deletePhone);

        WritePersonReturn deletePhoneResponse =
        (WritePersonReturn)client.process(callContext, (DataObject)deletePhone);

        dump("*** CHILD DELETED:", deletePhoneResponse);

        readPersonParameters.setReadSystemFields(false);

        dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

        readPersonResponse = (ReadPersonReturn)client.process(callContext,
        (DataObject)readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject)person).detach();

        person.getChangeSummary().beginLogging();

        ((DataObject)person.getPerson()).delete();

        person.getChangeSummary().endLogging();

        WritePerson deletePerson = (WritePerson)dataFactory.create(WritePerson.class);
        deletePerson.setObject(person);
        WritePersonParameters deletePersonParameters =
        (WritePersonParameters)dataFactory.create(WritePersonParameters.class);
        deletePersonParameters.setSystemName(systemName);
        deletePerson.setParameters(deletePersonParameters);

        dump("*** DELETE PERSON...", deletePerson);

        WritePersonReturn deletePersonResponse =
        (WritePersonReturn)client.process(callContext, (DataObject)deletePerson);

        dump("*** PERSON DELETED:", deletePersonResponse);

        dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

        try {
            readPersonResponse = (ReadPersonReturn)client.process(callContext,
            (DataObject)readPerson);

            dump("*** READ RESULT:", readPersonResponse);
        } catch (CompositeServiceException e) {
            out.println("*** READ RESULT: " + e.getLocalizedMessage());
        }
    }
}

```



```

    }

    private void dump(String title, Object object) {
        DataObject dataObject = (DataObject)object;
        String xml = helperContext.getXMLHelper().save(
            dataObject,
            dataObject.getType().getURI(),
            dataObject.getType().getName());
        out.println(title);
        out.println(xml);
        out.println();
    }
}

```

CHAPITRE 3

Appels de services d'entité commerciale Representational State Transfer

Ce chapitre comprend les rubriques suivantes :

- [Présentation des API REST de services d'entité d'entreprise, 26](#)
- [Méthodes REST prises en charge, 27](#)
- [Méthode d'authentification, 27](#)
- [Cookies d'authentification pour la connexion à partir d'applications tierces, 28](#)
- [Fichier Web Application Description Language, 28](#)
- [Uniform Resource Locator REST, 29](#)
- [Configuration de l'en-tête et du corps, 30](#)
- [Paramètres de requête standard, 33](#)
- [Date et heure au format UTC, 33](#)
- [Configuration de WebLogic pour exécuter des appels REST de services d'entité commerciale, 34](#)
- [Affichage des paramètres d'entrée et de sortie, 35](#)
- [Modèle JavaScript, 35](#)
- [Exemple de code JavaScript, 36](#)
- [Référence d'API REST pour les services d'entité d'entreprise , 38](#)

Présentation des API REST de services d'entité d'entreprise

Les appels de point d'extrémité REST rendent tous les services d'entité commerciale disponibles en tant que services Web.

Vous pouvez effectuer des appels REST pour créer, mettre à jour, supprimer et rechercher des enregistrements d'objets de base et les enregistrements enfants associés dans une entité d'entreprise. Vous pouvez effectuer différentes opérations, par exemple fusionner, annuler la fusion et faire correspondre des enregistrements. Vous pouvez aussi effectuer des appels REST pour créer, mettre à jour et rechercher des

tâches et pour exécuter des tâches. Vous pouvez aussi effectuer des appels REST pour créer, mettre à jour et supprimer des fichiers, tels que des pièces jointes pour des tâches ou des enregistrements.

Un appel de services d'entité d'entreprise REST est une demande de service Web sous forme d'URL (Uniform Resource Locator). MDM Hub attribue une URL unique à chaque objet de base d'une entité d'entreprise. Vous pouvez utiliser l'URL unique pour identifier l'objet de base à mettre à jour ou supprimer.

Remarque: Avant d'utiliser les API REST pour appeler les services d'entité d'entreprise, validez le stockage de référence opérationnelle.

Méthodes REST prises en charge

Les API REST des services d'entité d'entreprise utilisent les méthodes HTTP standard pour effectuer des opérations sur les ressources telles que les enregistrements, les tâches et les fichiers.

Les API REST de services d'entité d'entreprise prennent en charge les méthodes de demande HTTP suivantes :

Méthode	Description
GET	Récupère les informations sur un enregistrement, une tâche ou un fichier.
POST	Crée une tâche, un enregistrement racine, un enregistrement enfant ou un fichier. Remarque: Le nom du stockage de référence opérationnelle dans une demande POST est sensible à la casse. Si la casse du nom du stockage de référence opérationnelle ne correspond pas au nom dans le MDM Hub, une erreur se produit.
PUT	Met à jour un enregistrement racine, un enregistrement enfant, une tâche ou un fichier.
PATCH	Met partiellement à jour une tâche.
DELETE	Supprime un enregistrement racine, un enregistrement enfant ou un fichier.

Méthode d'authentification

Les points d'extrémité REST des services d'entité commerciale utilisent la méthode d'authentification HTTP de base pour authentifier les utilisateurs. Lorsque vous vous connectez pour la première fois à un service d'entité commerciale à l'aide de votre navigateur, vous devez indiquer votre nom d'utilisateur et votre mot de passe MDM Hub. Une fois l'authentification réussie, vous pouvez utiliser les API REST des services d'entité commerciale pour effectuer des opérations.

Le navigateur met en cache les justificatifs d'identité utilisateur et s'en sert pour les demandes suivantes de connexion à un service d'entité commerciale.

Cookies d'authentification pour la connexion à partir d'applications tierces

Utilisez les cookies d'authentification pour authentifier les utilisateurs d'Informatica MDM Hub et appeler les services d'entité commerciale d'applications tierces. Vous pouvez obtenir un cookie basé sur les justificatifs d'identité d'un utilisateur authentifié. Enregistrez le cookie et utilisez-le pour appeler les API REST. Vous n'avez pas besoin de coder en dur le nom d'utilisateur et le mot de passe.

Effectuez la demande POST suivante pour vous connecter à la vue Entity 360 à l'aide de votre nom d'utilisateur et de votre mot de passe :

```
POST http://<host>:<port>/e360/com.informatica.tools.mdm.web.auth/login
{
  user: 'admin',
  password: 'user password'
}
```

Lorsque l'opération de connexion a réussi, le serveur renvoie le cookie d'authentification dans le champ en-tête `set-cookie`. L'exemple de code suivant montre un champ `set-cookie` dans l'en-tête de réponse :

```
Set-Cookie: auth_hash_cookie="admin===QTclRkNGQkNCMzc1RjIyOQ==";
Version=1; Path=/
```

Stockez le hachage et utilisez-le dans l'en-tête de demande de vos appels d'API. Vous n'avez pas besoin d'indiquer de nom d'utilisateur et de mot de passe pour les appels d'API.

L'exemple suivant montre comment utiliser un cookie d'authentification dans l'en-tête de demande d'API :

```
GET http://<IP of host>:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
Cookie: auth_hash_cookie="admin===QTclRkNGQkNCMzc1RjIyOQ=="
```

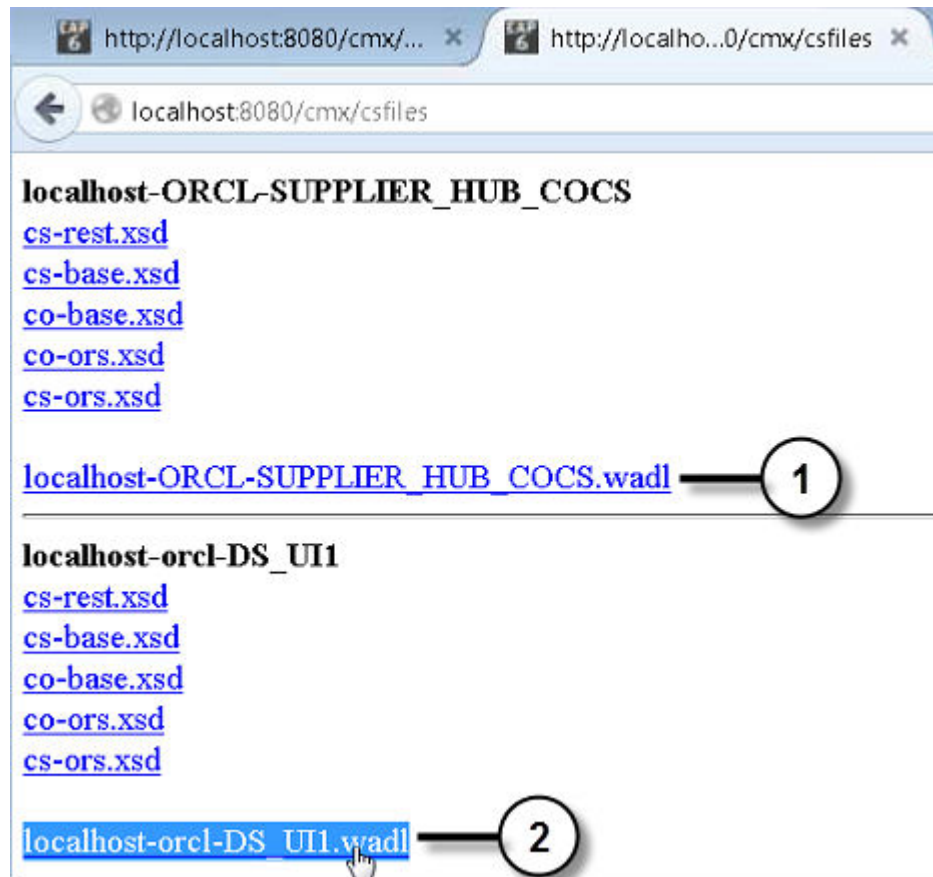
Fichier Web Application Description Language

Les fichiers Web Application Description Language (WADL) contiennent les descriptions XML des services Web REST, toutes les URL REST et tous les paramètres REST. MDM Hub génère un fichier WADL pour chaque stockage de référence opérationnelle.

Les fichiers WADL de chaque stockage de référence opérationnelle se trouvent dans l'emplacement suivant :

```
http://<hôte>:<port>/cmx/csfiles
```

L'image suivante montre l'emplacement où vous pouvez télécharger le fichier WADL pour les stockages de référence opérationnelle :



1. Lien de téléchargement du fichier WADL pour le stockage de référence opérationnelle SUPPLIER_HUB_COCS
2. Lien de téléchargement du fichier WADL pour le stockage de référence opérationnelle DS_UI1

Uniform Resource Locator REST

Vous utilisez une URL REST pour effectuer des appels REST pour des services d'entité d'entreprise.

L'URL REST a la syntaxe suivante :

`http://<hôte>:<port>/<contexte>/<identifiant de base de données>/<chemin>`

L'URL contient les champs suivants :

hôte

Hôte qui exécute la base de données.

port

Numéro de port utilisé par l'écouteur de la base de données.

contexte

Le contexte des API d'entité d'entreprise, de recherche, de requête, de correspondance, de tâche et de hiérarchie est `cmx/cs`.

Le contexte de l'API des colonnes de correspondance est `cmx`.

Le contexte des API de fichier est `cmx/file`.

Le contexte des API de tâche est `cmx/task`.

Le contexte des API d'administration de tâches en bloc est `cmx/task/operations`.

Le contexte des API de relation en bloc est `cmx/bulk`.

Le contexte des API de rapport de graphe est `cmx/report`.

Remarque: Dans un environnement MDM hébergé, incluez le nom du locataire dans le contexte. Par exemple, le contexte peut être `<tenant name>/cmx/cs` ou `<tenant name>/cmx/file`.

database ID

ID du stockage d'enregistrement opérationnel tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.

chemin

Objet sur lequel vous souhaitez utiliser l'API, comme des enregistrements, des tâches ou des fichiers.

Si l'URL correspond à un enregistrement racine, le chemin est le nom de l'objet racine suivi d'un identificateur unique.

`Person/798243.json` est un exemple de chemin pour un enregistrement racine `Person`.

Si l'URL correspond à un enregistrement qui est un enfant direct de l'objet racine, le chemin inclut également le nom de l'enregistrement enfant et un identificateur unique.

Exemple de chemin pour un enregistrement d'adresse de facturation qui est un enfant d'un enregistrement racine `Person` :

`Person/798243/BillAddresses/121522.json`.

Si l'URL correspond à un enregistrement enfant situé à un niveau 2 ou supérieur, le chemin inclut aussi le niveau.

L'URL suivante est un exemple d'URL REST pour un enregistrement enfant situé à un niveau 2 :

`http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/798243/BillAddresses/121522.json?depth=2`

Remarque: Les paramètres sont sensibles à la casse. Vérifiez que la casse des noms de paramètres dans l'URL REST correspond à celle des noms de paramètres dans la configuration REST.

Configuration de l'en-tête et du corps

Une opération REST associe une méthode HTTP à l'URL complète de la ressource. Pour une demande complète, associez l'opération REST aux en-têtes HTTP appropriés et aux données requises. Une demande REST contient un en-tête et un corps. Vous pouvez utiliser le format JSON ou XML pour définir une demande.

En-tête de demande

Utilisez un en-tête de demande pour définir les paramètres ou les métadonnées d'exécution de l'opération REST. L'en-tête comporte une série de paires valeur-champ. La ligne de demande d'API contient la méthode et l'URL. Spécifiez les champs d'en-tête après la ligne de demande.

Pour construire l'en-tête de demande d'API REST, ajoutez les champs d'en-tête après la ligne de demande `<METHOD> <<host>:<port>/<context>/<database ID>/<Path>`, comme indiqué dans l'exemple suivant :

```
<METHOD> <<host>:<port>/<context>/<database ID>/<Path>
Content-Type: application/<json/xml>
Accept: application/<json/xml>
```

Le tableau suivant décrit des champs d'en-tête de demande fréquemment utilisés :

Composant de la demande	Description
Content-Type	Type de média des données dans la demande. Lorsque vous incluez un corps dans la demande REST, vous devez spécifier le type de média du corps dans le champ d'en-tête Content-Type. Incluez le champ d'en-tête Content-Type dans les demandes PUT et POST.
Accept	Type de média des données dans la réponse. Pour spécifier le format de la demande, utilisez <code>application/<json/xml></code> dans l'en-tête ou ajoutez <code>.json</code> ou <code>.xml</code> dans l'URL. La valeur par défaut est <code>.XML</code> .

Corps de demande

Utilisez le corps de demande d'API REST pour envoyer des données dans la demande. Un corps de demande est utilisé avec une méthode à laquelle un corps peut être associé, par exemple les méthodes POST et PUT. Le corps de données est écrit après les lignes d'en-tête. Si le message de demande contient un corps, utilisez le champ d'en-tête Type de contenu pour spécifier le format du corps dans l'en-tête de demande.

Les fichiers XML Schema Definition (XSD) décrivent les éléments et attributs que vous pouvez utiliser. Le contenu du corps de demande dépend des types d'éléments que vous définissez dans les fichiers XSD.

Les fichiers XSD se trouvent dans l'emplacement suivant :

```
http://<hôte>:<port>/cmx/csfiles
```

Format XML

Lorsque vous utilisez le format de demande XML, définissez l'objet de demande sous forme d'ensemble de balises englobant.

Utilisez le format XML suivant pour définir un objet de demande :

```
<request object>
  <attribute1>value1</attribute1>
  <attribute2>value2</attribute2>
</request object>
```

L'exemple suivant montre la représentation XML d'un objet de demande :

```
<task>
  <taskType>
    <name>UpdateWithApprovalWorkflow</name>
  </taskType>
  <taskId>urn:b4p2:5149</taskId>
  <owner>manager</owner>
  <title>Smoke test task 222</title>
  <comments>Smoke testing</comments>
```

```

<dueDate>2015-06-15T00:00:00</dueDate>
<status>OPEN</status>
<priority>NORMAL</priority>
<creator>admin</creator>
<createDate>2015-06-15T00:00:00</createDate>
<updateBy>admin</updateBy>
<lastUpdateDate>2015-06-15T00:00:00</lastUpdateDate>
<businessEntity>Person</businessEntity>
<orsId>localhost-orcl-DS_UI1</orsId>
<processId>IDDUpdateWithApprovalTask</processId>
<taskRecord>
  <businessEntity>
    <name>Person</name>
    <key>
      <rowid>123</rowid>
      <systemName></systemName>
      <sourceKey></sourceKey>
    </key>
  </businessEntity>
</taskRecord>
</task>

```

Format JSON

Lorsque vous utilisez le format de demande JSON, définissez l'objet de demande avec l'attribut type.

Utilisez le format JSON suivant pour spécifier un objet de demande :

```

{
  "type"="<request object>"
  "<attribute1>": "<value1>",
  "<attribute2>": "<value2>",
}

```

L'exemple suivant montre la représentation JSON d'un objet de demande :

```

{
  "type"="task"
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "urn:b4p2:5149",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updateBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: 'IDDUpdateWithApprovalTask',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}

```


Paramètres de requête standard

Les API REST des services d'entité commerciale utilisent des paramètres de requête standard pour filtrer, paginer et développer les résultats.

Utilisez un point d'interrogation (?) pour séparer les paramètres de requête des autres paramètres. Les paramètres de requête sont des paires clé-valeur séparées par le signe égal. Utilisez une esperluette (&) pour séparer une séquence de paramètres de requête.

L'URL de demande REST suivante montre comment utiliser les paramètres de requête :

```
/Person/123/Phone/SFA:456/PhoneUse?recordsToReturn=100&recordStates=ACTIVE,PENDING
```

Utilisez les paramètres de requête standard suivants :

Paramètre	Description
recordsToReturn	Spécifie le nombre de lignes à renvoyer. La valeur par défaut est 10.
firstRecord	Spécifie la première ligne dans le résultat. La valeur par défaut est 1. Utilisé dans les appels suivants pour lire plus de pages.
searchToken	Spécifie le jeton de recherche renvoyé avec la demande précédente. Vous pouvez utiliser le jeton de recherche pour extraire les pages suivantes des résultats de recherche. Par exemple, la requête suivante liste la première page : <code>/Person/123/Phone</code> La requête suivante renvoie la deuxième page : <code>/Person/123/Phone?searchToken=SVR1.AZAM5&firstRecord=10</code>
returnTotal	Si défini sur True, renvoie le nombre d'enregistrements dans le résultat. La valeur par défaut est False.
depth	Spécifie le nombre de niveaux enfants à inclure dans le résultat.

Date et heure au format UTC

Dans la demande comme dans la réponse, toutes les dates et les heures sont spécifiées au format UTC (temps universel coordonné), avec ou sans décalage pour un fuseau horaire spécifique.

Lorsque vous spécifiez une date et une heure dans un corps de demande, utilisez l'un des formats définis dans [Date and Time Formats \(NOTE-datetime\)](#) selon la norme ISO 8601.

Les directives ci-dessous sont extraites du document NOTE-datetime :

Type	Syntaxe	Exemple
Date : année	AAAA	1997
Date : année et mois	AAAA-MM	1997-07
Date : année, mois et jour	AAAA_MM-JJ	1997-07-16
Date avec heures et minutes	AAAA-MM-JJThh:mmTZD	1997-07-16T19:20+01:00

Type	Syntaxe	Exemple
Date avec heures, minutes et secondes	AAAA-MM-JJThh:mm:ssTZD	1997-07-16T19:20:30+01:00
Date avec heures, minutes, secondes et fractions de secondes	AAAA-MM-JJThh:mm:ss.sTZD	1997-07-16T19:20:30.45+01:00

où :

- AAAA = une année en quatre chiffres
- MM = un mois en deux chiffres, de 01 à 12
- JJ = un jour du mois en deux chiffres, de 01 à 31
- T = valeur littérale qui suit la date et introduit l'heure
- hh = deux chiffres pour l'heure, de 00 à 23
- mm = deux chiffres pour les minutes, de 00 à 59
- ss = deux chiffres pour les secondes, de 00 à 59
- s = un chiffre ou plus représentant la fraction décimale d'une seconde
- TZD = indicateur de fuseau horaire (Z ou +hh:mm ou -hh:mm)
 - Z pour heure UTC
 - +hh:mm pour un fuseau horaire local en avance sur UTC
 - -hh:mm pour un fuseau horaire local en retard par rapport à UTC

Configuration de WebLogic pour exécuter des appels REST de services d'entité commerciale

Comme les appels REST de services d'entité commerciale utilisent l'authentification HTTP de base, vous devez désactiver l'authentification du serveur WebLogic pour ce type d'appels. Pour configurer WebLogic afin d'exécuter des appels REST de services d'entité commerciale, modifiez le fichier `config.xml` de WebLogic.

1. Accédez au répertoire WebLogic suivant :

Sous UNIX.

```
<répertoire d'installation de WebLogic>/user_projects/domains/base_domain/config
```

Sous Windows.

```
<répertoire d'installation de WebLogic>\user_projects\domains\base_domain\config
```

2. Ouvrez le fichier suivant dans un éditeur de texte :

```
config.xml
```

3. Ajoutez le code XML suivant avant la balise de fermeture `</security-configuration>` :

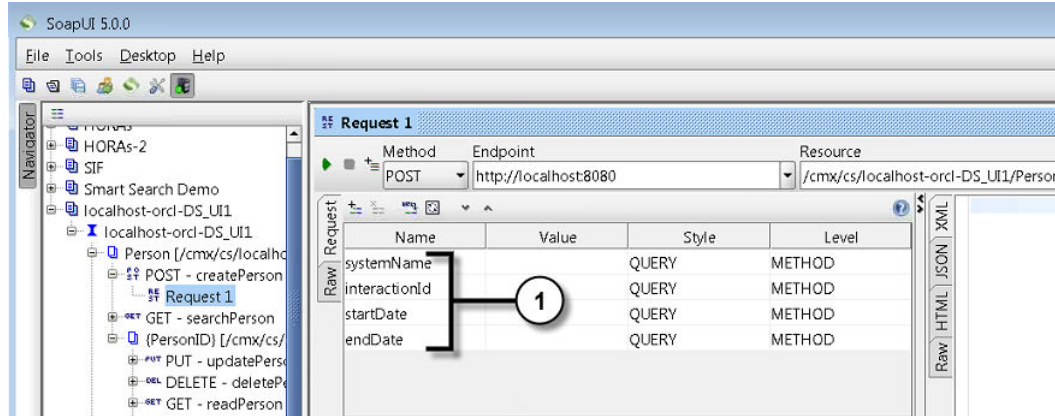
```
<enforce-valid-basic-auth-credentials>
false
</enforce-valid-basic-auth-credentials>
```

Affichage des paramètres d'entrée et de sortie

Vous pouvez utiliser un outil de test fonctionnel comme SoapUI pour afficher les paramètres d'entrée et de sortie de l'API REST.

Téléchargez le fichier WADL et importez-le dans l'outil de test fonctionnel pour créer un projet REST.

L'image suivante montre les paramètres d'entrée dans SoapUI pour l'API REST createPerson :



1. Paramètres d'entrée pour l'API REST createPerson

Remarque: Il est possible qu'un fichier WADL créé dans un environnement WebSphere ne soit pas importé correctement dans SoapUI. Si les paramètres d'entrée ne s'affichent pas dans SoapUI, modifiez le fichier WADL pour supprimer l'attribut `xmlns` de chaque élément `param`, puis importez à nouveau le fichier WADL.

Modèle JavaScript

L'exemple de code suivant montre un modèle de base que vous pouvez modifier pour créer un code JavaScript pour les appels de services d'entité commerciale REST. Vous devez utiliser la bibliothèque JavaScript jQuery.

```
(function ($) {  
    window.CSClient = window.CSClient || {  
        baseUrl: "/cmx/cs/" + "[siperian-client.orsId]",  
        user: "[siperian-client.username]",  
        pass: "[siperian-client.password]",  
  
        process: function (method, url, body, params) {  
            var fullUrl = this.baseUrl + url + ".json?" + $.param(params);  
            return $.ajax({  
                method: method,  
                contentType: "application/json",  
                url: fullUrl,  
                data: JSON.stringify(body),  
                beforeSend: function (xhr) {  
                    xhr.setRequestHeader("Authorization", "Basic " + btoa(CSClient.user  
+ ":" + CSClient.pass));  
                }  
            });  
        },  
  
        readCo: function (url, params) {  
            return this.process("GET", url, null, params);  
        },  
        createCo: function (url, body, params) {
```

```

        return this.process("POST", url, body, params);
    },
    updateCo: function (url, body, params) {
        return this.process("PUT", url, body, params);
    },
    deleteCo: function (url, params) {
        return this.process("DELETE", url, null, params);
    }
};
})(jQuery);

```

Exemple de code JavaScript

Le kit de ressources contient un exemple de code source Java qui montre comment effectuer des appels de services d'entité commerciale REST.

Cet exemple de code figure dans le fichier suivant :

```

<répertoire d'installation de MDM Hub>\hub\resourcekit\samples\COS\source\resources\webapp
\rest-api.html

```

Le code suivant montre des appels d'API REST pour créer un enregistrement racine Person, ajouter plusieurs enregistrements enfants, supprimer un enregistrement enfant et supprimer l'enregistrement Person et tous les enregistrements enfants :

```

<html>
<head>
  <script type="text/javascript" src="jquery-1.11.1.js"></script>
  <script type="text/javascript" src="cs-client.js"></script>
</head>
<body>

<script type="text/javascript" language="javascript">
  $(document).ready(function () {

    $("#run").click(function () {

      log = function(msg, json) {
        $('#log').before("<hr/><b>" + msg + "</b>");
        $('#log').before("<pre>" + JSON.stringify(json, undefined, 2) + "</
pre>");
      };

      CSClient.createCo(
        "/Person",
        {
          firstName: "John",
          lastName: "Smith"
        },
        {
          systemName: "Admin"
        }
      ).then(
        function (result) {
          log("PERSON CREATED:", result);
          return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
              depth: 1
            }
          );
        }
      ).then(
        function (result) {

```

```

        log("READ CREATED PERSON:", result);
        return CSClient.updateCo(
            "/Person/" + result.rowidObject.trim(),
            {
                genderCd: {
                    genderCode: "M"
                },
                TelephoneNumbers: {
                    item: [
                        {
                            phoneNumber: "111-11-11"
                        },
                        {
                            phoneNumber: "222-22-22"
                        }
                    ]
                }
            },
            {
                systemName: "Admin"
            }
        );
    }
    ).then(
        function (result) {
            log("PERSON UPDATED:", result);
            return CSClient.readCo(
                "/Person/" + result.Person.rowidObject.trim(),
                {
                    depth: 3,
                    readSystemFields: true
                }
            );
        }
    ).then(
        function (result) {
            log("READ UPDATED PERSON:", result);
            return CSClient.deleteCo(
                "/Person/" + result.rowidObject.trim() + "/" +
                TelephoneNumbers/" + result.TelephoneNumbers.item[0].rowidObject.trim(),
                {
                    systemName: "Admin"
                }
            );
        }
    ).then(
        function (result) {
            log("TELEPHONE DELETED:", result);
            return CSClient.readCo(
                "/Person/" + result.Person.rowidObject.trim(),
                {
                    depth: 3
                }
            );
        }
    ).then(
        function (result) {
            log("READ PERSON AFTER TELEPHONE IS DELETED:", result);
            return CSClient.deleteCo(
                "/Person/" + result.rowidObject.trim(),
                {
                    systemName: "Admin"
                }
            );
        }
    ).then(
        function (result) {
            log("PERSON DELETED:", result);
            return CSClient.readCo(
                "/Person/" + result.Person.rowidObject.trim(),
                {

```

```

        depth: 1,
        recordStates: "ACTIVE,PENDING,DELETED",
        readSystemFields: true
    }
    );
}
).then(
    function (result) {
        log("READ PERSON AFTER DELETE (HSI -1):", result);
    }
);
});
});
</script>

<input type="button" id="run" value="Run..." />
<p/>
<div id="log"></div>
</body>
</html>

```

Référence d'API REST pour les services d'entité d'entreprise

La référence d'API REST pour les services d'entité d'entreprise répertorie les API REST et fournit une description de chaque API. Elle contient également des informations sur les URL, les paramètres de requête, les exemples de demandes et les exemples de réponses.

Obtenir les métadonnées

L'API REST Obtenir les métadonnées renvoie la structure de données d'une entité d'entreprise ou d'une relation d'entité d'entreprise.

L'API utilise la méthode GET pour renvoyer les métadonnées suivantes d'une entité d'entreprise :

- Structure de l'entité d'entreprise
- Liste de champs
- Types de champs tels que le type de données et la taille
- Paramètres de sécurité d'opérations telles que la création, la mise à jour et la fusion
- Libellés localisés pour les nœuds ou les champs
- Nom du code et champs d'affichage pour les champs de recherche
- Valeurs par défaut des champs et des champs de recherche

Remarque: Des valeurs par défaut multiples ne sont pas prises en charge avec des champs de recherche dépendants.

L'API renvoie les détails suivants concernant la relation d'entité d'entreprise :

- Nom de la relation et son libellé.
- Entités d'entreprise De (from) et À (to).
- Sens de la relation.

URL de demande

Le format de l'URL Obtenir les métadonnées d'une entité d'entreprise est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

Le format de l'URL Obtenir les métadonnées d'une relation est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<relationship>?action=meta
```

Utilisez le paramètre de requête `action=meta` pour récupérer les informations de métadonnées. Assurez-vous de spécifier correctement le nom de l'entité d'entreprise.

Effectuez une demande HTTP GET à l'URL Obtenir les métadonnées :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity/relationship>?
action=meta
```

Vous pouvez ajouter des en-têtes HTTP à la demande.

Exemple de demande d'API

L'exemple de demande suivant récupère les informations de métadonnées pour l'entité d'entreprise et le nœud racine `Person` :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

L'exemple de demande suivant inclut un en-tête pour récupérer les informations de métadonnées pour l'entité d'entreprise `Person` au format JSON :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
Accept: application/json
```

L'exemple de demande suivant récupère les informations de métadonnées de la relation `HouseholdContainsMemberPerson` :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductGroupProductGroup?action=meta
```

Exemple de réponse d'API

Les exemples de réponse d'une entité et d'une relation contiennent des autorisations de sécurité. La première section `operations` définit les autorisations possibles. La section `object` répertorie vos autorisations pour l'entité d'entreprise ou la relation complète. La section `fields` définit vos autorisations au niveau du champ.

L'exemple suivant montre une structure de données partielle pour l'entité d'entreprise `Personne` au format JSON.

```
{
  "operations": {
    "read": {
      "allowed": true
    },
    "search": {
      "allowed": true
    },
    "create": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUpdate",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
      }
    }
  }
}
```

```

    }
  },
  "update": {
    "allowed": true,
    "task": {
      "template": {
        "title": "Review changes in {taskRecord[0].label}",
        "priority": "NORMAL",
        "dueDate": "2018-04-24T09:28:13.455-04:00",
        "taskType": "AVOSBeUpdate",
        "comment": "This is urgent. Please review ASAP"
      },
      "comment": "AS_REQUIRED",
      "attachment": "OPTIONAL"
    }
  },
  "merge": {
    "allowed": true,
    "task": {
      "template": {
        "title": "Review changes in {taskRecord[0].label}",
        "priority": "NORMAL",
        "dueDate": "2018-04-24T09:28:13.455-04:00",
        "taskType": "AVOSBeMerge",
        "comment": "This is urgent. Please review ASAP"
      },
      "comment": "AS_REQUIRED",
      "attachment": "OPTIONAL"
    }
  },
  "delete": {
    "allowed": true
  },
  "unmerge": {
    "allowed": true,
    "task": {
      "template": {
        "title": "Review changes in {taskRecord[0].label}",
        "priority": "NORMAL",
        "dueDate": "2018-04-24T09:28:13.455-04:00",
        "taskType": "AVOSBeUnmerge",
        "comment": "This is urgent. Please review ASAP"
      },
      "comment": "AS_REQUIRED",
      "attachment": "OPTIONAL"
    }
  }
},
"objectType": "ENTITY",
"timeline": true,
"object": {
  "operations": {
    "read": {
      "allowed": true
    },
    "create": {
      "allowed": true
    },
    "update": {
      "allowed": true
    },
    "merge": {
      "allowed": true
    },
    "delete": {
      "allowed": true
    },
    "unmerge": {
      "allowed": true
    }
  }
},

```



```

"field": [
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "create": {
        "allowed": true
      },
      "update": {
        "allowed": true
      }
    },
    "allowedValues": [
      "Person"
    ],
    "searchable": {
      "filterable": true,
      "facet": true
    },
    "name": "partyType",
    "label": "Party Type",
    "dataType": "String",
    "length": 255
  },
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "create": {
        "allowed": true
      },
      "update": {
        "allowed": true
      }
    },
    "name": "lastName",
    "label": "Last Name",
    "dataType": "String",
    "length": 50
  },
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "create": {
        "allowed": true
      },
      "update": {
        "allowed": true
      }
    },
    "searchable": {
      "filterable": true,
      "facet": true
    },
    "name": "displayName",
    "label": "Display Name",
    "dataType": "String",
    "length": 200
  },
  ...
],
"name": "Person",
"label": "Person",
"many": false
}
}

```

Répertoire des métadonnées

L'API REST Répertoire des métadonnées renvoie une liste des entités d'entreprise ou des relations que vous avez définies. La réponse contient des informations de chronologie et de sécurité si celles-ci sont mentionnées dans les entités d'entreprise. Vous pouvez utiliser l'API pour récupérer la liste des relations qui démarrent d'une entité, se terminent à une entité ou démarrent et se terminent à des entités spécifiées.

L'API utilise la méthode GET.

URL Répertoire des métadonnées

Le format de l'URL Répertoire des métadonnées est le suivant pour les métadonnées d'entreprise :

```
http://<host>:<port>/<context>/<database ID>/meta/entity
```

Le format de l'URL Répertoire des métadonnées est le suivant pour les métadonnées de relation :

```
http://<host>:<port>/<context>/<database ID>/meta/relationship
```

Effectuez la demande HTTP GET suivante à l'URL Répertoire des métadonnées :

```
GET http://<host>:<port>/<context>/<database ID>/meta/entity|relationship
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de requête à l'URL de la demande pour filtrer les résultats de la recherche pour les relations d'entité d'entreprise. Vous pouvez spécifier le sens de recherche des relations depuis et vers l'entité d'entreprise.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
start	Facultatif. Spécifie l'entité d'où provient une relation. Par exemple, la requête <code>meta/relationship?start=Organization</code> renvoie toutes les relations qui commencent depuis l'entité d'entreprise <code>Organisation</code> .
finish	Facultatif. Spécifie l'entité à laquelle une relation se termine. Par exemple, la requête <code>meta/relationship?finish=Person</code> renvoie toutes les relations qui se terminent à l'entité d'entreprise <code>Personne</code> .

Vous pouvez spécifier les deux paramètres pour récupérer toutes les relations entre deux entités d'entreprise. Par exemple, la requête `meta/relationship?start=Organization&finish=Person` renvoie toutes les relations qui démarrent de l'entité d'entreprise `Organisation` et se terminent à l'entité d'entreprise `Personne`.

Exemple de demande d'API

L'exemple de demande suivant récupère la liste des entités d'entreprise configurées :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/entity
```

L'exemple de demande suivant récupère la liste des relations configurées :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/relationship
```

L'exemple de demande suivant récupère la liste des relations qui commencent à l'entité d'entreprise Organisation et se terminent à l'entité d'entreprise Personne :

```
GET http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/meta/relationship?
start=Organization&finish=Person
```

Exemple de réponse d'API

L'exemple suivant montre un extrait de la liste des relations configurées :

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "item": [
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "search": {
          "allowed": false
        },
        "create": {
          "allowed": true,
          "task": {
            "template": {
              "title": "Review changes in {taskRecord[0].label}",
              "priority": "NORMAL",
              "dueDate": "2018-04-24T09:31:48.167-04:00",
              "taskType": "AVOSBeUpdate",
              "comment": "This is urgent. Please review ASAP"
            },
            "comment": "AS_REQUIRED",
            "attachment": "OPTIONAL"
          }
        },
        "update": {
          "allowed": true,
          "task": {
            "template": {
              "title": "Review changes in {taskRecord[0].label}",
              "priority": "NORMAL",
              "dueDate": "2018-04-24T09:31:48.167-04:00",
              "taskType": "AVOSBeUpdate",
              "comment": "This is urgent. Please review ASAP"
            },
            "comment": "AS_REQUIRED",
            "attachment": "OPTIONAL"
          }
        },
        "merge": {
          "allowed": true,
          "task": {
            "template": {
              "title": "Review changes in {taskRecord[0].label}",
              "priority": "NORMAL",
              "dueDate": "2018-04-24T09:31:48.167-04:00",
              "taskType": "AVOSBeMerge",
              "comment": "This is urgent. Please review ASAP"
            },
            "comment": "AS_REQUIRED",
            "attachment": "OPTIONAL"
          }
        },
        "delete": {
          "allowed": true
        },
        "unmerge": {
          "allowed": true,

```

```

"task": {
  "template": {
    "title": "Review changes in {taskRecord[0].label}",
    "priority": "NORMAL",
    "dueDate": "2018-04-24T09:31:48.167-04:00",
    "taskType": "AVOSBeUnmerge",
    "comment": "This is urgent. Please review ASAP"
  },
  "comment": "AS REQUIRED",
  "attachment": "OPTIONAL"
}
},
"objectType": "ENTITY",
"timeline": false,
"object": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/CreditCard.json?action=meta",
      "rel": "entity"
    }
  ],
  "field": [
    {
      "name": "issuingCompany",
      "label": "Issuing Company",
      "dataType": "String",
      "length": 100
    },
    {
      "name": "expirationYear",
      "label": "Expiration Year",
      "dataType": "String",
      "length": 4
    },
    {
      "allowedValues": [
        "Credit Card"
      ],
      "name": "accountType",
      "label": "Account Type",
      "dataType": "String",
      "length": 255
    },
    {
      "name": "accountNumber",
      "label": "Account Number",
      "dataType": "String",
      "length": 20
    },
    {
      "name": "securityCode",
      "label": "Security Code",
      "dataType": "String",
      "length": 4
    },
    {
      "name": "expirationMonth",
      "label": "Expiration Month",
      "dataType": "String",
      "length": 2
    },
    {
      "name": "cardholderName",
      "label": "Card Holder Name",
      "dataType": "String",
      "length": 100
    },
    {
      "name": "consolidationInd",

```

```

        "label": "Consolidation Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "creator",
        "label": "Creator",
        "dataType": "String",
        "length": 50,
        "readOnly": true,
        "system": true
    },
    {
        "name": "interactionId",
        "label": "Interaction Id",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "updatedBy",
        "label": "Updated By",
        "dataType": "String",
        "length": 50,
        "readOnly": true,
        "system": true
    },
    {
        "name": "lastUpdateDate",
        "label": "Last Update Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true
    },
    {
        "name": "lastRowidSystem",
        "label": "Last Rowid System",
        "dataType": "String",
        "length": 14,
        "readOnly": true,
        "system": true
    },
    {
        "name": "dirtyIndicator",
        "label": "Dirty Indicator",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "deletedBy",
        "label": "Deleted By",
        "dataType": "String",
        "length": 50,
        "readOnly": true,
        "system": true
    },
    {
        "name": "deletedInd",
        "label": "Deleted Indicator",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "hubStateInd",

```

```

        "label": "Hub State Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "deletedDate",
        "label": "Deleted Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true
    },
    {
        "name": "rowidObject",
        "label": "Rowid Object",
        "dataType": "String",
        "length": 14,
        "readOnly": true,
        "system": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "createDate",
        "label": "Create Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true
    }
],
"name": "CreditCard",
"label": "Credit Card",
"many": false
}
},
...
]
}

```

Répertoirer les colonnes de correspondance

L'API REST Répertoirer les colonnes de correspondance peut renvoyer une liste d'ensembles de règles de correspondance pour une entité d'entreprise spécifiée ou une liste de colonnes de correspondance pour un ensemble de règles de correspondance spécifié. Vous pouvez générer une liste de colonnes de correspondance et les utiliser avec l'API REST SearchMatch.

Pour plus d'informations sur la configuration des colonnes de correspondance et des ensembles de règles de correspondance, consultez le *Guide de configuration de MDM Multidomain*.

L'API utilise la méthode GET.

URL de demande

Le contexte de l'URL de l'option Répertoirer les colonnes de correspondance est `cmx`. Dans un environnement MDM hébergé, incluez le nom du locataire dans le contexte, tel que `<tenant name>/cmx`.

L'URL de l'option Répertoirer les colonnes de correspondance est aux formats suivants :

URL pour renvoyer tous les ensembles de règles de correspondance

Utilisez l'URL suivante pour répertorier tous les ensembles de règles de correspondance d'une entité d'entreprise spécifique :

```
http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>
```

Effectuez la demande HTTP GET suivante à l'URL de l'option Répertorier les colonnes de correspondance :

```
GET http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>
```

URL pour renvoyer les colonnes de correspondance utilisées dans un ensemble de règles de correspondance spécifié

Utilisez l'URL suivante pour répertorier toutes les colonnes de correspondance dans un ensemble de règles de correspondance spécifié :

```
http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>/<match rule set>
```

Exemple de demande d'API

Demande d'ensembles de règles de correspondance

L'exemple de demande suivant répertorie les ensembles de règles de correspondance pour l'entité d'entreprise Personne :

```
GET http://localhost:8080/cmx/queryTemplate/localhost-orcl-DS_UI1/Person
```

Demande de colonnes de correspondance

L'exemple de demande suivant répertorie les colonnes de correspondance utilisées dans l'ensemble de règles de correspondance IDL2 :

```
GET http://localhost:8080/cmx/queryTemplate/localhost-orcl-DS_UI1/Person/IDL2
```

Exemple de réponse d'API

L'exemple de réponse suivant contient les colonnes de correspondance comprises dans l'ensemble de règles de correspondance IDL2 :

```
{
  "queryTemplates":[
    {
      "businessEntity": "Person",
      "matchRuleSet": "IDL2",
      "type": "extended",
      "searchFields": [
        {
          "name": "displayName",
          "mandatory": true
        },
        {
          "name": "BillAddresses.Address.addressLine1",
          "mandatory": false
        },
        {
          "name": "ShipAddresses.Address.addressLine2",
          "mandatory": false
        },
        {
          "name": "ShipAddresses.Address.addressLine1",
          "mandatory": false
        }
      ]
    }
  ]
}
```

Lire l'enregistrement

L'API REST Lire l'enregistrement renvoie les détails d'un enregistrement racine dans l'entité d'entreprise. Vous pouvez utiliser l'API pour renvoyer les détails des enregistrements enfants d'un enregistrement racine. Vous pouvez aussi l'utiliser pour afficher les métadonnées de contenu d'un enregistrement.

L'API utilise la méthode GET.

Vous pouvez trier l'ensemble de résultats de façon à afficher les informations par ordre croissant ou décroissant. Utilisez la méthode POST lorsque vous avez besoin de paramètres supplémentaires ou plus complexes. Par exemple, lorsque vous voulez récupérer les données dans lesquelles les éléments enfants sont triés selon un ensemble de champs.

URL de demande

Utilisez l'ID de ligne ou le nom du système source et la clé source pour spécifier l'enregistrement dans l'URL de demande.

L'URL Lire l'enregistrement peut être aux formats suivants :

URL avec l'ID de ligne

Utilisez le format d'URL suivant lorsque vous spécifiez l'ID de ligne :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

URL avec le nom du système source et la clé source

Utilisez le format d'URL suivant lorsque vous spécifiez le nom du système source et la clé source :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:<source key>
```

URL avec le nom du système et l'identificateur d'entreprise global (GBID) d'un objet

Utilisez le format d'URL suivant lorsque vous spécifiez le nom du système source et le GBID :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:uid:<gbid>
```

URL avec le GBID uniquement

Utilisez le format d'URL suivant lorsque vous spécifiez uniquement le GBID :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:uid:<gbid>
```

URL avec plusieurs GBID

Utilisez le format d'URL suivant lorsque vous spécifiez plusieurs GBID :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:one:<gbid>,another:<gbid>
```

URL pour renvoyer les détails des nœuds enfants

Utilisez le format d'URL suivant pour renvoyer les détails des nœuds enfants :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>?depth=n
```


URL pour renvoyer les détails d'un nœud enfant

Utilisez le format d'URL suivant pour renvoyer les détails d'un nœud enfant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>/  
children=<comma separated list of child node names or paths>
```

Par exemple, children= BillAddresses/Address,Email

URL pour renvoyer les détails d'un nœud spécifique

Utilisez le format d'URL suivant pour renvoyer les détails d'un nœud spécifique :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>/  
<node name>
```

URL pour renvoyer les détails des enfants d'un nœud spécifique

Utilisez le format d'URL suivant pour renvoyer les détails des enfants d'un nœud spécifique :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>/  
<node name>?children=<child node name>
```

URL pour renvoyer les métadonnées de contenu d'un enregistrement

Utilisez le format d'URL suivant pour renvoyer les métadonnées de contenu d'un enregistrement :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root  
record>?contentMetadata=<content metadata type>
```

Par exemple, vous pouvez récupérer des correspondances pour les enregistrements enfants à l'aide de la demande GET suivante :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the  
root record>?contentMetadata=MATCH
```

URL pour trier les éléments enfants par champ

Utilisez le format d'URL suivant pour trier les éléments enfants par champ :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root  
record>/<node name>?order=-<field name>
```

Utilisez le caractère - comme suffixe pour spécifier l'ordre décroissant.

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les détails de l'enregistrement.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
depth	Nombre de niveaux enfants à renvoyer. Indiquez 2 pour renvoyer le nœud racine et ses enfants directs et 3 pour renvoyer le nœud racine, ses enfants directs et ses petits-enfants. La valeur par défaut est 1.
effectiveDate	Date pour laquelle vous souhaitez récupérer les données.
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. La valeur par défaut est False.

Paramètre	Description
recordStates	Etat de l'enregistrement. Fournissez une liste d'états séparés par des virgules. Les états d'enregistrements pris en charge sont ACTIVE, PENDING et DELETED. La valeur par défaut est ACTIVE.
contentMetadata	Métadonnées de l'enregistrement. Fournissez une liste de valeurs séparées par des virgules. Par exemple, XREF, PENDING_XREF, DELETED_XREF, HISTORY, XREF_HISTORY et MATCH. Lorsque vous sélectionnez MATCH, la réponse contient une liste d'enregistrements correspondants qui sont récupérés de la table _MTCH.
historyDate	Date à laquelle vous souhaitez récupérer les données d'historique. La réponse contient les données d'enregistrement pour la date spécifiée récupérée de la table _HIST. Vous pouvez utiliser historyDate combiné au paramètre contentMetadata pour récupérer les métadonnées d'historique. Définissez la valeur de contentMetadata sur XREF, BVT ou TRUST. <ul style="list-style-type: none"> - XREF. La réponse contient des données de références croisées d'historique de la table _HXRF. - BVT. La réponse contient la meilleure version de la vérité de l'historique de la table _HCTL. - TRUST. La réponse contient les paramètres d'approbation d'historique des tables _HCTL et _HVXR.
children	Liste séparée par des virgules de noms ou de chemins de nœud enfants.
suppressLinks	Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur True pour supprimer tous les liens parent-enfant dans la réponse. La valeur par défaut est False. Par exemple, la requête <code>Person/1242?depth=10&suppressLinks=true</code> permet d'afficher les détails d'enregistrements jusqu'à 10 niveaux enfant, sans lien parent-enfant visible dans la réponse.
order	Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +. Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis par le suivant. Par exemple, la requête <code>Person/1242/Names?order=-name</code> affiche le résultat contenant les noms dans l'ordre décroissant. La requête <code>Person/1242/BillAddresses?order=rowidObject,-effStartDate</code> affiche les adresses de facturation contenant le rowid par ordre croissant, puis la date de début effective par ordre décroissant.

L'exemple suivant montre comment filtrer les détails d'un enregistrement :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123/Phone/SFA:456/PhoneUse?
recordsToReturn=100&recordStates=ACTIVE,PENDING&contentMetadata=XREF
```

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Demande POST permettant de spécifier l'ordre de tri des éléments enfants

Une demande POST vous permet de trier l'ensemble de résultats en fonction de différents champs. Incluez les paramètres ou les champs dans le corps de la demande POST.

L'exemple de demande suivant montre comment utiliser une demande POST dans une opération de lecture et comment trier des données selon différents champs :

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/ReadPerson.json
{
  parameters:
  {
    coFilter: {
      object: {
        name:"Person",
        key: {
          rowid: 1242
        },
        order: "lastName",
        object:[
          {name:"Names", order:"-name"},
          {name:"Phone", order:"phoneNum, -phoneCountryCd",
            object:[{name:"PhonePermissions", order:"-column1"}]}
        ]
      }
    }
  }
}
```

Remarque: À chaque niveau d'entité d'entreprise et pour chaque type d'enfant, un seul type d'ordre de tri est autorisé.

Éléments à prendre en compte pour l'ordre de tri

L'API Lire l'enregistrement prend en charge l'ordre de tri par un ou plusieurs champs pour chaque nœud enfant d'entité d'entreprise. La section suivante décrit certaines considérations que vous devez connaître lorsque vous spécifiez l'ordre de tri.

- Si vous spécifiez l'ordre de tri pour le petit-enfant et pas pour l'enfant, l'élément petit-enfant est trié selon l'ordre spécifié, tandis que l'élément enfant ne l'est pas. Exemple de demande :

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone/861/
PhonePermissions?order=-column1
```

Dans cet exemple de demande, un ordre de tri décroissant est spécifié pour le petit-enfant PhonePermissions et aucun ordre n'est spécifié pour l'élément enfant Phone. Phone n'est pas trié selon l'ordre de tri de PhonePermissions.

- Si vous spécifiez l'ordre de tri pour l'enfant et pas pour le petit-enfant, l'élément enfant est trié selon l'ordre de tri spécifié, tandis que l'élément petit-enfant ne l'est pas. Exemple de demande :

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
          [{name:"PhonePermissions"}]}
      ]
    }
  }
}}
```

Dans cet exemple de demande, l'ordre de tri est spécifié pour l'enfant Phone et pas pour le petit-enfant PhonePermissions. L'enfant Phone est trié selon l'ordre indiqué.

- Si vous spécifiez l'ordre de tri pour l'enfant et le petit-enfant, les deux sont triés selon cet ordre de tri. L'exemple de demande suivant spécifie l'ordre de tri pour Phone (enfant) et PhonePermissions (petit-enfant) :

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
        [{name:"PhonePermissions", order:"-column1"}]},
      ]}
    }}}}
```

- Les éléments enfant et petit-enfant peuvent uniquement être triés en fonction de leurs colonnes respectives. Dans l'exemple de demande suivant, Phone est trié par PhoneType et PhonePermissions est trié par Colonne 1. PhoneType est une colonne de Phone (enfant) et Colonne 1 est une colonne de PhonePermissions (petit-enfant).

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone?order=-PhoneType
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone/861/
phonePermissions?order=column1
```

- À chaque niveau d'entité d'entreprise et pour chaque type d'enfant, un seul type d'ordre de tri est autorisé. Dans la demande suivante, vous avez spécifié plusieurs ordres de tri pour les enfants PhonePermissions de différents parents. Toutefois, étant donné que l'ordre de tri est d'abord spécifié par ordre décroissant, les enfants PhonePermissions des deux parents (rowid 861 et rowid 862) sont triés par ordre décroissant.

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", key: { rowid:861 }, order:"+phoneCountryCd, -phoneNum", object:
        [{name:"PhonePermissions", order:"-column1"}]},
        {name:"Phone", key: {rowid:862}, order:"phoneNum, -phoneCountryCd", object:
        [{name:"PhonePermissions", order:"column1"}]}
      ]}
    }}}}
```

Exemples de demandes d'API

L'exemple de demande suivant renvoie les détails de l'enregistrement racine dans l'entité d'entreprise Person.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102
```

L'exemple de demande suivant renvoie les détails d'un enregistrement enfant avec l'ID de ligne 2. L'objet de base enfant est genderCd et l'enregistrement enfant est au niveau 2.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/genderCd/2?depth=2
```

L'exemple de demande suivant utilise le nom du système et la clé source pour renvoyer les détails de l'enregistrement racine :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/SFA:9000000000
```

L'exemple de demande suivant renvoie les détails de l'enregistrement racine et ses enregistrements XREF :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?contentMetadata=XREF
```

L'exemple de demande suivant renvoie les détails de l'enregistrement racine contenant les noms par ordre décroissant :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Names?order=-name
```

L'exemple de demande suivant renvoie les adresses de facturation contenant le rowid par ordre croissant puis, la date de début effective par ordre décroissant :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/BillAddresses?
order=rowidObject,-effStartDate
```

Exemple de réponse d'API

L'exemple suivant affiche les détails de l'enregistrement racine dans l'entité d'entreprise Person.

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?
depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/request/hm_icons/person_small.jpeg?
ors=localhost-orcl-DS_UI1",
      "rel": "icon"
    }
  ],
  "rowidObject": "102",
  "label": "DARWENT, JIMMY",
  "partyType": "Person",
  "statusCd": "A",
  "lastName": "DARWENT",
  "middleName": "N",
  "firstName": "JIMMY",
  "displayName": "JIMMY N DARWENT",
  "genderCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd?depth=2",
        "rel": "children"
      }
    ],
    "genderCode": "M"
  },
  "generationSuffixCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd?depth=2",
        "rel": "children"
      }
    ]
  }
}
```

```

    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd",
      "rel": "self"
    }
  ],
  "generationSuffixCode": "I"
}
}

```

L'exemple suivant affiche les détails d'un enregistrement enfant dans l'objet de base genderCd.

```

{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/",
      "rel": "parent"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2",
      "rel": "self"
    }
  ],
  "rowidObject": "2",
  "label": "LU Gender",
  "genderDisp": "MALE",
  "genderCode": "M"
}

```

Créer un enregistrement

L'API REST Créer un enregistrement crée un enregistrement dans l'entité d'entreprise spécifiée. Envoyez les données de l'enregistrement dans le corps de demande. Utilisez l'API Promouvoir pour promouvoir et ajouter l'enregistrement dans l'entité d'entreprise.

L'API utilise la méthode POST pour créer un enregistrement.

URL de demande

L'URL Créer un enregistrement est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the
source system>

```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez la demande HTTP POST suivante à l'URL Créer un enregistrement :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of
the source system>

```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of
the source system>
Content-Type: application/<json/xml>

```

Paramètres d'URL

Le nom du système source est un paramètre requis dans l'URL de demande.

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans l'URL :

Paramètre	Description
systemName	Nom du système source.
interactionId	ID de l'interaction. Vous pouvez regrouper plusieurs demandes dans une interaction unique. Toutes les modifications sont effectuées avec l'ID d'interaction.
startDate et endDate	Spécifie la période pendant laquelle l'enregistrement est effectif. Indiquez ces paramètres pour un objet de base dont la chronologie est activée.
validateOnly	Indique si le service d'entité d'entreprise d'écriture valide les données entrantes. La valeur par défaut est False.
recordState	Etat de l'enregistrement. Utilisez le paramètre pour spécifier l'état initial de l'enregistrement. Utilisez ACTIVE ou PENDING. La valeur par défaut est ACTIVE.
taskComment	Ajoute un commentaire à la tâche de flux de travail déclenchée par l'API.
taskAttachments	Si le paramètre Joindre des fichiers à des tâches est activé, joignez un fichier à la tâche de flux de travail déclenchée par l'API.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Corps de demande

Envoyez les données de l'enregistrement dans le corps de demande REST. Utilisez le format JSON ou XML pour envoyer les données. Utilisez l'API Obtenir les métadonnées pour obtenir la structure de l'entité d'entreprise et fournir les valeurs de paramètres requises dans le corps de demande.

En-tête et corps de réponse

Lorsque la réponse a réussi, l'API renvoie l'ID d'interaction et l'ID de processus dans l'en-tête de réponse et les détails de l'enregistrement dans le corps de réponse.

Si le processus génère un ID d'interaction et s'en sert pour créer l'enregistrement, l'API renvoie l'ID d'interaction. Si le processus démarre un flux de travail au lieu d'enregistrer directement l'enregistrement dans la base de données, l'API renvoie l'ID du processus de flux de travail.

L'exemple suivant montre un en-tête de réponse avec un ID d'interaction et un ID de processus :

```
BES-interactionId: 72200000242000  
BES-processId: 15948
```

Le corps de réponse contient l'enregistrement avec les ID de ligne générés.

Exemple de demande d'API

L'exemple de demande suivant crée un enregistrement dans l'entité d'entreprise Personne. La demande ajoute un commentaire et une pièce jointe à la tâche de flux de travail déclenchée par l'API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
systemName=Admin&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9
Content-Type: application/json

{
  firstName: "John",
  lastName: "Smith",
  Phone: {
    item: [
      {
        phoneNumber: "111-11-11"
      }
    ]
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant montre l'en-tête et le corps de réponse après la création réussie d'un enregistrement :

```
BES-interactionId: 72200000242000
BES-processId: 15948
Content-Type: application/json
{
  "Person": {
    "key": {
      "rowid": "2198246",
      "sourceKey": "72200000241000"
    },
    "rowidObject": "2198246",
    "Phone": {
      "item": [
        {
          "key": {
            "rowid": "260961",
            "sourceKey": "72200000243000"
          },
          "rowidObject": "260961"
        }
      ]
    }
  }
}
```

Mettre à jour l'enregistrement

L'API REST Mettre à jour l'enregistrement met à jour l'enregistrement racine spécifié et ses enregistrements enfants. Envoyez l'ID de l'enregistrement dans l'URL de demande. Envoyez le résumé des modifications dans le corps de demande.

Après la modification, si l'état de l'enregistrement est En attente, utilisez l'API Promouvoir pour promouvoir les modifications. Par exemple, si la mise à jour a déclenché un flux de travail de vérification, l'enregistrement est en attente jusqu'à la fin de la vérification.

L'API utilise la méthode POST.

Pour les enregistrements racine, vous pouvez choisir d'utiliser la version PUT simplifiée dans laquelle le corps de la demande contient le champ modifié et non un résumé de la modification. Pour mettre à jour les enregistrements racine et les enregistrements enfants et petits-enfants associés, utilisez la version POST.

Vous avez également la possibilité d'utiliser la version PUT avec PKEY ou de spécifier l'élément rowidObject de l'enregistrement enfant.

URL de demande

L'URL Mettre à jour l'enregistrement est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?systemName=<name of the source system>
```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez la demande HTTP POST suivante à l'URL de l'option Mettre à jour l'enregistrement :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?systemName=<name of the source system>
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?systemName=<name of the source system>
Content-Type: application/<json/xml>
```

Paramètres de la requête

Le nom du système source est un paramètre de requête requis.

Vous pouvez utiliser les paramètres de requête suivants dans la demande :

Paramètre	Description
systemName	Nom du système source.
interactionId	ID de l'interaction. Vous pouvez regrouper plusieurs demandes dans une interaction unique. Toutes les modifications sont effectuées avec l'ID d'interaction.
startdate et enddate	Spécifie la période pendant laquelle l'enregistrement est effectif. Indiquez ces paramètres pour un objet de base dont la chronologie est activée.
validateOnly	Indique si le service d'entité d'entreprise d'écriture valide les données entrantes. La valeur par défaut est False.
recordState	Définit l'état de l'enregistrement. Utilisez ACTIF, EN ATTENTE ou SUPPRIME. Par exemple, si vous définissez recordState=ACTIVE et que la demande s'exécute sur un enregistrement supprimé de manière réversible, elle restaure l'enregistrement à l'état actif.
taskComment	Ajoute un commentaire à la tâche de flux de travail déclenchée par l'API.
taskAttachments	Si le paramètre Joindre des fichiers à des tâches est activé, joignez un fichier à la tâche de flux de travail déclenchée par l'API.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Corps de demande

Envoyez les données à mettre à jour dans le corps de demande REST. Utilisez le format JSON ou XML pour envoyer les données.

Indiquez les nouvelles valeurs de paramètres. Utilisez le paramètre \$original pour indiquer l'ancienne valeur d'un paramètre à mettre à jour.

Vous pouvez également utiliser les propriétés suivantes avec les enregistrements enfants :

Propriétés/ éléments	Type	Description
MATCH	objet	Si vous voulez ajouter ou supprimer des candidats de correspondance de la table de correspondance de l'enregistrement enfant, ajoutez l'objet MATCH à l'enregistrement enfant.
MERGE	objet	Si vous voulez fusionner des enregistrements enfants ou supprimer des candidats à la fusion, ajoutez l'objet MERGE à l'enregistrement enfant.

L'exemple de code JSON suivant change le prénom de l'enregistrement racine en Bob :

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
    firstName: "John"
  }
}
```

L'exemple de code JSON suivant supprime un candidat de correspondance d'un enregistrement enfant Adresse et définit la fusion de deux enregistrements enfants Téléphone :

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
    firstName: "John"
  }
  Address: {    // remove A3 from the matches for A2 in the Address_MTCH table
    item: [
      {
        rowidObject: "A2",
        MATCH: {
          item: [    // to remove matched child records for A2, specify null
            null
          ],
          $original: {
            item: [{key: {rowid: 'A3'}}]
          }
        }
      }
    ]
  }
  Telephone: {    // override the matches for the telephone child records
    item:[
      {
        rowid: "T1",
        MERGE: {
```

```

        item: [ // to remove merge candidates for T1, specify null
            null,
            null
        ],
        $original: {
            item: [
                {rowid: "T2"},
                {rowid: "T3"}
            ]
        }
    },
    {
        rowid: "T4",
        MERGE: {
            item: [ // to add or override matches, specify matched records
                {rowid: "T2"}
            ],
            $original: {
                item: [
                    null
                ]
            }
        }
    }
]
}

```

En-tête de réponse

Lorsque la réponse a réussi, l'API renvoie l'ID d'interaction et l'ID de processus dans l'en-tête de réponse et les détails de l'enregistrement dans le corps de réponse.

Si le processus génère un ID d'interaction et s'en sert pour créer l'enregistrement, l'API renvoie l'ID d'interaction. Si le processus démarre un flux de travail au lieu d'enregistrer directement l'enregistrement dans la base de données, l'API renvoie l'ID du processus de flux de travail.

L'exemple suivant montre un en-tête de réponse avec un ID d'interaction et un ID de processus :

```

BES-interactionId: 72200000242000
BES-processId: 15948

```

Le corps de réponse contient l'enregistrement avec les ID de ligne générés.

Exemple de demande d'API

L'exemple de demande suivant met à jour un enregistrement racine et son enregistrement enfant dans une entité d'entreprise. Personne est l'entité commerciale et Téléphone est un objet de base enfant. La demande ajoute un commentaire et une pièce jointe à la tâche de flux de travail déclenchée par l'API.

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?systemName=Admin
{
    rowidObject: "233",
    firstName: "BOB",
    lastName: "LLOYD",
    Phone: {
        item: [
            {
                rowidObject: "164",
                phoneNumber: "777-77-77",
                $original: {
                    phoneNumber: "(336) 366-4936"
                }
            }
        ]
    },
}

```

```

    $original: {
      firstName: "DUNN"
    }
  }
}

```

Exemple de réponse d'API

L'exemple de réponse suivant montre l'en-tête et le corps de réponse après la mise à jour réussie d'un enregistrement :

```

BES-interactionId: 72300000001000
BES-processId: 16302
{
  Person: {
    key: {
      rowid: "233",
      sourceKey: "SYS:233"
    },
    rowidObject: "233",
    preferredPhone: {
      key: {}
    }
  }
}

```

Supprimer un enregistrement

L'API REST Supprimer un enregistrement supprime un enregistrement racine dans une entité d'entreprise. Utilisez cette API pour supprimer les enregistrements enfants d'un enregistrement racine.

L'API utilise la méthode DELETE pour supprimer un enregistrement.

URL de demande

L'URL Supprimer un enregistrement est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the root
record>?systemName=Admin

```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez la demande HTTP DELETE suivante à l'URL Supprimer un enregistrement :

```

DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the
record>?systemName=Admin

```

Utilisez le format d'URL suivant pour supprimer un enregistrement enfant d'un enregistrement racine :

```

DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the
record>/<child base object>/<rowID of the child record>?systemName=Admin

```

Paramètre de requête

Le nom du système source est un paramètre d'URL requis. Utilisez le paramètre systemName pour spécifier le système source.

Exemple de demande d'API

L'exemple de demande suivant supprime un enregistrement racine de l'entité commerciale Person :

```

DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/292258?systemName=Admin

```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la réponse après la suppression réussie d'un enregistrement racine dans l'entité commerciale Person :

```
{
  "Person": {
    "key": {
      "rowid": "292258",
      "sourceKey": "WRK50000_7016"
    },
    "rowidObject": "292258"
  }
}
```

Lister l'enregistrement

L'API REST Lister l'enregistrement renvoie la liste des valeurs de recherche ou des valeurs de clé étrangère. Les recherches fournissent des données de référence et renvoient une liste de valeurs possibles pour une colonne donnée.

L'API utilise la méthode GET.

Utilisez cette API pour obtenir également les valeurs et les descriptions des codes de recherche. Vous pouvez spécifier l'ordre de tri des recherches. Utilisez la méthode POST lorsque vous avez besoin de paramètres supplémentaires ou plus complexes.

URL de demande

L'URL REST Répertoire l'enregistrement est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Utilisez le format d'URL suivant pour répertorier les codes des valeurs de recherche :

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list&idlabel=<lookup code>%3A<lookup display name>
```

Remarque: Utilisez l'API de métadonnées Get pour obtenir l'URL exacte permettant de répertorier les valeurs de recherche.

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les résultats.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
suppressLinks	Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur True pour supprimer tous les liens parent-enfant dans la réponse. La valeur par défaut est False. Par exemple, la requête <code>Person/1242?depth=10&suppressLinks=true</code> permet d'afficher les détails d'enregistrements jusqu'à 10 niveaux enfant, sans lien parent-enfant visible dans la réponse.
order	Permet de répertorier les valeurs de recherche par ordre croissant ou décroissant. Utilisez le caractère + comme préfixe pour spécifier l'ordre croissant et le caractère - pour spécifier l'ordre décroissant. Par défaut, si vous ne spécifiez pas de préfixe, l'ensemble de résultats est trié par ordre croissant. Par exemple, la requête <code>LUNamePrefix?action=list&order=-namePrefixDisp</code> répertorie les préfixes de nom qui sont triés selon les noms d'affichage par ordre décroissant.

Demande POST permettant de spécifier l'ordre de tri

Une demande POST vous permet de spécifier l'ordre de tri des valeurs de recherche. Incluez les paramètres ou les champs dans le corps de la demande POST.

L'exemple suivant montre comment utiliser une demande POST dans une opération de liste :

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/ListCO.json
{
  parameters:
  {
    coFilter: {
      object: {
        name:"LUCountry",
        order: "-countryNameDisp"
      }
    }
  }
}
```

Exemple de demande d'API

L'exemple de demande suivant liste les valeurs de recherche :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?action=list
```

L'exemple de demande suivant répertorie les codes des valeurs de recherche de genre :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?
action=list&idlabel=genderCode%3AgenderDisp
```

L'exemple de demande suivant répertorie les préfixes des noms, avec les noms d'affichage des préfixes par ordre décroissant :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUNamePrefix?action=list&order=-
namePrefixDisp
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des valeurs de recherche :

```
{
  "firstRecord": 1,
  "pageSize": 2147483647,
  "searchToken": "SVR1.AU5LC",
  "item": [
    {
      "rowidObject": "1",
      "genderDisp": "UNKNOWN",
      "genderCode": "N"
    },
    {
      "rowidObject": "2",
      "genderDisp": "MALE",
      "genderCode": "M"
    },
    {
      "rowidObject": "3",
      "genderDisp": "FEMALE",
      "genderCode": "F"
    }
  ]
}
```

L'exemple de réponse suivant affiche le code des valeurs de recherche :

```
{
  "item": [
    {
      "id": "F",
      "label": "FEMALE"
    },
    {
      "id": "M",
      "label": "MALE"
    },
    {
      "id": "N",
      "label": "UNKNOWN"
    }
  ],
  "firstRecord": 1,
  "recordCount": 0,
  "pageSize": 2147483647,
  "searchToken": "SVR1.AU5LD"
}
```

Rechercher un enregistrement

L'API REST Rechercher un enregistrement recherche des valeurs indexées dans un enregistrement racine de recherche et dans tous ses enregistrements enfants. Vous pouvez utiliser des filtres et des facettes pour afficher un sous-ensemble des résultats de la recherche. Les facettes groupent les résultats de la recherche en catégories, alors que les filtres affinent les résultats de la recherche. L'API effectue une recherche dans tous les champs configurés comme champs de recherche et renvoie les enregistrements correspondant aux critères de recherche.

L'API utilise la méthode GET pour effectuer une recherche dans les index des champs de recherche.

URL de demande

L'URL Rechercher un enregistrement est aux formats suivants :

URL pour une recherche simple

Utilisez l'URL suivante pour une recherche simple :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

Effectuez la demande HTTP GET suivante à l'URL Rechercher un enregistrement :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

URL pour une recherche de champ

Utilisez l'URL suivante pour une recherche de champ :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>=<'<business entity field value>'>
```

Si vous spécifiez une valeur négative numérique pour un champ d'entité d'entreprise, tel que -120, le classement des enregistrements renvoyés est susceptible d'être affecté.

URL pour une recherche avec facettes

Utilisez l'URL suivante pour une recherche simple avec facettes :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&facets=<field name>
```

Utilisez l'URL suivante pour une recherche de champ avec facettes :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>=<'<business entity field value>'>&facets=<field name>
```

URL pour une recherche avec filtres

Utilisez l'URL suivante pour une recherche simple avec filtres :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&filters=<field name1>=<field value1> AND <field name2>=<field value2> ...
```

Utilisez le paramètre q ou fq dans la recherche.

Codage d'URL

Utilisez le codage d'URL sachant que l'URL inclut des caractères tels que des espaces et des guillemets simples.

L'exemple suivant montre la représentation codée de l'URL Rechercher un enregistrement :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<field name>%3D%27<value of the field>
```


Paramètres de la requête

Utilisez les paramètres de requête `q` ou `fq` pour indiquer la valeur de chaîne pour la recherche. Les paramètres de requête `q` et `fq` s'excluent mutuellement. Utilisez le paramètre `fq` pour une recherche de champ. Utilisez l'opérateur logique AND pour plusieurs conditions.

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans l'URL :

Paramètre	Description
<code>q</code>	<p>Spécifie la valeur de chaîne ou le terme de recherche. La requête recherche les occurrences du terme de recherche partout dans l'enregistrement. Utilisé dans une recherche simple.</p> <p>Par exemple, la requête <code>Person?q=STEVE</code> recherche les enregistrements contenant le terme STEVE.</p> <p>Pour rechercher simultanément deux termes ou plus, placez-les entre guillemets doubles. Utilisez le caractère <code>+</code> avant chaque terme si vous voulez que les résultats de la recherche contiennent le terme. Si la valeur de champ contient un espace, placez-la entre guillemets simples.</p> <p>Utilisez la requête suivante pour rechercher une correspondance exacte avec WILLIAM JOHN LAWSON :</p> <pre>Person?q="WILLIAM JOHN LAWSON"</pre> <p>Utilisez la requête suivante pour rechercher WILLIAM, JOHN ou LAWSON :</p> <pre>Person?q=WILLIAM JOHN LAWSON</pre> <p>Utilisez la requête suivante pour rechercher WILLIAM, JOHN et LAWSON :</p> <pre>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</pre>
<code>fq</code>	<p>Spécifie la valeur de chaîne ou le terme de recherche dans un champ particulier. La requête recherche le terme uniquement dans cette partie de l'enregistrement. Utilisé dans une recherche ciblée basée sur des champs indexés.</p> <p>Par exemple, la requête <code>Person?fq=displayname=STEVE</code> recherche les enregistrements contenant le nom d'affichage STEVE.</p>
<code>facets</code>	<p>Spécifie les champs qui doivent être traités comme facettes ou les catégories utilisées pour grouper les résultats de la recherche. Spécifiez uniquement les champs de recherche. Utilisé avec les paramètres <code>q</code> et <code>fq</code>. La syntaxe est la suivante :</p> <pre>&facets=FieldName1,FieldName2,FieldNameN.</pre> <p>Par exemple, la requête <code>Person?q=STEVE&facets=department</code> recherche les personnes dont le nom d'affichage est STEVE et groupe les résultats de la recherche par service. La recherche affiche les enregistrements des personnes dont le nom d'affichage est STEVE et ces enregistrements sont groupés par service.</p>
<code>filters</code>	<p>Spécifie les champs utilisés pour affiner les résultats de la recherche. Spécifiez uniquement les champs filtrables. Utilisé avec les paramètres <code>q</code> et <code>fq</code>.</p> <p>Par exemple, la requête <code>Person?fq=STEVE&filters=birthdate='1980-11-27T08:00:00Z'</code> recherche les personnes dont le nom d'affichage est STEVE et filtre les résultats de la recherche par date de naissance. La recherche affiche les enregistrements des personnes dont le nom d'affichage est STEVE et dont la date de naissance est le 27 novembre 1980.</p> <p>Remarque: Spécifiez une date entre guillemets simples.</p>

Paramètre	Description
depth	<p>Spécifie le nombre de niveaux enfants à renvoyer. Indiquez 2 pour renvoyer le nœud racine et ses enfants directs et 3 pour renvoyer le nœud racine, ses enfants directs et ses petits-enfants. Indiquez 1 pour renvoyer le nœud racine uniquement. Par défaut, aucun niveau n'est spécifié.</p> <p>Si aucun niveau n'est spécifié, les résultats de recherche renvoient le nœud racine et les enfants dans lesquels une correspondance du terme de recherche est trouvée.</p> <p>Par exemple, la requête <code>Person?q=STEVE&depth=2</code> recherche les enregistrements contenant le terme STEVE et renvoie des informations sur l'enregistrement racine et ses enfants directs.</p>
queryOperator	<p>Indique si la recherche doit trouver l'une des chaînes contenue dans le terme de recherche ou toutes les chaînes contenues dans le terme de recherche.</p> <p>Ce paramètre prend l'une des valeurs suivantes :</p> <ul style="list-style-type: none"> - OR. Recherche n'importe quelle chaîne indiquée dans le paramètre <code>f</code> ou <code>fq</code>. - AND. Recherche toutes les chaînes indiquées dans le paramètre <code>f</code> ou <code>fq</code>. <p>Si vous ne spécifiez pas ce paramètre, la valeur par défaut est OR.</p> <p>Par exemple, la requête <code>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</code> recherche les enregistrements contenant WILLIAM, JOHN et LAWSON.</p>
suppressLinks	<p>Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur True pour supprimer tous les liens parent-enfant dans la réponse. La valeur par défaut est False.</p> <p>Par exemple, la requête <code>Person?q=STEVE&suppressLinks=true</code> recherche les enregistrements contenant le terme STEVE et renvoie la réponse dans laquelle aucun lien parent-enfant n'est visible.</p>
readSystemFields	<p>Indique si les champs système doivent être renvoyés dans le résultat. La valeur par défaut est False.</p>
ordre	<p>Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +.</p> <p>Si vous voulez faire appel à un champ enfant pour trier les résultats, utiliser le nom complet du champ. Par exemple, <code>BillAddresses.Address.cityName</code>.</p> <p>Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis selon le suivant. Par exemple, la requête <code>Person?order=displayName,-BillAddresses.Address.cityName</code> trie les résultats par nom d'affichage par ordre croissant, puis par nom de ville par ordre décroissant.</p>
maxRecordsToSort	<p>Nombre maximal de résultats de recherche à trier. La valeur par défaut est 1000.</p>

Spécifiez une plage avec le paramètre `filters` :

Vous pouvez utiliser le paramètre `filters` pour affiner les résultats de la recherche dans une plage spécifiée. Vous pouvez spécifier la plage pour les champs filtrables des types de données numériques et de date.

Utilisez le format suivant pour le type de données entières :

```
fieldName=[fromValue,toValue]
```

La plage est générée entre `fromValue` et `toValue`. Vérifiez que `fromValue` est inférieure à `toValue`. Par exemple, la requête `filters=age=[35,45]` affine les résultats de la recherche et recherche les enregistrements dans le groupe d'âge 35-45.

Utilisez le format suivant pour le type de données de date :

```
fieldName=[fromDate,toDate]
```

La plage est générée entre fromDate et toDate. Par exemple, la requête

`filters=birthdate=[2000-06-12T12:30:00Z,2015-06-12T12:30:00Z]` spécifie la date de naissance entre le 12 juin 2000 et le 12 juin 2015.

Remarque: Lorsque vous spécifiez un filtre de correspondance exacte de date, placez-le entre guillemets simples. Lorsque vous spécifiez une plage de dates, n'utilisez pas de guillemets.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

Demande avec le paramètre q

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom STEVE.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?q=STEVE
```

Demande avec le paramètre fq

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom d'affichage STEVE. Le champ Nom d'affichage est un champ indexé.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE
```

Demande avec option de tri

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Personne contenant le nom d'affichage STEVE et trie les résultats par ville par ordre croissant :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&order=BillAddresses.Address.cityName
```

Demande avec le paramètre fq et l'opérateur logique AND

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom d'affichage STEVE et l'ID de taxe DM106 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE AND taxId=DM106
```

Demande avec une facette

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom d'affichage STEVE et affine les résultats en les groupant par service :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&facets=department
```

Demande avec un filtre (filtre exact)

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom d'affichage STEVE et les filtre par la ville et le pays spécifiés :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&filters=cityName=Canberra AND country=Australia
```

Demande avec une plage de filtres

L'exemple de demande suivant recherche des enregistrements dans l'entité d'entreprise Person contenant le nom d'affichage STEVE et les filtre par le groupe d'âge 35-45 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&filters=age=[35,45] AND cityName=Canberra
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche le résultat d'une recherche par le nom Steve :

```
{
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 10,
  "item": [
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
Person/1443",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
Person/1443?depth=2",
            "rel": "children"
          }
        ],
        "rowidObject": "1443",
        "label": "CRAIG,STEVE",
        "partyType": "Person",
        "lastName": "CRAIG",
        "firstName": "STEVE",
        "taxID": "stevecraig",
        "displayName": "STEVE CRAIG"
      }
    },
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
Person/285",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
Person/285?depth=2",
            "rel": "children"
          }
        ],
        "rowidObject": "285",
        "label": "PEARSON,STEVE",
        "partyType": "Person",
        "lastName": "PEARSON",
        "firstName": "STEVE",
        "displayName": "STEVE PEARSON"
      }
    }
  ]
}
```

Générateur de suggestions

L'API REST Générateur de suggestions renvoie une liste de termes apparentés pour une chaîne de recherche, basée sur les données présentes dans votre base de données. Utilisez l'API pour accepter les caractères que vous saisissez dans un champ d'interface utilisateur et renvoyer des suggestions de remplissage automatique. Vous pouvez rechercher et sélectionner la chaîne dans la liste de suggestions. Utilisez l'API Générateur de suggestions pour les champs de recherche.

L'API utilise la méthode GET.

Remarque: Pour utiliser l'API afin de fournir une liste de suggestions de remplissage automatique pour un champ de recherche, définissez la propriété Générateur de suggestions du champ sur True et créez un nouvel index pour les données.

URL de demande

L'URL REST Générateur de suggestions est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Paramètres de la requête

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
suggest	Requis. Spécifie la chaîne pour laquelle vous recherchez des suggestions.
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
buildIndex	Facultatif. Indique si l'index doit être créé. Lorsque vous redémarrez le système, définissez ce paramètre sur True pour créer explicitement l'index pour la collection. Ce paramètre peut être obsolète dans une version ultérieure.

Exemple de demande d'API

L'exemple de demande suivant renvoie une liste de suggestions que vous pouvez utiliser dans une interface utilisateur :

```
GET http://localhost:8080/cmx/cs/localhost-inf-a-DS_UI1/Person.json?suggest=Abhinav
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des suggestions :

```
{
  term: [2]
  "abhinav goel"
  "abhinav gupta"
}
```

SearchQuery

L'API REST SearchQuery recherche des enregistrements qui sont une correspondance exacte des valeurs de champ spécifiées dans une requête. Vous pouvez utiliser l'API SearchQuery pour récupérer tous les enregistrements d'une entité d'entreprise spécifique ou pour récupérer des enregistrements en fonction des valeurs de champ spécifiques. Contrairement à l'API Rechercher des enregistrements qui recherche les valeurs spécifiées dans tous les champs de recherche, l'API SearchQuery recherche les valeurs spécifiées dans les champs spécifiques.

Vous pouvez filtrer les résultats de la requête pour afficher les valeurs spécifiques dans un enregistrement d'entité d'entreprise racine et dans les enregistrements enfants. Vous pouvez utiliser les opérateurs AND, IN et RANGE dans la requête.

Pour inclure des champs spécifiques dans les résultats de la requête, spécifiez les champs ou une vue Entité d'entreprise qui comprend les champs. Vous pouvez trier les résultats de la requête pour afficher les enregistrements par ordre croissant ou décroissant.

L'API utilise la méthode GET.

URL de demande

Le contexte de l'URL de SearchQuery est `cmx/cs`. Dans un environnement MDM hébergé, incluez le nom du locataire dans le contexte, tel que `<tenant name>/cmx/cs`.

L'URL de SearchQuery est aux formats suivants :

URL pour renvoyer tous les enregistrements d'un type d'entité d'entreprise spécifique

Utilisez l'URL suivante pour rechercher tous les enregistrements du type d'entité d'entreprise spécifié :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=query
```

Effectuez la demande HTTP GET suivante à l'URL de SearchQuery :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?action=query
```

URL pour renvoyer tous les détails des enregistrements qui correspondent aux valeurs de champ spécifiées

Utilisez l'URL suivante pour rechercher des enregistrements qui correspondent aux valeurs de champ que vous spécifiez :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=query&filter=<business entity field name 1>='<business entity field value 1>'
AND <business entity field name 2>='<business entity field value 2>'...AND <business
entity field name n>='<business entity field value n>'
```

URL pour renvoyer les détails spécifiques des enregistrements qui correspondent aux valeurs de champ spécifiées

Utilisez l'URL suivante pour rechercher des enregistrements et afficher des champs d'enregistrements spécifiques dans les résultats de la recherche :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=query&filter=<business entity field name 1>='<business entity field value 1>'
AND <business entity field name 2>='<business entity field value 2>'...AND <business
entity field name n>='<business entity field value n>'&outputView=<business entity
view>
```

Paramètres de la requête

Définissez la requête sous la forme d'une liste de paires champ-valeur.

Le tableau suivant décrit les paramètres de la requête que vous pouvez utiliser dans l'URL :

Paramètre	Description
action	<p>Requis. Renvoie tous les enregistrements du type d'entité d'entreprise spécifié dans les résultats de la requête. Définissez le paramètre sur <code>query</code> et utilisez-le avec le paramètre <code>filter</code>. En cas d'utilisation sans le paramètre <code>filter</code>, la requête recherche tous les enregistrements du type d'entité d'entreprise spécifié.</p> <p>Par exemple, utilisez la requête suivante pour rechercher tous les enregistrements de l'entité d'entreprise Personne :</p> <pre>Person?action=query</pre>
filter	<p>Requis. Spécifie une liste de paires champ-valeur séparées par des opérateurs. Les opérateurs valides sont AND, IN et RANGE.</p> <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements Personne contenant le prénom STEVE et le nom SMITH :</p> <pre>Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'</pre>
depth	<p>Spécifie le nombre de niveaux d'enregistrements enfants à renvoyer. Par exemple, vous pouvez spécifier les niveaux suivants :</p> <ul style="list-style-type: none">- 1. Renvoie l'enregistrement racine.- 2. Renvoie l'enregistrement racine et ses enregistrements enfants directs.- 3. Renvoie l'enregistrement racine, les enregistrements enfants directs et les enregistrements petits-enfants. <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et renvoyez des informations sur l'enregistrement racine et ses enregistrements enfants directs :</p> <pre>Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'&depth=2</pre>
suppressLinks	<p>Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur True pour supprimer tous les liens parent-enfant dans la réponse. La valeur par défaut est False.</p> <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et renvoyez une réponse dans laquelle aucun lien parent-enfant n'est visible :</p> <pre>Person?action=query&filter=firstName='STEVE'&suppressLinks=true</pre>
readSystemFields	<p>Indique si les champs système doivent être renvoyés dans le résultat. La valeur par défaut est False.</p>
fields	<p>Spécifie les champs à afficher dans les résultats de la requête.</p>
outputView	<p>Spécifie la vue Entité d'entreprise à utiliser pour afficher les résultats de la requête. Lorsque vous configurez la vue Entité d'entreprise pour les résultats de la requête, incluez les champs à afficher dans les résultats de la requête.</p>
order	<p>Spécifie l'ordre de tri des résultats de la requête. Utilisez le caractère plus (+) comme préfixe pour spécifier l'ordre croissant et le caractère moins (-) comme préfixe pour spécifier l'ordre décroissant. Par défaut, le résultat de la requête est dans l'ordre croissant.</p> <p>Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est classé selon le premier paramètre de la liste, puis selon le suivant.</p>

Vous pouvez utiliser les opérateurs suivants dans le paramètre filter :

AND

Recherche les enregistrements avec toutes les valeurs de champ répertoriées dans le paramètre filter.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et le nom SMITH :

```
Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'
```

IN

Recherche les enregistrements avec l'une des valeurs répertoriées dans le paramètre filter.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE ou JOHN :

```
Person?action=query&filter=firstName IN [STEVE,JOHN]
```

RANGE

Recherche les enregistrements dans une plage spécifiée. Vous pouvez spécifier une plage pour les champs des types de données numeric et date.

Utilisez le format suivant pour le type de données entières :

```
<business entity field name>=[fromValue,toValue]
```

La plage est générée entre fromValue et toValue. Vérifiez que fromValue est inférieure à toValue.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements dans la tranche d'âge 35 à 45 :

```
Person?action=query&filter=firstName IN [STEVE,JOHN] AND age=[35,45]
```

Utilisez le format suivant pour le type de données date :

```
<business entity field name>=[fromDate,toDate]
```

La plage se situe entre fromDate et toDate.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant la date de naissance entre le 12 juin 2000 et le 12 juin 2015 :

```
Person?action=query&filter=birthDate=[2000-06-12T12:30:00Z,2015-06-12T12:30:00Z]
```

Exemple de demande d'API

L'exemple de demande suivant recherche dans l'entité d'entreprise Personne les enregistrements contenant le prénom STEVE et le nom SMITH :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
action=query&filter=firstName='STEVE' AND lastName='SMITH'&outputView=PersonAddressView
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les résultats d'une requête pour les enregistrements Personne contenant le prénom STEVE et le nom SMITH :

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1T8UU",
  "facet": [],
  "item": [
    {
      "Person": {
```



```

        "rowidObject": "268",
        "label": "Person: SMITH, STEVE,268",
        "partyType": "Person",
        "lastName": "SMITH",
        "displayName": "STEVE SMITH",
        "firstName": "STEVE"
    }
},
{
    "Person": {
        "rowidObject": "448",
        "label": "Person: SMITH, STEVE,448",
        "partyType": "Person",
        "lastName": "SMITH",
        "displayName": "STEVE SMITH",
        "firstName": "STEVE"
    }
}
]
}

```

Exportation des résultats de SearchQuery dans un fichier CSV

Pour exporter les résultats d'une demande de SearchQuery dans un fichier CSV, spécifiez le nom de l'entité d'entreprise en tant que fichier .CSV dans le chemin de l'URL de la demande. Vous pouvez utiliser tous les paramètres de la requête dans l'URL de la demande.

Par exemple, utilisez l'URL de la demande suivante afin d'exporter les résultats d'une recherche pour les enregistrements qui correspondent aux valeurs de champ que vous spécifiez :

```

http://<host>:<port>/<context>/<database ID>/<business entity>.CSV?
action=query&filter=<business entity field name 1>='<business entity field value 1>' AND
<business entity field name 2>='<business entity field value 2>'...AND <business entity
field name n>='<business entity field value n>'

```

Exemple de demande d'API

L'exemple de demande suivant recherche les enregistrements contenant le prénom STEVE et le nom SMITH et renvoie les résultats de la requête au format CSV :

```

GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person.CSV?
action=query&filter=firstName='STEVE' AND
lastName='SMITH'&fields=recordID,firstName,lastName

```

Exemple de réponse d'API

L'exemple suivant indique les résultats de la requête au format CSV pour une requête contenant le prénom STEVE et le nom SMITH :

```

recordID,firstName,lastName
00023,Steve,Smith
00048,Steve,Smith

```

SearchMatch

L'API REST SearchMatch recherche des enregistrements qui sont des correspondances approximatives. L'API identifie les enregistrements correspondants selon les champs de l'entité d'entreprise spécifique, qui sont configurés comme colonnes de correspondance dans MDM Hub. Avant d'utiliser l'API SearchMatch, utilisez l'API Répertoire des colonnes de correspondance pour identifier les colonnes de correspondance d'une entité d'entreprise.

Vous pouvez éventuellement spécifier un ensemble de règles de correspondance au lieu de colonnes de correspondance. Pour utiliser un ensemble de règles de correspondance pour la requête, veillez à ce que l'option Activer la recherche par règles soit activée pour cet ensemble. Pour plus d'informations sur la

configuration des colonnes de correspondance et des ensembles de règles de correspondance, consultez le *Guide de configuration de MDM Multidomain*.

Vous pouvez utiliser les opérateurs AND, IN et RANGE dans la requête.

Pour inclure des champs spécifiques dans les résultats de la requête, spécifiez les champs ou une vue Entité d'entreprise qui comprend les champs. Vous pouvez spécifier l'ordre de tri pour afficher les enregistrements dans les résultats de la requête par ordre croissant ou décroissant.

L'API utilise la méthode GET pour interroger les champs de l'entité d'entreprise et renvoie les enregistrements qui sont des correspondances approximatives avec leurs scores de correspondance et règles de correspondance associées.

URL de demande

Le contexte de l'URL de SearchMatch est cmx/cs. Dans un environnement MDM hébergé, incluez le nom du locataire dans le contexte, tel que <tenant name>/cmx/cs.

L'URL de SearchMatch est aux formats suivants :

URL pour renvoyer les enregistrements correspondants selon les valeurs des champs spécifiques configurés comme colonnes de correspondance

Utilisez l'URL suivante pour rechercher des enregistrements qui correspondent aux valeurs de champ que vous spécifiez :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>='<business entity field
value 1>',<business entity field name 2>='<business entity field value
2>',...<business entity field name n>='<business entity field value n>'
```

Effectuez la demande HTTP GET suivante à l'URL de SearchMatch :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>='<business entity field
value 1>',<business entity field name 2>='<business entity field value
2>',...<business entity field name n>='<business entity field value n>'
```

URL pour renvoyer les enregistrements correspondants selon un ensemble de règles de correspondance

Utilisez l'URL suivante pour rechercher des enregistrements correspondants selon un ensemble de règles de correspondance que vous spécifiez :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>='<business entity field
value 1>',<business entity field name 2>='<business entity field value
2>',...<business entity field name n>='<business entity field value
n>'&matchRuleSet=<match rule set name>
```

Paramètres de la requête

Utilisez le paramètre `fuzzyFilter` pour spécifier les valeurs de champ pour lesquels effectuer une requête.

Utilisez le paramètre `fuzzyFilter` avec le paramètre `action`.

Le tableau suivant décrit les paramètres de la requête que vous pouvez utiliser dans l'URL :

Paramètre	Description
action	<p>Requis. Renvoie les enregistrements correspondants de l'entité d'entreprise spécifiée. Définissez le paramètre sur <code>match</code> et utilisez-le avec le paramètre <code>fuzzyFilter</code>.</p> <p>Par exemple, utilisez la requête suivante pour rechercher une personne contenant le prénom STEVE :</p> <pre>Person?action=match&fuzzyFilter=STEVE</pre>
fuzzyFilter	<p>Requis. Spécifie une liste de paires nom de champ et valeur de champ séparée par des virgules à utiliser pour rechercher les enregistrements d'un type d'entité d'entreprise spécifique.</p> <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE, qui possède une adresse à Toronto :</p> <pre>Person?action=match&fuzzyFilter=firstName=STEVE,Address.Address.City=TORONTO</pre>
matchRuleSet	<p>Spécifie un ensemble de règles de correspondance selon lequel identifier les enregistrements correspondants.</p> <p>Si vous ne possédez aucun ensemble de règles de correspondance spécifique, spécifiez <code>NONE</code>. Les règles de correspondance automatique et manuelle sont utilisées.</p>
filter	<p>Spécifie les valeurs de champ à utiliser pour filtrer les résultats d'une recherche approximative.</p> <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE, qui habite à Toronto :</p> <pre>Person? action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=city=Toronto</pre>
depth	<p>Spécifie le nombre de niveaux d'enregistrements enfants à renvoyer. Par exemple, vous pouvez spécifier les niveaux suivants :</p> <ul style="list-style-type: none">- 1. Renvoie l'enregistrement racine.- 2. Renvoie l'enregistrement racine et ses enregistrements enfants directs.- 3. Renvoie l'enregistrement racine, les enregistrements enfants directs et les enregistrements petits-enfants. <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et renvoyez des informations sur l'enregistrement racine et ses enregistrements enfants directs :</p> <pre>Person?action=match&fuzzyFilter=firstName='STEVE'&filter=city=Toronto</pre>
suppressLinks	<p>Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur <code>True</code> pour supprimer tous les liens parent-enfant dans la réponse. La valeur par défaut est <code>False</code>.</p> <p>Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et renvoyez une réponse dans laquelle aucun lien parent-enfant n'est visible :</p> <pre>Person?action=match&fuzzyFilter=firstName='STEVE'&suppressLinks=true</pre>
readSystemFields	<p>Indique si les champs système doivent être renvoyés dans le résultat. La valeur par défaut est <code>False</code>.</p>

Paramètre	Description
fields	Spécifie les champs à afficher dans les résultats de la requête.
outputView	Spécifie la vue Entité d'entreprise à utiliser pour afficher les résultats de la requête. Lorsque vous configurez la vue Entité d'entreprise pour les résultats de la requête, incluez les champs à afficher dans les résultats de la requête.

Vous pouvez utiliser les opérateurs suivants dans le paramètre filter :

AND

Recherche les enregistrements avec toutes les valeurs de champ répertoriées dans le paramètre filter.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE et le nom SMITH :

```
Person?
action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=city=Toronto AND
gender=Male
```

IN

Recherche les enregistrements avec l'une des valeurs répertoriées dans le paramètre filter.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant le prénom STEVE ou le nom JOHN, qui habite à Toronto ou à Ottawa :

```
Person?action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=city in
[Toronto,Ottawa]
```

RANGE

Recherche les enregistrements dans une plage spécifiée. Vous pouvez spécifier une plage pour les champs des types de données numeric et date.

Utilisez le format suivant pour le type de données entières :

```
<business entity field name>=[fromValue,toValue]
```

La plage est générée entre fromValue et toValue. Vérifiez que fromValue est inférieure à toValue.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements dans la tranche d'âge 35 à 45 :

```
Person?action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=age=[35,45]
```

Utilisez le format suivant pour le type de données date :

```
<business entity field name>=[fromDate,toDate]
```

La plage se situe entre fromDate et toDate.

Par exemple, utilisez la requête suivante pour rechercher des enregistrements contenant la date de naissance entre le 12 juin 2000 et le 12 juin 2015 :

```
Person?
action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=birthDate=[2000-06
-12T12:30:00Z,2015-06-12T12:30:00Z]
```

Exemple de demande d'API

L'exemple de demande suivant recherche dans l'entité d'entreprise Personne les enregistrements contenant le prénom STEVE en utilisant l'ensemble de règles de correspondance IDL2 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
action=match&fuzzyFilter=firstName=STEVE&matchRuleSet=IDL2
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les résultats d'une requête pour les enregistrements contenant le prénom STEVE selon l'ensemble de règles de correspondance IDL2 :

```
{
  "link": [],
  "firstRecord": 1,
  "recordCount": 3,
  "pageSize": 10,
  "searchToken": "SVR1.17LJ2",
  "matchedEntity": [
    {
      "businessEntity": {
        "Person": {
          "rowidObject": "145",
          "label": "SAMUEL,STEVE",
          "partyType": "Person",
          "lastName": "SAMUEL",
          "displayName": "MR STEVE SAMUEL",
          "statusCd": "A",
          "firstName": "STEVE",
          "genderCd": {
            "genderCode": "M"
          },
          "namePrefixCd": {
            "namePrefixCode": "MR"
          }
        }
      },
      "matchRule": "IDL2|1",
      "matchScore": "93",
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
          "rel": "matchRule"
        }
      ]
    },
    {
      "businessEntity": {
        "Person": {
          "rowidObject": "268",
          "label": "SMITH,STEVE",
          "partyType": "Person",
          "lastName": "SMITH",
          "displayName": "STEVE SMITH",
          "firstName": "SAM"
        }
      },
      "matchRule": "IDL2|1",
      "matchScore": "98",
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
          "rel": "matchRule"
        }
      ]
    }
  ]
}
```

```

{
  "businessEntity": {
    "Person": {
      "rowidObject": "448",
      "label": "SMITH,STEVEN",
      "partyType": "Person",
      "lastName": "SMITH",
      "displayName": "SAM STEVEN",
      "firstName": "STEVEN"
    }
  },
  "matchRule": "IDL2|1",
  "matchScore": "98",
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
      "rel": "matchRule"
    }
  ]
},
"facet": []
}

```

Exportation des résultats de SearchMatch dans un fichier CSV

Pour exporter les résultats d'une demande de SearchMatch en tant que fichier CSV, spécifiez le nom de l'entité d'entreprise en tant que fichier .CSV dans le chemin de l'URL de la demande. Vous pouvez utiliser tous les paramètres de la requête dans l'URL de la demande.

Par exemple, utilisez l'URL de la demande suivante pour exporter les résultats d'une recherche pour les enregistrements correspondants en fonction d'un ensemble de règles de correspondance :

```

http://<host>:<port>/<context>/<database ID>/<business entity>.CSV?
action=match&fuzzyFilter=<business entity field name 1>=<business entity field value 1>,<business entity field name 2>=<business entity field value 2>,...<business entity field name n>=<business entity field value n>&matchRuleSet=<match rule set name>

```

Exemple de demande d'API

L'exemple de demande suivant recherche les enregistrements qui correspondent au prénom STEVE et au nom SMITH et renvoie les résultats de la requête au format CSV :

```

GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person.CSV?
action=match&fuzzyFilter=firstName=STEVE,lastName=SMITH&fields=recordID,firstName,lastName

```

Exemple de réponse d'API

L'exemple suivant indique les résultats de la requête pour les enregistrements qui correspondent au prénom STEVE et au nom SMITH au format CSV :

```

recordID,firstName,lastName
00023,Steve,Smith
00035,Steven,Smith
00048,Steve,Smith
00079,Steve,Smithson

```

Obtenir les métadonnées BPM

L'API REST Obtenir les métadonnées BPM renvoie les types de tâches et deux indicateurs qui indiquent si l'outil de flux de travail BPM peut exécuter le service Obtenir le lignage des tâches ainsi que les services d'administration.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les métadonnées BPM est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les métadonnées BPM :

```
GET http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Exemple de demande d'API

L'exemple de demande suivant renvoie des informations sur les types de tâches et l'outil de flux de travail BPM :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/BPMMetadata
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les types de tâches et la valeur des deux indicateurs pour l'outil de flux de travail BPM :

```
{
  "parameters": {
    "doesSupportAdministration": true,
    "doesSupportLineage": true,
    "doesSupportAttachments": true,
    "maximumAttachmentFileSizeInMb": 20,
    "taskTypes": {
      "taskTypes": [
        {
          "name": "Merge",
          "label": "Merge"
        },
        {
          "name": "FinalReview",
          "label": "FinalReview"
        },
        {
          "name": "Update",
          "label": "Update"
        },
        {
          "name": "Notification",
          "label": "Notification"
        },
        {
          "name": "ReviewNoApprove",
          "label": "ReviewNoApprove"
        },
        {
          "name": "Unmerge",
          "label": "Unmerge"
        }
      ]
    }
  }
}
```

Lister les tâches

L'API Lister les tâches renvoie une liste de tâches de flux de travail. Un flux de travail définit les activités d'un processus d'entreprise ainsi que les chemins d'exécution entre les activités. Chaque activité est appelée une tâche.

L'API utilise la méthode GET pour renvoyer une liste de tâches triée et paginée.

URL de demande

L'URL Lister les tâches est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/task
```

Effectuez la demande HTTP GET suivante à l'URL Lister les tâches :

```
GET http://<host>:<port>/<context>/<database ID>/task
```

Vous pouvez ajouter des en-têtes HTTP à la demande.

Paramètres de la requête

Utilisez les champs de données de tâche comme paramètres de la requête pour filtrer la liste des tâches.

Vous pouvez utiliser les paramètres de requête suivants :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Utilisez l'attribut de nom pour spécifier le type de tâche. Pour plus d'informations sur les types de tâches, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
taskId	ID de la tâche.
processId	ID du processus de flux de travail qui contient la tâche.
owner	Utilisateur qui exécute la tâche.
title	Brève description de la tâche.
status	Statut de la tâche dans le flux de travail. Utilisez une des deux valeurs suivantes : <ul style="list-style-type: none">- Ouvert : la tâche n'a pas démarré ou est en cours.- Fermé : la tâche est terminée ou est annulée.
priority	Niveau d'importance de la tâche. Utilisez l'une des valeurs suivantes : élevé, normal et faible.
creator	Utilisateur qui crée la tâche.
createDateBefore et createDateAfter	Plage de dates : Vous pouvez filtrer les tâches selon le champ createDate.
dueDateBefore et dueDateAfter	Plage de dates : Vous pouvez filtrer les tâches selon le champ dueDate.

Paramètre	Description
scope	Affiche les tâches disponibles en fonction de votre rôle utilisateur. Utilisez la valeur <code>admin</code> pour filtrer la liste des tâches en fonction de votre rôle. Remarque: Les utilisateurs dont le rôle Administrateur de tâches a été attribué peuvent gérer toutes les tâches. Les utilisateurs qui ont reçu d'autres rôles utilisateur ne peuvent quant à eux gérer que les tâches à la disposition de leur rôle.
q	Spécifie le terme de recherche. La requête recherche les occurrences du terme de recherche dans des tâches liées aux enregistrements d'entité d'entreprise. Par exemple, la requête <code>q=KAREN</code> recherche des tâches comportant le terme de recherche KAREN.

Utilisez les paramètres de la requête comme paires nom-valeur dans l'URL de la demande.

L'exemple suivant montre comment utiliser les paramètres de la requête pour filtrer les tâches :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
scope=admin&recordsToReturn=100&owner=sergey&status=OPEN&q=KAREN
```

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Paramètres de tri

Pour déterminer l'ordre de tri dans la réponse d'API REST, utilisez le paramètre de tri générique et fournissez une liste de champs de tâche séparés par des virgules. Vous pouvez spécifier l'ordre de tri pour chaque champ. Utilisez un tiret (-) pour spécifier l'ordre décroissant. L'ordre de tri par défaut est l'ordre croissant.

L'exemple suivant montre comment trier les résultats :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
recordsToReturn=100&owner=sergey&status=OPEN&sort=-priority
```

Exemple de demande d'API

L'exemple de demande suivant récupère la liste de tâches :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task
```

La demande ne contient pas de corps.

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des tâches au format JSON :

```
{
  "firstRecord": 1,
  "recordCount": 10,
  "pageSize": 10,
  "task": [
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:15443",
```

```

        "rel": "self"
    }
},
"taskType": {
    "name": "ReviewNoApprove",
    "label": "Review No approve",
    "taskKind": "REVIEW",
    "taskAction": [
        {
            "name": "Escalate",
            "label": "Escalate",
            "nextTaskType": "AVOSBeFinalReview",
            "comment": "AS REQUIRED",
            "attachment": "NEVER",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
        },
        {
            "name": "Reject",
            "label": "Reject",
            "nextTaskType": "AVOSBeUpdate",
            "comment": "MANDATORY",
            "attachment": "MANDATORY",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
        },
        {
            "name": "Disclaim",
            "label": "Disclaim",
            "nextTaskType": "AVOSBeReviewNoApprove",
            "comment": "AS REQUIRED",
            "attachment": "NEVER",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
        }
    ],
    "pendingBVT": true,
    "updateType": "PENDING"
},
"taskId": "urn:b4p2:15443",
"title": "Review changes in SMITH,SMITH",
"dueDate": "2015-07-15T21:45:59-07:00",
"status": "OPEN",
"priority": "NORMAL",
"businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15440",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:15440",
    "title": "Review changes in SMITH,JOHN",
    "dueDate": "2015-07-15T21:37:50-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},

```

```

    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15437",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "taskAction": [
          {
            "name": "Reject",
            "label": "Reject",
            "nextTaskType": "AVOSBeUpdate",
            "comment": "AS_REQUIRED",
            "attachment": "MANDATORY",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
          }
        ],
        "pendingBVT": true,
        "updateType": "PENDING"
      },
      "taskId": "urn:b4p2:15437",
      "title": "Review changes in SMITH,JOHN",
      "dueDate": "2015-07-15T21:34:32-07:00",
      "status": "OPEN",
      "priority": "NORMAL",
      "businessEntity": "Person"
    },
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14820",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
      },
      "taskId": "urn:b4p2:14820",
      "title": "Review changes in STAS,STAS",
      "dueDate": "2015-07-14T10:40:51-07:00",
      "status": "OPEN",
      "priority": "NORMAL",
      "businessEntity": "Person"
    },
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14809",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
      },

```

```

        "taskId": "urn:b4p2:14809",
        "title": "Review changes in ,93C8ORSCOFSA687",
        "dueDate": "2015-07-14T08:28:15-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14609",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14609",
        "title": "Review changes in A8,A8",
        "dueDate": "2015-07-13T08:40:11-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14425",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14425",
        "title": "Review changes in A7,A7",
        "dueDate": "2015-07-10T14:11:02-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14422",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14422",
        "title": "Review changes in A6,A6",
        "dueDate": "2015-07-10T13:54:09-07:00",
        "status": "OPEN",
        "priority": "NORMAL",

```

```

        "businessEntity": "Person"
      },
      {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14415",
            "rel": "self"
          }
        ],
        "taskType": {
          "name": "ReviewNoApprove",
          "label": "Review No approve",
          "taskKind": "REVIEW",
          "pendingBVT": true,
          "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14415",
        "title": "Review changes in A5,A5",
        "dueDate": "2015-07-10T13:51:12-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
      },
      {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14355",
            "rel": "self"
          }
        ],
        "taskType": {
          "name": "Notification",
          "label": "Notification",
          "taskKind": "REVIEW",
          "pendingBVT": false,
          "updateType": "ACTIVE"
        },
        "taskId": "urn:b4p2:14355",
        "title": "Review changes in A4,A4",
        "dueDate": "2015-07-10T10:31:57-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
      }
    ]
  }
}

```

Lire la tâche

L'API REST Lire la tâche renvoie les détails d'une tâche tels que le type de tâche, la priorité et le statut.

L'API utilise la méthode GET.

URL de demande

L'URL Lire la tâche est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Remarque: Utilisez l'API Lister les tâches pour obtenir l'ID de la tâche.

Effectuez la demande HTTP GET suivante à l'URL Lire la tâche :

```
GET http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Exemple de demande d'API

L'exemple de demande suivant renvoie les détails d'une tâche :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:16605
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les détails de la tâche :

```
{
  "taskType": {
    "name": "ReviewNoApprove",
    "label": "Review No Approve",
    "taskKind": "REVIEW",
    "taskAction": [
      {
        "name": "Escalate",
        "label": "Escalate",
        "nextTaskType": "AVOSBeFinalReview",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      },
      {
        "name": "Reject",
        "label": "Reject",
        "nextTaskType": "AVOSBeUpdate",
        "comment": "MANDATORY",
        "attachment": "MANDATORY",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      },
      {
        "name": "Disclaim",
        "label": "Disclaim",
        "nextTaskType": "AVOSBeReviewNoApprove",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      }
    ],
    "pendingBVT": true,
    "updateType": "PENDING"
  },
  "taskId": "urn:b4p2:16605",
  "processId": "16603",
  "title": "Review changes in HERNANDEZ,ALEJANDRO",
  "dueDate": "2015-07-23T01:18:39.125-07:00",
  "status": "OPEN",
  "priority": "NORMAL",
  "taskRecord": [
    {
      "businessEntity": {
        "key": {
          "rowid": "114",
          "sourceKey": "SYS:114"
        },
        "name": "Person"
      }
    },
    {
      "businessEntity": {
        "key": {
          "rowid": "114",
          "sourceKey": "SYS:114"
        },
        "name": "Person"
      }
    }
  ]
}
```

```

        "sourceKey": "SYS:114",
        "rowidXref": "4680363"
    },
    "name": "Person.XREF"
}
}
],
"creator": "avos",
"createDate": "2015-07-16T01:18:46.148-07:00",
"attachments": [
    {
        "id": "urn.b4p2:22203::file1.txt",
        "name": "file1.txt",
        "contentType": "text/plain",
        "creator": "admin",
        "createDate": "2018-02-26T14:31:05.590-05:00"
    }
],
"businessEntity": "Person",
"interactionId": "72340000003000",
"orsId": "localhost-orcl-DS_UI1"
}

```

Créer une tâche

L'API REST Créer une tâche crée une tâche et lance un flux de travail.

L'API utilise la méthode POST pour créer une tâche et renvoie l'ID du processus de flux de travail qui contient la tâche.

URL de demande

L'URL Créer une tâche est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/task
```

Effectuez la demande HTTP POST suivante à l'URL Créer une tâche :

```
POST http://<host>:<port>/<context>/<database ID>/task
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
POST http://<host>:<port>/<context>/<database ID>/task
Content-Type: application/json/xml
```

Corps de demande

Spécifiez les attributs de tâche lorsque vous créez la tâche. Utilisez le format JSON ou XML pour envoyer les données de tâche dans la demande.

Le tableau suivant décrit les paramètres de tâche dans le corps de demande :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Utilisez l'attribut de nom pour spécifier le type de tâche. Pour plus d'informations sur les types de tâches, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
owner	Utilisateur auquel le créateur attribue la tâche.
title	Brève description de la tâche.

Paramètre	Description
comments	Commentaires sur la tâche.
pièces jointes	Pièces jointes de la tâche.
dueDate	Date à laquelle le propriétaire doit terminer la tâche.
status	Statut de la tâche dans le flux de travail. Utilisez une des deux valeurs suivantes : - Ouvert : la tâche n'a pas démarré ou est en cours. - Fermé : la tâche est terminée ou est annulée.
priorité	Niveau d'importance de la tâche. Utilisez l'une des valeurs suivantes : élevé, normal et faible. La valeur par défaut est Normal.
creator	Utilisateur qui crée la tâche.
createDate	Date de création de la tâche.
orsId	ID du stockage de référence opérationnelle (Operational Reference Store - ORS) tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.
processId	Identificateur du type de tâche ActiveVOS®. Pour plus d'informations, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
taskRecord	Enregistrement racine de l'objet d'entreprise ou enregistrement de références croisées associé à la tâche. Utilisez l'ID de ligne ou le système source et la clé source pour spécifier l'enregistrement.
businessEntity	Nom de l'entité d'entreprise à laquelle appartient l'enregistrement taskRecord.
interactionId	ID de l'interaction. L'identificateur d'interaction vous permet de conserver une relation de contexte de tâche entre une tâche et les enregistrements.
groupes	Attribuez une tâche à tous les utilisateurs d'un groupe spécifié. Définissez les groupes d'utilisateurs dans la console MDM Hub. Spécifiez les groupes sous la forme d'un tableau.

L'exemple de code suivant utilise l'ID de ligne pour spécifier l'enregistrement de tâche :

```
taskRecord: [{
  businessEntity:{
    name: "Person",
    key:{
      rowid: "233",
    }
  }
}]
```

Le corps de demande est au format suivant :

```
{
  taskType: {name:"name of the task"},
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
}
```



```

    creator: "use who creates the task",
    createDate: "date on which the task is created",
    updatedBy: "user who last updated the task",
    lastUpdateDate: "date on which the task was last updated",
    businessEntity: "name of the business entity",
    interactionID: "ID of an interaction",
    groups: ["group name A", "group name B", ...],
    orsId: "database ID",
    processId: "ActiveVOS task type ID",
    taskRecord: [{
      businessEntity:{
        name: "name of the business entity",
        key:{
          rowid: "rowId of the record", //Use the rowId or the source system and
          source key to identify the record.
        }
      }
    }]
  }
}

```

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande suivant crée une tâche pour un enregistrement racine :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task

{
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "",
  owner: "manager",
  title: "Smoke test task",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: "IDDUpdateWithApprovalTask",
  taskRecord: [{
    businessEntity:{
      name: "Person",
      key:{
        rowid: "123"
      }
    }
  }]
}

```

Exemple de réponse d'API

L'exemple suivant affiche la réponse lorsqu'une tâche est créée avec succès. L'API renvoie l'ID du processus de flux de travail qui contient la tâche.

```

{
  "parameters": {
    "processId": "15827"
  }
}

```

Mettre à jour la tâche

L'API REST Mettre à jour la tâche met à jour une seule tâche.

L'API utilise la méthode PATCH pour mettre à jour des champs de tâche et la méthode PUT pour mettre à jour la tâche complète. Indiquez l'ID de la tâche en tant que paramètre d'URL.

URL de demande

L'URL Mettre à jour la tâche est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Remarque: Utilisez l'API Lister les tâches pour obtenir l'ID de la tâche.

Effectuez la demande HTTP PUT suivante à l'URL Mettre à jour la tâche pour mettre à jour la tâche complète :

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Effectuez la demande HTTP PATCH suivante à l'URL Mettre à jour la tâche pour mettre à jour des champs de tâche :

```
PATCH http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>  
Content-Type: application/<json/xml>
```

Corps de demande

Utilisez l'API Lire la tâche pour obtenir les détails de la tâche. Spécifiez les attributs de tâche lorsque vous mettez à jour la tâche. Utilisez le format JSON ou XML pour envoyer les données à mettre à jour dans la demande.

Le tableau suivant décrit les paramètres de tâche dans le corps de demande :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Utilisez l'attribut de nom pour spécifier le type de tâche. Pour plus d'informations sur les types de tâches, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
taskId	ID de la tâche.
owner	Utilisateur qui exécute la tâche.
title	Brève description de la tâche.
comments	Commentaires sur la tâche.
pièces jointes	Pièces jointes de la tâche.
dueDate	Date à laquelle le propriétaire doit terminer la tâche.
status	Statut de la tâche dans le flux de travail. Utilisez une des deux valeurs suivantes : <ul style="list-style-type: none">- Ouvert : la tâche n'a pas démarré ou est en cours.- Fermé : la tâche est terminée ou est annulée.

Paramètre	Description
priorité	Niveau d'importance de la tâche. Utilisez l'une des valeurs suivantes : élevé, normal et faible. La valeur par défaut est Normal.
creator	Utilisateur qui crée la tâche.
createDate	Date de création de la tâche.
updatedBy	Utilisateur qui met à jour la tâche.
lastUpdateDate	Date de dernière mise à jour de la tâche.
orsId	ID du stockage d'enregistrement opérationnel tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.
processId	ID du processus de flux de travail qui contient la tâche.
taskRecord	Enregistrement racine ou enregistrement de références croisées associé à la tâche. Utilisez l'ID de ligne ou le système source et la clé source pour spécifier l'enregistrement.
businessEntity name	Nom de l'entité d'entreprise à laquelle appartient l'enregistrement taskRecord.

L'exemple de code suivant utilise l'ID de ligne pour spécifier l'enregistrement de tâche :

```
taskRecord: [{
  businessEntity:{
    name: 'Person',
    key:{
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

Pour une demande PATCH, le corps de demande contient les champs de tâche que vous voulez modifier. Vous pouvez modifier le titre de la tâche, la priorité, la date d'échéance et le propriétaire.

Pour une demande PUT, le corps de demande contient tous les champs de tâche. Utilisez le corps de demande suivant pour une demande PUT :

```
{
  taskType: {name:"name of the task"},
  taskId: "ID of the task",
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
  creator: "use who creates the task",
  createDate: "date on which the task is created",
  updatedBy: "user who last updated the task",
  lastUpdateDate: "date on which the task was last updated",
  businessEntity: "name of the business entity",
  orsId: "database ID",
  processId: 'ActiveVOS task type ID',
  taskRecord: [{
```

```

        businessEntity:{
            name: 'name of the business entity',
            key:{
                rowid: 'rowId of the record',//Use the rowId or the source system and source
key to identify the record.
                systemName: '',
                sourceKey: ''
            }
        }
    ]]
}

```

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande PUT suivant met à jour une tâche complète :

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934
{
    taskType: {name:"UpdateWithApprovalWorkflow"},
    taskId: "urn:b4p2:15934",
    owner: "John",
    title: "Smoke test task - updated",
    comments: "Smoke testing - updated",
    attachments: [
        {
            id: "TEMP_SVR1.1VDVS"
        }
    ],
    dueDate: "2015-08-15T00:00:00",
    status: "OPEN",
    priority: "HIGH",
    creator: "admin",
    createDate: "2015-06-15T00:00:00",
    updatedBy: "admin",
    lastUpdateDate: "2015-06-15T00:00:00",
    businessEntity: "Person",
    orsId: "localhost-orcl-DS_UI1",
    processId: '3719',
    taskRecord: [{
        businessEntity:{
            name: 'Person',
            key:{
                rowid: '123',
                systemName: '',
                sourceKey: ''
            }
        }
    }]
}

```

L'exemple de demande PATCH suivant met à jour des champs de tâche :

```

PATCH http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934
{
    processId: "3719",
    priority: "HIGH",
    owner: "John"
}

```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK lorsqu'une tâche est mise à jour avec succès. Le corps de réponse est vide.

Tâche terminée

L'API REST Tâche terminée ferme un flux de travail de tâches une fois toutes les tâches terminées. Utilisez l'API pour fermer un flux de travail après avoir traité tous les enregistrements associés aux tâches. Par exemple, lorsque vous sélectionnez Fusionner les candidats, vous pouvez créer une tâche qui lance le flux de travail de fusion. La tâche de fusion est terminée une fois que vous avez prévisualisé chaque candidat et que vous l'avez soit fusionné, soit marqué comme non-correspondance. Utilisez l'API pour fermer le flux de travail de fusion.

L'API utilise la méthode PUT.

URL de demande

L'URL Tâche terminée est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Effectuez la demande HTTP PUT suivante à l'URL Tâche terminée :

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
Content-Type: application/<json/xml>
```

Corps de demande

Envoyez les détails de la tâche dans le corps de demande. Utilisez l'API Lire la tâche pour obtenir les détails de la tâche.

Le tableau suivant décrit les paramètres de tâche dans le corps de demande :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Utilisez l'attribut de nom pour spécifier le type de tâche. Pour plus d'informations sur les types de tâches, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
taskId	ID de la tâche.
owner	Utilisateur qui exécute la tâche.
title	Brève description de la tâche.
comments	Commentaires sur la tâche.
dueDate	Date à laquelle le propriétaire doit terminer la tâche.
status	Statut de la tâche dans le flux de travail. Utilisez une des deux valeurs suivantes : <ul style="list-style-type: none">- Ouvert : la tâche n'a pas démarré ou est en cours.- Fermé : la tâche est terminée ou est annulée.

Paramètre	Description
priorité	Niveau d'importance de la tâche.
creator	Utilisateur qui crée la tâche.
createDate	Date de création de la tâche.
updatedBy	Utilisateur qui met à jour la tâche.
lastUpdateDate	Date de dernière mise à jour de la tâche.
orsId	ID du stockage d'enregistrement opérationnel tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.
processId	ID du processus de flux de travail qui contient la tâche.
taskRecord	Enregistrement racine ou enregistrement de références croisées associé à la tâche. Utilisez l'ID de ligne ou le système source et la clé source pour spécifier l'enregistrement.
businessEntity name	Nom de l'entité d'entreprise à laquelle appartient l'enregistrement taskRecord.

L'exemple de code suivant utilise l'ID de ligne pour spécifier l'enregistrement de tâche :

```
taskRecord: [{
  businessEntity:{
    name: 'Person',
    key:{
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande suivant exécute le flux de travail de fusion :

```
PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:20210?
action=complete

{
  "taskType": {"name": "Merge"},
  "taskId": "urn:b4p2:20210",
  "owner": "admin",
  "dueDate": "2015-08-14T17:00:00-07:00",
  "status": "OPEN",
  "priority": "NORMAL",
  "creator": "admin",
  "createDate": "2015-06-15T00:00:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-06-15T00:00:00",
  "businessEntity": "Person",
  "orsId": "localhost-orcl-DS_UI1",
  "processId": "20208",
  "taskRecord": [{
    "businessEntity": {
      "name": "Person",
```

```

    "key":{
      "rowid": "233",
      "systemName": "",
      "sourceKey": ""
    }
  }
}
]]
}

```

Exemple de réponse d'API

L'API renvoie une réponse 200 OK lorsqu'un flux de travail de tâches est terminé avec succès. Le corps de réponse est vide.

Exécuter l'action de tâche

L'API REST Exécuter l'action de tâche redéfinit une tâche dans le flux de travail pour un traitement ultérieur. Chaque type de tâche comprend un ensemble d'actions de tâche et un flux de travail qui indique la séquence des tâches. Lorsque vous exécutez une action de tâche, la tâche passe à l'étape suivante dans le flux de travail. Si une action de tâche n'a pas de tâche suivante, le flux de travail se termine lorsque vous exécutez l'action de tâche.

L'API utilise la méthode POST pour exécuter des actions, par exemple pour approuver, remonter ou annuler une tâche.

URL de demande

L'URL suivante spécifie le format de l'URL Exécuter l'action de tâche :

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Remarque: Utilisez l'API Lister les tâches pour obtenir l'ID de la tâche.

Effectuez la demande HTTP POST suivante à l'URL Exécuter l'action de tâche :

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Si vous voulez modifier la tâche avant d'exécuter l'action de tâche, ajoutez l'en-tête Content-Type pour spécifier le type de média des données de la demande :

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
Content-Type: application/json+xml
```

Corps de demande

Indiquez les données de tâche dans le corps de demande si vous voulez modifier les détails de la tâche avant d'exécuter l'action de tâche.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Utilisez l'attribut de nom pour spécifier le type de tâche. Pour plus d'informations sur les types de tâches, consultez le <i>Guide d'implémentation de Data Director pour MDM Multidomain</i> .
taskId	ID de la tâche.
owner	Utilisateur qui exécute la tâche.

Paramètre	Description
title	Brève description de la tâche.
comments	Commentaires sur la tâche.
pièces jointes	Pièces jointes de la tâche.
dueDate	Date à laquelle le propriétaire doit terminer la tâche.
status	Statut de la tâche dans le flux de travail. Utilisez une des deux valeurs suivantes : - Ouvert : la tâche n'a pas démarré ou est en cours. - Fermé : la tâche est terminée ou est annulée.
priorité	Niveau d'importance de la tâche. Utilisez l'une des valeurs suivantes : élevé, normal et faible.
creator	Utilisateur qui crée la tâche.
createDate	Date de création de la tâche.
updatedBy	Utilisateur qui met à jour la tâche.
lastUpdateDate	Date de dernière mise à jour de la tâche.
businessEntity	Nom de l'entité d'entreprise.
orsId	ID du stockage d'enregistrement opérationnel tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.
processId	ID du processus de flux de travail qui contient la tâche.
taskRecord	Enregistrement racine ou enregistrement de références croisées associé à la tâche. Utilisez l'ID de ligne ou le système source et la clé source pour spécifier l'enregistrement.
businessEntity name	Nom de l'entité d'entreprise à laquelle appartient l'enregistrement taskRecord.

L'exemple de code suivant utilise l'ID de ligne pour spécifier l'enregistrement de tâche :

```
taskRecord: [{
  businessEntity:{
    name: 'Person',
    key:{
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```


LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande suivant annule une tâche et termine le flux de travail :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934?
taskAction=Cancel

{
  taskType: {
    name: "UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
  taskId: "urn:b4p2:15934",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: '3685',
  taskRecord: [{
    businessEntity: {
      name: 'Person',
      key: {
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  }]
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK lorsqu'une action de tâche est exécutée avec succès. Le corps de réponse est vide.

Lister les utilisateurs pouvant être attribués

L'API REST Lister les utilisateurs pouvant être attribués renvoie la liste des utilisateurs auxquels vous pouvez attribuer les tâches appartenant à un type de tâche. Utilisez l'API pour obtenir les utilisateurs cible pour une tâche.

L'API utilise la méthode GET.

URL de demande

L'URL Lister les utilisateurs pouvant être attribués est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/user?taskType=<task type>&businessEntity=<business entity name>
```

Effectuez la demande HTTP GET suivante à l'URL Lister les utilisateurs pouvant être attribués :

```
GET http://<host>:<port>/<context>/<database ID>/user?taskType=<task type>&businessEntity=<business entity name>
```

Paramètres de la requête

Le tableau suivant répertorie les paramètres requis dans l'URL :

Paramètre	Description
taskType	Ensemble d'actions que vous pouvez effectuer sur un enregistrement. Les types de tâches incluent la mise à jour avec approbation, la mise à jour avec approbation facultative, la fusion, l'annulation de la fusion, la vérification d'absence d'approbation, la vérification finale et la mise à jour d'un enregistrement rejeté.
businessEntity	Nom de l'entité d'entreprise.

Exemple de demande d'API

L'exemple de demande suivant récupère une liste des utilisateurs pouvant être attribués :

```
GET http://localhost:8080/cmxc/cs/localhost-orcl-DS_UI1/user.json?taskType=ReviewNoApprove&businessEntity=Person
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des utilisateurs pouvant être attribués pour le type de tâche ReviewNoApprove :

```
{
  "users": {
    "user": [
      {
        "userName": "admin"
      }
    ]
  },
  "roles": {}
}
```

Réclamer des tâches en bloc

L'API REST Réclamer des tâches en bloc réclame plusieurs tâches. Utilisez l'API pour réclamer des tâches pour lesquelles vous êtes propriétaire potentiel.

L'API utilise la méthode POST.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Réclamer des tâches en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Réclamer des tâches en bloc :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches à réclamer dans le corps de la demande.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
opType	Opération à effectuer sur les tâches. Spécifiez Réclamer pour réclamer les tâches.

Exemple de demande d'API

L'exemple de demande suivant réclame plusieurs tâches :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2315"
    },
    {
      "taskId": "urn:b4p2:2312"
    },
    {
      "taskId": "urn:b4p2:2309"
    },
    {
      "taskId": "urn:b4p2:2306"
    },
    {
      "taskId": "urn:b4p2:2303"
    }
  ],
  "opType": "Claim"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste des tâches réclamées :

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2315",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2312",
      "status": "Completed"
    },
    {

```

```

        "taskId": "urn:b4p2:2309",
        "status": "Completed"
    },
    {
        "taskId": "urn:b4p2:2306",
        "status": "Completed"
    },
    {
        "taskId": "urn:b4p2:2303",
        "status": "Completed"
    }
]
}

```

Libérer des tâches en bloc

L'API REST Libérer des tâches en bloc libère plusieurs tâches dans le pool de tâches non attribuées.

L'API utilise la méthode POST.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Libérer des tâches en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Libérer des tâches en bloc :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches à libérer dans le pool du corps de demande.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
comments	Facultatif. Commentaires sur les tâches.
opType	Opération à effectuer sur les tâches. Spécifiez Libérer pour libérer les tâches dans le pool des tâches disponibles.

Exemple de demande d'API

L'exemple de demande suivant libère plusieurs tâches et ajoute le même commentaire à chaque tâche :

```

POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2318"
    },
    {
      "taskId": "urn:b4p2:2209"
    }
  ]
}

```

```

    "comments": "John is on vacation. Releasing these tasks back to the pool of
available tasks.",
    "opType": "Release"
}

```

L'exemple de demande suivant libère plusieurs tâches et ajoute un commentaire différent à chaque tâche :

```

POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2318",
      "comments": "Jane is on vacation."
    },
    {
      "taskId": "urn:b4p2:2209",
      "comments": "Joe is on vacation."
    }
  ],
  "opType": "Release"
}

```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste des tâches libérées :

```

{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2318",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2209",
      "status": "Completed"
    }
  ]
}

```

Attribuer des tâches en bloc

L'API REST Attribuer des tâches en bloc permet d'attribuer ou de réattribuer des tâches aux utilisateurs. Cette API permet d'attribuer ou de réattribuer des tâches à tout utilisateur qui est propriétaire potentiel de la tâche.

Pour déterminer les propriétaires potentiels des tâches, utilisez l'API Répertoire des propriétaires potentiels des tâches. Pour plus d'informations, consultez la rubrique ["Répertoire des propriétaires potentiels des tâches" à la page 108](#).

L'API utilise la méthode POST.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Attribuer des tâches en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Attribuer des tâches en bloc :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches et les utilisateurs à attribuer aux tâches dans le corps de la demande.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
comments	Facultatif. Commentaires sur les tâches.
assignTo	Utilisateur responsable de la tâche. Remarque: Vous ne pouvez attribuer des tâches qu'aux utilisateurs qui sont des propriétaires potentiels de la tâche. Pour déterminer les propriétaires potentiels des tâches, utilisez l'API Répertoire des propriétaires potentiels des tâches. Pour plus d'informations, consultez la rubrique "Répertoire des propriétaires potentiels des tâches" à la page 108 .
opType	Opération à effectuer sur les tâches. Spécifiez <code>Attribuer</code> pour attribuer les tâches.

Exemple de demande d'API

L'exemple de demande suivant attribue un utilisateur à plusieurs tâches et ajoute le même commentaire à chaque tâche :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ],
  "assignTo": "srmngr1",
  "comments": "Please take a look at these tasks instead.",
  "opType": "Assign"
}
```

L'exemple de demande suivant attribue les tâches à des utilisateurs différents et ajoute un commentaire différent à chaque tâche :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "assignTo": "srmngr1",
      "comments": "Please help John take a look at this task."
    },
    {
      "taskId": "urn:b4p2:5352",
      "assignTo": "srmngr2",
      "comments": "Please help Jane take a look at this task."
    }
  ],
  "opType": "Assign"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste des tâches attribuées :

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:5352",
      "status": "Completed"
    }
  ]
}
```

Modifier des tâches en bloc

L'API REST Modifier des tâches en bloc permet de modifier les détails de plusieurs tâches.

L'API utilise la méthode POST pour mettre à jour des champs de tâches.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Modifier des tâches en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Modifier des tâches en bloc :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches à modifier et fournissez les données de tâches dans le corps de la demande.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
dueDate	Facultatif. Date à laquelle le propriétaire doit terminer la tâche.
comments	Facultatif. Commentaires sur la tâche.
title	Facultatif. Brève description de la tâche.
priority	Facultatif. Niveau d'importance de la tâche. Utilisez l'une des valeurs suivantes : élevé, normal et faible.
assignTo	Facultatif. Utilisateur responsable de la tâche.

Paramètre	Description
fileId	Facultatif. ID du fichier à joindre à la tâche. Pour joindre un fichier à une tâche, vous devez le charger dans un espace de stockage temporaire. Pour plus d'informations, consultez la rubrique Annexe B, "Utilisation d'API REST pour le chargement de fichiers" à la page 299 .
opType	Opération à effectuer sur les tâches. Spécifiez <code>Modifier</code> pour modifier les tâches.

Exemple de demande d'API

L'exemple de demande suivant modifie plusieurs tâches en utilisant les mêmes détails de tâches :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
    {
      "taskId": "urn:b4p2:4384"
    }
  ],
  "comments": "Please work on this task ASAP. There's more information in the attachment.",
  "priority": "High",
  "title": "Test Demo Review Person",
  "dueDate": "2019-05-01 00:00:00",
  "fileId": "TEMP_SVR1.1IOPK",
  "opType": "Edit"
}
```

L'exemple de demande suivant modifie plusieurs tâches en utilisant des détails de tâches différents :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "comments": "Added a screenshot to help clarify.",
      "priority": "High",
      "title": "Test Demo Review Person",
      "dueDate": "2019-05-01 00:00:00",
      "fileId": "TEMP_SVR1.1IOPK"
    },
    {
      "taskId": "urn:b4p2:4384",
      "priority": "High"
    }
  ],
  "opType": "Edit"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste des tâches modifiées :

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "status": "Completed"
    }
  ]
}
```



```

    },
    {
      "taskId": "urn:b4p2:4384",
      "status": "Completed"
    }
  ]
}

```

Action de tâches en bloc

L'API REST Action de tâches en bloc exécute des actions sur plusieurs tâches. Lorsque vous exécutez une action de tâche, la tâche passe à l'étape suivante dans le flux de travail. Si une action de tâche ne comporte aucune tâche suivante, le flux de travail se termine lorsque vous exécutez l'action de tâche.

L'API utilise la méthode POST pour exécuter des actions, par exemple pour approuver, remonter ou rejeter une tâche.

Pour déterminer les actions que vous pouvez effectuer sur une tâche, utilisez l'API Obtenir les actions de tâche. Pour plus d'informations, consultez la rubrique ["Obtenir les actions de tâches" à la page 106](#).

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Action de tâches en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Action de tâches en bloc :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches sur lesquelles vous souhaitez agir et les actions à exécuter.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
actionName	Action de tâche à exécuter. Utilisez l'API Obtenir les actions de tâches pour déterminer celles que vous pouvez effectuer sur les tâches.
opType	Opération à effectuer sur les tâches. Spécifiez <code>Action</code> pour exécuter une action de tâche.

Exemple de demande d'API

L'exemple de demande suivant approuve les tâches :

```

POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ]
}

```

```

    }
  ],
  "actionName": "Approve",
  "opType": "Action"
}

```

L'exemple de demande suivant approuve une tâche et la remonte :

```

POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "actionName": "Approve",
      "taskId": "urn:b4p2:5353"
    },
    {
      "actionName": "Escalate",
      "taskId": "urn:b4p2:5354"
    }
  ],
  "opType": "Action"
}

```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste des tâches modifiées :

```

{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:5352",
      "status": "Completed"
    }
  ]
}

```

Obtenir les actions de tâches

L'API REST Obtenir les actions de tâches récupère une liste des actions que vous pouvez effectuer sur des tâches.

L'API utilise la méthode POST pour récupérer une liste des actions de tâches.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Obtenir les actions de tâches est au format suivant :

```
http://<host>:<port>/<context>/<database ID>
```

Effectuez la demande HTTP POST suivante à l'URL Obtenir les actions de tâches :

```
POST http://<host>:<port>/<context>/<database ID>
```

Corps de demande

Spécifiez les tâches pour lesquelles vous souhaitez connaître les actions que vous pouvez effectuer.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
opType	Opération à effectuer sur les tâches. Spécifiez <code>GetTaskActions</code> pour récupérer une liste d'actions que vous pouvez effectuer sur la tâche.

Exemple de demande d'API

L'exemple de demande suivant récupère les actions que vous pouvez effectuer sur les tâches :

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ],
  "opType": "GetTaskActions"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche une liste des tâches et des actions que vous pouvez effectuer sur chaque tâche.

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed",
      "taskActions":
      [
        {
          "name": "Approve",
          "displayName": "Approve",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeNotification"
        },
        {
          "name": "Reject",
          "displayName": "Reject",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeUpdate"
        },
        {
          "name": "Disclaim",
          "displayName": "Disclaim",

```

```

        "description": null,
        "manualReassign": true,
        "closeTaskView": true,
        "cancelTask": false,
        "nextTaskType": "AVOSBeFinalReview"
    }
]
},
{
    "taskId": "urn:b4p2:5352",
    "status": "Completed",
    "taskActions": [
        {
            "name": "Approve",
            "displayName": "Approve",
            "description": null,
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false,
            "nextTaskType": "AVOSBeNotification"
        },
        {
            "name": "Reject",
            "displayName": "Reject",
            "description": null,
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false,
            "nextTaskType": "AVOSBeUpdate"
        },
        {
            "name": "Disclaim",
            "displayName": "Disclaim",
            "description": null,
            "manualReassign": true,
            "closeTaskView": true,
            "cancelTask": false,
            "nextTaskType": "AVOSBeFinalReview"
        }
    ]
}
]
}

```

Répertorier les propriétaires potentiels des tâches

L'API REST Répertorier les propriétaires potentiels des tâches récupère les propriétaires potentiels de plusieurs tâches.

L'API utilise la méthode GET.

Remarque: Vous pouvez spécifier un maximum de 100 tâches dans une demande.

URL de demande

L'URL Répertorier les propriétaires potentiels des tâches est au format suivant :

`http://<host>:<port>/<context>/<database ID>`

Effectuez la demande HTTP GET suivante à l'URL Répertorier les propriétaires potentiels des tâches :

GET `http://<host>:<port>/<context>/<database ID>`

Corps de demande

Spécifiez les tâches pour lesquelles vous voulez récupérer les propriétaires potentiels.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
opType	Opération à effectuer sur les tâches. Spécifiez <code>PotentialOwners</code> pour obtenir une liste de propriétaires potentiels pour les tâches.

Exemple de demande d'API

L'exemple de réponse suivant récupère les propriétaires potentiels de plusieurs tâches :

```
GET http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
    {
      "taskId": "urn:b4p2:4382"
    }
  ],
  "opType": "PotentialOwners"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les propriétaires potentiels de chaque tâche :

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    },
    {
      "taskId": "urn:b4p2:4382",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ]
      ]
    }
  ]
}
```

```

        [
            "srmngr1",
            "senior manager"
        ]
    ]
}

```

Répertorier les propriétaires potentiels d'une tâche

L'API REST Répertorier les propriétaires potentiels d'une tâche récupère les propriétaires potentiels d'une tâche.

L'API utilise la méthode GET.

URL de demande

L'URL Répertorier les propriétaires potentiels d'une tâche est au format suivant :

```
http://<host>:<port>/<context>/list/owners/<database ID>/<taskId>
```

Effectuez la demande HTTP GET suivante à l'URL Répertorier les propriétaires potentiels d'une tâche :

```
GET http://<host>:<port>/<context>/list/owners/<database ID>
```

Corps de demande

Spécifiez la tâche pour laquelle vous voulez récupérer les propriétaires potentiels.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
taskId	ID de la tâche.
opType	Opération à effectuer sur la tâche. Spécifiez <code>PotentialOwners</code> pour obtenir une liste de propriétaires potentiels pour la tâche.

Exemple de demande d'API

L'exemple de réponse suivant récupère les propriétaires potentiels d'une tâche :

```

GET http://localhost:8080/cmx/task/operations/list/owners/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
    {
      "taskId": "urn:b4p2:4382"
    }
  ],
  "opType": "PotentialOwners"
}

```

Exemple de réponse d'API

L'exemple de réponse suivant présente les propriétaires potentiels d'une tâche :

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    },
    {
      "taskId": "urn:b4p2:4382",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    }
  ]
}
```

Lister les métadonnées des fichiers

L'API REST Lister les métadonnées des fichiers renvoie une liste des métadonnées de fichier dans un stockage.

Utilisez l'API REST Lister les métadonnées des fichiers avec un stockage BPM ou TEMP.

L'API utilise la méthode GET.

URL de demande

L'URL Lister les métadonnées des fichiers est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Effectuez la demande HTTP GET suivante à l'URL Lister les métadonnées des fichiers :

```
GET http://<host>:<port>/<context>/<database ID>/<storage>
```

Exemple de demande d'API

L'exemple de demande suivant récupère une liste des métadonnées de fichier dans un stockage TEMP :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
```

Exemple de réponse d'API

L'exemple de réponse suivant montre une liste des métadonnées de fichier :

```
{
  files: [
    {
      "fileId": "TEMP_SVR1.1VDVS",
      "fileName": "file1.txt",
      "fileType": "text",
      "fileContentType": "text/plain",
    },
    {
      "fileId": "TEMP_SVR1.2ESDS",
      "fileName": "image1.png",
      "fileType": "image",
      "fileContentType": "image/png",
    },
    ...
  ]
}
```

Créer les métadonnées du fichier

L'API REST Créer les métadonnées du fichier crée des métadonnées pour un fichier et renvoie un ID de fichier pour le fichier.

Vous pouvez utiliser l'ID de fichier pour charger, joindre, mettre à jour, télécharger et supprimer le fichier.

Utilisez l'API REST Créer les métadonnées du fichier avec un stockage DB ou TEMP.

L'API utilise la méthode POST.

URL de demande

L'URL Créer les métadonnées du fichier est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Effectuez la demande HTTP POST suivante à l'URL Créer les métadonnées du fichier :

```
POST http://<host>:<port>/<context>/<database ID>/<storage>
```


Corps de demande

Indiquez les métadonnées du fichier.

Le tableau suivant décrit les paramètres des métadonnées du fichier dans le corps de demande :

Paramètre	Description
fileName	Nom du fichier. Par exemple, <code>file.txt</code> .
fileType	Catégorie du type de fichier. Par exemple, <code>text</code> ou <code>image</code> .
fileContentType	Type de contenu du fichier. Le type de contenu se compose d'un type et d'un sous-type qui sont séparés par une barre oblique (/). Par exemple : <code>image/png</code> .

Remarque: Les paramètres requis pour l'API REST Créer les métadonnées du fichier sont spécifiques au stockage.

Exemple de demande d'API

L'exemple de demande suivant crée les métadonnées d'un fichier dans un stockage TEMP :

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP

{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

Exemple de réponse d'API

L'exemple suivant montre la réponse lors de création réussie des métadonnées d'un fichier dans un stockage TEMP. L'API renvoie l'ID de fichier du fichier.

```
TEMP_SVR1.1VDVS
```

Remarque: Le format de l'ID de fichier est : <storage type>_<uniqueID>.

Obtenir les métadonnées du fichier

L'API REST Obtenir les métadonnées du fichier renvoie les métadonnées d'un fichier associé à un ID de fichier.

Utilisez l'API REST Obtenir les métadonnées du fichier avec un stockage BPM, BUNDLE, DB ou TEMP.

L'API utilise la méthode GET.

URL de demande

Le format de l'URL Obtenir les métadonnées du fichier est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les métadonnées du fichier :

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Exemple de demande d'API

L'exemple de demande suivant renvoie les métadonnées d'un fichier avec l'ID de fichier, TEMP_SVR1.1VDVS, dans le stockage TEMP :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

L'exemple de demande suivant renvoie les métadonnées du fichier de groupement de ressources avec l'ID de fichier, besMetadata, dans le stockage BUNDLE :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Exemple de réponse d'API

L'exemple de réponse suivant montre les métadonnées d'un fichier avec l'ID de fichier, TEMP_SVR1.1VDVS, dans un stockage TEMP :

```
{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

L'exemple de réponse suivant montre les métadonnées du fichier de groupement de ressources, besMetadata, dans un stockage BUNDLE :

```
{
  "fileName": "besMetadata.zip",
  "fileType": "BES Metadata Bundle",
  "fileContentType": "application/zip",
  "digest": "a08c5d97da7e6a780ed7c427ff14a8d2d396438cd65b654ad67424e226f64a41"
}
```

Mettre à jour les métadonnées du fichier

L'API REST Mettre à jour les métadonnées du fichier met à jour les métadonnées d'un fichier associé à un ID de fichier.

Utilisez l'API REST Mettre à jour les métadonnées du fichier avec un stockage DB ou TEMP.

L'API utilise la méthode PUT.

URL de demande

L'URL Mettre à jour les métadonnées du fichier est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Effectuez la demande HTTP PUT suivante à l'URL Mettre à jour les métadonnées du fichier :

```
PUT
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Exemple de demande d'API

L'exemple de demande suivant met à jour les métadonnées d'un fichier avec l'ID de fichier, TEMP_SVR1.1VDVS, dans un stockage TEMP :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

```
{
  "fileName": "file2.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

L'exemple de demande suivant met à jour les métadonnées d'un fichier avec l'ID de fichier, DB_SVR1.OJU1, dans un stockage DB :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
{
  "fileName": "Document_2.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK lorsque les métadonnées de fichier sont mises à jour avec succès. Le corps de réponse est vide.

Charger le contenu du fichier

L'API REST Charger le contenu du fichier charge le contenu d'un fichier associé à un ID de fichier.

Utilisez l'API REST Charger le contenu du fichier avec un stockage BUNDLE, DB ou TEMP.

L'API utilise la méthode PUT.

URL de demande

L'URL Charger le contenu du fichier est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Effectuez la demande HTTP PUT suivante à l'URL Charger le contenu du fichier :

```
PUT http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Exemple de demande d'API

L'exemple de demande suivant charge le contenu d'un fichier avec l'ID de fichier, TEMP_SVR1.1VDVS, dans un stockage TEMP :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/content
Test attachment content: file 1
```

L'exemple de demande suivant charge le contenu d'un fichier portant l'ID de fichier, DB_SVR1.OJU1, dans un stockage DB :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

L'exemple de demande suivant charge le contenu d'un fichier de groupement de ressources portant l'ID de fichier, besMetadata, dans un stockage BUNDLE :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK lorsque le contenu d'un fichier est chargé avec succès. Le corps de réponse est vide.

Obtenir le contenu du fichier

L'API REST Obtenir le contenu du fichier renvoie le contenu d'un fichier associé à un ID de fichier.

Utilisez l'API REST Obtenir le contenu du fichier avec un stockage BPM, BUNDLE, DB ou TEMP.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir le contenu du fichier est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir le contenu du fichier :

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Exemple de demande d'API

L'exemple de demande suivant renvoie le contenu d'un fichier portant l'ID de fichier, `urn:b4p2:22203::file1.txt`, dans un stockage BPM :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BPM/urn:b4p2:22203::file1.txt/content
```

Remarque: Utilisez l'API REST Lire la tâche pour récupérer l'ID de fichier d'une pièce jointe de tâche dans un stockage BPM.

L'exemple de demande suivant renvoie le contenu d'un fichier portant l'ID de fichier, `DB_SVR1.OJU1`, dans un stockage DB :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
```

Remarque: Utilisez l'API REST Lire l'enregistrement pour récupérer l'ID de fichier du fichier que vous joignez à un enregistrement.

L'exemple de demande suivant renvoie le contenu du fichier de groupement de ressources portant l'ID de fichier, `besMetadata`, dans un stockage BUNDLE :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
```

Exemple de réponse d'API

L'exemple de réponse suivant montre le contenu d'un fichier TXT dans un stockage BPM :

```
Test attachment content: file 1
```

L'exemple de réponse suivant montre le contenu du fichier de groupement de ressources dans un stockage BUNDLE :

```
Content-Disposition → attachment; filename=besMetadata.zip
Content-Type → application/octet-stream
Transfer-Encoding → chunked
```

Supprimer un fichier

L'API REST Supprimer un fichier supprime le fichier associé à un ID de fichier, qui comprend les métadonnées et le contenu du fichier.

Utilisez l'API REST Supprimer un fichier avec un stockage BUNDLE, DB ou TEMP.

L'API utilise la méthode DELETE.

URL de demande

L'URL Supprimer un fichier est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Effectuez la demande HTTP DELETE suivante à l'URL Supprimer un fichier :

```
DELETE http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Exemple de demande d'API

L'exemple de demande suivant supprime le fichier associé à l'ID de fichier, `TEMP_SVR1.1VDVS`, dans un stockage `TEMP`, y compris les métadonnées et le contenu du fichier :

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

L'exemple de demande suivant supprime le fichier associé à l'ID de fichier, `DB_SVR1.OJU1`, dans un stockage `DB`, y compris les métadonnées et le contenu du fichier :

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
```

L'exemple de demande suivant supprime le fichier de groupement de ressources portant l'ID de fichier, `besMetadata`, dans un stockage `BUNDLE`, y compris les métadonnées et le contenu du fichier :

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK lorsqu'un fichier est supprimé avec succès. Le corps de réponse est vide.

Prévisualiser les enregistrements remplacés

L'API REST Prévisualiser les enregistrements remplacés renvoie des enregistrements d'entité d'entreprise modifiés et remplacés par de nouvelles valeurs.

L'API utilise la méthode POST.

URL de demande

L'URL Prévisualiser les enregistrements remplacés est au format suivant :

```
http://<host>:<port>/find-replace/<orsId>/<entity>/<preview{?firstRecord,order,recordsToReturn,returnTotal}>
```

Effectuez la demande HTTP POST suivante à l'URL Prévisualiser les enregistrements remplacés :

```
POST http://<host>:<port>/find-replace/<orsId>/<entity>/<preview{?firstRecord,order,recordsToReturn,returnTotal}>
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les enregistrements d'entité d'entreprise modifiés et remplacés par de nouvelles valeurs.

Le tableau suivant décrit les paramètres dans le corps de la demande.

Paramètre	Type	Description
entity	Chemin	Nom des enregistrements d'entité d'entreprise à renvoyer.
orsId	Chemin	ID du stockage de référence opérationnelle de la base de données permettant de renvoyer les enregistrements d'entité d'entreprise remplacés.
firstRecord	Requête	Facultatif. Ligne de la liste d'enregistrements.
order	Requête	Facultatif. Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +. Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis par le suivant.
recordsToReturn	Requête	Facultatif. Spécifie le nombre de lignes à renvoyer.
returnTotal	Requête	Facultatif. Si ce paramètre est défini sur <code>True</code> , il renvoie le nombre d'enregistrements dans le résultat. La valeur par défaut est <code>False</code> .

Exemple de demande d'API

L'exemple de demande suivant renvoie des enregistrements d'entité d'entreprise modifiés et remplacés par de nouvelles valeurs :

```
POST /cmx/find-replace/localhost-orcl-MDM_SAMPLE/Person/preview?returnTotal=true HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 234

{
  "scope" : {
    "type" : "query",
    "filter" : "lastName='jones'"
  },
  "findReplace" : {
    "field" : "Emails.selectAddrTypeCd",
    "find" : null,
    "replacement" : {
      "electronicType" : "EMAIL"
    }
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les enregistrements d'entité d'entreprise remplacés :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
Date: Mon, 13 Jan 2020 18:07:02 GMT
X-Powered-By: Undertow/1
```

Content-Type: application/json;charset=UTF-8
Content-Length: 4891

```
{
  "link" : [ ],
  "firstRecord" : 1,
  "recordCount" : 6,
  "pageSize" : 10,
  "foundRecords" : 2,
  "item" : [ {
    "origin" : {
      "key" : {
        "rowid" : "401924"
      },
      "name" : "Person",
      "label" : "jones, alice",
      "object" : {
        "key" : {
          "rowid" : "426"
        },
        "name" : "Emails",
        "label" : "Emails"
      }
    },
    "found" : false,
    "record" : {
      "rowidObject" : "426",
      "creator" : "admin",
      "createDate" : "2020-01-13T12:40:17.811-05:00",
      "updatedAt" : "admin",
      "lastUpdateDate" : "2020-01-13T12:40:18.583-05:00",
      "consolidationInd" : 4,
      "lastRowidSystem" : "SYS0",
      "hubStateInd" : 1,
      "label" : "Emails",
      "electronicAddress" : "alice@mail.com",
      "electAddrTypeCd" : {
        "electronicType" : "EMAIL"
      }
    }
  }, {
    "origin" : {
      "key" : {
        "rowid" : "401925"
      },
      "name" : "Person",
      "label" : "jones, bob",
      "object" : {
        "key" : {
          "rowid" : "427"
        },
        "name" : "Emails",
        "label" : "Emails"
      }
    },
    "found" : false,
    "record" : {
      "rowidObject" : "427",
      "creator" : "admin",
      "createDate" : "2020-01-13T12:40:18.006-05:00",
      "updatedAt" : "admin",
      "lastUpdateDate" : "2020-01-13T12:40:18.655-05:00",
      "consolidationInd" : 4,
      "lastRowidSystem" : "SYS0",
      "hubStateInd" : 1,
      "label" : "Emails",
      "electronicAddress" : "bob@mail.com",
      "electAddrTypeCd" : {
        "electronicType" : "EMAIL"
      }
    }
  }
}
```

```

}, {
  "origin" : {
    "key" : {
      "rowid" : "401926"
    },
    "name" : "Person",
    "label" : "jones, charlie",
    "object" : {
      "key" : {
        "rowid" : "428"
      },
      "name" : "Emails",
      "label" : "Emails"
    }
  },
  "found" : false,
  "record" : {
    "rowidObject" : "428",
    "creator" : "admin",
    "createDate" : "2020-01-13T12:40:18.216-05:00",
    "updatedBy" : "admin",
    "lastUpdateDate" : "2020-01-13T12:40:18.216-05:00",
    "consolidationInd" : 4,
    "lastRowidSystem" : "SYS0",
    "hubStateInd" : 1,
    "label" : "Emails",
    "electronicAddress" : "charlie@mail.com",
    "electAddrTypeCd" : {
      "electronicType" : "EMAIL"
    }
  }
}, {
  "origin" : {
    "key" : {
      "rowid" : "401935"
    },
    "name" : "Person",
    "label" : "jones, alice",
    "object" : {
      "key" : {
        "rowid" : "437"
      },
      "name" : "Emails",
      "label" : "Emails"
    }
  },
  "found" : true,
  "record" : {
    "rowidObject" : "437",
    "creator" : "admin",
    "createDate" : "2020-01-13T13:07:02.187-05:00",
    "updatedBy" : "admin",
    "lastUpdateDate" : "2020-01-13T13:07:02.187-05:00",
    "consolidationInd" : 4,
    "lastRowidSystem" : "SYS0",
    "hubStateInd" : 1,
    "label" : "Emails",
    "electronicAddress" : "alice@mail.com"
  }
}, {
  "origin" : {
    "key" : {
      "rowid" : "401936"
    },
    "name" : "Person",
    "label" : "jones, bob",
    "object" : {
      "key" : {
        "rowid" : "438"
      },
      "name" : "Emails",

```



```

        "label" : "Emails"
    }
}, {
    "found" : true,
    "record" : {
        "rowidObject" : "438",
        "creator" : "admin",
        "createDate" : "2020-01-13T13:07:02.389-05:00",
        "updatedBy" : "admin",
        "lastUpdateDate" : "2020-01-13T13:07:02.390-05:00",
        "consolidationInd" : 4,
        "lastRowidSystem" : "SYS0",
        "hubStateInd" : 1,
        "label" : "Emails",
        "electronicAddress" : "bob@mail.com"
    }
}, {
    "origin" : {
        "key" : {
            "rowid" : "401937"
        },
        "name" : "Person",
        "label" : "jones, charlie",
        "object" : {
            "key" : {
                "rowid" : "439"
            },
            "name" : "Emails",
            "label" : "Emails"
        }
    },
    "found" : false,
    "record" : {
        "rowidObject" : "439",
        "creator" : "admin",
        "createDate" : "2020-01-13T13:07:02.587-05:00",
        "updatedBy" : "admin",
        "lastUpdateDate" : "2020-01-13T13:07:02.587-05:00",
        "consolidationInd" : 4,
        "lastRowidSystem" : "SYS0",
        "hubStateInd" : 1,
        "label" : "Emails",
        "electronicAddress" : "charlie@mail.com",
        "electAddrTypeCd" : {
            "electronicType" : "EMAIL"
        }
    }
}
} ]
}

```

Mettre à jour l'option Rechercher et remplacer

L'API REST Mettre à jour les enregistrements remplacés renvoie des enregistrements filtrés pour les entités d'entreprise à modifier et à remplacer par de nouvelles valeurs.

L'API utilise la méthode POST.

URL de demande

L'URL Mettre à jour les enregistrements remplacés est au format suivant :

```
http://<host>:<port>/find-replace/<orsId>/<entity>/{?systemName}
```

Effectuez la demande HTTP POST suivante à l'URL Mettre à jour les enregistrements remplacés :

```
POST http://<host>:<port>/find-replace/<orsId>/<entity>/{?systemName}
```

Paramètres

Spécifiez les paramètres permettant de mettre à jour les enregistrements filtrés des entités d'entreprise.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
entity	Chemin	Nom des enregistrements d'entité d'entreprise à renvoyer.
orsId	Chemin	ID du stockage de référence opérationnelle de la base de données permettant de renvoyer les enregistrements d'entité d'entreprise remplacés.
systemName	Requête	Facultatif. Nom du système source.
If-Unmodified-Since	En-tête	Facultatif. Permet d'ignorer les enregistrements auparavant remplacés.

Exemple de demande d'API

L'exemple de demande suivant met à jour les enregistrements filtrés pour les entités d'entreprise à modifier et à remplacer par de nouvelles valeurs :

```
POST /cmx/find-replace/localhost-orcl-MDM_SAMPLE/Person?systemName=Admin HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 234

{
  "scope" : {
    "type" : "query",
    "filter" : "lastName='jones'"
  },
  "findReplace" : {
    "field" : "Emails.selectAddrTypeCd",
    "find" : null,
    "replacement" : {
      "electronicType" : "EMAIL"
    }
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les enregistrements filtrés mis à jour des entités d'entreprise :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 1339

{
  "successful" : 2,
  "failed" : 0,
  "skipped" : 0,
  "item" : [ {
    "object" : {
      "Person" : {
        "key" : {
          "rowid" : "401935"
        },
        "rowidObject" : "401935"
      }
    }
  } ]
}
```

```

      "Emails" : {
        "link" : [ ],
        "item" : [ {
          "key" : {
            "rowid" : "437",
            "sourceKey" : "480000041000"
          },
          "rowidObject" : "437",
          "electAddrTypeCd" : {
            "key" : {
              "rowid" : "1"
            },
            "rowidObject" : "1"
          }
        } ]
      }
    },
    "parameters" : { }
  }, {
    "object" : {
      "Person" : {
        "key" : {
          "rowid" : "401936"
        },
        "rowidObject" : "401936",
        "Emails" : {
          "link" : [ ],
          "item" : [ {
            "key" : {
              "rowid" : "438",
              "sourceKey" : "480000043000"
            },
            "rowidObject" : "438",
            "electAddrTypeCd" : {
              "key" : {
                "rowid" : "1"
              },
              "rowidObject" : "1"
            }
          } ]
        }
      }
    },
    "parameters" : { }
  } ]
}

```

Importer le nouveau fichier

L'API REST Importer le nouveau fichier démarre le travail d'importation d'un nouveau fichier et importe les entités d'entreprise à partir d'un fichier.

L'API utilise la méthode POST.

URL de demande

L'URL Importer le nouveau fichier est au format suivant :

http://<host>:<port>/<context>/<orsId>/<job>

Effectuez la demande HTTP POST suivante à l'URL Importer le nouveau fichier :

POST http://<host>:<port>/<context>/<orsId>/<job>

Paramètres

Spécifiez les paramètres suivants de l'importation d'un nouveau fichier dans le corps de la demande.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsID	Chemin	ID de base de données du stockage de référence opérationnelle dans lequel importer le nouveau fichier.
fileID	Requête	Facultatif. ID du fichier importé. Utilisez cette option en cas de mise à jour d'enregistrements existants.
mappingID	Requête	Facultatif. ID de mappage du fichier importé. Utilisez cette option en cas de mise à jour d'un fichier précédemment importé.
systemName	Requête	Facultatif. Nom de l'utilisateur système qui importe le fichier.

Exemple de demande d'API

L'exemple de demande suivant démarre le travail d'importation d'un nouveau fichier et importe des entités d'entreprise :

```
POST /cmx/beimport/localhost-orcl-MDM_SAMPLE/job HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 94

{
  "fileId" : "TEMP_SVR1.1G7UW",
  "mappingId" : "SVR1.1G7UX",
  "systemName" : "Admin"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant indique que le nouveau fichier a été importé :

```
HTTP/1.1 201 Created
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Location: http://localhost:8080/cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/group/SVR1.1G7UY
```

Importer le fichier correspondant

L'API REST Importer le fichier correspondant fait correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier. Une fois l'opération d'importation terminée, les entités d'entreprise correspondantes sont mises à jour avec les nouvelles données.

Vous pouvez importer des entités d'entreprise dupliquées, les entités d'entreprises uniques ou les entités d'entreprise dupliquées et uniques.

L'API utilise la méthode POST.

URL de demande

L'URL Importer le fichier correspondant est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Effectuez la demande HTTP POST suivante à l'URL Importer le fichier correspondant :

```
POST http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Paramètres

Spécifiez les paramètres permettant de faire correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant d'importer le fichier après avoir fait correspondre les enregistrements existants.
filter	Requête	Facultatif. Utilisez les valeurs de filtre suivantes : withMatches : entrez <code>True</code> si les enregistrements correspondent aux enregistrements actuels. withoutMatches : entrez <code>False</code> si les enregistrements ne correspondent pas aux enregistrements actuels.
jobGroupControlId	Requête	Facultatif. ID de groupe du travail de correspondance.
systemName	Requête	Facultatif. Nom du système source.

Exemple de demande d'API

L'exemple de demande suivant fait correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier :

```
POST /cmx/beimport/{ors}/aftermatch
{
  jobGroupControlId: "...",
  systemName: "SFA",
  filter: {
    withMatches: true,
    withoutMatches: false
  }
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK après la correspondance réussie de nouvelles entités d'entreprise avec les actuelles et l'importation du fichier. Le corps de la réponse est vide.

Obtenir les propriétés du fichier

L'API REST Obtenir les propriétés du fichier renvoie les propriétés du fichier importé. Par exemple, l'API renvoie les propriétés du fichier de délimiteur, de qualificatif de texte et de séparateur décimal si vous avez sélectionné les propriétés lors de l'importation du fichier.

L'API utilise la méthode GET.

Remarque: L'API renvoie les propriétés du fichier existantes avec des délimiteurs uniquement s'il y a au moins cinq lignes.

URL de demande

L'URL Obtenir les propriétés du fichier est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties{query}>
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les propriétés du fichier :

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/<properties{query}>
```

Paramètres

Spécifiez les paramètres permettant d'obtenir les propriétés du fichier importé.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle.
fileId	Chemin	ID du fichier importé.
suggest	Requête	Facultatif. Si ce paramètre est défini sur True, la demande renvoie les propriétés selon le fichier importé. Sinon, la demande renvoie les propriétés du fichier suggérées en fonction des données dans le fichier importé.

Exemple de demande d'API

L'exemple de demande suivant détecte et renvoie les propriétés du fichier importé :

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties?suggest=true
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente l'en-tête et le corps après avoir renvoyé les propriétés du fichier importé :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json; charset=UTF-8
Content-Length: 370

{
  "confidence" : "HIGH",
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
  }
}
```

Enregistrer les propriétés du fichier

L'API REST Enregistrer les propriétés du fichier analyse et enregistre les propriétés du fichier importé.

L'API utilise la méthode PUT.

URL de demande

L'URL Enregistrer les propriétés du fichier est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Effectuez la demande HTTP PUT suivante à l'URL Enregistrer les propriétés du fichier :

```
PUT http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Paramètres

Spécifiez les paramètres permettant d'enregistrer les propriétés du fichier importé.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle.
fileId	Chemin	ID du fichier importé.
headerRow	Requête	Facultatif. Numéro d'en-tête de ligne souhaité pour une colonne.
dataStartAtRow	Requête	Facultatif. Ligne du fichier source à partir de laquelle vous voulez importer des données.
locale	Requête	Facultatif. Langue de localisation.
dataPattern	Requête	Facultatif. Format de date des champs de date dans le fichier d'importation.
decimalSeparator	Requête	Facultatif. Séparateur décimal. Entrez une virgule ou un point.
thousandsSeparator	Requête	Facultatif. Séparateur des milliers. Entrez une virgule ou un point.
delimiter	Requête	Facultatif. Caractère qui permet de séparer les valeurs de données dans le fichier d'importation. Entrez un délimiteur prédéfini ou définissez-en un personnalisé.
encoding	Requête	Facultatif. Norme d'encodage Unicode.
textQualifier	Requête	Facultatif. Symboles utilisés dans le fichier pour encadrer une chaîne.

Exemple de demande d'API

L'exemple de demande suivant analyse et enregistre les propriétés du fichier :

```
PUT /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 344

{
```

```

    "type" : "CSV",
    "properties" : {
      "headerRow" : 1,
      "dataStartsAtRow" : 2,
      "regionalSettings" : {
        "locale" : "en-CA",
        "datePattern" : "MM-dd-yyyy",
        "decimalSeparator" : ".",
        "thousandsSeparator" : ",",
      },
      "delimiter" : ",",
      "encoding" : "UTF-8",
      "textQualifier" : "\""
    }
  }
}

```

Exemple de réponse d'API

L'exemple de réponse suivant indique que les propriétés du fichier sont enregistrées.

```

HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1

```

Renvoyer les propriétés des fichiers

L'API REST Renvoyer les propriétés des fichiers renvoie les propriétés des fichiers qui sont conservées.

L'API utilise la méthode GET.

URL de demande

L'URL Renvoyer les propriétés du fichier est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Effectuez la demande HTTP GET suivante à l'URL Renvoyer les propriétés du fichier :

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les propriétés de fichiers qui sont conservées.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID du stockage de référence opérationnelle à partir duquel récupérer les propriétés des fichiers.
fileId	Chemin	ID du fichier importé.
suggest	Requête	Facultatif. Si ce paramètre est défini sur True, la demande renvoie les propriétés des fichiers qui sont conservées. Sinon, la demande renvoie toutes les propriétés des fichiers.

Exemple de demande d'API

L'exemple de demande suivant renvoie les propriétés de fichiers qui sont conservées.

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties?suggest=true
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les propriétés de fichiers qui sont conservées :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json; charset=UTF-8
Content-Length: 370

{
  "confidence" : "HIGH",
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
  }
}
```

Prévisualiser les fichiers analysés

L'API REST Prévisualiser les fichiers analysés analyse les propriétés des fichiers et renvoie des enregistrements plats selon les propriétés fournies dans le corps de la demande. L'API analyse les fichiers CSV et Excel.

L'API utilise la méthode POST.

URL de demande

L'URL de l'API Prévisualiser les fichiers analysés est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Effectuez la demande HTTP POST suivante à l'URL de l'API Prévisualiser les fichiers analysés :

```
POST http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Paramètres

Spécifiez les paramètres permettant d'analyser les propriétés du fichier et de renvoyer des enregistrements plats selon les propriétés fournies dans le corps de la demande.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsld	Chemin	ID de base de données du stockage de référence opérationnelle du fichier stocké dans la base de données.
fileld	Chemin	ID du fichier importé.
pagesize	Requête	Facultatif. Nombre de lignes d'enregistrements à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant analyse les propriétés du fichier et renvoie des enregistrements plats selon les propriétés fournies dans le corps de la demande :

```
POST /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/preview HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 345

{
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "d MMM, yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les enregistrements plats récupérés selon les propriétés fournies dans le corps de la demande :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 381

{
  "header" : [ "first name", "last name", "job title", "gender", "birth_date" ],
  "data" : [ [ "alice", "smith", "developer", "F", "01-01-1950" ], [ "bob", "smith",
"tester", "M", "02-02-1960" ], [ "charlie", "smith", "manager", "X", "03-03-1970" ],
[ "dave", "smith", "accountant", "M", "ten years ago" ], [ "eve", "smith", "", "F",
"04-04-1990" ] ],
  "numberOfRows" : 5
}
```

Obtenir les fichiers analysés

L'API REST Obtenir les fichiers analysés analyse les propriétés des fichiers et renvoie des enregistrements plats selon les propriétés. L'API analyse les fichiers CSV et Excel.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les propriétés des fichiers analysés est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les propriétés des fichiers analysés :

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Paramètres

Spécifiez les paramètres permettant d'analyser les propriétés du fichier et de renvoyer des enregistrements plats selon les propriétés du fichier.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle.
fileId	Chemin	ID du fichier importé.
pagesize	Requête	Facultatif. Nombre de lignes d'enregistrements à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant analyse les fichiers et renvoie des enregistrements plats en fonction des propriétés associées à un fichier :

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/preview HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les enregistrements plats analysés selon les propriétés associées à un fichier :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json; charset=UTF-8
Content-Length: 381

{
  "header" : [ "first_name", "last_name", "job_title", "gender", "birth_date" ],
  "data" : [ [ "alice", "smith", "developer", "F", "01-01-1950" ], [ "bob", "smith",
"tester", "M", "02-02-1960" ], [ "charlie", "smith", "manager", "X", "03-03-1970" ],
[ "dave", "smith", "accountant", "M", "ten years ago" ], [ "eve", "smith", "", "F",
"04-04-1990" ] ],
  "numberOfRows" : 5
}
```

Erreurs d'analyse de l'importation du fichier

L'API REST Erreurs d'analyse de l'importation du fichier renvoie les erreurs d'analyse de l'opération d'importation du fichier dans un format de fichier CSV.

L'API utilise la méthode GET.

URL de demande

L'URL Erreurs d'analyse de l'importation du fichier est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<parse-errors>/{jobGroupControlId}{?entity}
```

Effectuez la demande HTTP GET suivante à l'URL Erreurs d'analyse de l'importation du fichier :

```
GET http://<host>:<port>/<context>/<orsId>/<parse-errors>/{jobGroupControlId}{?entity}
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les erreurs d'analyse à partir de l'opération d'importation de fichier au format CSV.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de renvoyer les erreurs d'analyse à partir de l'opération d'importation de fichier.
jobGroupControlId	Requête	Facultatif. ID de groupe du travail de lots.
entity	Requête	Facultatif. Nom de l'entité d'entreprise.

Exemple de demande d'API

L'exemple de demande suivant renvoie les erreurs d'analyse à partir de l'opération d'importation du fichier :

```
GET /cmx/beimport/localhost-orcl-MDM_SAMPLE/parse-errors/SVR1.1G7UY HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les erreurs d'analyse à partir de l'opération d'importation du fichier :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: text/csv
Content-Length: 200

first_name,last_name,job_title,gender,birth_date,ROW,COLUMN,ERROR,ENTITY
dave,smith,accountant,M,ten years ago,3,birth_date,Failed to convert value [ten years ago] for field [birthdate].,PersonView
```

Erreurs de chargement de l'importation du fichier

L'API REST Erreurs de chargement de l'importation du fichier renvoie les erreurs de chargement de l'opération d'importation du fichier dans un format de fichier CSV.

L'API utilise la méthode GET.

URL de demande

L'URL Erreurs de chargement de l'importation du fichier est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<load-errors>/{jobGroupControlId}{?entity}
```

Effectuez la demande HTTP GET suivante à l'URL Erreurs de chargement de l'importation du fichier :

```
GET http://<host>:<port>/<context>/<orsId>/<load-errors>/{jobGroupControlId}{?entity}
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les erreurs de chargement à partir de l'opération d'importation de fichier au format CSV.

Le tableau suivant décrit les paramètres dans le corps de la demande.

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de renvoyer les erreurs de chargement à partir de l'opération d'importation de fichier.
jobGroupControlId	Requête	Facultatif. ID de groupe du travail de lots.
entity	Requête	Facultatif. Nom de l'entité d'entreprise.

Exemple de demande d'API

L'exemple de demande suivant renvoie les erreurs de chargement à partir de l'opération d'importation du fichier :

```
GET /cmx/beimport/localhost-orcl-MDM_SAMPLE/load-errors/SVR1.1G7UY HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les erreurs de chargement à partir de l'opération d'importation du fichier :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: text/csv
Content-Length: 257

first_name,last_name,job_title,gender,birth_date,ROW,COLUMN,ERROR,ENTITY
charlie,smith,manager,X,03-03-1970,2,,SIP-50112: Could not run WriteCO business entity
service. A record for [genderCd] could not be found in business entity
[LUGender]..PersonView
```

Prévisualiser la promotion

L'API REST Prévisualiser la promotion renvoie un aperçu d'un enregistrement obtenu si vous promouvez les modifications en attente.

L'API utilise la méthode GET. Vous pouvez prévisualiser un enregistrement auquel vous appliquez les modifications en attente. La réponse d'API contient l'enregistrement avec les nouvelles valeurs et un résumé des modifications avec les anciennes valeurs. L'API ne renvoie pas d'informations sur les données que vous supprimez. Indiquez l'ID d'interaction des modifications en attente dans l'URL.

URL de demande

L'URL Prévisualiser la promotion est au format suivant :

```
http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?
action=previewPromote&interactionID=<interaction ID>
```

Effectuez la demande HTTP GET suivante à l'URL Prévisualiser la promotion :

```
GET http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?
action=previewPromote&interactionID=<interaction ID>
```

Paramètres de la requête

L'ID d'interaction des modifications en attente est un paramètre requis dans l'URL.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
contentMetadata	Métadonnées de l'aperçu de fusion. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none">- BVT. Spécifiez le rowid de l'enregistrement qui contient la valeur la plus fiable à utiliser dans l'aperçu de fusion. Renvoie des informations sur l'enregistrement de références croisées et l'identificateur de l'enregistrement d'origine.- MERGE. Spécifiez les rowids des enregistrements à fusionner. Renvoie les informations sur la manière dont les enregistrements descendants ont été fusionnés.
interactionId	ID d'interaction des modifications en attente.
effectiveDate	Facultatif. Date à laquelle vous souhaitez prévisualiser les modifications. Utilisez ce paramètre pour les objets de base dont la chronologie est activée.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande suivant crée un aperçu d'un enregistrement racine dans l'entité commerciale Person :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
action=previewPromote&interactionId=72300000001000
```

Exemple de réponse d'API

L'exemple de réponse suivant renvoie un aperçu de l'enregistrement avec les nouvelles valeurs et un résumé des modifications des anciennes valeurs :

```
{
  "rowidObject": "233",
  "creator": "admin",
  "createDate": "2008-08-12T02:15:02-07:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-07-14T03:42:38.778-07:00",
  "consolidationInd": "1",
  "lastRowidSystem": "SYS0",
  "dirtyIndicator": "0",
  "interactionId": "72300000001000",
  "hubStateInd": "1",
  "label": "LLOYD,BOB",
  "partyType": "Person",
```

```

      "lastName": "LLOYD",
      "firstName": "BOB",
      "displayName": "BOB LLOYD",
      "preferredPhone": {
        "rowidObject": "164",
        "$original": {
          "rowidObject": "164"
        }
      },
      "$original": {
        "label": "DUNN, LLOYD",
        "lastName": "DUNN",
        "firstName": "LLOYD",
        "displayName": "LLOYD DUNN"
      }
    }
  }
}

```

Promouvoir

L'API REST Promouvoir promeut toutes les modifications en attente apportées à un enregistrement en fonction de l'ID d'interaction de la demande de modification.

L'API utilise la méthode POST. Indiquez l'ID d'interaction comme un paramètre de requête.

URL de demande

L'URL Promouvoir est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root
record>?action=promote&interactionId=<interaction ID>

```

Effectuez la demande HTTP POST suivante à l'URL Promouvoir :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root
record>?action=promote&interactionId=<interaction ID>

```

Paramètre de requête

L'ID d'interaction des modifications en attente est un paramètre requis. L'API utilise l'ID d'interaction pour rechercher tous les enregistrements associés aux modifications en attente.

Exemple de demande d'API

L'exemple de demande suivant promeut toutes les modifications en attente en fonction de l'ID d'interaction de la demande de modification :

```

POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1038246?
action=promote&interactionId=69120000294000

```

Exemple de réponse d'API

L'exemple de réponse suivant contient l'ID de ligne de l'enregistrement après la promotion des modifications en attente :

```

{
  Person: {
    rowidObject: "1038246"
  }
}

```

Supprimer les modifications en attente

L'API REST Supprimer les modifications en attente supprime toutes les modifications en attente apportées à un enregistrement en fonction de l'ID d'interaction de la demande de modification.

L'API utilise la méthode DELETE et renvoie l'ID de ligne de l'enregistrement.

URL de demande

L'URL Supprimer les modifications en attente est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?  
action=deletePending&interactionId=<interaction ID>
```

Effectuez la demande DELETE suivante à l'URL Supprimer les modifications en attente :

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?  
action=deletePending&interactionId=<interaction ID>
```

Paramètre de requête

Indiquez l'ID d'interaction des modifications en attente que vous voulez supprimer.

Exemple de demande d'API

L'exemple de demande suivant supprime toutes les modifications en attente en fonction de l'ID d'interaction de la demande de modification :

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?  
action=deletePending&interactionId=72300000001000
```

Exemple de réponse d'API

L'exemple de réponse suivant contient l'ID de ligne de l'enregistrement après la suppression des modifications en attente :

```
{  
  Person: {  
    rowidObject: "233"  
  }  
}
```

Prévisualiser la fusion

L'API REST Prévisualiser la fusion renvoie un aperçu d'un enregistrement racine consolidé si vous fusionnez deux enregistrements racines ou plus.

L'API utilise la méthode POST et accepte une liste d'enregistrements racines et de remplacements au niveau du champ pour renvoyer un aperçu de l'enregistrement fusionné. L'ID de ligne de l'enregistrement cible est un paramètre requis.

URL de demande

L'URL Prévisualiser la fusion est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?  
action=previewMerge
```

Le format suivant de l'URL Prévisualiser la fusion spécifie le nombre de niveaux enfants à renvoyer :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?  
action=previewMerge&depth=2
```


Effectuez la demande HTTP POST suivante à l'URL Prévisualiser la fusion :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge
```

Remarque: Dans le corps de demande, ajoutez la propriété de clés et spécifiez les enregistrements racines à fusionner avec l'enregistrement cible.

Pour remplacer les correspondances des enregistrements enfants, ajoutez le paramètre contentMetadata :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge&contentMetadata=MERGE/<json/xml>
```

Remarque: dans le corps de demande, ajoutez la propriété de remplacement et spécifiez les remplacements de fusion.

Pour spécifier le type de média des données à envoyer avec la demande, ajoutez l'en-tête Content-Type :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge
Content-Type: application/<json/xml>
```

Paramètres de la requête

L'ID de ligne de l'enregistrement cible est un paramètre requis.

Vous pouvez utiliser les paramètres de requête suivants :

Paramètre	Description
contentMetadata	Métadonnées de l'aperçu de fusion. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none">- BVT. Renvoie des informations sur l'enregistrement de références croisées qui l'emporte ainsi que l'identificateur de l'enregistrement d'origine.- MERGE. Renvoie les informations sur la manière dont les enregistrements descendants ont été fusionnés.
depth	Nombre de niveaux enfants à renvoyer.
effectiveDate	Date à laquelle vous souhaitez générer l'aperçu.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Corps de demande

Avant de commencer, utilisez l'API Lire les enregistrements correspondants pour déterminer les enregistrements qui peuvent être fusionnés avec l'enregistrement racine initial. Envoyez la liste d'enregistrements dans le corps de demande de l'API Aperçu de fusion.

Vous pouvez remplacer les valeurs de champ dans l'enregistrement racine. Par exemple, si aucun enregistrement racine correspondant ne contient l'orthographe correcte du prénom, vous pouvez spécifier le prénom correct dans le corps de la demande. En outre, vous pouvez supprimer les enregistrements correspondants ou en spécifier d'autres.

Utilisez les propriétés suivantes dans le corps de demande :

Propriétés/ éléments	Type	Description
keys	tableau	Requis. Liste classée d'enregistrements racine correspondants à fusionner. Vous pouvez identifier les enregistrements à l'aide de l'ID de ligne ou d'une combinaison entre le système source et la clé source.
overrides	objet	Remplace les valeurs de champ dans un enregistrement racine et les correspondances des enregistrements enfants.
MERGE	objet	Remplace les valeurs de champ dans les enregistrements enfants à fusionner. Ajoutez le type d'enregistrement enfant dans l'objet <code>overrides</code> , puis ajoutez l'objet <code>MERGE</code> .

L'exemple de code JSON suivant identifie un enregistrement racine à fusionner avec l'enregistrement racine cible :

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
}
```

Le code suivant montre comment remplacer un champ dans l'enregistrement racine Partie et comment remplacer les candidats à la fusion pour les enregistrements enfants Téléphone :

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
  overrides: {
    Party: {
      rowidObject: "P1",
      firstName: "Serge", //override the value for the first name
      Telephone: { // override which Telephone child records to merge
        item:[
          {
            rowidObject: "T1",
            MERGE: {
              item: [ // to remove the original merge candidates, specify null
                null,
                null
              ],
              $original: {
                item: [
                  {key:{rowid: "T2"}},
                  {key:{rowid: "T3"}}
                ]
              }
            }
          },
          {
            rowidObject: "T4",
            MERGE: {
              item: [ // to add or change merge candidates, specify matched records
                {key:{rowid: "T2"}}
              ],
              $original: {
                item: [
                  null
                ]
              }
            }
          }
        ]
      }
    }
  }
}
```

```

    }
  ]
}
}
}

```

Exemple de demande d'API

L'exemple de demande suivant renvoie l'aperçu d'un enregistrement consolidé :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=previewMerge

{
  keys: [
    {
      rowid: "2478246"
    },
    {
      rowid: "2478230"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}

```

Exemple de réponse d'API

L'exemple de réponse suivant affiche l'aperçu de l'enregistrement consolidé :

```

{
  "Person": {
    "rowidObject": "2478245",
    "partyType": "Person",
    "lastName": "Smith",
    "firstName": "Charlie",
    "displayName": "ALICE SMITH"
  }
}

```

Mettre à jour la fusion en attente

Pour enregistrer les modifications apportées aux enregistrements qui font partie d'une tâche de fusion en attente, telle que les remplacements de la valeur d'enregistrement, utilisez l'API REST Mettre à jour la fusion en attente. L'API enregistre les modifications selon l'ID d'interaction des enregistrements.

L'API utilise la méthode POST.

URL de demande

Le composant de chemin de l'URL de la demande doit inclure l'ID de ligne de l'enregistrement cible.

L'URL de l'option Mettre à jour la fusion en attente est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>
/<rowid of the root record>?
action=updatePendingMerge&interactionId=<interaction ID>

```

Effectuez la demande HTTP POST suivante à l'URL de l'option Mettre à jour la fusion en attente :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>
/<rowid of the root record>?

```

```
action=updatePendingMerge&interactionId=<interaction ID>
```

Dans le corps de la demande, ajoutez les clés et spécifiez les enregistrements racine à fusionner avec l'enregistrement racine cible. Vous pouvez également spécifier les enregistrements enfants pour lesquels remplacer les correspondances.

Paramètre de requête

Le tableau suivant décrit les paramètres de la requête que vous pouvez utiliser dans l'URL :

Paramètre	Description
action	Requis. Enregistrez les modifications, telles que les remplacements au niveau des champs, dans une tâche de fusion en attente. Définissez le paramètre sur <code>updatePendingMerge</code> et utilisez-le avec le paramètre <code>interactionId</code> . Par exemple, utilisez la requête suivante pour enregistrer les modifications dans les enregistrements de l'entité d'entreprise Personne, qui sont en attente de l'action de fusion : <code>Person?action=updatePendingMerge&interactionId</code>
interactionId	Requis. ID d'interaction de la tâche de fusion en attente.

Corps de demande

Avant d'utiliser l'API Mettre à jour la fusion en attente, utilisez l'API REST Lire les enregistrements correspondants pour déterminer les enregistrements correspondants que vous pouvez fusionner avec l'enregistrement racine cible. Envoyez la liste d'enregistrements dans le corps de la demande de l'API Mettre à jour la fusion en attente.

Vous pouvez remplacer les valeurs de champ dans l'enregistrement racine. Par exemple, si aucun enregistrement racine correspondant ne contient l'orthographe correcte du prénom, vous pouvez spécifier le prénom correct dans le corps de la demande. En outre, vous pouvez supprimer les enregistrements correspondants ou en spécifier d'autres.

Utilisez les propriétés suivantes dans le corps de la demande :

Propriétés/éléments	Type	Description
keys	tableau	Requis. Liste classée d'enregistrements racine correspondants à fusionner. Vous pouvez identifier les enregistrements à l'aide de l'ID de ligne ou d'une combinaison entre le système source et la clé source.
overrides	objet	Remplace les valeurs de champ dans un enregistrement racine et les correspondances des enregistrements enfants.
MERGE	objet	Remplace les valeurs de champ dans les enregistrements enfants à fusionner. Ajoutez le type d'enregistrement enfant dans l'objet <code>overrides</code> , puis ajoutez l'objet <code>MERGE</code> .

L'exemple de code JSON suivant identifie deux enregistrements racine à fusionner avec l'enregistrement racine cible :

```
{
  keys: [
    {rowid: "2478246"},
    {rowid: "2478230"}
  ]
}
```

```

    ]
  }
}

```

L'exemple suivant de corps de la demande montre comment remplacer un champ dans l'enregistrement racine Partie et comment remplacer les enregistrements correspondants des enregistrements enfants Téléphone :

```

{
  keys: [
    {
      rowid: "2478246"
    }
  ]
  overrides: {
    Party: {
      rowidObject: "2478230",
      firstName: "Charlie", //Override the value for the first name
      Telephone: { // Specifies the Telephone child records to merge
        item:[
          {
            rowidObject: "2511",
            MERGE: {
              item: [ // To remove the original merge candidates, specify null
                null,
                null
              ],
              $original: {
                item: [
                  {key:{rowid: "2822"}},
                  {key:{rowid: "2733"}}
                ]
              }
            }
          },
          {
            rowidObject: "2644",
            MERGE: {
              item: [ // To add or change merge candidates, specify matched records
                {key:{rowid: "2822"}}
              ],
              $original: {
                item: [
                  null
                ]
              }
            }
          }
        ]
      }
    }
  }
}

```

Exemple de demande d'API

L'exemple de demande suivant remplace le champ Prénom de l'enregistrement cible par la valeur Charlie :

POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/2478245?
action=updatePendingMerge&interactionId=3982462873645

```

{
  keys: [
    {
      rowid: "2478246"
    },
    {
      rowid: "2478230"
    }
  ],
  overrides: {
    Person: {

```

```

        firstName: "Charlie"
    }
}

```

Exemple de réponse d'API

L'exemple de réponse suivant contient l'ID de ligne de l'enregistrement cible pour lequel l'action de fusion est en attente :

```

{
  "Person": {
    "key": {
      "rowid": "2478245"
    },
    "rowidObject": "2478245"
  }
}

```

Fusion en attente

L'API REST Fusion en attente met à jour toutes les tâches de fusion en attente relatives à un enregistrement en fonction de l'ID d'interaction de la demande de modification. La fonction de fusion en attente vous permet de différer les opérations de fusion jusqu'à ce que le processus de flux de travail approuve toutes les tâches de fusion.

L'API utilise la méthode POST et renvoie l'ID de ligne de l'enregistrement.

URL de demande

L'URL Fusion en attente est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PendingMerge&interactionId=<interaction ID>

```

Effectuez la demande HTTP POST suivante à l'URL Fusion en attente :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge

```

Paramètre de requête

L'ID d'interaction de la fusion en attente est un paramètre requis.

Exemple de demande d'API

L'exemple de demande suivant met à jour toutes les tâches de fusion en attente associées à l'ID d'interaction :

```

POST /Person/123?action=pendingMerge&interactionId=123

```

Exemple de réponse d'API

L'exemple de réponse suivant contient les ID de ligne des objets de base racines concernés :

```

{
  keys: [{rowid: "456"}, {rowid: "789"}],
  overrides: {...}
}

```

PromoteMerge

L'API REST Promouvoir la fusion exécute toutes les tâches de fusion en attente associées à l'ID d'interaction de la demande de modification.

L'API utilise la méthode POST et renvoie l'ID de ligne de l'enregistrement gagnant.

URL de demande

L'URL Promouvoir la fusion est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PromoteMerge&interactionId=<interaction ID>
```

Effectuez la demande HTTP POST suivante à l'URL Promouvoir la fusion :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
```

Paramètre de requête

L'ID d'interaction de la tâche de fusion en attente est un paramètre requis. L'API utilise l'ID d'interaction pour rechercher toutes les tâches de fusion en attente et exécute les fusions.

Exemple de demande d'API

L'exemple de demande suivant promeut toutes les tâches de fusion en attente associées à l'ID d'interaction :

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Exemple de réponse d'API

L'exemple de réponse suivant contient les ID de ligne des enregistrements suite à la promotion des tâches de fusion en attente :

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Transformation du fichier

L'API Transformation du fichier récupère la liste des entités d'entreprise à partir du fichier importé avant le mappage des entités d'entreprise aux champs cible.

L'API utilise la méthode POST.

URL de demande

L'API Transformation des fichiers est au format suivant :

```
http://<host>:<port>/cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Effectuez la demande HTTP POST suivante à l'URL de l'API Transformation des fichiers :

```
POST /cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Paramètres

Spécifiez les propriétés permettant de prévisualiser la liste des entités d'entreprise à partir du fichier importé avant de mapper les entités d'entreprise aux champs cible.

Le tableau suivant décrit les paramètres.

Paramètre	Description
fileId	ID de fichier permettant de générer l'aperçu.
orsId	ID du stockage de référence opérationnelle de la base de données à partir duquel prévisualiser le mappage.
entity	Facultatif. Nombre d'entités d'entreprise ou de relations à prévisualiser.
pageSize	Facultatif. Nombre de pages à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant récupère la liste des entités d'entreprise mappées à partir du fichier importé avant le mappage des entités d'entreprise :

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?entity=Person&fileId=TEMP_SVR1.1G7UW
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 431

{
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "d MMM, yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "Person",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    } ]
  } ]
}
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des entités d'entreprise mappées à partir du fichier importé avant le mappage des entités d'entreprise aux champs cible.

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
Content-Length: 825

{
  "link" : [ ],
  "item" : [ {
```



```

    "object" : {
      "Person" : {
        "label" : "smith, alice",
        "lastName" : "smith",
        "firstName" : "alice"
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, bob",
          "lastName" : "smith",
          "firstName" : "bob"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, charlie",
          "lastName" : "smith",
          "firstName" : "charlie"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, dave",
          "lastName" : "smith",
          "firstName" : "dave"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, eve",
          "lastName" : "smith",
          "firstName" : "eve"
        }
      }
    }
  ]
}

```

Suggérer le mappage

L'API REST Suggérer le mappage suggère le mappage des entités d'entreprise ou des relations dans le fichier importé.

L'API utilise la méthode GET.

URL de demande

L'API Suggérer le mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<suggest{?entity,fileId,purpose}>
```

Effectuez la demande HTTP GET suivante à l'URL de l'API Suggérer le mappage :

```
GET http://<host>:<port>/<context>/<orsId>/<suggest{?entity,fileId,purpose}>
```

Corps de demande

Spécifiez les paramètres permettant de suggérer le mappage des entités d'entreprise ou des relations dans le fichier importé.

Le tableau suivant décrit les paramètres dans le corps de la demande.

Paramètre	Type	Description
orsld	Chemin	ID de la base de données du stockage de référence opérationnelle à partir duquel prévisualiser le mappage.
fileId	Requête	Facultatif. ID du fichier importé.
purpose	Requête	Facultatif. Objectif du mappage et son mode d'utilisation.
entity	Requête	Facultatif. Nombre d'entités d'entreprise ou de relations à prévisualiser. Si vous spécifiez le paramètre d'entité, l'API détecte et renvoie le mappage de l'entité spécifiée uniquement.

Exemple de demande d'API

L'exemple de demande suivant renvoie le mappage suggéré des entités d'entreprise ou des relations dans le fichier importé :

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/suggest?fileId=TEMP_SVR1.1G7UW HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente le mappage suggéré des entités d'entreprise ou des relations dans le fichier importé :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 1315

{
  "name" : "my_mapping",
  "systemName" : "Admin",
  "mapping" : {
    "confidence" : "HIGH",
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
  },
  "objects" : [ {
    "confidence" : "HIGH",
    "name" : "PersonView",
    "fields" : [ {
      "confidence" : "HIGH",
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "confidence" : "HIGH",
      "name" : "jobTitle",
      "fileColumn" : "job_title",
      "skipNulls" : false
    } ],
  }, {
```

```

        "confidence" : "HIGH",
        "name" : "lastName",
        "fileColumn" : "last_name",
        "skipNulls" : false
    }, {
        "confidence" : "HIGH",
        "name" : "birthdate",
        "fileColumn" : "birth_date",
        "skipNulls" : false
    }, {
        "confidence" : "HIGH",
        "name" : "genderCd",
        "fileColumn" : "gender",
        "skipNulls" : false
    } ],
    "children" : [ ]
} ],
"purpose" : "IMPORT"
},
"creator" : "admin",
"createDate" : "2020-01-13T12:39:37.186-05:00",
"updatedBy" : "admin",
"lastUpdateDate" : "2020-01-13T12:39:37.186-05:00",
"system" : false
}

```

Fusionner les enregistrements

L'API REST Fusionner les enregistrements fusionne deux enregistrements racines ou plus pour créer un enregistrement consolidé unique. L'ID de ligne de l'enregistrement consolidé est l'ID de ligne de l'enregistrement dans lequel vous fusionnez les autres enregistrements.

L'API utilise la méthode POST. Vous pouvez spécifier des remplacements au niveau du champ pour l'enregistrement fusionné dans le corps de demande.

URL de demande

L'URL Fusionner les enregistrements est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=Merge

```

Effectuez la demande HTTP POST suivante à l'URL Fusionner les enregistrements :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge

```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
Content-Type: application/<json/xml>

```

Paramètres de la requête

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans l'URL :

Paramètre	Description
taskComment	Ajoute un commentaire à la tâche de flux de travail déclenchée par l'API.
taskAttachments	Si le paramètre Joindre des fichiers à des tâches est activé, joignez un fichier à la tâche de flux de travail déclenchée par l'API.

Corps de demande

Avant de commencer, utilisez l'API Aperçu de fusion pour prévisualiser les résultats de la fusion des enregistrements racines sélectionnés. Si vous êtes satisfait de ce qui s'affiche, utilisez les mêmes propriétés dans le corps de demande de l'API Fusionner les enregistrements.

Vous pouvez remplacer les valeurs de champ dans l'enregistrement racine. Par exemple, si aucun enregistrement racine correspondant ne contient l'orthographe correcte du prénom, vous pouvez spécifier le prénom correct dans le corps de la demande. En outre, vous pouvez supprimer les enregistrements correspondants ou en spécifier d'autres.

Utilisez les propriétés suivantes dans le corps de demande :

Propriétés/éléments	Type	Description
keys	tableau	Requis. Liste classée d'enregistrements racine correspondants à fusionner. Vous pouvez identifier les enregistrements à l'aide de l'ID de ligne ou d'une combinaison entre le système source et la clé source.
overrides	objet	Remplace les valeurs de champ dans un enregistrement racine et les correspondances des enregistrements enfants.
MERGE	objet	Remplace les valeurs de champ dans les enregistrements enfants à fusionner. Ajoutez le type d'enregistrement enfant dans l'objet <code>overrides</code> , puis ajoutez l'objet <code>MERGE</code> .

L'exemple de code JSON suivant identifie deux enregistrements racines à fusionner avec l'enregistrement racine cible :

```
{
  keys: [
    {rowid: "2478246"},
    {rowid: "2478269"}
  ]
}
```

Si vous voulez voir un exemple d'utilisation des propriétés `overrides` et `MERGE` avec l'API Fusionner les enregistrements, consultez le corps de demande de l'API Aperçu de fusion.

Exemple de demande d'API

L'exemple de demande suivant fusionne les enregistrements pour créer un enregistrement consolidé. La demande ajoute un commentaire et une pièce jointe à la tâche de flux de travail déclenchée par l'API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskComment=Read my comment&taskAttachments=TEMP_SVR1.290T8,TEMP_SVR1.290T9
Content-Type: application/<json/xml>

{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant montre l'enregistrement consolidé :

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/
2478245",
        "rel": "self"
      }
    ],
    "key": {
      "rowid": "2478245"
    },
    "rowidObject": "2478245"
  }
}
```

Annuler la fusion des enregistrements

L'API REST Annuler la fusion des enregistrements annule la fusion d'un enregistrement racine dans les enregistrements racines individuels qui existaient avant la fusion des enregistrements.

L'API utilise la méthode POST.

URL de demande

L'URL Annuler la fusion des enregistrements est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=unmerge
```

Effectuez la demande HTTP POST suivante à l'URL Annuler la fusion des enregistrements :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
Content-Type: application/json/xml
```

Paramètres de la requête

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans l'URL :

Paramètre	Description
taskComment	Ajoute un commentaire à la tâche de flux de travail déclenchée par l'API.
taskAttachments	Si le paramètre Joindre des fichiers à des tâches est activé, joignez un fichier à la tâche de flux de travail déclenchée par l'API.

Corps de demande

Envoyez la liste des enregistrements dont vous voulez annuler la fusion dans l'enregistrement consolidé dans le corps de demande. Utilisez l'ID de ligne xref ou le système source et la clé source pour spécifier les enregistrements.

Utilisez l'API Lire l'enregistrement pour obtenir l'ID de ligne xref de l'enregistrement dont vous voulez annuler la fusion. L'exemple de demande suivant récupère les métadonnées XREF d'un enregistrement :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2638243?
contentMetadata=XREF
```

Exemple de demande d'API

L'exemple de demande suivant annule la fusion d'un enregistrement de l'enregistrement consolidé. La demande ajoute un commentaire et une pièce jointe à la tâche de flux de travail déclenchée par l'API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9

{
    rowid: "4880369"
}
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche l'enregistrement dont vous avez annulé la fusion dans l'enregistrement consolidé :

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/
2478249",
        "rel": "self"
      }
    ],
    "key": {
      "rowid": "2478249"
    },
    "rowidObject": "2478249"
  }
}
```

Créer un mappage

L'API REST Créer un mappage crée un mappage lors de l'importation d'un nouveau fichier dans la base de données.

L'API utilise la méthode POST.

URL de demande

L'URL Créer un mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<mapping>
```

Effectuez la demande HTTP POST suivante à l'URL Créer un mappage :

```
POST //<host>:<port>/<context>/<orsId>/<mapping>
```

Paramètres

Spécifiez les paramètres permettant de créer un mappage dans le corps de la demande. Créez un mappage dans la demande à l'aide du format JSON ou XML.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsID	Chemin	ID de base de données du stockage de référence opérationnelle permettant de créer un mappage.
mapping	Requête	Facultatif. Entrez les propriétés du mappage.
confidence	Requête	Facultatif. Niveau de confiance du mappage. Utilisez l'une des valeurs suivantes : Élevé, Moyen ou Faible.
creator	Requête	Facultatif. Votre nom d'utilisateur.
createDate	Requête	Facultatif. Date de création du mappage.
UpdatedBy	Requête	Facultatif. Si vous mettez à jour un mappage existant, entrez votre nom d'utilisateur.
lastUpdateDate	Requête	Facultatif. Date de mise à jour du mappage par l'utilisateur.
modified	Requête	Facultatif. Utilisez les valeurs suivantes : True si vous modifiez un mappage existant ou False si vous créez un mappage.
name	Requête	Facultatif. Nom du mappage.
score	Requête	Facultatif. Score de correspondance du mappage.
system	Requête	Facultatif. Utilisez les valeurs suivantes : True : si le système crée le mappage. False : si un utilisateur crée manuellement le mappage.
systemName	Requête	Facultatif. Nom du système qui crée le mappage.
objects	Requête	Facultatif. Entrez les champs souhaités pour l'objet de base d'entité d'entreprise ou de relation.

Exemple de demande d'API

L'exemple de demande suivant crée un mappage :

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 737

{
  "mapping" : {
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "dd-MM-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "objects" : [ {
```

```

    "name" : "PersonView",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "jobTitle",
      "fileColumn" : "job_title",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    }, {
      "name" : "birthdate",
      "fileColumn" : "birth_date",
      "skipNulls" : false
    } ]
  } ],
  "purpose" : "IMPORT"
}

```

Exemple de réponse d'API

L'exemple de réponse suivant présente le mappage créé dans la base de données.

```

HTTP/1.1 201 Created
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Location: http://localhost:8080/cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX
Content-Type: application/json;charset=UTF-8
Content-Length: 27

{
  "id" : "SVR1.1G7UX"
}

```

Prévisualiser le mappage

L'API REST Prévisualiser le mappage récupère la liste des entités d'entreprise mappées à partir du fichier importé.

L'API utilise la méthode GET.

URL de demande

L'API Prévisualiser le mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<preview{?entity,fileId,mapping,pageSize}>
```

Effectuez la demande HTTP GET suivante à l'URL de l'API Prévisualiser le mappage :

```
GET http://<host>:<port>/<context>/<orsId>/<preview{?entity,fileId,mapping,pageSize}>
```


Paramètres

Spécifiez les paramètres permettant de prévisualiser la liste des entités d'entreprise mappées à partir du fichier importé.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsld	Chemin	ID du stockage de référence opérationnelle de la base de données à partir duquel prévisualiser le mappage.
fileId	Requête	ID de fichier. Si spécifié, l'API récupère les entités d'entreprise mappées à partir de l'ID du fichier spécifié.
mappingId	Requête	Facultatif. Entrez l'ID de mappage du fichier importé.
entity	Requête	Facultatif. Entrez le nombre d'entités d'entreprise ou de relations à prévisualiser.
pageSize	Requête	Facultatif. Entrez le nombre de pages ou d'enregistrements à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant fournit un aperçu des entités d'entreprise mappées à partir du fichier importé :

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?
mapping=SVR1.1G7UX&entity=PersonView&fileId=TEMP_SVR1.1G7UW HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente un aperçu des entités d'entreprise mappées à partir du fichier importé :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
Content-Length: 1751

{
  "link" : [ ],
  "item" : [ {
    "object" : {
      "PersonView" : {
        "firstName" : "alice",
        "lastName" : "smith",
        "birthdate" : "1950-01-01T00:00:00-05:00",
        "genderCd" : {
          "genderCode" : "F",
          "genderDisp" : "F"
        },
        "jobTitle" : "developer"
      }
    }
  }, {
    "object" : {
      "PersonView" : {
        "firstName" : "bob",
        "lastName" : "smith",
```

```

        "birthdate" : "1960-02-02T00:00:00-05:00",
        "genderCd" : {
            "genderCode" : "M",
            "genderDisp" : "M"
        },
        "jobTitle" : "tester"
    }
}, {
    "object" : {
        "PersonView" : {
            "firstName" : "charlie",
            "lastName" : "smith",
            "birthdate" : "1970-03-03T00:00:00-05:00",
            "genderCd" : {
                "genderCode" : "X",
                "genderDisp" : "X"
            },
            "jobTitle" : "manager"
        }
    }
}, {
    "object" : {
        "PersonView" : {
            "firstName" : "dave",
            "lastName" : "smith",
            "genderCd" : {
                "genderCode" : "M",
                "genderDisp" : "M"
            },
            "jobTitle" : "accountant"
        }
    }
},
"errors" : [ {
    "code" : "SIP-55206",
    "message" : "Failed to convert value [ten years ago] for field [birthdate].",
    "row" : 3,
    "column" : "birth_date",
    "field" : "birthdate"
} ]
}, {
    "object" : {
        "PersonView" : {
            "firstName" : "eve",
            "lastName" : "smith",
            "birthdate" : "1990-04-04T00:00:00-04:00",
            "genderCd" : {
                "genderCode" : "F",
                "genderDisp" : "F"
            },
            "jobTitle" : ""
        }
    }
} ]
}

```

Rechercher le mappage

L'API REST Rechercher le mappage recherche le mappage existant défini par l'utilisateur ou par le système selon l'ID de mappage.

L'API utilise la méthode GET.

URL de demande

L'URL Rechercher le mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<mapping{?  
createdBy,fileId,firstRecord,purpose,recordsToReturn,returnTotal}>
```

Effectuez la demande HTTP GET suivante à l'URL Rechercher le mappage :

```
GET http://<host>:<port>/<context>/<orsId>/<mapping {?  
createdBy,fileId,firstRecord,purpose,recordsToReturn,returnTotal}>
```

Paramètres

Spécifiez les paramètres permettant de rechercher le mappage existant défini par l'utilisateur ou par le système selon l'ID de mappage.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Chemin	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de rechercher le mappage.
fileId	Requête	ID de fichier. Si spécifié, l'API récupère le mappage à partir de l'ID de fichier spécifié.
createdBy	Requête	Entrez le créateur du mappage.
firstRecord	Requête	Facultatif. Entrez cette valeur pour ne renvoyer que la première ligne de l'enregistrement.
purpose	Requête	Facultatif. Utilisez les valeurs suivantes : Importer : si vous créez un mappage. Correspondance : si vous faites correspondre un mappage existant.
recordsToReturn	Requête	Facultatif. Entrez le nombre d'enregistrements à renvoyer.
returnTotal	Requête	Facultatif. Entrez le nombre total d'enregistrements.

Exemple de demande d'API

L'exemple de demande suivant recherche le mappage existant défini par l'utilisateur ou par le système selon l'ID de mappage :

```
GET http://<host>:<port>/cmx/flat2be/{orsId}/mapping{?  
createdBy,fileId,firstRecord,purpose,recordsToReturn,returnTotal}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente le mappage existant défini par l'utilisateur ou par le système selon l'ID de mappage :

```
{  
  mappings: [  
    {  
      id: 123, name: "....", ...  
    },  
    ...  
  ]  
}
```

```
}  
}
```

Lire le mappage

L'API REST Lire le mappage renvoie le mappage existant des entités d'entreprise ou des relations selon l'ID de mappage.

L'API utilise la méthode GET.

URL de demande

L'URL Lire le mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<mapping>/<id><fileId>
```

Effectuez la demande HTTP GET suivante à l'URL Lire le mappage :

```
GET http://<host>:<port>/<context>/<orsId>/<mapping>/<id><fileId>
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les entités d'entreprise ou les relations mappées à partir de l'ID de fichier importé.

Le tableau suivant décrit les paramètres dans le corps de la demande.

Paramètre	Type	Description
mappingId	Chemin	Nom du mappage donné par l'utilisateur.
orsId	Chemin	ID de base de données du stockage de référence opérationnelle à partir duquel récupérer le mappage.
fileId	Requête	Facultatif. Spécifiez l'ID de fichier associé au mappage existant.

Exemple de demande d'API

L'exemple de demande suivant renvoie les entités d'entreprise ou relations mappées selon l'ID de mappage :

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX HTTP/1.1  
Content-Type: application/json  
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant renvoie les entités d'entreprise ou relations existantes selon l'ID de mappage :

```
HTTP/1.1 200 OK  
Server: JBoss-EAP/7  
Content-Disposition: inline;filename=f.txt  
X-Powered-By: Undertow/1  
Content-Type: application/json; charset=UTF-8  
Content-Length: 993  
  
{  
  "id" : "SVR1.1G7UX",  
  "systemName" : "Admin",  
  "mapping" : {  
    "regionalSettings" : {  
      "locale" : "en-CA",
```

```

        "datePattern" : "dd-MM-yyyy",
        "decimalSeparator" : ".",
        "thousandsSeparator" : ",",
    },
    "objects" : [ {
        "name" : "PersonView",
        "fields" : [ {
            "name" : "firstName",
            "fileColumn" : "first_name",
            "skipNulls" : false
        }, {
            "name" : "jobTitle",
            "fileColumn" : "job_title",
            "skipNulls" : false
        }, {
            "name" : "lastName",
            "fileColumn" : "last_name",
            "skipNulls" : false
        }, {
            "name" : "birthdate",
            "fileColumn" : "birth_date",
            "skipNulls" : false
        } ],
        "children" : [ ]
    } ],
    "purpose" : "IMPORT"
},
"creator" : "admin",
"createDate" : "2020-01-13T13:06:26.153-05:00",
"updatedBy" : "admin",
"lastUpdateDate" : "2020-01-13T13:06:26.153-05:00",
"system" : true
}

```

Mettre à jour le mappage

L'API REST Mettre à jour le mappage met à jour le mappage existant avec les nouvelles données.

L'API utilise la méthode PUT.

URL de demande

L'URL Mettre à jour le mappage est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<mapping>/<mappingId>
```

Effectuez la demande HTTP PUT suivante à l'URL Mettre à jour le mappage :

```
PUT http://<host>:<port>/<context>/<orsId>/<mapping>/<mappingId>
```

Paramètres

Spécifiez les paramètres permettant de mettre à jour le mappage défini par l'utilisateur dans le corps de la demande. Créez un mappage dans la demande à l'aide du format JSON ou XML.

Le tableau suivant décrit les paramètres dans le corps de la demande.

Paramètre	Type	Description
mappingId	Chemin	ID de mappage des données utilisateur.
orsId	Chemin	ID de base de données du stockage de référence opérationnelle.

Paramètre	Type	Description
mapping	Requête	Facultatif. Entrez les propriétés du mappage.
confidence	Requête	Facultatif. Niveau de confiance du mappage. Utilisez l'une des valeurs suivantes : Élevé, Moyen ou Faible.
creator	Requête	Facultatif. Votre nom d'utilisateur.
createDate	Requête	Facultatif. Date de création du mappage.
UpdatedBy	Requête	Facultatif. Si vous mettez à jour un mappage existant, entrez votre nom d'utilisateur.
lastUpdateDate	Requête	Facultatif. Date de mise à jour du mappage par l'utilisateur.
modified	Requête	Facultatif. Utilisez les valeurs suivantes : True si vous modifiez un mappage existant ou False si vous créez un mappage.
name	Requête	Facultatif. Nom du mappage.
score	Requête	Facultatif. Score de correspondance du mappage.
system	Requête	Facultatif. Utilisez les valeurs suivantes : True : si le système crée le mappage. False : si un utilisateur crée manuellement le mappage.
systemName	Requête	Facultatif. Affiche le nom du système qui crée le mappage.
objects	Requête	Facultatif. Entrez les champs souhaités pour l'objet de base d'entité d'entreprise ou de relation.
purpose	Requête	Facultatif. Utilisez les valeurs suivantes : Importer : si vous créez un mappage. Correspondance : si vous faites correspondre un mappage existant.

Exemple de demande d'API

L'exemple de demande suivant met à jour le mappage existant avec les nouvelles données.

```
PUT /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 889

{
  "system" : false,
  "name" : "my_mapping",
  "mapping" : {
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "dd-MM-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ",",
    },
    "objects" : [ {
      "name" : "PersonView",
      "fields" : [ {
        "name" : "firstName",
```

```

        "fileColumn" : "first_name",
        "skipNulls" : false
    }, {
        "name" : "jobTitle",
        "fileColumn" : "job_title",
        "skipNulls" : false
    }, {
        "name" : "lastName",
        "fileColumn" : "last_name",
        "skipNulls" : false
    }, {
        "name" : "birthdate",
        "fileColumn" : "birth_date",
        "skipNulls" : false
    }, {
        "name" : "genderCd",
        "fileColumn" : "gender",
        "skipNulls" : false
    } ]
    } ],
    "purpose" : "IMPORT"
}

```

Exemple de réponse d'API

Ce qui suit est l'exemple de réponse après la mise à jour du mappage existant avec les nouvelles données :

```

HTTP/1.1 204 No Content
Server: JBoss-EAP/7
X-Powered-By: Undertow/1

```

Transformation du fichier

L'API Transformation du fichier récupère la liste des entités d'entreprise à partir du fichier importé avant le mappage des entités d'entreprise aux champs cible.

L'API utilise la méthode POST.

URL de demande

L'API Transformation des fichiers est au format suivant :

```
http://<host>:<port>/cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Effectuez la demande HTTP POST suivante à l'URL de l'API Transformation des fichiers :

```
POST /cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Paramètres

Spécifiez les propriétés permettant de prévisualiser la liste des entités d'entreprise à partir du fichier importé avant de mapper les entités d'entreprise aux champs cible.

Le tableau suivant décrit les paramètres.

Paramètre	Description
fileId	ID de fichier permettant de générer l'aperçu.
orsId	ID du stockage de référence opérationnelle de la base de données à partir duquel prévisualiser le mappage.

Paramètre	Description
entity	Facultatif. Nombre d'entités d'entreprise ou de relations à prévisualiser.
pageSize	Facultatif. Nombre de pages à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant récupère la liste des entités d'entreprise mappées à partir du fichier importé avant le mappage des entités d'entreprise :

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?entity=Person&fileId=TEMP_SVR1.1G7UW
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 431

{
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "d MMM, yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "Person",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    } ]
  } ]
}
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche la liste des entités d'entreprise mappées à partir du fichier importé avant le mappage des entités d'entreprise aux champs cible.

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
Content-Length: 825

{
  "link" : [ ],
  "item" : [ {
    "object" : {
      "Person" : {
        "label" : "smith, alice",
        "lastName" : "smith",
        "firstName" : "alice"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, bob",
        "lastName" : "smith",
        "firstName" : "bob"
      }
    }
  } ]
}
```



```

    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, charlie",
        "lastName" : "smith",
        "firstName" : "charlie"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, dave",
        "lastName" : "smith",
        "firstName" : "dave"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, eve",
        "lastName" : "smith",
        "firstName" : "eve"
      }
    }
  }
]

```

Lire la relation

L'API REST Lire la relation renvoie les détails d'un enregistrement de relation tels que les types de partie, les ID de ligne et les noms d'affichage des deux enregistrements.

L'API utilise la méthode GET.

URL de demande

L'URL Lire la relation est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the relationship record>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the relationship record>
```

Paramètres de la requête

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
suppressLinks	Facultatif. Indique si les liens parent-enfant sont visibles dans la réponse de l'API. Définissez le paramètre sur True pour supprimer tous les liens parent-enfant dans la réponse. Définissez le paramètre sur False pour ne pas afficher les liens dans la réponse de l'API. La valeur par défaut est False.
depth	Facultatif. Nombre de niveaux enfants à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant renvoie les détails de l'enregistrement de la relation dont l'ID de ligne est 85, dont le type est `ProductGroupProductGroupIsParentOfProductProducts` :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85
```

L'exemple de demande suivant renvoie les détails avec une profondeur de 2 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85?depth=2
```

Exemple de réponse d'API

L'exemple suivant montre les détails de l'enregistrement de la relation dont l'ID de ligne est 85, dont le type est `ProductGroupProductGroupIsParentOfProductProducts` :

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json?depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "85",
  "label": "Product Group Product Group is parent of Product Products",
  "rowidRelType": "9",
  "rowidHierarchy": "3",
  "from": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json?depth=2",
        "rel": "children"
      }
    ],
    "rowidObject": "86",
    "label": "Product Group",
    "productType": "Product Group",
    "productNumber": "Presenter2",
    "productName": "Presenter",
    "productDesc": "Presenter Family",
    "productTypeCd": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
          "rel": "parent"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json?depth=2",
          "rel": "children"
        }
      ]
    }
  }
}
```

```

    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json",
      "rel": "self"
    }
  ],
  "productType": "Product Group"
}
},
"to": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "parent"
    }
  ],
  "rowidObject": "66",
  "label": "Products",
  "productType": "Product",
  "productNumber": "931307-0403",
  "productName": "2.4 GHz Cordless Presenter",
  "productDesc": "A cordless presenter to streamline your delivery.",
  "productTypeCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json?depth=2",
        "rel": "children"
      }
    ],
    "productType": "Presenter"
  }
}
}
}

```

Créer une relation

L'API REST Créer une relation crée une relation entre les enregistrements spécifiés. Pour créer une relation entre des enregistrements, des relations doivent exister entre les entités d'entreprise auxquelles les enregistrements appartiennent. Par exemple, si vous voulez spécifier une relation entre Informatica et John Smith, une relation doit exister entre les entités d'entreprise Organisation et Personne. Vous devez envoyer les données de la relation dans le corps de demande.

L'API utilise les méthodes PUT et POST.

URL de demande

L'URL REST Créer une relation est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the source system>
```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez la demande HTTP POST ou PUT suivante à l'URL :

```
POST http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the source system>
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
Content-Type: application/<json/xml>
```

Paramètres d'URL

Le nom du système source est un paramètre requis dans l'URL de demande.

Corps de demande

Envoyez les données pour l'enregistrement de relation dans le corps de demande REST. Utilisez le format JSON ou XML pour envoyer les données. Indiquez les valeurs des paramètres requis dans le corps de demande.

Exemple de demande d'API

L'exemple de demande suivant crée la relation `OrganizationEmploysPerson` entre une entité d'entreprise Organisation dont l'ID de ligne est 101 et une entité d'entreprise Personne dont l'ID de ligne est 1101 :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson?systemName=SFA
Content-Type: application/json

{
  "from": {
    "rowidObject": "101"
  },
  "to": {
    "rowidObject": "1101"
  },
  "relName": "Documentation",
  "relDesc": "Writer"
}
```

La relation `OrganizationEmploysPerson` définit une relation d'une entité d'entreprise Organisation à une entité d'entreprise Personne. L'élément `from` spécifie l'enregistrement où commence la relation et l'élément `to` spécifie l'enregistrement à laquelle elle se termine.

Exemple de réponse d'API

L'exemple de réponse suivant montre l'en-tête et le corps de la réponse après la création réussie d'une relation entre une entité d'entreprise Organisation dont l'ID de ligne est 101 et une entité d'entreprise Personne dont l'ID de ligne est 1101 :

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
    }
  }
}
```

```

        "sourceKey": "SVR1.1E7UW"
    }-
    "rowidObject": "414721"
    "from": {
        "key": {
            "rowid": "101 "
        }-
        "rowidObject": "101 "
    }-
    "to": {
        "key": {
            "rowid": "1101 "
        }-
    }-
    "rowidObject": "1101 "
    }-
}

```

Mettre à jour une relation

L'API REST Mettre à jour une relation met à jour la relation entre deux enregistrements. L'API met à jour les attributs supplémentaires définis pour la relation.

L'API utilise les méthodes POST et PUT.

URL de demande

L'URL Mettre à jour la relation est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez l'appel HTTP POST ou PUT suivant à l'URL :

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande.

Corps de demande

Envoyez les mises à jour à l'enregistrement de relation dans le corps de demande. Utilisez le format JSON ou XML pour envoyer les données. Indiquez les valeurs des paramètres requis dans le corps de demande.

Exemple de demande d'API

La relation dont l'ID de ligne est 414721 est une relation OrganizationEmploysPerson entre l'entité Organisation dont l'ID de ligne est 101 et l'entité Personne dont l'ID de ligne est 1101.

L'exemple de demande suivant met à jour l'enregistrement de relation dont l'ID de ligne est 414721 :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?
systemName=SFA
Content-Type: application/json
{
    "from": {
        "rowidObject": "101"
    }
}

```

```

    },
    "to": {
      "rowidObject": "1101      "
    },
    "relName": "Development",
    "relDesc": "Software Engineer",

    "$original":
    { "relName": "Documentation",
      "relDesc": "Writer" }
  }
}

```

Exemple de réponse d'API

L'exemple de réponse suivant est envoyé suite à la mise à jour réussie de la relation dont l'ID de ligne est 414721 :

```

{
  "OrganizationEmployesPerson": {
    "key": {
      "rowid": "414721"
      "sourceKey": "SVR1.1E7UW"
    }-
    "rowidObject": "414721"
    "from": {
      "key": {
        "rowid": "101"
      }-
      "rowidObject": "101"
    }-
    "to": {
      "key": {
        "rowid": "1101 "
      }-
      "rowidObject": "1101 "
    }-
  }-
}

```

Supprimer une relation

L'API REST Supprimer une relation supprime la relation entre deux enregistrements.

L'API utilise la méthode DELETE.

URL de demande

L'URL Supprimer la relation est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the relationship
record>?systemName=<name of the source system>

```

Remarque: Le nom du système source est un paramètre requis dans l'URL.

Effectuez la demande HTTP DELETE suivante à l'URL Supprimer :

```

DELETE http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the
relationship record>?systemName=<name of the source system>

```

Paramètre de requête

Le nom du système source est un paramètre d'URL requis. Utilisez le paramètre `systemName` pour spécifier le système source.

Exemple de demande d'API

L'exemple de demande suivant supprime l'enregistrement de relation dont l'ID de ligne est 414721 :

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?systemName=SFA
```

Exemple de réponse d'API

L'exemple de réponse suivant montre la réponse après la suppression réussie de l'enregistrement de relation dont l'ID de ligne est 414721 :

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
    }-
    "rowidObject": "414721"
  }-
}
```

Obtenir les enregistrements associés

L'API REST Obtenir les enregistrements associés renvoie une liste d'enregistrements associés à un enregistrement racine spécifique en fonction des relations configurées. L'API renvoie également les détails des relations.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les enregistrements associés est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les enregistrements associés :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de requête à l'URL de la demande.

Le tableau suivant répertorie les paramètres de requête que vous pouvez utiliser :

Paramètre	Description
recordsToReturn	Nombre d'enregistrements de l'enfant <code>plusieurs</code> que vous voulez lire.
searchToken	Jeton de recherche permettant d'obtenir les pages suivantes de l'ensemble de résultats.
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur <code>True</code> pour obtenir le nombre d'enregistrements dans l'ensemble de résultats. La valeur par défaut est <code>False</code> .

Paramètres de filtre

Vous pouvez ajouter des paramètres à l'URL afin de filtrer les enregistrements associés.

Le tableau suivant répertorie les paramètres de filtre que vous pouvez utiliser :

Paramètre	Description
recordStates	Liste d'états des enregistrements à récupérer séparés par des virgules. Les états d'enregistrements pris en charge sont ACTIVE, PENDING et DELETED. La valeur par défaut est ACTIVE. Par exemple, la requête <code>/Party/123?action=related&recordStates=ACTIVE,PENDING</code> renvoie des enregistrements actifs ou en attente.
entityLabel	Libellé de l'entité.
relationshipLabel	Libellé de la relation.
entityType	Liste de types d'entités séparés par des virgules. Par exemple, la liste <code>entityType=Person,Organization</code> renvoie des types d'entités <code>Personne</code> et <code>Organisation</code> .
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, la liste <code>relationshipType=Employee,Employer</code> renvoie les enregistrements associés des types de relations <code>Employé</code> et <code>Employeur</code> .

Remarque: Si vous spécifiez plusieurs conditions de filtre, le résultat contient tous les enregistrements qui satisfont à la condition AND.

Corps de réponse

Le corps de réponse contient la liste des enregistrements associés, les détails des enregistrements associés, les relations et un jeton de recherche. Utilisez le jeton de recherche pour récupérer les pages suivantes des résultats.

Exemple de demande d'API

L'exemple de demande suivant récupère les enregistrements et les relations associés configurés pour l'entité d'entreprise Organisation dont l'ID de ligne est 101 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Organization/101?action=related
```

Exemple de réponse d'API

L'exemple de réponse suivant montre les enregistrements et les relations associés pour le type d'entité d'entreprise Organisation dont l'ID de ligne est 101 :

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1H7YB",
  "relatedEntity": [
    {
      "businessEntity": {
        "SecurePerson": {
          "link": [
            {
              "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/"
            }
          ]
        }
      }
    }
  ]
}
```



```

SecurePerson/1101",
    "rel": "self"
  }
],
"rowidObject": "1101",
"creator": "admin",
"createDate": "2008-11-11T21:22:20-08:00",
"updatedBy": "admin",
"lastUpdateDate": "2012-03-29T19:03:19-07:00",
"consolidationInd": "1",
"lastRowidSystem": "SYS0",
"dirtyIndicator": "0",
"interactionId": "20003000",
"hubStateInd": "1",
"partyType": "Person",
"lastName": "Obama",
"firstName": "Barack"
},
"entityLabel": "Obama,Barack",
"relationshipLabel": "Organization employees SecurePerson",
"relationship": {
  "rowidObject": "414721",
  "creator": "admin",
  "createDate": "2016-10-17T01:58:12.436-07:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
  "consolidationInd": "4",
  "lastRowidSystem": "SFA",
  "interactionId": "1476866426786",
  "hubStateInd": "1",
  "rowidRelType": "101",
  "rowidHierarchy": "1",
  "relName": "Documentation",
  "relDesc": "Writer"
},
"entityType": "SecurePerson",
"relationshipType": "OrganizationEmployeesSecurePerson"
},
{
  "businessEntity": {
    "SecurePerson": {
      "link": [
        {
          "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/
SecurePerson/114",
          "rel": "self"
        }
      ],
      "rowidObject": "114",
      "creator": "admin",
      "createDate": "2008-08-11T23:00:55-07:00",
      "updatedBy": "Admin",
      "lastUpdateDate": "2008-08-12T02:59:17-07:00",
      "consolidationInd": "1",
      "lastRowidSystem": "Legacy",
      "dirtyIndicator": "0",
      "hubStateInd": "1",
      "partyType": "Person",
      "lastName": "HERNANDEZ",
      "displayName": "ALEJANDRO HERNANDEZ",
      "firstName": "ALEJANDRO"
    }
  },
  "entityLabel": "HERNANDEZ,ALEJANDRO",
  "relationshipLabel": "Organization employees SecurePerson",
  "relationship": {
    "rowidObject": "434721",
    "creator": "admin",
    "createDate": "2016-10-19T01:49:03.415-07:00",
    "updatedBy": "admin",

```

```

        "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
        "consolidationInd": "4",
        "lastRowidSystem": "SFA",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
    },
    "entityType": "SecurePerson",
    "relationshipType": "OrganizationEmployeeSecurePerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_uil/Person/1101",
                    "rel": "self"
                }
            ],
            "rowidObject": "1101",
            "creator": "admin",
            "createDate": "2008-11-11T21:22:20-08:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2012-03-29T19:03:19-07:00",
            "consolidationInd": "1",
            "lastRowidSystem": "SYS0",
            "dirtyIndicator": "0",
            "interactionId": "20003000",
            "hubStateInd": "1",
            "partyType": "Person",
            "lastName": "Obama",
            "firstName": "Barack"
        }
    },
    "entityLabel": "Obama,Barack",
    "relationshipLabel": "Organization employes Person",
    "relationship": {
        "rowidObject": "414721",
        "creator": "admin",
        "createDate": "2016-10-17T01:58:12.436-07:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
        "consolidationInd": "4",
        "lastRowidSystem": "SFA",
        "interactionId": "1476866426786",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
    },
    "entityType": "Person",
    "relationshipType": "OrganizationEmployeePerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_uil/Person/114",
                    "rel": "self"
                }
            ],
            "rowidObject": "114",
            "creator": "admin",
            "createDate": "2008-08-11T23:00:55-07:00",
            "updatedBy": "Admin",
            "lastUpdateDate": "2008-08-12T02:59:17-07:00",
            "consolidationInd": "1",

```

```

        "lastRowidSystem": "Legacy",
        "dirtyIndicator": "0",
        "hubStateInd": "1",
        "partyType": "Person",
        "lastName": "HERNANDEZ",
        "displayName": "ALEJANDRO HERNANDEZ",
        "statusCd": "A",
        "firstName": "ALEJANDRO"
    },
    "entityLabel": "HERNANDEZ,ALEJANDRO",
    "relationshipLabel": "Organization employes Person",
    "relationship": {
        "rowidObject": "434721",
        "creator": "admin",
        "createDate": "2016-10-19T01:49:03.415-07:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
        "consolidationInd": "4",
        "lastRowidSystem": "SFA",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
    },
    "entityType": "Person",
    "relationshipType": "OrganizationEmployeesPerson"
}
]
}

```

Exporter les entités d'entreprise liées

Entrez une brève description de votre concept ici (obligatoire).

Ceci est le début du concept.

URL de demande

Entrez une brève description de votre concept ici (obligatoire).

Ceci est le début du concept.

Exemple de demande d'API

Entrez une brève description de votre concept ici (obligatoire).

Ceci est le début du concept.

Exemple de réponse d'API

Entrez une brève description de votre concept ici (obligatoire).

Ceci est le début du concept.

Liste Hiérarchies

L'API REST Répertoire des hiérarchies renvoie toutes les hiérarchies. Vous pouvez également utiliser l'API permettant de renvoyer les hiérarchies qui contiennent une entité d'entreprise spécifiée à tous les niveaux.

L'API utilise la méthode GET.

URL de demande

L'URL Répertoire les hiérarchies peut être aux formats suivants :

URL permettant de renvoyer toutes les hiérarchies

Utilisez l'URL suivante pour répertoire toutes les hiérarchies :

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy
```

URL permettant de renvoyer toutes les hiérarchies qui contiennent une entité d'entreprise spécifiée

Utilisez l'URL suivante pour répertoire toutes les hiérarchies qui contiennent une entité d'entreprise spécifiée à tous les niveaux :

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy?entityName=<entity name>
```

Exemple de demande d'API

L'exemple de demande suivant répertoire toutes les hiérarchies :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy
```

L'exemple de demande suivant répertoire toutes les hiérarchies contenant l'entité d'entreprise Produit :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy?entityName=Product
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les hiérarchies contenant l'entité d'entreprise Produit :

```
{
  "link": [],
  "item": [
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "update": {
          "allowed": true
        }
      },
      "root": "ProductGroup",
      "name": "Product"
    }
  ]
}
```

Obtenir les métadonnées de la hiérarchie

L'API REST Obtenir les métadonnées de la hiérarchie renvoie les métadonnées d'une hiérarchie.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les métadonnées de la hiérarchie est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy/<hierarchy name>
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les métadonnées de la hiérarchie :

```
GET http://<host>:<port>/<context>/<database ID>/metadata/hierarchy/<hierarchy name>
```

Exemple de demande d'API

L'exemple de demande suivant renvoie les métadonnées pour une hiérarchie :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy/ODI
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les métadonnées pour une hiérarchie :

```
{
  "relationship": [
    {
      "field": [
        {
          "allowedValues": [
            "2"
          ],
          "name": "rowidRelType",
          "label": "Rowid Rel Type",
          "dataType": "String",
          "length": 14,
          "readOnly": false,
          "required": false,
          "system": false,
          "trust": false,
          "applyNullValues": false,
          "displayFormat": "DATETIME_LONG_FORMAT",
          "filterable": true,
          "sortable": true,
          "lookup": {
            "link": [
              {
                "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/id-
label/IsOdiParentOf/rowidRelType",
                "rel": "lookup"
              },
              {
                "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/
object/IsOdiParentOf/rowidRelType",
                "rel": "list"
              }
            ],
            "object": "RboRelType",
            "key": "rowidObject",
            "value": "displayName"
          }
        }
      ],
      "allowedValues": [
        "1"
      ],
      "name": "rowidHierarchy",
      "label": "Rowid Hierarchy",
      "dataType": "String",
      "length": 14,
      "readOnly": false,
      "required": false,
      "system": false,
      "trust": false,
      "applyNullValues": false,
      "displayFormat": "DATETIME_LONG_FORMAT",
      "filterable": true,
      "sortable": true,
      "lookup": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/id-
label/IsOdiParentOf/rowidHierarchy",
            "rel": "lookup"
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/
object/IsOdiParentOf/rowidHierarchy",
      "rel": "list"
    }
  ],
  "object": "RboHierarchy",
  "key": "rowidObject",
  "value": "displayName"
}
},
{
  "name": "relName",
  "label": "Rel Name",
  "dataType": "String",
  "length": 50,
  "trust": false,
  "applyNullValues": false,
  "displayFormat": "DATETIME_LONG_FORMAT",
  "filterable": true,
  "sortable": true
},
{
  "name": "relDesc",
  "label": "Rel Desc",
  "dataType": "String",
  "length": 200,
  "trust": false,
  "applyNullValues": false,
  "displayFormat": "DATETIME_LONG_FORMAT",
  "filterable": true,
  "sortable": true
},
{
  "name": "consolidationInd",
  "label": "Consolidation Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "creator",
  "label": "Creator",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "interactionId",
  "label": "Interaction Id",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{

```

```

    "name": "updatedBy",
    "label": "Updated By",
    "dataType": "String",
    "length": 50,
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  },
  {
    "name": "lastUpdateDate",
    "label": "Last Update Date",
    "dataType": "Date",
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  },
  {
    "name": "lastRowidSystem",
    "label": "Last Rowid System",
    "dataType": "String",
    "length": 14,
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  },
  {
    "name": "dirtyIndicator",
    "label": "Dirty Indicator",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  },
  {
    "name": "deletedBy",
    "label": "Deleted By",
    "dataType": "String",
    "length": 50,
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  },
  {
    "name": "deletedInd",
    "label": "Deleted Indicator",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true,
    "trust": false,
    "applyNullValues": false,
    "filterable": true,
    "sortable": true
  }
]

```

```

        "name": "hubStateInd",
        "label": "Hub State Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true,
        "trust": false,
        "applyNullValues": false,
        "filterable": true,
        "sortable": true
    },
    {
        "name": "deletedDate",
        "label": "Deleted Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true,
        "trust": false,
        "applyNullValues": false,
        "filterable": true,
        "sortable": true
    },
    {
        "name": "rowidObject",
        "label": "Rowid Object",
        "dataType": "String",
        "length": 14,
        "readOnly": true,
        "system": true,
        "trust": false,
        "applyNullValues": false,
        "filterable": true,
        "sortable": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true,
        "trust": false,
        "applyNullValues": false,
        "filterable": true,
        "sortable": true
    },
    {
        "name": "createDate",
        "label": "Create Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true,
        "trust": false,
        "applyNullValues": false,
        "filterable": true,
        "sortable": true
    }
],
"contentMetadata": [
    {
        "operations": {
            "read": {
                "allowed": true
            },
            "create": {
                "allowed": true
            },
            "update": {
                "allowed": true
            },
            "delete": {

```



```

        "allowed":true
      }
    },
    "name":"XREF"
  }
],
"name":"IsOdiParentOf",
"label":"is ODI parent of",
"color":"#990066",
"direction":"ENTITY_1_TO_ENTITY_2",
"bidirectional":false,
"from":{
  "dataType":"BusinessEntity",
  "required":true,
  "lookup":{
    "object":"Organization"
  }
},
"to":{
  "dataType":"BusinessEntity",
  "required":true,
  "lookup":{
    "object":"Organization"
  }
},
"hierarchy":"ODI"
}
],
"operations":{
  "read":{
    "allowed":true
  },
  "update":{
    "allowed":true
  }
},
"root":"Organization",
"name":"ODI"
}

```

Obtenir le chemin de la hiérarchie

L'API REST Obtenir les chemins de la hiérarchie renvoie les chemins d'un enregistrement d'entité d'entreprise vers la racine de la hiérarchie.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les chemins de la hiérarchie est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/path
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les chemins de la hiérarchie :

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/path
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, <code>relationshipType=Employee,Employer</code> list renvoie les enregistrements associés des types de relations <code>Employee</code> et <code>Employer</code> .
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur <code>True</code> pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

L'exemple de demande suivant répertorie les chemins d'un enregistrement d'entité d'entreprise vers la racine de la hiérarchie :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/  
Organization/35/path
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les chemins de l'enregistrement d'entité d'entreprise vers la racine de la hiérarchie :

```
{  
  "link": [],  
  "item": [  
    {  
      "businessEntity": {  
        "organization": {  
          "link": [  
            {  
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/  
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",  
              "rel": "icon"  
            },  
            {  
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/  
Organization/584",  
              "rel": "self"  
            }  
          ]  
        }  
      }  
    }  
  ]  
}
```

```

    ],
    "rowidObject": "584",
    "label": "Time Warner Inc",
    "partyType": "Organization",
    "displayName": "Time Warner Inc",
    "dunsNumber": "799527630"
  }
},
"entityLabel": "Time Warner Inc",
"entityType": "Organization",
"depth": 2,
"object": {
  "businessEntity": {
    "Organization": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel": "icon"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/55",
          "rel": "self"
        }
      ]
    },
    "rowidObject": "55",
    "label": "Historic TW Inc",
    "partyType": "Organization",
    "displayName": "Historic TW Inc",
    "dunsNumber": "958466278"
  }
},
"entityLabel": "Historic TW Inc",
"relationshipLabel": "is ODI parent of",
"relationship": {
  "rowidObject": "834",
  "label": "is ODI parent of",
  "rowidRelType": "2",
  "rowidHierarchy": "1",
  "relName": "Parent",
  "relDesc": "Parent",
  "from": {
    "rowidObject": "584"
  },
  "to": {
    "rowidObject": "55"
  }
},
"entityType": "Organization",
"relationshipType": "IsOdiParentOf",
"object": {
  "businessEntity": {
    "Organization": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel": "icon"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/35",
          "rel": "self"
        }
      ]
    },
    "rowidObject": "35",
    "effectivePeriod": {
  },
  "label": "People Magazine",

```

```

        "partyType": "Organization",
        "displayName": "People Magazine",
        "dunsNumber": "011584096"
      },
      {
        "entityLabel": "People Magazine",
        "relationshipLabel": "is ODI parent of",
        "relationship": {
          "rowidObject": "889",
          "label": "is ODI parent of",
          "rowidRelType": "2",
          "rowidHierarchy": "1",
          "relName": "Parent",
          "relDesc": "Parent",
          "from": {
            "rowidObject": "55"
          },
          "to": {
            "rowidObject": "35"
          }
        }
      },
      {
        "entityType": "Organization",
        "relationshipType": "IsOdiParentOf"
      }
    ]
  }
}

```

Obtenir les parents

L'API REST Obtenir les parents renvoie les relations parentes directes pour un enregistrement d'entité d'entreprise.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les parents est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/parent
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les parents :

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/parent
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .

Paramètre	Description
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, <code>relationshipType=Employee,Employer list</code> renvoie les enregistrements associés des types de relations <code>Employee</code> et <code>Employer</code> .
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur <code>True</code> pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

L'exemple de demande suivant répertorie les relations parentes directes pour un enregistrement d'entité d'entreprise :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/
Organization/55/parent
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les relations parentes directes pour un enregistrement d'entité d'entreprise :

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1G7VH",
  "relatedEntity": [
    {
      "businessEntity": {
        "organization": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/584",
              "rel": "self"
            },
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
              "rel": "icon"
            }
          ]
        },
        "rowidObject": "584",
        "label": "Time Warner Inc",
        "partyType": "Organization",
        "displayName": "Time Warner Inc",
        "dunsNumber": "799527630"
      }
    },
    {
      "entityLabel": "Time Warner Inc",
      "relationshipLabel": "is ODI parent of",
      "relationship": {
```

```

        "rowidObject": "834",
        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "Parent",
        "relDesc": "Parent",
        "from": {
            "rowidObject": "584"
        },
        "to": {
            "rowidObject": "55"
        }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
}
]
}

```

Obtenir les enfants

L'API REST Obtenir les enfants renvoie les relations enfants pour un enregistrement d'entité d'entreprise.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les enfants est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/children

```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les enfants :

```

GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/children

```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.

Paramètre	Description
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, relationshipType=Employee,Employer list renvoie les enregistrements associés des types de relations Employee et Employer.
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur True pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

L'exemple de demande suivant répertorie les relations enfants pour un enregistrement d'entité d'entreprise :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/
Organization/55/children
```

Exemple de réponse d'API

L'exemple de réponse suivant présente les relations enfants pour un enregistrement d'entité d'entreprise :

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1G7VF",
  "relatedEntity": [
    {
      "businessEntity": {
        "Organization": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
              "rel": "icon"
            },
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/593",
              "rel": "self"
            }
          ]
        },
        "rowidObject": "593",
        "label": "Turner Broadcasting System Inc",
        "partyType": "Organization",
        "displayName": "Turner Broadcasting System Inc",
        "dunsNumber": "003319068"
      }
    },
    {
      "entityLabel": "Turner Broadcasting System Inc",
      "relationshipLabel": "is ODI parent of",
      "relationship": {
        "rowidObject": "887",
        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "Parent",
        "relDesc": "Parent",
        "from": {
          "rowidObject": "55"
        },
        "to": {
          "rowidObject": "593"
        }
      }
    }
  ]
}
```

```

    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
  },
  {
    "businessEntity": {
      "Organization": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
            "rel": "icon"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/414",
            "rel": "self"
          }
        ],
        "rowidObject": "414",
        "label": "Historic TW Inc",
        "partyType": "Organization",
        "displayName": "Historic TW Inc",
        "dunsNumber": "007305290"
      }
    },
    "entityLabel": "Historic TW Inc",
    "relationshipLabel": "is ODI parent of",
    "relationship": {
      "rowidObject": "888",
      "label": "is ODI parent of",
      "rowidRelType": "2",
      "rowidHierarchy": "1",
      "relName": "HQ",
      "relDesc": "HQ",
      "from": {
        "rowidObject": "55"
      },
      "to": {
        "rowidObject": "414"
      }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
  },
  {
    "businessEntity": {
      "Organization": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/35",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
            "rel": "icon"
          }
        ],
        "rowidObject": "35",
        "label": "People Magazine",
        "partyType": "Organization",
        "displayName": "People Magazine",
        "dunsNumber": "011584096"
      }
    },
    "entityLabel": "People Magazine",
    "relationshipLabel": "is ODI parent of",
    "relationship": {
      "rowidObject": "889",

```



```

        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "Parent",
        "relDesc": "Parent",
        "from": {
            "rowidObject": "55"
        },
        "to": {
            "rowidObject": "35"
        }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
},
{
    "businessEntity": {
        "Organization": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/391",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel": "icon"
                }
            ],
            "rowidObject": "391",
            "label": "Historic TW Inc",
            "partyType": "Organization",
            "displayName": "Historic TW Inc",
            "dunsNumber": "011816704"
        }
    },
    "entityLabel": "Historic TW Inc",
    "relationshipLabel": "is ODI parent of",
    "relationship": {
        "rowidObject": "890",
        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "HQ",
        "relDesc": "HQ",
        "from": {
            "rowidObject": "55"
        },
        "to": {
            "rowidObject": "391"
        }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
},
{
    "businessEntity": {
        "Organization": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/448",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel": "icon"
                }
            ],

```

```

        "rowidObject": "448",
        "label": "Historic TW Inc",
        "partyType": "Organization",
        "displayName": "Historic TW Inc",
        "dunsNumber": "069110711"
    },
    {
        "entityLabel": "Historic TW Inc",
        "relationshipLabel": "is ODI parent of",
        "relationship": {
            "rowidObject": "891",
            "label": "is ODI parent of",
            "rowidRelType": "2",
            "rowidHierarchy": "1",
            "relName": "HQ",
            "relDesc": "HQ",
            "from": {
                "rowidObject": "55"
            },
            "to": {
                "rowidObject": "448"
            }
        },
        "entityType": "Organization",
        "relationshipType": "IsOdiParentOf"
    },
    {
        "businessEntity": {
            "Organization": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                        "rel": "icon"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/545",
                        "rel": "self"
                    }
                ],
                "rowidObject": "545",
                "label": "Historic TW Inc",
                "partyType": "Organization",
                "displayName": "Historic TW Inc",
                "dunsNumber": "072077139"
            }
        },
        "entityLabel": "Historic TW Inc",
        "relationshipLabel": "is ODI parent of",
        "relationship": {
            "rowidObject": "892",
            "label": "is ODI parent of",
            "rowidRelType": "2",
            "rowidHierarchy": "1",
            "relName": "HQ",
            "relDesc": "HQ",
            "from": {
                "rowidObject": "55"
            },
            "to": {
                "rowidObject": "545"
            }
        },
        "entityType": "Organization",
        "relationshipType": "IsOdiParentOf"
    },
    {
        "businessEntity": {
            "Organization": {
                "link": [

```

```

        {
            "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/349",
            "rel":"self"
        },
        {
            "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
            "rel":"icon"
        }
    ],
    "rowidObject":"349",
    "label":"Historic TW Inc",
    "partyType":"Organization",
    "displayName":"Historic TW Inc",
    "dunsNumber":"074056123"
},
{
    "entityLabel":"Historic TW Inc",
    "relationshipLabel":"is ODI parent of",
    "relationship":{
        "rowidObject":"893",
        "label":"is ODI parent of",
        "rowidRelType":"2",
        "rowidHierarchy":"1",
        "relName":"HQ",
        "relDesc":"HQ",
        "from":{
            "rowidObject":"55"
        },
        "to":{
            "rowidObject":"349"
        }
    },
    "entityType":"Organization",
    "relationshipType":"IsOdiParentOf"
},
{
    "businessEntity":{
        "Organization":{
            "link":{
                {
                    "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                },
                {
                    "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/298",
                    "rel":"self"
                }
            ],
            "rowidObject":"298",
            "label":"WF Cinema Holdings LP",
            "partyType":"Organization",
            "displayName":"WF Cinema Holdings LP",
            "dunsNumber":"075408711"
        }
    },
    "entityLabel":"WF Cinema Holdings LP",
    "relationshipLabel":"is ODI parent of",
    "relationship":{
        "rowidObject":"894",
        "label":"is ODI parent of",
        "rowidRelType":"2",
        "rowidHierarchy":"1",
        "relName":"Parent",
        "relDesc":"Parent",
        "from":{
            "rowidObject":"55"
        },
    },

```

```

        "to":{
          "rowidObject":"298"
        },
        "entityType":"Organization",
        "relationshipType":"IsOdiParentOf"
      },
      {
        "businessEntity":{
          "Organization":{
            "link":[
              {
                "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                "rel":"icon"
              },
              {
                "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/596",
                "rel":"self"
              }
            ],
            "rowidObject":"596",
            "label":"Historic TW Inc",
            "partyType":"Organization",
            "displayName":"Historic TW Inc",
            "dunsNumber":"076749246"
          }
        },
        "entityLabel":"Historic TW Inc",
        "relationshipLabel":"is ODI parent of",
        "relationship":{
          "rowidObject":"895",
          "label":"is ODI parent of",
          "rowidRelType":"2",
          "rowidHierarchy":"1",
          "relName":"HQ",
          "relDesc":"HQ",
          "from":{
            "rowidObject":"55"
          },
          "to":{
            "rowidObject":"596"
          }
        },
        "entityType":"Organization",
        "relationshipType":"IsOdiParentOf"
      },
      {
        "businessEntity":{
          "Organization":{
            "link":[
              {
                "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                "rel":"icon"
              },
              {
                "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/699",
                "rel":"self"
              }
            ],
            "rowidObject":"699",
            "label":"Historic TW Inc",
            "partyType":"Organization",
            "displayName":"Historic TW Inc",
            "dunsNumber":"083115568"
          }
        },
        "entityLabel":"Historic TW Inc",

```

```

    "relationshipLabel": "is ODI parent of",
    "relationship": {
      "rowidObject": "896",
      "label": "is ODI parent of",
      "rowidRelType": "2",
      "rowidHierarchy": "1",
      "relName": "HQ",
      "relDesc": "HQ",
      "from": {
        "rowidObject": "55"
      },
      "to": {
        "rowidObject": "699"
      }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
  }
]
}

```

Exporter la hiérarchie

L'API REST Exporter la hiérarchie exporte une hiérarchie au format CSV.

L'API utilise la méthode GET.

URL de demande

L'URL Exporter la hiérarchie est au format suivant :

```

http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/exportHierarchy

```

Effectuez la demande HTTP GET suivante à l'URL Exporter la hiérarchie :

```

GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/exportHierarchy

```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.

Paramètre	Description
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, relationshipType=Employee,Employer list renvoie les enregistrements associés des types de relations Employee et Employer.
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur True pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

L'exemple de demande suivant exporte la hiérarchie d'un enregistrement d'entité d'entreprise :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/
Organization/584/exportHierarchy
```

Exemple de réponse d'API

L'exemple de réponse suivant présente la hiérarchie exportée au format CSV :

```
Level,Business Entity,ID,Label,Parent ID,Parent Label,Type of Relationship to
Parent,issuingCompany,expirationYear,accountType,accountNumber,securityCode,expirationMon
th,cardholderName,productType,productNumber,productName,inceptionDate,productDesc,groupTy
pe,groupName,partyType,lastName,displayName,middleName,birthdate,firstName,dunsNumber
Root,Organization,584,Time Warner Inc,,,,,,,,,,,,,Organization,,Time Warner Inc,,,,
799527630
1,Organization,55,Historic TW Inc,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,958466278
1,Organization,206,Courtroom Television Network LLC,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Courtroom Television Network LLC,,,,043905707
1,Organization,674,Rebellion Pictures LLC,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Rebellion Pictures LLC,,,,557414278
1,Organization,881,"Time Warner Cable Programming, Inc",584,Time Warner Inc,is ODI
parent of,,,,,,,,,,,,,Organization,,,"Time Warner Cable Programming, Inc",,,,787474949
...
```

Exporter les enfants et les parents directs

L'API REST Exporter les enfants et les parents directs exporte les relations enfants et parentes directes d'un enregistrement d'entité d'entreprise au format CSV.

L'API utilise la méthode GET.

URL de demande

L'URL Exporter les enfants et les parents directs est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/exportRelated
```

Effectuez la demande HTTP GET suivante à l'URL Exporter les enfants et les parents directs :

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/exportRelated
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, <code>relationshipType=Employee,Employer</code> list renvoie les enregistrements associés des types de relations <code>Employee</code> et <code>Employer</code> .
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur <code>True</code> pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

L'exemple de demande suivant exporte les relations enfants et parentes directes pour un enregistrement d'entité d'entreprise :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/
Organization/55/exportRelated
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les relations enfants et parentes directes exportées pour un enregistrement d'entité d'entreprise au format CSV :

```
Level,Business Entity,ID,Label,Parent ID,Parent Label,Type of Relationship to
Parent,issuingCompany,expirationYear,accountType,accountNumber,securityCode,expirationMon
th,cardholderName,productType,productNumber,productName,inceptionDate,productDesc,groupTy
pe,groupName,partyType,lastName,displayName,middleName,birthdate,firstName,dunsNumber
Root,Organization,584,Time Warner Inc,,,,,,,,,,,,,Organization,,Time Warner Inc,,,,
799527630
1,Organization,55,Historic TW Inc,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,958466278
2,Organization,593,Turner Broadcasting System Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Turner Broadcasting System Inc,,,,003319068
2,Organization,414,Historic TW Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,007305290
2,Organization,35,People Magazine,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,People Magazine,,,,011584096
2,Organization,391,Historic TW Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,011816704
...
```

Obtenir les modifications de la hiérarchie

L'API Obtenir les modifications de la hiérarchie récupère les modifications en attente d'une hiérarchie.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les modifications de la hiérarchie est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/hierarchy/{hierarchy name}/changes
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les modifications de la hiérarchie :

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/{hierarchy name}/changes
```

Paramètres de la requête

Vous pouvez ajouter les paramètres de la requête à l'URL de la demande pour filtrer les relations.

Le tableau suivant décrit les paramètres de la requête :

Paramètre	Description
effectiveDate	Date à laquelle vous souhaitez récupérer les relations.
interactionId	ID de l'interaction.
readLabelsOnly	Indique s'il faut renvoyer uniquement les libellés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
readSystemFields	Indique si les champs système doivent être renvoyés dans le résultat. Les valeurs sont <code>True</code> ou <code>False</code> .
recordsToReturn	Spécifie le nombre de lignes à renvoyer.
rejectInteractionId	ID de l'interaction pour le rejet des modifications.
relationshipType	Liste de types de relations séparés par des virgules. Par exemple, <code>relationshipType=Employee,Employer</code> list renvoie les enregistrements associés des types de relations <code>Employee</code> et <code>Employer</code> .
returnTotal	Renvoie le nombre d'enregistrements dans l'ensemble de résultats. Définissez ce paramètre sur <code>True</code> pour obtenir le nombre d'enregistrements dans l'ensemble de résultats.
rootId	ID de la racine de la hiérarchie.

Exemple de demande d'API

La demande suivante permet de renvoyer les modifications en attente d'une hiérarchie filtrée selon l'ID d'interaction, l'ID racine et l'ID d'interaction et l'ID d'interaction de rejet :

```
GET http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/hierarchy/Product/changes?
interactionId=480003&rootId=165&rejectInteractionId=480004
```


Exemple de réponse d'API

L'exemple de réponse suivant présente les modifications en attente :

```
{
  "interactionId":"480003",
  "rejectInteractionId":"480004",
  "type":"hierarchy",
  "name":"Product",
  "objects":[
    {
      "before":[
        ],
      "after":[
        {
          "businessEntity":{
            "ProductGroup":{
              "link":[
                {
                  "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                  "rel":"icon"
                }
              ],
              "rowidObject":"165",
              "creator":"admin",
              "createDate":"2020-01-13T13:07:35.128-05:00",
              "updatedBy":"admin",
              "lastUpdateDate":"2020-01-13T13:07:35.128-05:00",
              "consolidationInd":4,
              "lastRowidSystem":"PRODUCT",
              "hubStateInd":1,
              "label":"Vehicles",
              "productType":"Product Group",
              "productName":"Vehicles"
            }
          },
          "entityLabel":"Vehicles",
          "entityType":"ProductGroup",
          "depth":1,
          "object":{
            "businessEntity":{
              "ProductGroup":{
                "link":[
                  {
                    "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                  }
                ],
                "rowidObject":"166",
                "label":"Scooter",
                "productType":"Product Group",
                "productName":"Scooter"
              }
            },
            "entityLabel":"Scooter",
            "relationshipLabel":"Product Group is parent of Product Group",
            "relationship":{
              "rowidObject":"124",
              "creator":"datasteward",
              "createDate":"2020-01-13T13:07:35.864-05:00",
              "updatedBy":"datasteward",
              "lastUpdateDate":"2020-01-13T13:07:35.865-05:00",
              "consolidationInd":4,
              "lastRowidSystem":"PRODUCT",
              "interactionId":"480003",
              "hubStateInd":0,
              "label":"Product Group is parent of Product Group",
              "rowidRelType":"7"
            }
          }
        }
      ]
    }
  ]
}
```

```

        "rowidHierarchy": "3",
        "from": {
            "rowidObject": "165",
        },
        "to": {
            "rowidObject": "166",
        }
    },
    "entityType": "ProductGroup",
    "relationshipType": "ProductGroupIsParentOfProductGroup"
},
{
    "businessEntity": {
        "ProductGroup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel": "icon"
                }
            ]
        },
        "rowidObject": "165",
        "creator": "admin",
        "createDate": "2020-01-13T13:07:35.128-05:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-01-13T13:07:35.128-05:00",
        "consolidationInd": 4,
        "lastRowidSystem": "PRODUCT",
        "hubStateInd": 1,
        "label": "Vehicles",
        "productType": "Product Group",
        "productName": "Vehicles"
    }
},
    "entityLabel": "Vehicles",
    "entityType": "ProductGroup",
    "depth": 2,
    "object": {
        "businessEntity": {
            "ProductGroup": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                        "rel": "icon"
                    }
                ]
            },
            "rowidObject": "166",
            "creator": "admin",
            "createDate": "2020-01-13T13:07:35.250-05:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-01-13T13:07:35.250-05:00",
            "consolidationInd": 4,
            "lastRowidSystem": "PRODUCT",
            "hubStateInd": 1,
            "label": "Scooter",
            "productType": "Product Group",
            "productName": "Scooter"
        }
    },
    "entityLabel": "Scooter",
    "relationshipLabel": "Product Group is parent of Product Group",
    "relationship": {
        "rowidObject": "124",
        "creator": "datasteward",
        "createDate": "2020-01-13T13:07:35.864-05:00",
        "updatedBy": "datasteward",
        "lastUpdateDate": "2020-01-13T13:07:35.865-05:00",
        "consolidationInd": 4,
        "lastRowidSystem": "PRODUCT",
    },

```

```

        "interactionId": "480003",
        "hubStateInd": 0,
        "label": "Product Group is parent of Product Group",
        "rowidRelType": "7",
        "rowidHierarchy": "3",
        "from": {
            "rowidObject": "165"
        },
        "to": {
            "rowidObject": "166"
        }
    },
    "entityType": "ProductGroup",
    "relationshipType": "ProductGroupIsParentOfProductGroup",
    "object": {
        "businessEntity": {
            "ProductGroup": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmz/request/hm_icons/
hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "167",
                "label": "E-Scooter",
                "productType": "Product Group",
                "productName": "E-Scooter"
            }
        },
        "entityLabel": "E-Scooter",
        "relationshipLabel": "Product Group is parent of Product Group",
        "relationship": {
            "rowidObject": "125",
            "creator": "datasteward",
            "createDate": "2020-01-13T13:07:36.033-05:00",
            "updatedBy": "datasteward",
            "lastUpdateDate": "2020-01-13T13:07:36.034-05:00",
            "consolidationInd": 4,
            "lastRowidSystem": "PRODUCT",
            "interactionId": "480003",
            "hubStateInd": 0,
            "label": "Product Group is parent of Product Group",
            "rowidRelType": "7",
            "rowidHierarchy": "3",
            "from": {
                "rowidObject": "166"
            },
            "to": {
                "rowidObject": "167"
            }
        },
        "entityType": "ProductGroup",
        "relationshipType": "ProductGroupIsParentOfProductGroup"
    }
},
{
    "businessEntity": {
        "ProductGroup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmz/request/hm_icons/hierarchymanager/
Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel": "icon"
                }
            ],
            "rowidObject": "165",
            "creator": "admin",
            "createDate": "2020-01-13T13:07:35.128-05:00",
            "updatedBy": "admin",

```

```

        "lastUpdateDate": "2020-01-13T13:07:35.128-05:00",
        "consolidationInd": 4,
        "lastRowidSystem": "PRODUCT",
        "hubStateInd": 1,
        "label": "Vehicles",
        "productType": "Product Group",
        "productName": "Vehicles"
    },
    {
        "entityLabel": "Vehicles",
        "entityType": "ProductGroup",
        "depth": 3,
        "object": {
            "businessEntity": {
                "ProductGroup": {
                    "link": [
                        {
                            "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                            "rel": "icon"
                        }
                    ]
                },
                "rowidObject": "166",
                "creator": "admin",
                "createDate": "2020-01-13T13:07:35.250-05:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-01-13T13:07:35.250-05:00",
                "consolidationInd": 4,
                "lastRowidSystem": "PRODUCT",
                "hubStateInd": 1,
                "label": "Scooter",
                "productType": "Product Group",
                "productName": "Scooter"
            }
        },
        "entityLabel": "Scooter",
        "relationshipLabel": "Product Group is parent of Product Group",
        "relationship": {
            "rowidObject": "124",
            "creator": "datasteward",
            "createDate": "2020-01-13T13:07:35.864-05:00",
            "updatedBy": "datasteward",
            "lastUpdateDate": "2020-01-13T13:07:35.865-05:00",
            "consolidationInd": 4,
            "lastRowidSystem": "PRODUCT",
            "interactionId": "480003",
            "hubStateInd": 0,
            "label": "Product Group is parent of Product Group",
            "rowidRelType": "7",
            "rowidHierarchy": "3",
            "from": {
                "rowidObject": "165",
                "label": "Vehicles",
                "productType": "Product Group",
                "productName": "Vehicles"
            },
            "to": {
                "rowidObject": "166",
                "label": "Scooter",
                "productType": "Product Group",
                "productName": "Scooter"
            }
        },
        "entityType": "ProductGroup",
        "relationshipType": "ProductGroupIsParentOfProductGroup",
        "object": {
            "businessEntity": {
                "ProductGroup": {
                    "link": [
                        {
                            "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                            "rel": "icon"
                        }
                    ]
                },
                "rowidObject": "167",
                "creator": "admin",
                "createDate": "2020-01-13T13:07:35.864-05:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-01-13T13:07:35.865-05:00",
                "consolidationInd": 4,
                "lastRowidSystem": "PRODUCT",
                "hubStateInd": 1,
                "label": "Scooter",
                "productType": "Product Group",
                "productName": "Scooter"
            }
        }
    }
]

```

```

        "createDate": "2020-01-13T13:07:35.364-05:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-01-13T13:07:35.364-05:00",
        "consolidationInd": 4,
        "lastRowidSystem": "PRODUCT",
        "hubStateInd": 1,
        "label": "E-Scooter",
        "productType": "Product Group",
        "productName": "E-Scooter"
    }
},
"entityLabel": "E-Scooter",
"relationshipLabel": "Product Group is parent of Product Group",
"relationship": {
    "rowidObject": "125",
    "creator": "datasteward",
    "createDate": "2020-01-13T13:07:36.033-05:00",
    "updatedBy": "datasteward",
    "lastUpdateDate": "2020-01-13T13:07:36.034-05:00",
    "consolidationInd": 4,
    "lastRowidSystem": "PRODUCT",
    "interactionId": "480003",
    "hubStateInd": 0,
    "label": "Product Group is parent of Product Group",
    "rowidRelType": "7",
    "rowidHierarchy": "3",
    "from": {
        "rowidObject": "166"
    },
    "to": {
        "rowidObject": "167"
    }
},
"entityType": "ProductGroup",
"relationshipType": "ProductGroupIsParentOfProductGroup",
"object": {
    "businessEntity": {
        "Products": {
            "link": [
                {
                    "href": "http://localhost:8080/cmz/request/hm_icons/HM_Icons/BulletSquareBlue/bullet_square_blue.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel": "icon"
                }
            ],
            "rowidObject": "168",
            "label": "Xiaomi Mi Electric Scooter",
            "productType": "Product",
            "productName": "Xiaomi Mi Electric Scooter"
        }
    }
},
"entityLabel": "Xiaomi Mi Electric Scooter",
"relationshipLabel": "Product Group is parent of Product",
"relationship": {
    "rowidObject": "126",
    "creator": "datasteward",
    "createDate": "2020-01-13T13:07:36.096-05:00",
    "updatedBy": "datasteward",
    "lastUpdateDate": "2020-01-13T13:07:36.096-05:00",
    "consolidationInd": 4,
    "lastRowidSystem": "PRODUCT",
    "interactionId": "480003",
    "hubStateInd": 0,
    "label": "Product Group is parent of Product",
    "rowidRelType": "9",
    "rowidHierarchy": "3",
    "from": {
        "rowidObject": "167"
    },
    "to": {
        "rowidObject": "168"
    }
}

```



```

{
  "operation": "createRelationship",
  "relationship": "ProductGroupIsParentOfProductGroup",
  "payload": {
    "from": {
      "rowidObject": "165"
    },
    "to": {
      "rowidObject": "166"
    }
  }
},
{
  "operation": "createRelationship",
  "relationship": "ProductGroupIsParentOfProductGroup",
  "payload": {
    "from": {
      "rowidObject": "166"
    },
    "to": {
      "rowidObject": "167"
    }
  }
},
{
  "operation": "createRelationship",
  "relationship": "ProductGroupIsParentOfProduct",
  "payload": {
    "from": {
      "rowidObject": "167"
    },
    "to": {
      "rowidObject": "168"
    }
  },
  "effectivePeriod": {
    "startDate": "2019-01-01T11:11:53.974-04:00",
    "endDate": "2039-12-31T11:11:53.974-04:00"
  }
}
]
}

```

Exemple de réponse d'API

L'exemple de réponse suivant présente l'ID d'interaction associé aux modifications, que vous pouvez utiliser pour promouvoir et rejeter les modifications :

```

{
  "processId": "14068",
  "interactionId": "480003",
  "rejectInteractionId": "480004",
  "items": [
    {
      "payload": {
        "key": {
          "rowid": "124",
          "sourceKey": "SVR1.1G9AG"
        },
        "rowidObject": "124",
        "from": {
          "key": {
            "rowid": "165"
          },
          "rowidObject": "165"
        },
        "to": {
          "key": {
            "rowid": "166"
          }
        }
      }
    }
  ]
}

```

```

        },
        "rowidObject": "166"
    }
}
},
{
    "payload": {
        "key": {
            "rowid": "125",
            "sourceKey": "SVR1.1G9AH"
        },
        "rowidObject": "125",
        "from": {
            "key": {
                "rowid": "166"
            },
            "rowidObject": "166"
        },
        "to": {
            "key": {
                "rowid": "167"
            },
            "rowidObject": "167"
        }
    }
},
{
    "payload": {
        "key": {
            "rowid": "126",
            "sourceKey": "SVR1.1G9AI"
        },
        "rowidObject": "126",
        "from": {
            "key": {
                "rowid": "167"
            },
            "rowidObject": "167"
        },
        "to": {
            "key": {
                "rowid": "168"
            },
            "rowidObject": "168"
        }
    }
}
]
}

```

Promouvoir en bloc

L'API REST Promouvoir en bloc promeut les modifications du réseau ou de la hiérarchie.

L'API utilise la méthode POST.

URL de demande

L'URL Promouvoir en bloc est au format suivant :

http://<host>:<port>/<context>/<database ID>/promote

Effectuez la demande HTTP GET suivante à l'URL Promouvoir en bloc :

POST http://<host>:<port>/<context>/<database ID>/promote

Exemple de demande d'API

L'exemple de demande suivant promeut les modifications :

```
POST http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/promote
{
  "interactionId": "67080007",
  "rejectInteractionId": "67080008",
  "objects": [
    {
      "type": "Hierarchy",
      "name": "HPerson"
    }
  ]
}
```

Exemple de réponse d'API

L'exemple de réponse suivant présente la réponse après la promotion des modifications :

```
{
  "interactionId": "68520005",
  "rejectInteractionId": "68520006",
  "type": "hierarchy",
  "name": "HPerson",
  "objects": [
    {
      "before": [
        {
          "businessEntity": {
            "Person": {
              "link": [
                {
                  "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                  "rel": "icon"
                }
              ],
              "rowidObject": "2568254",
              "creator": "admin",
              "createDate": "2020-04-01T14:38:06.577-04:00",
              "updatedBy": "admin",
              "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
              "consolidationInd": 4,
              "lastRowidSystem": "SFA",
              "hubStateInd": 1,
              "effectivePeriod": {},
              "label": "LN, FN",
              "partyType": "Person",
              "lastName": "LN",
              "displayName": "FN LN",
              "generationSuffixCd": "X",
              "firstName": "FN"
            }
          },
          "entityLabel": "LN, FN",
          "entityType": "Person",
          "depth": 0
        }
      ],
      "after": [
        {
          "businessEntity": {
            "Person": {
              "link": [
                {
                  "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                  "rel": "icon"
                }
              ]
            }
          }
        }
      ]
    }
  ]
}
```

```

    ],
    "rowidObject": "2568253",
    "creator": "admin",
    "createDate": "2020-04-01T14:38:03.956-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SFA",
    "hubStateInd": 1,
    "label": "LN, FN",
    "partyType": "Person",
    "lastName": "LN",
    "displayName": "FN LN",
    "generationSuffixCd": "X",
    "firstName": "FN"
  }
},
"entityLabel": "Person",
"entityType": "Person",
"depth": 1,
"object": {
  "businessEntity": {
    "Automobile": {
      "rowidObject": "1450063",
      "creator": "admin",
      "createDate": "2020-04-01T14:38:09.630-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SFA",
      "hubStateInd": 1,
      "effectivePeriod": {},
      "label": "Subaru",
      "model": "MODEL",
      "drivetrainCode": "4WD",
      "doorsCode": "2DR",
      "make": "Subaru"
    }
  },
  "entityLabel": "Subaru",
  "relationshipLabel": "PersonToAutomobile Relationship",
  "relationship": {
    "rowidObject": "1450083",
    "creator": "ds1",
    "createDate": "2020-04-01T14:40:14.921-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2020-04-01T14:40:23.812-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "interactionId": "68520005",
    "hubStateInd": 1,
    "label": "PersonToAutomobile Relationship",
    "from": {
      "rowidObject": "2568253"
    },
    "to": {
      "rowidObject": "1450063"
    }
  },
  "entityType": "Automobile",
  "relationshipType": "PersonToAutomobile2"
}
},
{
  "businessEntity": {
    "Person": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
          "rel": "icon"
        }
      ]
    }
  }
}

```

```

    },
    "rowidObject": "2568253",
    "creator": "admin",
    "createDate": "2020-04-01T14:38:03.956-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SFA",
    "hubStateInd": 1,
    "label": "LN, FN",
    "partyType": "Person",
    "lastName": "LN",
    "displayName": "FN LN",
    "generationSuffixCd": "X",
    "firstName": "FN"
  },
  "entityLabel": "Person",
  "entityType": "Person",
  "depth": 1,
  "object": {
    "businessEntity": {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
            "rel": "icon"
          }
        ]
      }
    },
    "rowidObject": "2568254",
    "creator": "admin",
    "createDate": "2020-04-01T14:38:06.577-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SFA",
    "hubStateInd": 1,
    "effectivePeriod": {},
    "label": "LN, FN",
    "partyType": "Person",
    "lastName": "LN",
    "displayName": "FN LN",
    "generationSuffixCd": "X",
    "firstName": "FN"
  }
},
"entityLabel": "LN, FN",
"relationshipLabel": "Person associate Person",
"relationship": {
  "rowidObject": "1634729",
  "creator": "dsl",
  "createDate": "2020-04-01T14:40:14.870-04:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2020-04-01T14:40:23.872-04:00",
  "consolidationInd": 4,
  "lastRowidSystem": "SYS0",
  "interactionId": "68520005",
  "hubStateInd": 1,
  "label": "Person associate Person",
  "rowidRelType": "103",
  "rowidHierarchy": "1",
  "note": "note DS",
  "from": {
    "rowidObject": "2568253"
  },
  "to": {
    "rowidObject": "2568254"
  }
}
},

```

```

        "entityType": "Person",
        "relationshipType": "PersonAssociatePerson2"
    }
}
],
"approved": [
    {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568253",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:03.956-04:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SFA",
                "hubStateInd": 1,
                "label": "LN, FN",
                "partyType": "Person",
                "lastName": "LN",
                "displayName": "FN LN",
                "generationSuffixCd": "X",
                "firstName": "FN"
            }
        },
        "entityLabel": "Person",
        "entityType": "Person",
        "depth": 1,
        "object": {
            "businessEntity": {
                "Automobile": {
                    "rowidObject": "1450063",
                    "creator": "admin",
                    "createDate": "2020-04-01T14:38:09.630-04:00",
                    "updatedBy": "admin",
                    "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
                    "consolidationInd": 4,
                    "lastRowidSystem": "SFA",
                    "hubStateInd": 1,
                    "effectivePeriod": {},
                    "label": "Subaru",
                    "model": "MODEL",
                    "drivetrainCode": "4WD",
                    "doorsCode": "2DR",
                    "make": "Subaru"
                }
            },
            "entityLabel": "Subaru",
            "relationshipLabel": "PersonToAutomobile Relationship",
            "relationship": {
                "rowidObject": "1450083",
                "creator": "ds1",
                "createDate": "2020-04-01T14:40:14.921-04:00",
                "updatedBy": "ds1",
                "lastUpdateDate": "2020-04-01T14:40:14.922-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SYS0",
                "interactionId": "68520005",
                "hubStateInd": 0,
                "label": "PersonToAutomobile Relationship",
                "from": {
                    "rowidObject": "2568253"
                },
                "to": {

```

```

        "rowidObject": "1450063",
    },
    "entityType": "Automobile",
    "relationshipType": "PersonToAutomobile2"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                    "rel": "icon"
                }
            ],
            "rowidObject": "2568253",
            "creator": "admin",
            "createDate": "2020-04-01T14:38:03.956-04:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
            "consolidationInd": 4,
            "lastRowidSystem": "SFA",
            "hubStateInd": 1,
            "label": "LN, FN",
            "partyType": "Person",
            "lastName": "LN",
            "displayName": "FN LN",
            "generationSuffixCd": "X",
            "firstName": "FN"
        }
    },
    "entityLabel": "Person",
    "entityType": "Person",
    "depth": 1,
    "object": {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568254",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:06.577-04:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SFA",
                "hubStateInd": 1,
                "effectivePeriod": {},
                "label": "LN, FN",
                "partyType": "Person",
                "lastName": "LN",
                "displayName": "FN LN",
                "generationSuffixCd": "X",
                "firstName": "FN"
            }
        },
        "entityLabel": "LN, FN",
        "relationshipLabel": "Person associate Person",
        "relationship": {
            "rowidObject": "1634729",
            "creator": "ds1",
            "createDate": "2020-04-01T14:40:14.870-04:00",
            "updatedBy": "ds1",
            "lastUpdateDate": "2020-04-01T14:40:14.871-04:00",

```

```

        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520005",
        "hubStateInd": 0,
        "label": "Person associate Person",
        "rowidRelType": "103",
        "rowidHierarchy": "1",
        "note": "note DS",
        "from": {
            "rowidObject": "2568253"
        },
        "to": {
            "rowidObject": "2568254"
        }
    },
    "entityType": "Person",
    "relationshipType": "PersonAssociatePerson2"
}
]
}

```

Rejeter en bloc

L'API REST Rejeter en bloc rejette les modifications du réseau ou de la hiérarchie.

L'API utilise la méthode POST.

URL de demande

L'URL Rejeter en bloc est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/reject
```

Effectuez la demande HTTP GET suivante à l'URL Rejeter en bloc :

```
POST http://<host>:<port>/<context>/<database ID>/reject
```

Exemple de demande d'API

L'exemple de demande suivant rejette les modifications :

```

POST http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/reject
{
  "interactionId":"67080007",
  "rejectInteractionId":"67080008",
  "objects":[
    {
      "type":"Hierarchy",
      "name":"HPerson"
    }
  ]
}

```

Exemple de réponse d'API

L'exemple suivant présente la réponse après le rejet des modifications :

```

{
  "interactionId": "68520003",
  "rejectInteractionId": "68520004",
  "type": "hierarchy",
  "name": "HPerson",
  "objects": [

```

```

{
  "before": [
    {
      "businessEntity": {
        "Person": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
              "rel": "icon"
            }
          ],
          "rowidObject": "2568254",
          "creator": "admin",
          "createDate": "2020-04-01T14:38:06.577-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SFA",
          "hubStateInd": 1,
          "effectivePeriod": {},
          "label": "LN, FN",
          "partyType": "Person",
          "lastName": "LN",
          "displayName": "FN LN",
          "generationSuffixCd": "X",
          "firstName": "FN"
        }
      },
      "entityLabel": "LN, FN",
      "entityType": "Person",
      "depth": 0
    }
  ],
  "after": [
    {
      "businessEntity": {
        "Person": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
              "rel": "icon"
            }
          ],
          "rowidObject": "2568254",
          "creator": "admin",
          "createDate": "2020-04-01T14:38:06.577-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SFA",
          "hubStateInd": 1,
          "effectivePeriod": {},
          "label": "LN, FN",
          "partyType": "Person",
          "lastName": "LN",
          "displayName": "FN LN",
          "generationSuffixCd": "X",
          "firstName": "FN"
        }
      },
      "entityLabel": "LN, FN",
      "entityType": "Person",
      "depth": 0
    }
  ],
  "approved": [
    {
      "businessEntity": {
        "Person": {

```

```

        "link": [
            {
                "href": "/request/hm_icons/hierarchymanager/Person/
Person_Small.png?ors={ors}",
                "rel": "icon"
            }
        ],
        "rowidObject": "2568253",
        "creator": "admin",
        "createDate": "2020-04-01T14:38:03.956-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "label": "LN, FN",
        "partyType": "Person",
        "lastName": "LN",
        "displayName": "FN LN",
        "generationSuffixCd": "X",
        "firstName": "FN"
    }
},
"entityLabel": "Person",
"entityType": "Person",
"depth": 1,
"object": {
    "businessEntity": {
        "Automobile": {
            "rowidObject": "1450063",
            "creator": "admin",
            "createDate": "2020-04-01T14:38:09.630-04:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
            "consolidationInd": 4,
            "lastRowidSystem": "SFA",
            "hubStateInd": 1,
            "effectivePeriod": {},
            "label": "Subaru",
            "model": "MODEL",
            "drivetrainCode": "4WD",
            "doorsCode": "2DR",
            "make": "Subaru"
        }
    },
    "entityLabel": "Subaru",
    "relationshipLabel": "PersonToAutomobile Relationship",
    "relationship": {
        "rowidObject": "1450082",
        "creator": "ds1",
        "createDate": "2020-04-01T14:38:13.916-04:00",
        "updatedBy": "ds1",
        "lastUpdateDate": "2020-04-01T14:38:13.916-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520003",
        "hubStateInd": 0,
        "label": "PersonToAutomobile Relationship",
        "from": {
            "rowidObject": "2568253"
        },
        "to": {
            "rowidObject": "1450063"
        }
    },
    "entityType": "Automobile",
    "relationshipType": "PersonToAutomobile2"
}
},
{
    "businessEntity": {

```



```

        "Person": {
            "link": [
                {
                    "href": "/request/hm_icons/hierarchymanager/Person/
Person_Small.png?ors={ors}",
                    "rel": "icon"
                }
            ],
            "rowidObject": "2568253",
            "creator": "admin",
            "createDate": "2020-04-01T14:38:03.956-04:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
            "consolidationInd": 4,
            "lastRowidSystem": "SFA",
            "hubStateInd": 1,
            "label": "LN, FN",
            "partyType": "Person",
            "lastName": "LN",
            "displayName": "FN LN",
            "generationSuffixCd": "X",
            "firstName": "FN"
        }
    },
    "entityLabel": "Person",
    "entityType": "Person",
    "depth": 1,
    "object": {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "/request/hm_icons/hierarchymanager/
Person/Person_Small.png?ors={ors}",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568254",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:06.577-04:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SFA",
                "hubStateInd": 1,
                "effectivePeriod": {},
                "label": "LN, FN",
                "partyType": "Person",
                "lastName": "LN",
                "displayName": "FN LN",
                "generationSuffixCd": "X",
                "firstName": "FN"
            }
        }
    },
    "entityLabel": "LN, FN",
    "relationshipLabel": "Person associate Person",
    "relationship": {
        "rowidObject": "1634728",
        "creator": "ds1",
        "createDate": "2020-04-01T14:38:13.871-04:00",
        "updatedBy": "ds1",
        "lastUpdateDate": "2020-04-01T14:38:13.871-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520003",
        "hubStateInd": 0,
        "label": "Person associate Person",
        "rowidRelType": "103",
        "rowidHierarchy": "1",
        "note": "note DS",
        "from": {

```

```

        "rowidObject": "2568253"
      },
      "to": {
        "rowidObject": "2568254"
      }
    },
    "entityType": "Person",
    "relationshipType": "PersonAssociatePerson2"
  }
}
]
}

```

Démarrer la correspondance

L'API REST Démarrer la correspondance démarre le travail de correspondance d'une entité d'entreprise. Vous devez importer le fichier comportant les entités d'entreprise avant d'appeler l'API Démarrer la correspondance.

L'API utilise la méthode POST.

URL de demande

L'URL Démarrer la correspondance est au format suivant :

```
http://<host>:<port>/adhocmatch/<orsId>/job
```

Effectuez la demande HTTP POST suivante à l'URL Démarrer la correspondance :

```
POST http://<host>:<port>/adhocmatch/<orsId>/job
```

Paramètres

Spécifiez les paramètres permettant de démarrer un travail de correspondance d'entité d'entreprise.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de récupérer les entités d'entreprise correspondantes.
fileId	Corps	Facultatif. ID du fichier à importer.
mappingId	Corps	Facultatif. ID de mappage du fichier à importer.
matchRuleSet	Corps	Facultatif. Spécifiez l'ensemble de règles de correspondance à appliquer au travail de correspondance de l'entité d'entreprise.

Exemple de demande d'API

L'exemple de demande suivant renvoie les entités d'entreprise importées après avoir effectué une correspondance :

```

POST /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/execute/match HTTP/1.1
Content-Type: application/json
Host: localhost:8080

```

Content-Length: 64

```
{
  "tableName" : "C_PRODUCT",
  "matchRuleSetName" : "IDL"
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK après la correspondance des entités d'entreprise et l'importation d'un fichier. Le corps de la réponse est vide.

Lire les enregistrements correspondants

L'API REST Lire les enregistrements correspondants renvoie les enregistrements qui correspondent à un enregistrement racine spécifié. Vous pouvez vérifier la liste des enregistrements pour déterminer ceux qui peuvent être fusionnés avec l'enregistrement racine initial. Vous pouvez utiliser l'API Fusionner les enregistrements pour fusionner les enregistrements.

L'API utilise la méthode GET.

URL de demande

L'URL Lire les enregistrements correspondants est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Effectuez la demande HTTP GET suivante à l'URL Lire les enregistrements correspondants :

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Corps de réponse

Le corps de réponse contient le nombre d'enregistrements correspondant à l'enregistrement spécifié, les détails des enregistrements correspondants et un jeton de recherche. Utilisez le jeton de recherche pour récupérer les pages suivantes du résultat de correspondance.

Exemple de demande d'API

L'exemple de demande suivant recherche des enregistrements dans l'entité commerciale correspondant à un enregistrement :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1038245?action=matched
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les détails de l'enregistrement correspondant à l'enregistrement spécifié :

```
{
  "firstRecord": 1,
  "recordCount": 1,
  "pageSize": 10,
  "searchToken": "SVR1.AU5HE",
  "matchedEntity": [
    {
      "businessEntity": {
        "Person": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/"
            }
          ]
        }
      }
    }
  ]
}
```

```

    Person/1038246",
    "rel": "self"
  }
],
"rowidObject": "1038246",
"creator": "admin",
"createDate": "2008-08-12T02:15:02-07:00",
"updatedBy": "Admin",
"lastUpdateDate": "2008-08-12T02:59:17-07:00",
"consolidationInd": "1",
"lastRowidSystem": "SFA",
"dirtyIndicator": "0",
"hubStateInd": "1",
"partyType": "Person",
"lastName": "BATES",
"firstName": "DAISY",
"displayName": "DAISY BATES"
}
},
"matchRule": "PUT"
}
]
}

```

Mettre à jour les enregistrements correspondants

L'API REST Mettre à jour les enregistrements correspondants crée ou met à jour un enregistrement dans la table de correspondance. La table de correspondance contient les paires d'enregistrements correspondants dans une entité commerciale après l'exécution d'un processus de correspondance sur l'entité commerciale. Utilisez l'API pour ajouter les enregistrements qualifiés pour une fusion avec l'enregistrement spécifié.

L'API utilise la méthode PUT.

URL de demande

L'URL Mettre à jour les enregistrements correspondants est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Effectuez la demande HTTP PUT suivante à l'URL Mettre à jour les enregistrements correspondants :

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
Content-Type: application/json+xml
```

Corps de demande

Envoyez la liste des enregistrements correspondants à l'enregistrement spécifié dans le corps de demande. Utilisez l'ID de ligne ou le système source et la clé source pour spécifier les enregistrements.

Exemple de demande d'API

L'exemple suivant ajoute un enregistrement dans la table de correspondance :

```

PUT http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1038245?action=matched
{
  keys: [
    {
      rowid: "1038246"
    }
  ]
}

```

```
} ]  
}
```

Exemple de réponse d'API

L'API renvoie une réponse 200 OK lorsqu'un enregistrement est créé avec succès dans la table de correspondance. Le corps de réponse est vide.

Supprimer les enregistrements correspondants

L'API REST Supprimer les enregistrements correspondants supprime les enregistrements correspondants associés à un enregistrement racine de la table de correspondance.

L'API utilise la méthode DELETE.

URL de demande

L'URL Supprimer les enregistrements correspondants est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Effectuez la demande HTTP DELETE suivante à l'URL Supprimer les enregistrements correspondants :

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?  
action=matched
```

Ajoutez l'en-tête Content-Type pour spécifier le type de média des données à envoyer avec la demande :

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?  
action=matched  
Content-Type: application/<json/xml>
```

Corps de demande

Envoyez la liste des enregistrements à supprimer de la table de correspondance dans le corps de demande.

Exemple de demande d'API

L'exemple de demande suivant supprime un enregistrement qui correspond à l'enregistrement racine spécifié de la table de correspondance :

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1038245?action=matched  
  
{  
  keys: [  
    {  
      rowid: "1038246"  
    }  
  ]  
}
```

Exemple de réponse d'API

L'API renvoie une réponse 200 OK lorsqu'un enregistrement est supprimé de la table de correspondance avec succès. Le corps de réponse est vide.

Obtenir les données de correspondance dans CSV

L'API REST Obtenir les données de correspondance dans CSV récupère les entités d'entreprise correspondantes et non correspondantes au format CSV après que le système fait correspondre les enregistrements des champs cible et du système source.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les données de correspondance est au format suivant :

```
http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?withMatches,withoutMatches}
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les résultats des données de correspondance :

```
GET http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?withMatches,withoutMatches}
```

Paramètres

Spécifiez les paramètres permettant de renvoyer les entités d'entreprise correspondantes et non correspondantes au format CSV.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
jobGroupId	Chemin	ID de groupe du travail de lots.
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de récupérer les enregistrements d'entité d'entreprise.
withMatches	Requête	Facultatif. Utilisez la valeur suivante : True : pour renvoyer des enregistrements correspondant aux données existantes.
withoutMatches	Requête	Facultatif. Utilisez la valeur suivante : False : pour renvoyer des enregistrements ne correspondant pas aux données existantes.

Exemple de demande d'API

L'exemple de demande suivant renvoie des entités d'entreprise correspondantes et non correspondantes au format CSV :

```
GET /cmx/adhocmatch/{ors}/result/{jobId}?withMatches=true&withoutMatches=true
Accept: application/csv
```

```
{
  item: [
    {
      businessEntity: {
        Person: {
          firstName: "John"
        }
      },
      matchedEntity: {
```

```

    {
      businessEntity: {Person: {rowidObject: 123}},
    },
    {
      matchRule: "MRS|1",
      matchScore: 0.9
    }
  ],
  {
    businessEntity: {
      Person: {
        firstName: "Alex"
      }
    }
  }
]
}

```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK et le fichier CSV avec les résultats de la correspondance.

Obtenir les données de correspondance dans JSON

L'API REST Obtenir les données de correspondance dans JSON récupère les entités d'entreprise correspondantes et non correspondantes au format CSV après que le système fait correspondre les enregistrements des champs cible et du système source.

L'API utilise la méthode GET.

URL de demande

L'URL Résultats de la correspondance est au format suivant :

```

http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?
withMatches,withoutMatches}

```

Effectuez la demande HTTP GET suivante à l'URL Résultats de la correspondance :

```

GET http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?
withMatches,withoutMatches}

```

Paramètres

Spécifiez les paramètres permettant de renvoyer les entités d'entreprise correspondantes et non correspondantes au format CSV.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
jobGroupld	Chemin	ID de groupe du travail de lots.
orsld	Chemin	ID de base de données du stockage de référence opérationnelle permettant de récupérer les enregistrements d'entité d'entreprise.

Paramètre	Type	Description
withMatches	Requête	Facultatif. Utilisez la valeur suivante : True : pour renvoyer des enregistrements correspondant aux données existantes.
withoutMatches	Requête	Facultatif. Utilisez la valeur suivante : False : pour renvoyer des enregistrements ne correspondant pas aux données existantes.

Exemple de demande d'API

L'exemple de demande suivant renvoie la liste des entités d'entreprise correspondantes et non correspondantes après que le système effectue une correspondance :

```
GET /cmx/adhocmatch/{ors}/result/{jobId}?withMatches=true&withoutMatches=true
Accept: application/json
```

```
{
  item: [
    {
      businessEntity: {
        Person: {
          firstName: "John"
        }
      },
      matchedEntity: {
        {
          businessEntity: {Person: {rowidObject: 123}},
        }
        matchRule: "MRS|1",
        matchScore: 0.9
      }
    },
    {
      businessEntity: {
        Person: {
          firstName: "Alex"
        }
      }
    }
  ]
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK et le fichier JSON avec les résultats de la correspondance.

Importer le fichier correspondant

L'API REST Importer le fichier correspondant fait correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier. Une fois l'opération d'importation terminée, les entités d'entreprise correspondantes sont mises à jour avec les nouvelles données.

Vous pouvez importer des entités d'entreprise dupliquées, les entités d'entreprises uniques ou les entités d'entreprise dupliquées et uniques.

L'API utilise la méthode POST.

URL de demande

L'URL Importer le fichier correspondant est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Effectuez la demande HTTP POST suivante à l'URL Importer le fichier correspondant :

```
POST http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Paramètres

Spécifiez les paramètres permettant de faire correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier.

Le tableau suivant décrit les paramètres :

Paramètre	Type	Description
orsId	Chemin	ID de base de données du stockage de référence opérationnelle permettant d'importer le fichier après avoir fait correspondre les enregistrements existants.
filter	Requête	Facultatif. Utilisez les valeurs de filtre suivantes : withMatches : entrez <code>True</code> si les enregistrements correspondent aux enregistrements actuels. withoutMatches : entrez <code>False</code> si les enregistrements ne correspondent pas aux enregistrements actuels.
jobGroupControlId	Requête	Facultatif. ID de groupe du travail de correspondance.
systemName	Requête	Facultatif. Nom du système source.

Exemple de demande d'API

L'exemple de demande suivant fait correspondre les nouvelles entités d'entreprise avec les actuelles avant d'importer le fichier :

```
POST /cmx/beimport/{ors}/aftermatch
{
  jobGroupControlId: "...",
  systemName: "SFA",
  filter: {
    withMatches: true,
    withoutMatches: false
  }
}
```

Exemple de réponse d'API

L'API renvoie un code de réponse 200 OK après la correspondance réussie de nouvelles entités d'entreprise avec les actuelles et l'importation du fichier. Le corps de la réponse est vide.

Obtenir les métadonnées du système source

L'API REST Obtenir les métadonnées du système source renvoie les métadonnées des systèmes source.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir les métadonnées du système source est au format suivant :

```
http://<host>:<port>/<context>/<orsId>/sourceSystem
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les métadonnées du système source :

```
GET http://<host>:<port>/<context>/<orsId>/sourceSystem
```

Paramètres

Spécifiez le système source à renvoyer dans la réponse.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Type	Description
orsId	Chemin	ID du stockage de référence opérationnelle de la base de données permettant de récupérer le système source.
entityId	Chemin	ID de l'entité d'entreprise.
entityName	Chemin	Nom de l'entité d'entreprise.
action	Requête	Action devant être effectuée par l'API. Ce qui suit est la valeur par défaut : getSourceSystems
aggregate	Requête	Facultatif. Données agrégées dans plusieurs XREF. Utilisez les valeurs suivantes : <ul style="list-style-type: none">- RECORD- NODE- ENTITY
children	Requête	Facultatif. Liste des noms de nœuds enfants. Si ce paramètre est spécifié, la réponse contient les noms des nœuds enfants.
compact	Requête	Facultatif. Spécifiez ce paramètre pour renvoyer un résultat compact.
contentMetadata	Requête	Facultatif. Métadonnées de la liste des événements d'historique. Utilisez les valeurs suivantes : XREF PENDING_XREF DELETED_XREF HISTORY MATCH BVT TRUST
contributorsOnly	Requête	Facultatif. Compte toutes les XREF ou ces XREF qui contribuent à la meilleure version de la vérité (BVT).
depth	Requête	Facultatif. Nombre de niveaux enfants à renvoyer.
describe	Requête	Facultatif. Ajoute la description du système source.

Paramètre	Type	Description
fields	Requête	Facultatif. Liste de champs d'entité d'entreprise séparés par des virgules. Si ce paramètre est spécifié, la réponse ne contient que les champs répertoriés.
filter	Requête	Facultatif. Expression de filtre.
firstRecord	Requête	Facultatif. Spécifie la première ligne dans le résultat.
order	Requête	Facultatif. Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +. Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis par le suivant.
recordStates	Requête	Facultatif. Enregistrez les états renvoyés dans la liste d'événements d'historique. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : - ACTIVE - PENDING - DELETED
recordsToReturn	Requête	Facultatif. Spécifie le nombre de lignes à renvoyer.
recordsTotal	Requête	Facultatif. Si ce paramètre est défini sur <code>True</code> , il renvoie le nombre d'enregistrements dans le résultat. La valeur par défaut est <code>False</code> .

Exemple de demande d'API

L'exemple de demande suivant renvoie les métadonnées du système source disponibles :

```
GET /cmx/metadata/localhost-orcl-MDM_SAMPLE/sourceSystem HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant renvoie les métadonnées du système source disponibles :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 328
```

```
{
  "totalCount" : 7,
  "firstRecord" : 1,
  "pageSize" : 7,
  "sourceSystems" : [ {
    "name" : "Product"
  }, {
    "name" : "Informatica Data Director"
  }, {
    "name" : "Legacy"
  }, {
    "name" : "SFA"
  }, {
    "name" : "Lookups"
  } ]
}
```

```

    }, {
      "name" : "OrgDataInc"
    }, {
      "name" : "Admin"
    } ]
  }
}

```

Obtenir les événements d'historique de l'enregistrement

L'API REST Obtenir les événements d'historique de l'enregistrement renvoie une liste d'événements d'historique ou de groupes d'événements d'historique associés à un enregistrement. Envoyez l'ID de l'enregistrement dans le corps de demande.

L'API utilise la méthode GET pour renvoyer les données suivantes pour chaque groupe d'événements d'historique :

- Date de début et de fin du groupe
- Nombre d'événements dans le groupe

L'API renvoie les données suivantes pour chaque événement d'historique :

- Identificateur d'événement d'historique
- Date de la modification
- Utilisateur ayant effectué la modification
- Liste des tables d'historique affectées par la modification
- Liste des nœuds d'enregistrement affectés par la modification

URL de demande

Le format de l'URL Obtenir les événements d'historique de l'enregistrement est le suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents

```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les événements d'historique de l'enregistrement :

```

GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents

```

Paramètres de la requête

L'identificateur de l'enregistrement est un paramètre requis. L'API utilise l'identificateur d'enregistrement pour rechercher tous les événements d'historique associés.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
startDate et endDate	Facultatif. Plage de dates au cours de laquelle vous souhaitez récupérer les données. Si vous spécifiez une plage de dates, la réponse contient uniquement les événements qui se sont produits au cours de cette plage.
granularity	Facultatif. Niveau de détails des événements d'historique du groupe. Si ce paramètre est spécifié, la réponse regroupe les événements d'historique. Dans le cas contraire, elle ne les regroupe pas. Utilisez l'une des valeurs suivantes : <ul style="list-style-type: none">- YEAR- QUARTER- MONTH- WEEK- DAY- HOUR- MINUTE- AUTO
recordStates	Facultatif. Enregistrez les états renvoyés dans la liste d'événements d'historique. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none">- ACTIVE- PENDING- DELETED
contentMetadata	Facultatif. Métadonnées de la liste d'événements d'historique. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none">- XREF- PENDING_XREF- DELETED_XREF- HISTORY- MATCH- BVT- TRUST
children	Facultatif. Liste séparée par des virgules des noms de nœuds enfants. Si ce paramètre est spécifié, la réponse contient les noms des nœuds enfants.
ordre	Facultatif. Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +. Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis par le suivant.
fields	Facultatif. Liste de champs d'entité d'entreprise séparés par des virgules. Si ce paramètre est spécifié, la réponse ne contient que les champs répertoriés.
filter	Facultatif. Expression de filtre.
depth	Facultatif. Nombre de niveaux enfants à renvoyer.

Paramètre	Description
recordsToReturn	Facultatif. Spécifie le nombre de lignes à renvoyer.
searchToken	Facultatif. Spécifie le jeton de recherche renvoyé avec la demande précédente.
returnTotal	Facultatif. Si ce paramètre est défini sur <code>True</code> , il renvoie le nombre d'enregistrements dans le résultat. La valeur par défaut est <code>False</code> .
firstRecord	Facultatif. Spécifie la première ligne dans le résultat.
changeType	Facultatif. Spécifie les types de modifications renvoyées dans le résultat. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none"> - BO - XREF - BVT - MERGE - MERGE_AS_SOURCE - MERGE_AS_TARGET - UNMERGE_AS_SOURCE - UNMERGE_AS_TARGET

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Exemple de demande d'API

L'exemple de demande suivant renvoie toutes les fusions d'un enregistrement, groupées par année, depuis le 1er janvier 2000 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123?
action=listHistoryEvents&startDate=2010-01-01&granularity=YEAR&depth=2&changeType=MERGE
```

Exemple de réponse d'API

L'exemple de réponse suivant répertorie les fusions de l'enregistrement spécifié depuis le 1er janvier 2000 :

```
{
  firstRecord: 1,
  recordCount: 2
  item: [
    {
      link: [ // you can use links directly to get event list
        {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2000-01-01&endDate=2001-01-01&depth=2&changeType=MERGE
"}
      ],
      startDate: "2000-01-01",
      endDate: "2001-01-01",
      eventCount: 123
    },
    // no events in 2001, 2002, ... 2009
    {
      link: [
        {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2010-01-01&endDate=2011-01-01&depth=2&changeType=MERGE
"}
      ],
      startDate: "2010-01-01",
```

```

        endDate: "2011-01-01",
        eventCount: 23
    }
    // no events in 2011, ..., 2016
}
]
}

```

Obtenir les détails de l'événement

L'API REST Obtenir les détails de l'événement renvoie les détails d'un événement d'historique spécifique associé à un enregistrement. L'API renvoie des détails tels que le type de modification effectuée et les valeurs précédant et suivant la modification. Envoyez les identificateurs de l'enregistrement et de l'événement d'historique dans le corps de demande.

L'API utilise la méthode GET.

URL de demande

Le format de l'URL Obtenir les détails de l'événement est le suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails

```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les détails de l'événement :

```

GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails

```

Paramètres de la requête

Avant de commencer, utilisez l'API Obtenir les événements d'historique de l'enregistrement pour répertorier les événements d'historique associés à un enregistrement. À partir des résultats renvoyés, utilisez l'ID d'événement d'historique et l'ID d'enregistrement comme paramètres de requête.

Le tableau suivant répertorie les paramètres de la requête :

Paramètre	Description
eventID	Requis. L'API Obtenir les événements d'historique de l'enregistrement renvoie les ID d'événement des événements d'historique.
recordStates	Facultatif. Enregistrez les états renvoyés dans la liste d'événements d'historique. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none"> - ACTIVE - PENDING - DELETED
contentMetadata	Facultatif. Métadonnées de la liste d'événements d'historique. Fournissez une liste de valeurs séparées par des virgules. Vous pouvez utiliser les valeurs suivantes : <ul style="list-style-type: none"> - XREF - PENDING_XREF - DELETED_XREF - HISTORY - MATCH - BVT - TRUST

Paramètre	Description
children	Facultatif. Liste séparée par des virgules des noms de nœuds enfants. Si ce paramètre est spécifié, la réponse contient les noms des nœuds enfants.
order	Facultatif. Liste de noms de champs séparés par des virgules et contenant un préfixe facultatif + ou -. Le préfixe + indique de classer les résultats par ordre croissant et le préfixe - par ordre décroissant. La valeur par défaut est +. Lorsque vous spécifiez plusieurs paramètres, l'ensemble de résultats est trié selon le premier paramètre de la liste, puis par le suivant.
fields	Facultatif. Liste de champs d'entité d'entreprise séparés par des virgules. Si ce paramètre est spécifié, la réponse ne contient que les champs répertoriés.
filter	Facultatif. Expression de filtre.
depth	Facultatif. Nombre de niveaux enfants à renvoyer.
recordsToReturn	Facultatif. Spécifie le nombre de lignes à renvoyer.
searchToken	Facultatif. Spécifie le jeton de recherche renvoyé avec la demande précédente.
returnTotal	Facultatif. Si ce paramètre est défini sur <code>True</code> , il renvoie le nombre d'enregistrements dans le résultat. La valeur par défaut est <code>False</code> .
firstRecord	Facultatif. Spécifie la première ligne dans le résultat.

Exemple de demande d'API

L'exemple de demande suivant renvoie les informations d'un événement d'historique :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123?
action=getHistoryEventDetails&eventId=2016-07-14T02%3A01%3A24.529%2B0000
```

Exemple de réponse d'API

L'exemple de réponse suivant affiche les détails de l'événement spécifié :

```
{
  "eventId": "2016-07-14T02:01:24.529+0000",
  "eventDate": "2016-07-14T02:01:24.529Z",
  "user": "admin",
  "changeType": [
    "BVT",
    "BO",
    "UNMERGE_AS_TARGET"
  ],
  "businessEntity": {
    "Person": {
      "rowidObject": "438243",
      "creator": "datasteward1",
      "createDate": "2016-07-08T20:42:47.402Z",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-07-14T01:42:50.841Z",
      "consolidationInd": 1,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "BE,AC",
      "partyType": "Person",
      "lastName": "BE",
      "displayName": "AC BE",

```



```

    "firstName": "AC"
  }
}

```

Liste Rapports

L'API REST Répertoire les rapports renvoie une liste des rapports enregistrés et leur configuration.

L'API utilise la méthode GET.

URL de demande

L'URL REST Répertoire les rapports est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/list
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/list
```

Paramètres de la requête

Vous pouvez utiliser le paramètre `show` pour spécifier si vous souhaitez répertorier tous les rapports ou uniquement les rapports racine.

Pour renvoyer tous les rapports, dont les rapports racine et les rapports d'exploration, utilisez le paramètre de requête suivant :

```
list?show=all
```

Pour ne renvoyer que les rapports racine, utilisez le paramètre de requête suivant :

```
list?show=root
```

Remarque: Si vous ne spécifiez pas le paramètre `show`, l'API renvoie les rapports racine.

Corps de la réponse

Le corps de la réponse contient la liste des rapports, dont les détails de chaque rapport.

Le tableau suivant décrit les paramètres de configuration des rapports dans le corps de la réponse :

Paramètre	Description
ROWID_RPT_CONFIG	ID de rapport.
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
RPT_NAME	Nom du rapport.

Paramètre	Description
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <code>Number of closed tasks</code> .
RPT_DESC	Description du rapport.
RPT_TYPE	Indique s'il s'agit d'un rapport d'exploration. Si la valeur est <code>Null</code> , il s'agit d'un rapport racine. Si la valeur n'est pas <code>Null</code> , il s'agit d'un rapport d'exploration.

Exemple de demande d'API

L'exemple de demande suivant renvoie tous les rapports enregistrés, y compris les rapports racine et les rapports d'exploration :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list?show=all
```

Exemple de réponse d'API

L'exemple de réponse suivant présente une liste de rapports enregistrés :

```
{
  "ROWID_RPT_CONFIG": "7 ",
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries",
  "RPT_TYPE": "null"
},
{
  "metadata": {
    "ROWID_RPT_CONFIG": "12 ",
    "DIMENSION_NAME_1": "City code",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Customer by City",
    "METRIC_NAME": "Customer by City",
    "RPT_DESC": "Customer by City",
    "RPT_TYPE": "null"
  }
}
```

Obtenir la configuration et les données du rapport

L'API REST Obtenir la configuration et les données du rapport renvoie la configuration et les données du rapport.

L'API utilise la méthode GET.

URL de demande

L'URL REST Obtenir la configuration et les données du rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/data/<reportId>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport pour lequel vous souhaitez la configuration et les données.

Corps de la réponse

La réponse inclut la configuration du rapport et les données.

Le tableau suivant décrit les paramètres dans le corps de la réponse :

Paramètre	Description
metadata	Comprend la configuration du rapport.
ROWID_RPT_CONFIG	ID de rapport.
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <i>Month</i> .
RPT_NAME	Nom du rapport.
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <i>Number of closed tasks</i> .
RPT_DESC	Description du rapport.
RPT_TYPE	Indique s'il s'agit d'un rapport d'exploration. Si la valeur est <i>Null</i> , il s'agit d'un rapport racine. Si la valeur n'est pas <i>Null</i> , il s'agit d'un rapport d'exploration.
fieldsMetadata	Répertorie les champs dans un tableau de données du rapport. Les valeurs incluent <i>DIMENSION_VALUE_1</i> , <i>DIMENSION_VALUE_2</i> , <i>TIMEPERIOD_VALUE</i> , <i>METRIC_VALUE</i> et <i>DRILLDOWN_RPT_ID</i> .
data	Répertorie les valeurs de chaque champ dans un tableau de données du rapport.

Exemple de demande d'API

L'exemple de demande suivant renvoie la configuration et les données d'un rapport :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/MDM.RPT.1
```

Exemple de réponse d'API

L'exemple de réponse suivant présente la configuration et les données d'un rapport :

```
{
  "metadata": {
    "ROWID_RPT_CONFIG": "11 ",
```

```

        "DIMENSION_NAME_1": "Country code",
        "DIMENSION_NAME_2": "null",
        "TIMEPERIOD_NAME": "null",
        "RPT_NAME": "Customer by Region",
        "METRIC_NAME": "Customer by Region",
        "RPT_DESC": "Customer by Region",
        "RPT_TYPE": "null",
        "fieldsMetadata": ["DIMENSION_VALUE_1", "DIMENSION_VALUE_2", "TIMEPERIOD_VALUE",
        "METRIC_VALUE", "DRILLDOWN_RPT_ID"]
    },
    "data": [
        [
            "CA ",
            "null",
            "null",
            "9",
            "123"
        ],
        [
            "IN ",
            "null",
            "null",
            "19",
            "456"
        ],
        [
            "US ",
            "United States",
            "null",
            "9",
            "678"
        ],
        [
            "IN ",
            "India ",
            "null",
            "19",
            null
        ]
    ]
}

```

Obtenir la configuration du rapport et les rapports d'exploration

L'API REST Obtenir la configuration du rapport et les rapports d'exploration renvoie la configuration du rapport et les rapports d'exploration associés.

L'API utilise la méthode GET.

URL de demande

L'URL REST Obtenir la configuration du rapport et les rapports d'exploration est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/meta/<report ID>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/meta/<report ID>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport pour lequel vous souhaitez la configuration et les rapports d'exploration associés.

Corps de la réponse

Le corps de la réponse inclut les paramètres suivants :

Paramètre	Description
ROWID_RPT_CONFIG	ID de rapport.
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
RPT_NAME	Nom du rapport.
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <code>Number of closed tasks</code> .
RPT_DESC	Description du rapport.
RPT_TYPE	Indique s'il s'agit d'un rapport d'exploration. Si la valeur est <code>Null</code> , il s'agit d'un rapport racine. Si la valeur n'est pas <code>Null</code> , il s'agit d'un rapport d'exploration.
DRILLDOWN	Répertorie les rapports d'exploration. L'ordre des rapports répertoriés reflète le niveau du rapport avec le rapport racine.
RPT_NAME	Nom du rapport d'exploration.
DETAILED_RPT_IDS	Répertorie les ID des détails de rapports. Comprend les ID de détails des rapports d'exploration pour le niveau d'exploration.
CONFIG_RPT_IDS	Répertorie les ID des rapports d'exploration pour le niveau d'exploration.

Exemple de demande d'API

L'exemple de demande suivant renvoie la configuration du rapport et les rapports d'exploration associés :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/meta/SVR1.2X9N0
```

Exemple de réponse d'API

L'exemple de réponse suivant présente la configuration du rapport et les rapports d'exploration associés :

```
{
  "ROWID_RPT_CONFIG": "7 ",
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries",
  "RPT_TYPE": "null",
```

```

"DRILLDOWN":
[
  {
    "RPT_NAME": "Customer by region - First level drilldown",
    "DETAILED_RPT_IDS":["1", "2", "3"],
    "CONFIG_RPT_IDS":["1"]
  },
  {
    "RPT_NAME": "Customer by city - Second level drilldown",
    "DETAILED_RPT_IDS":["5", "6", "7", "8", "9", "10"],
    "CONFIG_RPT_IDS":["2"]
  }
]
}

```

Enregistrer le rapport

L'API REST enregistre les rapports personnalisés. L'API renvoie l'ID du rapport que vous pouvez utiliser dans d'autres API.

L'API utilise la méthode POST.

Pour plus d'informations sur l'enregistrement des rapports prêts à l'emploi, consultez la rubrique ["Gestion des rapports prêts à l'emploi" à la page 312](#).

URL de demande

L'URL REST Enregistrer le rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/list
```

Effectuez la demande HTTP POST suivante à l'URL :

```
POST http://<host>:<port>/<context>/<database ID>/list
```

Corps de la demande

Spécifiez la configuration du rapport.

Le tableau suivant décrit les paramètres des détails du rapport dans le corps de la demande :

Paramètre	Description
ROWID_RPT_CONFIG	ID de rapport.
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <i>Month</i> .
RPT_NAME	Nom du rapport.

Paramètre	Description
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <code>Number of closed tasks</code> .
RPT_DESC	Description du rapport.
RPT_TYPE	Indique s'il s'agit d'un rapport d'exploration. Si la valeur est <code>Null</code> , il s'agit d'un rapport racine. Si la valeur n'est pas <code>Null</code> , il s'agit d'un rapport d'exploration.

Exemple de demande d'API

L'exemple de demande suivant enregistre un rapport :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries"
}
```

Exemple de réponse d'API

L'exemple suivant présente la réponse lors de l'enregistrement réussi d'un rapport :

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Subject area",
  "DIMENSION_NAME_2": "Source System",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "randomname750",
  "RPT_DESC": "Source system Metrics",
  "METRIC_NAME": "Number Of Records",
  "RPT_TYPE": null
}
```

L'API renvoie l'ID de rapport dans le paramètre `ROWID_RPT_CONFIG`.

Mettre à jour la configuration du rapport

L'API REST Mettre à jour la configuration du rapport met à jour la configuration d'un rapport.

L'API utilise la méthode POST.

URL de demande

L'URL REST Mettre à jour la configuration du rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/list/<report ID>
```

Effectuez la demande HTTP POST suivante à l'URL :

```
POST http://<host>:<port>/<context>/<database ID>/list/<report ID>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport pour lequel vous souhaitez mettre à jour la configuration.

Corps de demande

Spécifiez la configuration du rapport à mettre à jour.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
RPT_NAME	Nom du rapport.
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <code>Number of closed tasks</code> .
RPT_DESC	Description du rapport.

Exemple de demande d'API

L'exemple de demande suivant met à jour la configuration d'un rapport :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list/SVR1.2X9N0
{
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries"
}
```

Exemple de réponse d'API

L'API ne renvoie aucune réponse lors de la mise à jour de la configuration du rapport.

Ajouter ou mettre à jour les données du rapport

L'API REST Ajouter ou mettre à jour les données du rapport ajoute ou met à jour les entrées de données dans un rapport.

Remarque: Si le rapport contient des données du rapport, l'API remplace les données du rapport.

L'API utilise la méthode POST.

URL de demande

L'URL REST Ajouter ou mettre à jour les données du rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Effectuez la demande HTTP POST suivante à l'URL :

```
POST http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport pour lequel vous souhaitez ajouter ou mettre à jour des entrées de données.

Corps de la demande

Spécifiez le rapport pour lequel vous souhaitez ajouter ou mettre à jour des entrées de données.

Le tableau suivant décrit les paramètres des données du rapport dans le corps de la demande :

Paramètre	Description
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
METRIC_VALUE	Nombre qui représente les dimensions des données. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Une dimension du rapport est l'utilisateur John Smith et une autre est le type de tâche de vérification. La valeur de la métrologie peut être 5. Cela implique que John Smith a fermé cinq tâches du type de tâche de vérification.
DRILLDOWN_RPT_ID	ID d'un rapport d'exploration. Si la valeur est <code>Null</code> , aucun rapport d'exploration n'a été configuré.

Exemple de demande d'API

L'exemple de demande suivant ajoute ou met à jour les entrées de données dans un rapport :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "11:Org,ex.Phone",
    "DIMENSION_VALUE_2": "Cross Source Matches",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "5",
    "DRILLDOWN_RPT_ID": null
  },
  {
    "DIMENSION_VALUE_1": "9:Org Name,City,Zip",
    "DIMENSION_VALUE_2": "Cross Source Matches",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "40",
    "DRILLDOWN_RPT_ID": null
  }
]
```

```

    },
    {
      "DIMENSION_VALUE_1": "12:Org,ex.Email",
      "DIMENSION_VALUE_2": "CRM Matches",
      "TIMEPERIOD_VALUE": "null",
      "METRIC_VALUE": "7",
      "DRILLDOWN_RPT_ID": null
    }
  ]

```

Exemple de réponse d'API

L'API ne renvoie aucune réponse lors de l'ajout ou de la mise à jour des entrées de données dans le rapport.

Supprimer le rapport

L'API REST Supprimer le rapport supprime un rapport.

L'API utilise la méthode DELETE.

URL de demande

L'URL REST Supprimer le rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Effectuez la demande HTTP DELETE suivante à l'URL :

```
DELETE http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport à supprimer.

Exemple de demande d'API

L'exemple de demande suivant supprime un rapport :

```
DELETE http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Exemple de réponse d'API

L'API ne renvoie aucune réponse lors de la suppression du rapport.

Exécuter le travail de mise à jour du rapport

L'API REST Exécuter le travail de mise à jour du rapport démarre le travail de lots associé au rapport prêt à l'emploi.

L'API utilise la méthode GET.

URL de demande

L'URL REST Exécuter le travail de mise à jour du rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>
```

Paramètres de la requête

L'ID de rapport est un paramètre d'URL requis. Utilisez l'ID de rapport pour spécifier le rapport pour lequel vous souhaitez exécuter un travail de mise à jour.

Corps de réponse

La réponse peut contenir l'une des valeurs suivantes dans le paramètre `status` :

Valeur	Description
PROCESSING	Le travail a été déclenché. À la fin du travail, son ID est renvoyé.
COMPLETED_SUCCESSFULLY	Les données du rapport ont été calculées immédiatement sans exécution d'un travail.
DONE	Le travail n'a pas été démarré et un calcul a été effectué dans un mode synchrone.
FAILED	Le travail n'a pas démarré en raison d'une erreur. Les attributs <code>errorMessage</code> et <code>errorCode</code> expliquent l'erreur.

Exemple de demande d'API

L'exemple de demande suivant exécute un travail de mise à jour d'un rapport :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/SVR1.2X9N0
```

Exemple de réponse d'API

L'exemple suivant présente la réponse lors de l'exécution du travail de mise à jour d'un rapport :

```
{
  "status": "PROCESSING"
}
```

Obtenir le statut du travail de mise à jour du rapport

L'API REST Obtenir le statut du travail de mise à jour du rapport renvoie le statut d'un travail de mise à jour du rapport.

L'API utilise la méthode GET.

URL de demande

L'URL REST Obtenir le statut du travail de mise à jour du rapport est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>/status
```

Effectuez la demande HTTP GET suivante à l'URL :

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>/status
```

Paramètres de la requête

L'ID de l'enregistrement est un paramètre d'URL requis. Utilisez le paramètre `report ID` pour spécifier le rapport pour lequel vous voulez obtenir le statut du travail de mise à jour.

Corps de réponse

La réponse contient les paramètres suivants :

Paramètre	Description
status	Statut du travail de mise à jour du rapport. Les valeurs sont COMPLETED_SUCCESSFULLY, PROCESSING, FAILED ou ERROR.
lastUpdateDate	Date de dernière mise à jour pour le rapport. Définie pour les travaux de mise à jour du rapport ayant le statut COMPLETED_SUCCESSFULLY.
errorCode	Facultatif. Code d'erreur avec le travail de mise à jour du rapport. Défini pour les travaux de mise à jour du rapport ayant le statut ERROR.
errorMessage	Facultatif. Message d'erreur avec le travail de mise à jour du rapport. Défini pour les travaux de mise à jour du rapport ayant le statut ERROR.

Exemple de demande d'API

L'exemple de demande suivant renvoie le statut d'un travail de mise à jour du rapport :

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/SVR1.2X9N1/status
```

Exemple de réponse d'API

L'exemple suivant présente la réponse lors de la vérification du statut d'un travail de mise à jour du rapport :

```
{
  "status": "COMPLETED_SUCCESSFULLY",
  "jobId": "SVR1.2X9N1-",
  "lastUpdateDate": "2019-10-31T14:37:11.846-04:00",
  "startRunDate": "2019-10-31T14:37:09.120-04:00"
}
```

Répertorier les groupes de travaux

L'API REST Répertorier les groupes de travaux renvoie la liste des groupes de travaux existants et le statut. Vous pouvez filtrer les travaux selon le type de groupes de travaux. Par exemple, vous ne pouvez afficher que tous les travaux d'importation.

L'API utilise la méthode GET.

URL de demande

L'URL Obtenir la liste des travaux est au format suivant :

```
http://<host>:<port>/<jobcontrol>/<orsId>/<group?jobGroupType>
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir la liste des travaux :

```
GET http://<host>:<port>/<jobcontrol>/<orsId>/<group?jobGroupType>
```

Paramètres

Spécifiez les paramètres permettant de renvoyer la liste des groupes de travaux existants et le statut.

Le tableau suivant décrit les paramètres :

Paramètre	Chemin	Description
jobGroupControlId	Chemin	ID du groupe de travaux.
orclId	Chemin	ID de base de données du stockage de référence opérationnelle permettant de renvoyer la liste des groupes de travaux existants et le statut.
detailed	Requête	Facultatif. Utilisez l'une des valeurs suivantes : True : pour afficher une liste détaillée des groupes de travaux existants et le statut. False : pour afficher une courte liste des groupes de travaux existants et le statut.
firstRecord	Requête	Facultatif. Le premier enregistrement de la liste des groupes de travaux existants.
jobGroupType	Requête	Facultatif. Répertoriez les travaux selon le type de groupes de travaux. Utilisez l'une des valeurs suivantes : Importer Correspondance Import_After_Match Find_Replace
groupBy	Requête	Facultatif. Groupez les travaux selon l'entité d'entreprise.
recordsToReturn	Requête	Facultatif. Nombre d'enregistrements à renvoyer.
returnTotal	Requête	Facultatif. Définissez ce paramètre sur True. Nombre total d'enregistrements à renvoyer.

Exemple de demande d'API

L'exemple de demande suivant renvoie la liste des groupes de travaux existants et le statut :

```
GET /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/group/SVR1.1G7UY?groupBy=owningTable
HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant renvoie la liste des groupes de travaux existants et le statut :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
Content-Disposition: inline;filename=f.txt
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 1551

{
  "jobGroupControlId" : "SVR1.1G7UY",
  "jobGroup" : {
    "type" : "IMPORT",
```

```

    "name" : "BE_Import_RH_LE_PersonView[1]_16f9ffd55b5"
  },
  "startDate" : "2020-01-13T13:06:26.649-05:00",
  "endDate" : "2020-01-13T13:06:40.842-05:00",
  "status" : "COMPLETED WITH ERRORS",
  "statusMessage" : "Completed with errors/warnings",
  "errorMessages" : [ "Completed with errors/warnings", "Completed with errors/
warnings", "Completed with errors/warnings", "Completed with errors/warnings",
"Completed with errors/warnings", "Completed with errors/warnings" ],
  "jobItems" : [ {
    "startDate" : "2020-01-13T13:06:36.150-05:00",
    "endDate" : "2020-01-13T13:06:40.808-05:00",
    "status" : "COMPLETED WITH ERRORS",
    "owningTable" : "PersonView",
    "errorMessages" : [ "Completed with errors/warnings", "Completed with errors/
warnings", "Completed with errors/warnings", "Completed with errors/warnings",
"Completed with errors/warnings", "Completed with errors/warnings" ],
    "progress" : 100,
    "metrics" : {
      "200" : 6,
      "300" : 5,
      "201" : 5,
      "301" : 4,
      "202" : 1,
      "302" : 1
    }
  } ],
  "parameters" : {
    "mappingId" : "SVR1.1G7UX",
    "fileName" : "my.pdf",
    "systemName" : "Admin",
    "fileId" : "DB_SVR1.1G7UW"
  },
  "metricLabels" : {
    "200" : "Total lines in file",
    "201" : "Converted lines",
    "300" : "Total records",
    "202" : "Rejected lines",
    "301" : "Loaded records",
    "302" : "Rejected records"
  },
  "progress" : 100
}

```

Répertoire un groupe de travaux

L'API REST Répertoire un groupe de travaux renvoie des données pour chaque travail d'un groupe de lots.

L'API utilise la méthode GET.

URL de demande

L'URL Répertoire un groupe de travaux est au format suivant :

```
http://<host>:<port>/<jobcontrol>/<orsId>/<groupBy>
```

Effectuez la demande HTTP GET suivante à l'URL Répertoire un groupe de travaux :

```
GET http://<host>:<port>/<jobcontrol>/<orsId>/<groupBy>
```

Paramètres

Spécifiez les paramètres permettant de renvoyer des données pour chaque travail d'un groupe de lots.

Le tableau suivant décrit les paramètres dans le corps de demande :

Paramètre	Description
orsID	ID du stockage de référence opérationnelle de la base de données à partir duquel récupérer le mappage.
groupBy	Facultatif. Groupez les travaux selon le type de travail demandé.
jobGroupControlId	Spécifiez l'ID de contrôle de groupe de travaux de lots.

Exemple de demande d'API

L'exemple de demande suivant renvoie des données pour chaque travail d'un groupe de lots :

```
GET /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/group?jobGroupType=IMPORT HTTP/1.1
Host: localhost:8080
```

Exemple de réponse d'API

L'exemple de réponse suivant présente des données pour chaque travail d'un groupe de lots :

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json; charset=UTF-8
Content-Length: 564

{
  "records" : [ {
    "jobGroupControlId" : "SVR1.1G7UY",
    "startDate" : "2020-01-13T13:06:26.649-05:00",
    "status" : "PENDING",
    "parameters" : { },
    "metricLabels" : { }
  }, {
    "jobGroupControlId" : "SVR1.1G61Y",
    "startDate" : "2020-01-13T12:39:42.152-05:00",
    "endDate" : "2020-01-13T12:40:03.221-05:00",
    "status" : "COMPLETED_WITH_ERRORS",
    "statusMessage" : "Completed with errors/warnings",
    "parameters" : { },
    "metricLabels" : { }
  } ],
  "totalCount" : 2,
  "firstRecord" : 1,
  "pageSize" : 10
}
```

Obtenir les métadonnées DaaS

L'API REST Obtenir les métadonnées DaaS renvoie des informations sur le fournisseur DaaS telles que le nom, le type, l'entité d'entreprise avec lequel il travaille ainsi que la liste des champs requis.

L'API utilise la méthode GET.

URL de demande

Le format de l'URL Obtenir les métadonnées DaaS est le suivant :

```
http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Effectuez la demande HTTP GET suivante à l'URL Obtenir les métadonnées DaaS :

```
GET http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Paramètre de requête

Le paramètre `providerName` est un paramètre requis. Le paramètre est le nom du fournisseur DaaS configuré.

Exemple de demande d'API

L'exemple de demande suivant renvoie les informations de métadonnées du fournisseur DaaS `dcp` :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/dcp
```

L'exemple de demande suivant renvoie les informations de métadonnées du fournisseur DaaS `ondemand` :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/ondemand
```

Exemple de réponse d'API

L'exemple suivant montre les informations de métadonnées du fournisseur DaaS `dcp` au format JSON :

```
{
  "providerName": "dcp",
  "providerType": "READ",
  "businessEntity": "Organization",
  "systemName": "SFA",
  "requiredFields": [
    "dunsNumber"
  ]
}
```

L'exemple suivant montre les informations de métadonnées du fournisseur DaaS `ondemand` au format JSON :

```
{
  "providerName": "ondemand",
  "providerType": "SEARCH",
  "businessEntity": "Organization",
  "systemName": "SFA",
  "requiredFields": [
    "displayName"
  ]
}
```

Recherche DaaS

L'API REST de recherche DaaS utilise certains champs d'entrée d'une entité d'entreprise pour appeler un service DaaS externe et transforme la réponse en une liste d'enregistrements.

L'API utilise la méthode POST.

URL de demande

Le format de l'URL Recherche DaaS est le suivant :

```
http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```


Effectuez la demande HTTP POST suivante à l'URL Recherche DaaS :

```
POST http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```

Remarque: dans le corps de demande, indiquez les détails de l'entité d'entreprise à l'aide du champ requis.

Corps de demande

Le corps de demande doit contenir un élément XML ou JSON de type `Organization` ou `OrganizationView` de l'espace de noms `urn:co-ors.informatica.mdm`.

Exemple de demande d'API

L'exemple de demande suivant s'adresse au fournisseur DaaS `ondemandet` concerne la recherche de l'entité d'entreprise `Organisation` dont le nom d'affichage est `INFA` :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant montre les résultats d'une recherche renvoyés par un fournisseur DaaS pour l'entité d'entreprise `Organisation` dont le nom d'affichage est `INFA` :

```
{
  "link": [],
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "TelephoneNumbers": {
            "link": [],
            "item": [
              {
                "phoneNum": "9736252265"
              }
            ]
          },
          "Addresses": {
            "link": [],
            "item": [
              {
                "Address": {
                  "cityName": "ROCKAWAY",
                  "addressLine1": "11 WALL ST",
                  "postalCd": "07866",
                  "stateCd": {
                    "stateAbbreviation": "NJ"
                  }
                }
              }
            ]
          }
        }
      },
      "label": "001352574-INFA LAB INC",
      "systemName": "SFA"
    },
    {
      "businessEntity": {
        "Organization": {

```

```

        "displayName": "INFA INC",
        "dunsNumber": "007431013",
        "TelephoneNumbers": {
            "link": [],
            "item": [
                {
                    "phoneNum": "5629019971"
                }
            ]
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "LONG BEACH",
                        "addressLine1": "3569 GARDENIA AVE",
                        "postalCd": "90807",
                        "stateCd": {
                            "stateAbbreviation": "CA"
                        }
                    }
                }
            ]
        }
    },
    "label": "007431013-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "020591086",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "7186248777"
                    }
                ]
            }
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "BROOKLYN",
                        "addressLine1": "16 COURT ST STE 2004",
                        "postalCd": "11241",
                        "stateCd": {
                            "stateAbbreviation": "NY"
                        }
                    }
                }
            ]
        }
    }
},
    "label": "020591086-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "604057286",
            "TelephoneNumbers": {
                "link": [],
                "item": [

```

```

        {
            "phoneNum": "8473580802"
        }
    ],
    },
    "Addresses": {
        "link": [],
        "item": [
            {
                "Address": {
                    "cityName": "PALATINE",
                    "addressLine1": "THE HARRIS BANK BLDG STE 614,800E NW HWY",
                    "postalCd": "60074",
                    "stateCd": {
                        "stateAbbreviation": "IL"
                    }
                }
            }
        ]
    }
},
    "label": "604057286-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "032785606",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "5629019971"
                    }
                ]
            }
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "SIMI VALLEY",
                        "addressLine1": "3962 HEMWAY CT",
                        "postalCd": "93063",
                        "stateCd": {
                            "stateAbbreviation": "CA"
                        }
                    }
                }
            ]
        }
    }
},
    "label": "032785606-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA",
            "dunsNumber": "045228877",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "3304410033"
                    }
                ]
            }
        }
    }
},

```

```

    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "NORTON",
            "addressLine1": "4725 ROCK CUT RD",
            "postalCd": "44203",
            "stateCd": {
              "stateAbbreviation": "OH"
            }
          }
        }
      ]
    }
  },
  "label": "045228877-INFA",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "609028209",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "9084394655"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "CALIFON",
            "addressLine1": "21 FAIRMOUNT RD W",
            "postalCd": "07830",
            "stateCd": {
              "stateAbbreviation": "NJ"
            }
          }
        }
      ]
    }
  },
  "label": "609028209-INFA INC",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "195271796",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "7137824181"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {

```

```

        "cityName": "HOUSTON",
        "addressLine1": "1800 AUGUSTA DR STE 200",
        "postalCd": "77057",
        "stateCd": {
            "stateAbbreviation": "TX"
        }
    }
}
]
}
},
"label": "195271796-INFA INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "098605830",
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "BELLFLOWER",
                            "postalCd": "90707",
                            "stateCd": {
                                "stateAbbreviation": "CA"
                            }
                        }
                    }
                ]
            }
        }
    }
},
"label": "098605830-INFA INC",
"systemName": "SFA"
}
]
}

```

Lecture DaaS

L'API REST de lecture DaaS utilise certains champs d'une entité d'entreprise pour demander un service DaaS externe et transforme la réponse en un enregistrement complet.

L'API utilise la méthode POST. Spécifiez le champ requis dans la demande au fournisseur DaaS.

URL de demande

Le format de l'URL Lecture DaaS est le suivant :

```
http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Effectuez la demande POST suivante à l'URL Lecture DaaS :

```
POST http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Remarque: Dans le corps de demande, indiquez les détails de l'enregistrement à l'aide du champ requis.

Paramètre de requête

Le tableau suivant répertorie les paramètres de requête que vous pouvez utiliser :

Paramètre	Description
logChanges	Facultatif. Si la valeur est définie sur True, l'enregistrement qui en résulte comprend le résumé des modifications SDO (Service Data Object) transférées au service d'entité d'entreprise d'écriture. La valeur par défaut est False.

Corps de demande

Le corps de demande doit contenir un élément XML ou JSON de type `OrganizationView` de l'espace de noms `urn:coors.informatica.mdm`.

Exemple de demande d'API

L'exemple de demande suivant s'adresse au fournisseur DaaS `ondemandet` concerne la recherche de l'entité d'entreprise `Organisation` dont le nom d'affichage est `INFA` :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Exemple de réponse d'API

L'exemple de réponse suivant montre les détails renvoyés par le fournisseur DaaS pour l'organisation dont le numéro D-U-N-S est `001352574` :

```
{
  "Organization": {
    "displayName": "Infa Lab Inc",
    "dunsNumber": "001352574",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "phoneNum": "09736250550"
        }
      ]
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "Rockaway",
            "addressLine1": "11 WALL ST"
          }
        }
      ]
    }
  }
}
```

WriteMerge

L'API REST WriteMerge accepte une liste d'enregistrements récupérés d'un fournisseur DaaS, les conserve dans des références croisées séparées avec le système source approprié et les fusionne en une seule entité d'entreprise. Toutes les références croisées appartiennent au même enregistrement.

L'API utilise la méthode POST.

URL de demande

Le format de l'URL WriteMerge est le suivant :

```
http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Effectuez la demande HTTP POST suivante à l'URL WriteMerge :

```
POST http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Corps de demande

Le corps de demande doit contenir un élément XML ou JSON de type `DaaSEntity.Pager` de l'espace de noms `urn:cobase.informatica.mdm`.

En-tête de réponse

Lorsque la réponse est réussie, l'API renvoie `interactionId` et `processId` dans l'en-tête de la réponse.

Exemple de demande d'API

L'exemple de demande suivant utilise le résultat d'une recherche DaaS en tant qu'entrée pour créer un enregistrement de l'entité d'entreprise de type `Organisation` :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/write-merge/Organization
{
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "TelephoneNumbers": {
            "item": [
              {
                "phoneNum": "9736252265"
              }
            ]
          }
        }
      },
      "systemName": "DNB"
    },
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA INC",
          "dunsNumber": "007431013",
          "TelephoneNumbers": {
            "item": [
              {
                "phoneNum": "5629019971"
              }
            ]
          }
        }
      }
    }
  ],
}
```

```

        "systemName": "Admin"
    }
  ]
}

```

Exemple de réponse d'API

L'exemple de réponse suivant montre l'en-tête et le corps de réponse après la fusion réussie de la liste des enregistrements en un seul enregistrement :

```

{
  "Organization": {
    "key": {
      "rowid": "121921",
      "sourceKey": "140000000000"
    },
    "rowidObject": "121921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "21721",
            "sourceKey": "140000001000"
          },
          "rowidObject": "21721"
        }
      ]
    }
  }
}

```

Importation DaaS

L'API REST d'importation DaaS accepte les données au format XML et les convertit en un enregistrement.

L'API utilise la méthode POST.

URL de demande

Le format de l'URL Importer des données DaaS est le suivant :

```

http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```

Effectuez la demande HTTP POST suivante à l'URL Importer des données DaaS :

```

POST http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```


Paramètres de la requête

Le nom du système source est un paramètre requis.

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans la demande :

Paramètre	Description
systemName	Requis. Nom du système source qui effectue la modification des données.
interactionId	Facultatif. Identificateur d'interaction à attribuer à toutes les modifications. En général, le Hub génère l'identificateur lorsqu'il crée une modification en attente en tant que résultat d'un appel.
effectivePeriod	Facultatif. Contient la date de début et de fin. Spécifie la période pendant laquelle l'enregistrement est effectif. Indiquez ces paramètres pour un enregistrement dont la chronologie est activée.
validateOnly	Facultatif. Si ce paramètre est défini sur TRUE, seules les règles de validation sont appliquées à l'enregistrement modifié et les modifications ne sont pas persistantes.
recordState	Facultatif. État du Hub pour les enregistrements créés ou mis à jour. Les états d'enregistrements pris en charge sont ACTIVE et PENDING.
processId	Facultatif. ID du processus de flux de travail qui contient la tâche. Lorsqu'un flux de travail démarre suite à un appel de service, le paramètre contient l'identificateur du processus de flux de travail démarré.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Corps de demande

Le corps de demande doit contenir un élément XML ou JSON de type

DaaSChangeFamilyTreeMemberOrganizationToOrgView de l'espace de noms urn:co-ors.informatica.mdm.

En-tête de réponse

Lorsque la réponse a réussi, l'API renvoie l'ID d'interaction et l'ID de processus dans l'en-tête de réponse et les détails de l'enregistrement dans le corps de réponse.

Si le processus génère un ID d'interaction et s'en sert pour créer l'enregistrement, l'API renvoie l'ID d'interaction. Si le processus démarre un flux de travail au lieu d'enregistrer directement l'enregistrement dans la base de données, l'API renvoie l'ID du processus de flux de travail.

L'exemple suivant montre un en-tête de réponse avec un ID d'interaction et un ID de processus :

```
BES-interactionId: 72200000242000  
BES-processId: 15948
```

Exemple de demande d'API

La demande est toujours au format XML.

L'exemple de demande suivant montre des données XML de type ChildAssociation de l'espace de noms de liaison :

```
POST http://localhost:8080/cmxcsllocalhost-orcl-MDM_SAMPLE/daas/import/linkage2org?
systemName=Admin
<FamilyTreeMemberOrganization xmlns="http://services.dnb.com/LinkageServiceV2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ChildAssociation">
  <AssociationTypeText>ParentSubsidiary</AssociationTypeText>
  <OrganizationName>
    <OrganizationPrimaryName>
      <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
  </OrganizationName>
  <SubjectHeader>
    <DUNSNumber>697557825</DUNSNumber>
  </SubjectHeader>
  <Location>
    <PrimaryAddress>
      <StreetAddressLine>
        <LineText>2-5-1, ATAGO</LineText>
      </StreetAddressLine>
      <StreetAddressLine>
        <LineText>ATAGO GREEN HILLS MORI TOWER 26F.</LineText>
      </StreetAddressLine>
      <PrimaryTownName>MINATO-KU</PrimaryTownName>
      <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
      <TerritoryAbbreviatedName>TKY</TerritoryAbbreviatedName>
      <PostalCode>105-0002</PostalCode>
      <TerritoryOfficialName>TOKYO</TerritoryOfficialName>
    </PrimaryAddress>
  </Location>
  <OrganizationDetail>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Subsidiary</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Headquarters</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
  </OrganizationDetail>
  <Linkage>
    <LinkageSummary>
      <ChildrenSummary>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Affiliate</ChildrenTypeText>
        <ChildrenQuantity>29</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Branch</ChildrenTypeText>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>true</DirectChildrenIndicator>
      </ChildrenSummary>
      <SiblingCount>29</SiblingCount>
    </LinkageSummary>
    <GlobalUltimateOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
      <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <ParentOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </ParentOrganization>
    <FamilyTreeMemberOrganization>
      <AssociationTypeText>HeadquartersBranch</AssociationTypeText>
      <OrganizationName>
        <OrganizationPrimaryName>
```

```

        <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
</OrganizationName>
<SubjectHeader>
    <DUNSNumber>692640710</DUNSNumber>
    <SubjectHandling>
        <SubjectHandlingText DNBCodeValue="11028">De-listed</
SubjectHandlingText>
    </SubjectHandling>
</SubjectHeader>
<Location>
    <PrimaryAddress>
        <StreetAddressLine>
            <LineText>2-2-2, UMEDA, KITA-KU</LineText>
        </StreetAddressLine>
        <PrimaryTownName>OSAKA</PrimaryTownName>
        <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
        <TerritoryAbbreviatedName>OSK</TerritoryAbbreviatedName>
        <PostalCode>530-0001</PostalCode>
        <TerritoryOfficialName>OSAKA</TerritoryOfficialName>
    </PrimaryAddress>
</Location>
<OrganizationDetail>
    <FamilyTreeMemberRole>
        <FamilyTreeMemberRoleText>Branch</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
</OrganizationDetail>
<Linkage>
    <GlobalUltimateOrganization>
        <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <HeadquartersOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </HeadquartersOrganization>
    <FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>
<FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>

```

Exemple de réponse d'API

L'exemple de réponse suivant montre l'en-tête et le corps de la réponse après l'importation et la création réussies d'une entité d'entreprise Organisation :

```

BES-interactionId: 72202100242034
BES-processId: 156048
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921"
  }
}

```

Mise à jour DaaS

L'API REST de mise à jour DaaS accepte les données au format XML avant et après une modification. L'API applique les modifications à l'enregistrement.

L'API utilise la méthode POST.

URL de demande

Le format de l'URL Mise à jour DaaS est le suivant :

```
http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Effectuez la demande HTTP POST suivante à l'URL Mise à jour DaaS :

```
POST http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Paramètres de la requête

Le nom du système source est un paramètre requis.

Le tableau suivant répertorie les paramètres que vous pouvez utiliser dans la demande :

Paramètre	Description
systemName	Requis. Nom du système source qui effectue la modification des données.
interactionId	Facultatif. Identificateur d'interaction à attribuer à toutes les modifications. En général, le Hub génère l'identificateur lorsqu'il crée une modification en attente en tant que résultat d'un appel.
effectivePeriod	Facultatif. Contient la date de début et de fin. Spécifie la période pendant laquelle l'enregistrement est effectif. Indiquez ces paramètres pour un enregistrement dont la chronologie est activée.
validateOnly	Facultatif. Si ce paramètre est défini sur TRUE, seules les règles de validation sont appliquées à l'enregistrement modifié et les modifications ne sont pas persistantes.
recordState	Facultatif. État du Hub pour les enregistrements créés ou mis à jour. Les états d'enregistrements pris en charge sont ACTIVE et PENDING.
processId	Facultatif. ID du processus de flux de travail qui contient la tâche. Lorsque le flux de travail démarre suite à un appel de service, le paramètre contient l'identificateur du processus de flux de travail démarré.

LIENS CONNEXES :

- ["Date et heure au format UTC" à la page 33](#)

Corps de demande

Le corps de demande doit contenir un élément XML ou JSON de type

DaaSChangeFamilyTreeMemberOrganizationToOrgView de l'espace de noms urn:co-ors.informatica.mdm.

En-tête de réponse

Lorsque la réponse a réussi, l'API renvoie l'ID d'interaction et l'ID de processus dans l'en-tête de réponse et les détails de l'enregistrement dans le corps de réponse.

Si le processus génère un ID d'interaction et s'en sert pour créer l'enregistrement, l'API renvoie l'ID d'interaction. Si le processus démarre un flux de travail au lieu d'enregistrer directement l'enregistrement dans la base de données, l'API renvoie l'ID du processus de flux de travail.

L'exemple suivant montre un en-tête de réponse avec un ID d'interaction et un ID de processus :

```
BES-interactionId: 72200000242000
BES-processId: 15948
```

Exemple de demande d'API

L'API accepte deux réponses au format XML, l'une avant la modification et l'autre après. Dans la demande suivante, un nouveau numéro de téléphone est ajouté à l'organisation. Les données XML *before* ne contiennent pas de numéro de téléphone, alors que les données XML *after* en contiennent un.

La demande suivant contient le numéro de téléphone nouvellement ajouté :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/update/linkage2org?
systemName=Admin
<urn:DaaSChangelinkage2org xmlns:urn="urn:cs-ors.informatica.mdm" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:type="urn:DaaSChangelinkage2org">
  <urn:before xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
  </urn:before>

  <urn:after xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
    <Telecommunication>
      <TelephoneNumber>
        <TelecommunicationNumber>09736250550</TelecommunicationNumber>
        <InternationalDialingCode>1</InternationalDialingCode>
        <UnreachableIndicator>true</UnreachableIndicator>
      </TelephoneNumber>
    </Telecommunication>
  </urn:after>
</urn:DaaSChangelinkage2org>
```

Exemple de réponse d'API

L'exemple suivant montre le rowID du numéro de téléphone créé pour l'organisation :

```
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "1722",
            "sourceKey": "09736250550"
          },
          "rowidObject": "1722"
        }
      ]
    }
  }
}
```

CHAPITRE 4

Appels de services d'entité commerciale du protocole SOAP (Simple Object Access Protocol)

Ce chapitre comprend les rubriques suivantes :

- [Appels du protocole SOAP \(Simple Object Access Protocol\) pour les services d'entité commerciale, 254](#)
- [Méthode d'authentification, 255](#)
- [Cookies d'authentification pour la connexion à partir d'applications tierces, 255](#)
- [Fichier Web Services Description Language, 256](#)
- [URL SOAP, 257](#)
- [Demandes et réponses SOAP, 258](#)
- [Affichage des paramètres d'entrée et de sortie, 259](#)
- [Référence d'API SOAP, 260](#)
- [Exemple de demande et de réponse SOAP, 261](#)

Appels du protocole SOAP (Simple Object Access Protocol) pour les services d'entité commerciale

Les appels de point d'extrémité du protocole SOAP (Simple Object Access Protocol) rendent tous les services d'entité commerciale disponibles en tant que services Web. Vous pouvez effectuer des appels SOAP pour créer, supprimer, mettre à jour et rechercher des enregistrements dans une entité commerciale. Vous pouvez effectuer différentes opérations, par exemple fusionner, annuler la fusion et faire correspondre des enregistrements. Vous pouvez aussi effectuer des appels SOAP pour créer, mettre à jour et rechercher des tâches et pour exécuter des tâches.

Les points d'extrémité SOAP pour les services d'entité commerciale utilisent le point d'extrémité UsernameToken de Web Services Security (WS-Security) pour authentifier les utilisateurs.

Remarque: Avant d'utiliser les API SOAP pour appeler les services d'entité commerciale, validez le stockage de référence opérationnelle.

Méthode d'authentification

Tous les appels SOAP aux services d'entité commerciale requièrent une authentification utilisateur. Indiquez le nom d'utilisateur et le mot de passe dans l'en-tête du message SOAP de la demande de service Web.

L'élément d'en-tête SOAP Security contient les données de sécurité. L'élément Security contient l'élément UsernameToken constitué des éléments enfants suivants :

nom d'utilisateur

Nom d'utilisateur associé au jeton.

mot de passe

Mot de passe du nom d'utilisateur associé au jeton.

Envoie le nom d'utilisateur et le mot de passe vers l'élément UsernameToken.

L'exemple suivant montre l'élément d'en-tête Security dans le message SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
      secext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
</soapenv:Body>
.....
</soapenv:Body>
</soapenv:Envelope>
```

Cookies d'authentification pour la connexion à partir d'applications tierces

Utilisez les cookies d'authentification pour authentifier les utilisateurs d'Informatica MDM Hub et appeler les services d'entité commerciale d'applications tierces. Vous pouvez obtenir un cookie basé sur les justificatifs d'identité d'un utilisateur authentifié. Enregistrez le cookie et utilisez-le pour appeler les API SOAP. Vous n'avez pas besoin de coder en dur le nom d'utilisateur et le mot de passe.

Effectuez la demande POST suivante pour vous connecter à la vue Entity 360 à l'aide de votre nom d'utilisateur et de votre mot de passe :

```
POST http://<host>:<port>/e360/com.informatica.tools.mdm.web.auth/login
{
  user: 'admin',
  password: 'user password'
}
```

Lorsque l'opération de connexion a réussi, le serveur renvoie le cookie d'authentification dans le champ en-tête set-cookie. L'exemple de code suivant montre un champ set-cookie dans l'en-tête de réponse :

```
Set-Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzc1RjIyOQ==";
Version=1; Path=7
```

Stockez le hachage et utilisez-le dans l'en-tête de demande de vos appels d'API. Vous n'avez pas besoin d'indiquer de nom d'utilisateur et de mot de passe pour les appels d'API.

L'exemple suivant montre comment utiliser un cookie d'authentification dans l'en-tête de demande d'API :

```
GET http://<IP of host>/cmx/cs/localhost-orcl-DS_UI1  
Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzc1RjIyOQ=="
```

Fichier Web Services Description Language

Les fichiers Web Services Description Language (WSDL) contiennent les descriptions XML des services Web, les formats des demandes et des réponses SOAP et tous les paramètres. MDM Hub génère un fichier WSDL pour chaque stockage de référence opérationnelle.

Les fichiers WSDL de chaque stockage de référence opérationnelle se trouvent dans l'emplacement suivant :

```
http://<hôte>:<port>/cmx/csfiles
```


L'image suivante montre l'emplacement où vous pouvez télécharger le fichier WSDL pour les stockages de référence opérationnelle :



localhost-orcl-DS_UI2

[cs-rest.xsd](#)
[cs-base.xsd](#)
[co-base.xsd](#)
[co-types.xsd](#)
[co-ors.xsd](#)
[cs-ors.xsd](#)
[co-meta.xsd](#)
[task-base.xsd](#)

[localhost-orcl-DS_UI2.wadl](#)

[localhost-orcl-DS_UI2.wsdl](#)

localhost-orcl-DS_UI1

[cs-rest.xsd](#)
[cs-base.xsd](#)
[co-base.xsd](#)
[co-types.xsd](#)
[co-ors.xsd](#)
[cs-ors.xsd](#)
[co-meta.xsd](#)
[task-base.xsd](#)

[localhost-orcl-DS_UI1.wadl](#)

[localhost-orcl-DS_UI1.wsdl](#)

Cliquez sur le lien pour télécharger le fichier WSDL pour le stockage d'enregistrement opérationnel DS_UI1 ou DS_UI2.

URL SOAP

L'URL SOAP est l'adresse que vous utilisez pour vous connecter au serveur SOAP.

Une URL SOAP contient la syntaxe suivante :

`http://<hôte>:<port>/<contexte>/<identifiant de base de données>`

L'URL contient les champs suivants :

hôte

Hôte qui exécute la base de données.

port

Numéro de port utilisé par l'écouteur de la base de données.

contexte

Le contexte est toujours `cmx/services/BEServices`.

database ID

ID du stockage d'enregistrement opérationnel tel qu'il est enregistré dans l'outil Bases de données dans la console Hub.

L'exemple suivant montre une URL SOAP :

```
http://localhost:8080/cmx/services/BEServices/localhost-orcl-DS_UI1
```

Demandes et réponses SOAP

Utilisez le format de message SOAP XML pour envoyer des demandes via un client SOAP au service d'entité commerciale et pour recevoir les réponses du service d'entité commerciale au client. Le format des demandes et des réponses SOAP est identique.

Un message SOAP contient les éléments suivants :

Enveloppe

Définit le début et la fin du message.

En-tête

Facultatif. Contient des attributs supplémentaires tels que les détails d'authentification pour le traitement du message. Si l'élément En-tête est présent, il doit être le premier élément enfant de l'élément Enveloppe.

Corps

Contient les données XML traitées par le client ou le service Web.

Un message SOAP est au format suivant :

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
  </env:Header>

  <env:Body>
  </env:Body>

</env:Envelope>
```

Une demande SOAP est au format suivant :

```
POST /<host>:<port>/<context>/<database ID> HTTP/1.0
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
```

```

</env:Header>

<env:Body>
</env:Body>

</env:Envelope>

```

Une réponse SOAP est au format suivant :

```

HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

<env:Header>
</env:Header>

<env:Body>
</env:Body>

</env:Envelope>

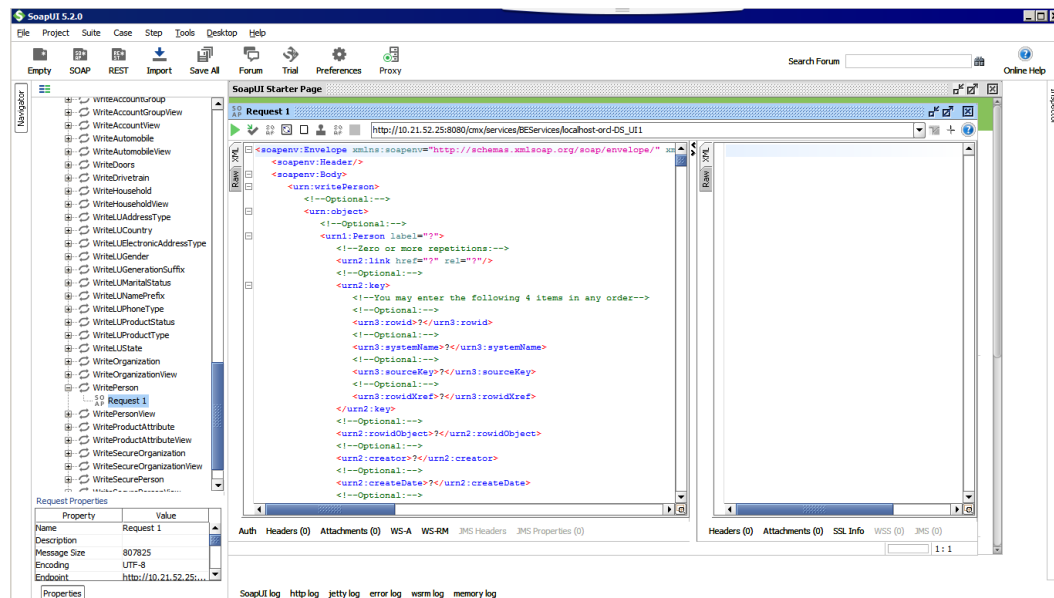
```

Affichage des paramètres d'entrée et de sortie

Vous pouvez utiliser un outil de test fonctionnel comme SoapUI pour afficher les paramètres d'entrée et de sortie de SOAP API.

Créez un projet SOAP et importez le fichier WSDL dans le projet. Les opérations que vous pouvez effectuer à l'aide des services d'entité commerciale apparaissent sous forme de nœuds dans SoapUI. Chaque opération contient un format de message de demande et de réponse. SoapUI crée un exemple de demande pour chaque opération lorsque vous importez le fichier WSDL.

Ouvrez le projet et double-cliquez sur une demande pour ouvrir l'éditeur de demande. L'image suivante montre les paramètres d'entrée dans SoapUI pour l'API SOAP WritePerson :



Référence d'API SOAP

La référence d'API SOAP pour les services d'entité d'entreprise répertorie les API SOAP et fournit une description de chaque API. Consultez aussi le fichier WSDL, qui contient des descriptions des services d'entité d'entreprise.

Utilisez les API SOAP suivantes pour effectuer des opérations sur des entités commerciales :

Obtenir les métadonnées

Renvoie la structure de données d'une entité d'entreprise.

Lister l'enregistrement

Renvoie la liste des valeurs de recherche ou des valeurs de clé étrangère.

Lire l'enregistrement

Renvoie les détails d'un enregistrement racine dans l'entité d'entreprise.

Créer un enregistrement

Crée un enregistrement dans l'entité d'entreprise spécifiée.

Mettre à jour l'enregistrement

Met à jour l'enregistrement racine spécifié et ses enregistrements enfants.

Supprimer un enregistrement

Supprime un enregistrement racine d'une entité d'entreprise.

Rechercher un enregistrement

Recherche une valeur de chaîne dans une entité d'entreprise racine de recherche et renvoie les enregistrements racines correspondant aux critères de recherche.

Prévisualiser la promotion

Renvoie un aperçu d'un enregistrement obtenu si vous promouvez les modifications en attente en fonction de l'ID d'interaction de la demande de modification.

Promouvoir

Promeut toutes les modifications en attente apportées à un enregistrement en fonction de l'ID d'interaction de la demande de modification.

Supprimer la promotion

Supprime toutes les modifications en attente apportées à un enregistrement en fonction de l'ID d'interaction de la demande de modification.

Prévisualiser la fusion

Renvoie un aperçu d'un enregistrement racine consolidé si vous fusionnez deux enregistrements racines ou plus.

Fusionner les enregistrements

Fusionne deux enregistrements racines ou plus pour créer un seul enregistrement consolidé.

Annuler la fusion des enregistrements

Annule la fusion d'un enregistrement racine dans les enregistrements racines individuels qui existaient avant la fusion des enregistrements.

Obtenir les enregistrements associés

Renvoie une liste des enregistrements associés en fonction des relations que vous configurez dans le gestionnaire de hiérarchies.

Lire les enregistrements correspondants

Renvoie les enregistrements correspondant à un enregistrement racine spécifié.

Mettre à jour les enregistrements correspondants

Crée ou met à jour un enregistrement dans la table de correspondance.

Supprimer les enregistrements correspondants

Supprime les enregistrements correspondants de la table de correspondance.

Obtenir les métadonnées BPM

Renvoie les types de tâches et deux indicateurs qui indiquent si l'adaptateur de flux de travail peut exécuter le service Obtenir le lignage des tâches ainsi que les services d'administration.

Lister les tâches

Renvoie une liste de tâches de flux de travail.

Lire la tâche

Renvoie les détails d'une tâche.

Créer une tâche

Crée une tâche et démarre un flux de travail.

Mettre à jour la tâche

Met à jour une seule tâche.

Exécuter l'action de tâche

Exécute une action de tâche et redéfinit la tâche sur le flux de travail pour un traitement ultérieur.

Lister les utilisateurs pouvant être attribués

Renvoie la liste des utilisateurs auxquels vous pouvez attribuer les tâches appartenant à un type de tâche.

Tâche terminée

Ferme un flux de travail de tâches une fois toutes les tâches terminées.

Exemple de demande et de réponse SOAP

L'exemple de demande SOAP suivant récupère une liste des utilisateurs pouvant être attribués :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
seceext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
  <soapenv:Body>
    <urn:listAssignableUsers>
      <!--Optional:-->
      <urn:parameters>
        <!--Optional:-->
        <urn:taskType>Update</urn:taskType>
      </urn:parameters>
    </urn:listAssignableUsers>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <!--Optional:-->
        <urn:businessEntity>Person</urn:businessEntity>
    </urn:parameters>
</urn:listAssignableUsers>
</soapenv:Body>
</soapenv:Envelope>

```

L'exemple de réponse SOAP suivant liste les utilisateurs pouvant être attribués :

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns6:listAssignableUsersReturn xmlns:ns1="urn:cs-base.informatica.mdm"
      xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns3="urn:co-ors.informatica.mdm"
      xmlns:ns4="urn:co-meta.informatica.mdm" xmlns:ns5="urn:task-base.informatica.mdm"
      xmlns:ns6="urn:cs-ors.informatica.mdm">
      <ns6:object>
        <ns1:users>
          <user>
            <userName>admin</userName>
          </user>
        </users>
        <ns1:roles/>
      </ns6:object>
    </ns6:listAssignableUsersReturn>
  </soapenv:Body>
</soapenv:Envelope>

```

CHAPITRE 5

Services d'enregistrements de références croisées et de calculs de MVV

Ce chapitre comprend les rubriques suivantes :

- [Aperçu des calculs de services d'enregistrements de références croisées et de la MVV, 263](#)
- [Obtenir les calculs de données de références croisées et d'examen de la MVV, 263](#)
- [Réponses de filtrage et de pagination, 268](#)
- [Établir la meilleure version de la vérité, 268](#)

Aperçu des calculs de services d'enregistrements de références croisées et de la MVV

Vous pouvez utiliser les calculs de services d'enregistrements de références croisées et de la meilleure version de la vérité (MVV) pour déterminer comment les données source forment l'enregistrement principal.

Vous pouvez utiliser ces services pour effectuer les tâches suivantes :

- Rassembler des informations sur les données source
- Déterminer la manière dont la meilleure version de la vérité a été déterminée
- Remplacer les calculs de la MVV pour vous assurer que les enregistrements principaux contiennent la meilleure version de la vérité

Obtenir les calculs de données de références croisées et d'examen de la MVV

Les enregistrements principaux de MDM Hub conservent la meilleure version de la vérité (MVV). MDM Hub consolide les données les plus fiables provenant de plusieurs systèmes source de chaque enregistrement principal pour obtenir la meilleure version de la vérité. MDM Hub stocke les données source dans des

enregistrements de références croisées. Vous pouvez utiliser les services d'entité d'entreprise pour lire les données des enregistrements de références croisées et déterminer la manière dont la MVV a été calculée.

Obtenir les enregistrements de références croisées

Vous pouvez utiliser un service d'entité d'entreprise pour obtenir les enregistrements de références croisées pour un enregistrement principal spécifique.

Le format de l'URL de l'API REST permettant d'obtenir les enregistrements de références croisées est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=XREF
```

L'exemple de demande suivant récupère les enregistrements de références croisées pour un enregistrement d'entité d'entreprise Personne dont l'ID de ligne est 123 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF
```

Obtenir une réponse des enregistrements de références croisées

L'exemple suivant montre les enregistrements de références croisées qui sont renvoyés pour l'enregistrement Personne dont l'ID de ligne est 123 :

```
GET /Person/123?contentMetadata=XREF
```

```
{
  "firstName": "Joe",
  "lastName": "Smith",
  "XREF": {
    "item": [
      {
        "rowidXref": 111,
        "firstName": "Joe",
        "lastName": "Smith",
      },
      {
        "rowidXref": 222,
        "firstName": "John",
        "lastName": "Smith"
      }
    ]
  }
}
```

Déterminer les contributeurs à l'enregistrement principal

Vous pouvez utiliser les services d'entité d'entreprise pour voir les champs d'enregistrements de références croisées qui contribuent à l'enregistrement principal. Chaque enregistrement qui contribue à un champ est identifié par l'ID de ligne de l'enregistrement de références croisées.

Le format de l'URL de l'API REST qui détermine les contributeurs à l'enregistrement principal est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=BVT
```

L'exemple de demande suivant récupère les informations de MVV pour un enregistrement Personne dont l'ID de ligne est 123 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=BVT
```


Déterminer les contributeurs à la réponse d'enregistrement principal

L'exemple suivant montre l'enregistrement de références croisées qui a contribué à chaque champ de l'enregistrement principal :

```
{
  "firstName": "Joe",
  "lastName": "Smith",
  "BVT": {
    "firstName": {
      "rowidXref": 111
    },
    "lastName": {
      "rowidXref": 222
    }
  }
}
```

Obtenir les scores d'approbation des champs Enregistrements des références croisées contributifs

Vous pouvez utiliser des services d'entité d'entreprise pour obtenir les scores d'approbation des champs d'enregistrements de références croisées qui contribuent à l'enregistrement principal.

Le format de l'URL de l'API REST qui détermine les contributeurs et obtient les scores d'approbation est le suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=TRUST
```

L'exemple de demande suivant fournit les scores d'approbation pour un enregistrement Personne dont l'ID de ligne est 123 :

```
GET http://localhost:8080/cmx/cs/ors/Person/123?contentMetadata=TRUST
```

Obtenir les scores d'approbation de la réponse des champs Enregistrements de références croisées contributifs

L'exemple de réponse suivant fournit des scores d'approbation pour chaque champ d'un enregistrement d'entité d'entreprise Personne :

```
{
  "firstName": "John",
  "lastName": "Smith",
  "TRUST": {
    "firstName": {
      "score": 75.0,
      "valid": true,
      "trustSetting": {
        // custom settings
      }
    },
    "lastName": {
      "score": 80.0,
      "valid": true,
      "trustSetting": {
        // custom settings
      }
    }
  }
}
```

Obtention des scores d'approbation de tous les champs Enregistrements de références croisées

Utilisez l'API REST dont le paramètre contentMetadata est défini sur XREF_TRUST pour obtenir les scores d'approbation et les pourcentages de mise au niveau inférieur de tous les champs d'enregistrements de références croisées.

URL de demande de l'API REST Lire permettant de déterminer les contributeurs et d'obtenir les scores d'approbation :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?  
contentMetadata=XREF_TRUST
```

L'exemple de demande suivant récupère les données de références croisées pour un enregistrement Personne dont l'ID de ligne est 123 :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF_TRUST
```

Obtenir les scores d'approbation de la réponse de tous les champs Enregistrements de références croisées

L'exemple suivant montre les scores d'approbation et les pourcentages de mise au niveau inférieur de tous les champs d'enregistrements de références croisées d'une entité d'entreprise Personne :

```
{  
  "firstName": "Sergey",  
  "lastName": "Ivanov",  
  "XREF": {  
    "item": [{  
      "rowidXref": 111,  
      "firstName": "Sergey",  
      "lastName": "Petrov",  
      "TRUST": {  
        "firstName": {  
          "score": 75.0,  
          "valid": true  
        },  
        "lastName": {  
          "score": 60.0,  
          "valid": false,  
          "downgradePerCent": 20.0  
        }  
      }  
    }, {  
      "rowidXref": 222,  
      "firstName": "Sergey",  
      "lastName": "Ivanov",  
      "TRUST": {  
        "firstName": {  
          "score": 10.0,  
          "valid": true  
        },  
        "lastName": {  
          "score": 80.0,  
          "valid": true  
        }  
      }  
    }  
  ]  
}
```

Obtenir les informations sur les systèmes source

Vous pouvez obtenir les informations sur les systèmes source dont proviennent les données de références croisées ainsi que sur le nombre d'enregistrements de références croisées auxquels contribuent les systèmes source pour l'intégralité de l'enregistrement, pour chaque nœud ou pour chaque enregistrement.

Les paramètres suivants peuvent être spécifiés dans la demande :

describe

Définir cette valeur sur `True` pour renvoyer la description du système source. Les valeurs peuvent être `True` ou `False`. La valeur par défaut est `False`.

aggregate

Définit le niveau pour lequel renvoyer les informations de système source. Les valeurs peuvent être `ENTITY`, `NODE` et `RECORD`. La valeur par défaut est `ENTITY`.

recordStates

Spécifie l'état de l'enregistrement pour lequel renvoyer les enregistrements. La valeur peut être `ACTIVE`, `PENDING` ou `DELETED`. La valeur par défaut est `ACTIVE`.

compact

Lorsque cette valeur est définie sur `No`, les données du niveau de l'entité sont renvoyées lorsque les paramètres d'agrégation indiquent la valeur `ENTITY` et les autres niveaux agrégés. La valeur peut être définie sur `Yes` ou `No`. Pour les demandes d'API REST uniquement. La valeur par défaut est `Yes`.

Exemple d'obtention d'informations sur les systèmes source

L'exemple de demande suivant obtient des informations sur les systèmes source au niveau de l'entité et du nœud pour l'entité d'entreprise Personne dont l'ID de ligne est 123 :

```
GET http://localhost:8080/cmx/cs/ors/Person/123?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

L'exemple de demande suivant obtient des informations sur les systèmes source au niveau de l'enregistrement ainsi que les descriptions du système source pour l'entité d'entreprise Personne dont l'ID de ligne est 456 :

```
GET http://localhost:8080/cmx/cs/ors/Person/123/Address/456?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

Obtenir les informations sur la réponse des systèmes source

L'exemple suivant montre des informations sur le niveau de l'entité et le niveau du nœud d'une entité d'entreprise Personne :

```
{
  {
    "name": "Admin",
    "xrefCount": 120
  },
  Person: {
    "rowidObject": "456",
    "sourceSystem": {
      {
        "name": "Admin",
        "xrefCount": 30
      }
    }
  }
}
```

Réponses de filtrage et de pagination

Vous pouvez sélectionner les champs à renvoyer dans les réponses, filtrer les résultats selon plusieurs critères et paginer les résultats.

Exemples de demandes de filtration

Le tableau suivant montre des exemples de demandes pour l'entité d'entreprise Personne avec une variété de filtres appliqués ainsi qu'une description des résultats renvoyés dans la réponse :

Demande	Description des résultats renvoyés
/Person/123	Tous les champs définis par l'utilisateur
/Person/123?readSystemFields=true	Tous les champs définis par l'utilisateur et tous les champs système
/Person/123?fields=firstName	Un champ défini par l'utilisateur
/Person/123?fields=updatedAt	Un champ système
Person/123?fields=firstName,updatedAt	Un champ défini par l'utilisateur et un champ système
/Person/123?fields=firstName&readsystemFields=true	Un champ défini par l'utilisateur et tous les champs système

Établir la meilleure version de la vérité

Une fois qu'un gestionnaire de données a examiné les données source dans les enregistrements de références croisées, il peut apporter des ajustements à la manière dont les données source sont consolidées afin de s'assurer que l'enregistrement principal représente la meilleure version de la vérité.

Vous pouvez utiliser les services d'entité d'entreprise pour effectuer les actions suivantes et établir la meilleure version de la vérité :

- Mettre à jour les paramètres d'approbation
- Supprimer des données source sans correspondance
- Sélectionner le champ de contribution correct
- Écrire la valeur correcte dans l'enregistrement principal

Sélectionner le champ de contribution correct

Si le champ contenant le score d'approbation le plus élevé ne contient pas la meilleure version de la vérité, un gestionnaire de données peut sélectionner le champ contenant les données correctes pour fournir les données à l'enregistrement principal.

Le format de l'URL et du corps de demande permettant de sélectionner le champ de contribution correct en fonction du nom du système et de la clé source est le suivant :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
{
```

```

    BVT: {
      <field name>: {
        systemName: "<source system name>",
        sourceKey: "<source key>"
      }
    }
  }
}

```

Le format de l'URL et du corps de demande permettant de sélectionner le champ de contribution correct en fonction de l'identificateur de l'enregistrement de références croisées est le suivant :

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
{
  BVT: {
    <field name>: {
      rowidXref: "<row ID>"
    }
  }
}

```

Sélectionner l'exemple de champ de contribution correct

L'URL suivante et le corps de demande sélectionnent le champ Prénom de l'enregistrement de références croisées depuis le système source Ventes avec une clé de source de 0001 pour contribuer à l'enregistrement principal :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      systemName: "Sales",
      sourceKey: "0001"
    }
  }
}

```

L'URL suivante et le corps de demande sélectionnent le champ Prénom de l'enregistrement de références dont l'ID de ligne est 789 pour contribuer à l'enregistrement principal :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      rowidXref: "789"
    }
  }
}

```

Écrire la valeur correcte dans l'enregistrement principal

Lorsque vous utilisez un appel de service d'entité d'entreprise pour qu'il écrive une valeur correcte dans un enregistrement principal, vous pouvez également établir les paramètres d'approbation pour la valeur. Si vous ne spécifiez pas de paramètres d'approbation, MDM Hub utilise les paramètres système de l'administrateur.

Le format de l'URL et du corps de demande permettant d'écrire la valeur correcte avec le paramètre d'approbation de l'administrateur est le suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
  "TRUST": {

```

```

    "<field name>": {
      "trustSetting" : {
        custom: false
      }
    }
  }
}

```

Le format de l'URL et du corps de demande permettant d'écrire la valeur correcte avec le paramètre d'approbation défini est le suivant :

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: true, // if custom=true, all other trustSetting fields
                      //are mandatory. If they are not set,
                      //the service will return an error.
        minimumTrust: <minimum trust percent>,
        maximumTrust: <maximum trust percent>,
        timeUnit: "<units for measuring trust decay>",
        maximumTimeUnits: <number of units>,
        graphType: "<name of graph type>"
      }
    }
  }
}

```

Paramètres d'approbation

Vous pouvez définir les paramètres d'approbation suivants :

minimumTrust

Niveau d'approbation d'une valeur de données lorsqu'elle est « ancienne » (après écoulement de la période de déclin). Cette valeur doit être inférieure ou égale à l'approbation maximum.

Remarque: Si les niveaux d'approbation minimum et maximum sont égaux, la courbe de déclin est une ligne horizontale et la période et le type de déclin n'ont aucun effet.

maximumTrust

Niveau d'approbation d'une valeur de données lorsqu'elle vient d'être modifiée. Par exemple, si le système source X change un champ de numéro de téléphone de 555-1234 en 555-4321, le niveau d'approbation maximum du système X est attribué à la nouvelle valeur pour le champ de numéro de téléphone. En définissant un niveau d'approbation maximum relativement élevé, vous vous assurez que les modifications apportées aux systèmes source seront généralement appliquées à l'objet de base.

timeUnit

Indique les unités utilisées dans le calcul de la période de déclin : jour, semaine, mois, trimestre ou année.

maximumTimeUnits

Indique le nombre (de jours, de semaines, de mois, de trimestres ou d'années) utilisé dans le calcul de la période de déclin.

graphType

Le déclin suit un modèle dans lequel le niveau de confiance diminue durant la période de déclin. Les types de graphiques peuvent être l'un des modèles de déclin suivants :

Paramètre de type de graphique	Description
LINEAR	Déclin le plus simple. Le déclin suit une ligne droite de l'approbation maximum à l'approbation minimum.
RISL	La majorité de la diminution se produit au début de la période de déclin. Le déclin décrit une courbe concave. Si un système source a ce type de graphique, une nouvelle valeur provenant du système sera probablement approuvée. Mais, il est possible que cette valeur soit remplacée.
SIRL	La majorité de la diminution se produit à la fin de la période de déclin. Le déclin décrit une courbe convexe. Si un système source a ce type de graphique, il est relativement improbable qu'un autre système remplace la valeur dans l'enregistrement principal avant que cette valeur approche de la fin de la période de déclin.

Exemple d'écriture de valeur correcte dans l'enregistrement principal

Exemple 1

Vous voulez modifier le nom dans l'enregistrement principal de Sam Brown en John Smith. La modification est attribuée au système source Ventes. Les paramètres d'approbation sont définis selon les paramètres d'approbation de l'administrateur.

Le code suivant montre l'URL et la commande de l'exemple 1.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Sales
{
  "firstName": "John",
  "lastName": "Smith"
  "$original": {
    "firstName": "Sam",
    "lastName": "Brown"
  },
  "TRUST": {
    "firstName": {
      "trustSetting": {
        custom: false
      }
    },
    "lastName": {
      "trustSetting": {
        custom: false
      }
    }
  }
}
```

Exemple 2

Vous voulez modifier le nom dans l'enregistrement principal de Sam Brown en John Smith. La modification est attribuée au système source SFA. Les paramètres d'approbation sont définis sur une approbation minimum de 60 et maximum de 90, et les déclin d'approbation définis de manière linéaire sur une période de trois mois.

Le code suivant montre l'URL et la commande de l'exemple 2.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=SFA
{
  "firstName": "John",
  "lastName": "Smith"
  "$original": {
    "firstName": "Sam",
    "lastName": "Brown"
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: true,
        minimumTrust: 60,
        maximumTrust: 90,
        timeUnit: "Month",
        maximumTimeUnits: 3,
        graphType: "LINEAR"
      }
    }
    "lastName": {
      "trustSetting" : {
        custom: true,
        minimumTrust: 60,
        maximumTrust: 90,
        timeUnit: "Month",
        maximumTimeUnits: 3,
        graphType: "LINEAR"
      }
    }
  }
}
```

Supprimer des données source sans correspondance

Si un enregistrement de références croisées est incorrectement associé à un enregistrement principal spécifique, un gestionnaire de données peut annuler la fusion de l'enregistrement de références croisées. Un nouvel enregistrement principal est créé à partir de l'enregistrement de références croisées pour lequel la fusion a été annulée.

Lors d'un appel d'annulation de fusion, cette annulation ne peut se produire que sur un seul enregistrement de références croisées. Si l'annulation de fusion concerne plusieurs enregistrements de références croisées, effectuez un appel pour chacun d'eux.

Si un déclencheur est configuré pour un événement d'annulation de fusion, une tâche d'annulation de fusion est alors créée. Dans le cas contraire, l'annulation de fusion est opérée sur l'enregistrement de références croisées

Le format de l'URL et de la commande permettant l'annulation de fusion pour un enregistrement en fonction du nom de système et de la clé source est le suivant :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
  key: {rowid: "<rowid value>", sourcekey: "<source key>", systemName: "<source system
name>" }
}
```

Le format de l'URL et de la commande permettant l'annulation de fusion pour un enregistrement en fonction de l'identificateur de l'enregistrement de références croisées est le suivant :

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
```



```
key: {rowid: "<rowid value>", rowidXref: "<row ID of xref>"}
}
```

Supprimer l'exemple de données source sans correspondance

Exemple d'API REST

Le code suivant montre l'URL et la commande permettant d'annuler la fusion d'un enregistrement de références croisées au niveau de l'enfant à partir d'un enregistrement d'adresse :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person/181921?
action=unmerge&systemName=Admin
{
  "name": "Person.Address",
  "key": {
    "rowid": "41721 ",
    "rowidXref": 41722
  }
}
```

Où :

- l'enregistrement de références croisées pour lequel annuler la fusion a un ID de ligne 41722
- l'ID de ligne de l'enregistrement principal duquel annuler la fusion avec l'enregistrement de références croisées est 41721
- l'ID de ligne de l'enregistrement racine est 181921

Exemple de SOAP/EJB

Le code suivant montre l'URL et la commande permettant d'annuler la fusion d'un enregistrement de références croisées au niveau de l'enfant à partir d'un enregistrement d'adresse :

```
<ns9:UnMerge xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns7="urn:co-
meta.informatica.mdm" xmlns:ns3="http://services.dnb.com/LinkageServiceV2.0"
xmlns:ns8="urn:task-base.informatica.mdm" xmlns:ns6="urn:co-ors.informatica.mdm"
xmlns:ns1="urn:cs-base.informatica.mdm" xmlns:ns9="urn:cs-ors.informatica.mdm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns9:UnMerge">
  <ns9:parameters>
    <ns9:businessEntityKey name="Person">
      <ns1:key>
        <ns1:rowid>181921</ns1:rowid>
      </ns1:key>
    </ns9:businessEntityKey>
    <ns9:unmergeKey name="Person.TelephoneNumbers">
      <ns1:key>
        <ns1:rowid>41721 </ns1:rowid>
        <ns1:rowidXref>41722</ns1:rowidXref>
      </ns1:key>
    </ns9:unmergeKey>
    <ns9:treeUnmerge>true</ns9:treeUnmerge>
  </ns9:parameters>
</ns9:UnMerge>
```

Où :

- l'enregistrement de références croisées pour lequel annuler la fusion a un ID de ligne 41722
- l'ID de ligne de l'enregistrement principal duquel annuler la fusion avec l'enregistrement de références croisées est 41721
- l'ID de ligne de l'enregistrement racine est 181921

Réponse d'annulation de fusion

La réponse d'annulation de fusion contient l'ID de ligne de l'objet de base créé depuis l'enregistrement de références croisées non fusionné.

Exemple de réponse 1

L'exemple suivant montre la réponse lorsqu'un enregistrement de références croisées vient d'un nœud racine Personne.

```
{
  Person: {
    rowidObject: "7777"
  }
}
```

Exemple de réponse 2

L'exemple suivant montre la réponse lorsqu'un enregistrement de références croisées vient d'un nœud racine Adresse.

```
{
  Person: {
    Address: {
      item: [
        rowidObject: "55555"
      ]
    }
  }
}
```

CHAPITRE 6

Prise en charge du service de liaison d'entreprise

Ce chapitre comprend les rubriques suivantes :

- [Présentation, 275](#)
- [Services d'entité d'entreprise d'importation et de mise à jour DaaS, 275](#)
- [Configuration de la prise en charge de la liaison, 276](#)
- [Application personnalisée pour le fractionnement des données de liaison, 276](#)

Présentation

Le service de liaison d'entreprise de Duns & Bradstreet (D&B) renvoie le parent d'une organisation requise et toutes ses entités associées. Vous pouvez utiliser le service de liaison de D&B via lequel vous obtenez des informations sur toutes les branches et divisions d'une organisation. Les données émanant du service de liaison peuvent être utilisées pour créer et mettre à jour des enregistrements.

Vous pouvez importer les données de liaison d'entreprise dans MDM Hub. Vous devez développer une application personnalisée pouvant utiliser le service de liaison du composant personnalisé du fournisseur DaaS dans une vue Entité.

Un service d'entité d'entreprise est requis pour importer les données d'un service D&B et créer un enregistrement avec ces données. Lorsque des données du stockage externe sont modifiées, vous devez être capable d'effectuer les modifications correspondantes dans votre enregistrement. D&B propose un service de surveillance qui vous avertit en cas de modification des données. Vous avez besoin d'un service capable d'accepter les données avant et après une modification et d'appliquer les modifications à l'enregistrement correspondant.

Services d'entité d'entreprise d'importation et de mise à jour DaaS

Le service d'entité d'entreprise d'importation DaaS accepte les données du service de liaison au format XML et les convertit en un enregistrement. Le service d'entité d'entreprise de mise à jour DaaS accepte les données du service externe sous la forme de deux fichiers XML. Les deux fichiers XML correspondent aux

données avant et après modification. Le service de mise à jour applique les modifications à l'enregistrement correspondant.

LIENS CONNEXES :

- ["Importation DaaS" à la page 248](#)
- ["Mise à jour DaaS" à la page 251](#)

Configuration de la prise en charge de la liaison

Le service de liaison de D&B vous permet de créer et de mettre à jour des enregistrements. Pour ce faire, vous devez ajouter une configuration dans l'outil d'approvisionnement et créer une application personnalisée de façon à fractionner la réponse du service de liaison.

Effectuez les tâches suivantes pour configurer la prise en charge du service de liaison de D&B :

1. Utilisez l'outil d'approvisionnement pour charger le fichier WSDL du service de liaison.
2. Utilisez l'outil d'approvisionnement pour configurer un document XML sur la transformation d'entité d'entreprise et le présenter en tant que service. Lorsque vous présentez la transformation en tant que service, le processus crée les services d'entité d'entreprise d'importation et de mise à jour DaaS.
3. Créez une application personnalisée qui peut demander des données au service de liaison et fractionner la réponse entre détails d'enregistrement et détails de liaison.
4. Développez une interface utilisateur qui envoie la demande à l'application personnalisée.

Remarque: Pour plus d'informations sur le chargement du fichier WSDL et la configuration d'un document XML sur une transformation d'entité d'entreprise, consultez le chapitre Intégration de Data as a Service dans le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

Application personnalisée pour le fractionnement des données de liaison

Pour utiliser les services de liaison de D&B, vous devez concevoir une application personnalisée capable de fractionner les informations de liaison en détails d'enregistrement et détails de liaison.

L'application personnalisée doit effectuer les fonctions suivantes :

1. Accepter la demande d'un service de liaison depuis la vue Entité.
2. Envoyer la demande à D&B et recevoir la réponse.
3. Convertir la réponse au format XML.
4. Fractionner la réponse en détails d'enregistrement et de liaison.
5. Envoyer les informations XML aux services d'entité d'entreprise pour qu'elles soient enregistrées en tant qu'enregistrement dans la base de données.
6. Surveiller les modifications apportées aux données et appeler la fonction Notification de changement de liste du service externe.

CHAPITRE 7

Appels externes pour le nettoyage, l'analyse et la transformation de données

Ce chapitre comprend les rubriques suivantes :

- [Présentation, 277](#)
- [Événements pris en charge, 278](#)
- [Configuration des appels externes, 278](#)
- [Exemple : validation et logique personnalisées des services d'entité d'entreprise, 279](#)

Présentation

Les fournisseurs externes proposent des services Web de nettoyage, d'analyse et de transformation des données d'enregistrements. Utilisez ces services Web externes pour la validation personnalisée, comme vérifier si le champ d'adresse est vide lorsque vous ajoutez un enregistrement. Ils vous permettent également de personnaliser la logique afin de transformer les données d'enregistrements. Par exemple, lorsque vous fusionnez deux enregistrements, vous pouvez fusionner les adresses, mais pas les numéros de téléphone.

Un service Web externe présente une ou plusieurs opérations que les services d'entité d'entreprise peuvent appeler. Chaque opération contient un type de demande et de réponse. Les services d'entité d'entreprise envoient les données d'enregistrements et les paramètres de service requis aux services externes. Vous pouvez configurer des appels vers les services Web externes pour certaines étapes de la logique d'exécution. En fonction de la logique que vous implémentez, les demandes sont acheminées à partir de Data Director pour mettre à jour les données d'enregistrements. Les services externes peuvent modifier les données en cas de besoin.

Dans l'outil d'approvisionnement, configurez l'entité d'entreprise et les événements pour lesquels vous voulez appeler le service externe. Dans l'outil d'approvisionnement, chargez le fichier WSDL du service externe et enregistrez le service SOAP et l'opération. Liez le service à des entités d'entreprise et à des événements spécifiques.

Utilisez le fichier WSDL du kit de ressources pour comprendre le service, les opérations, les méthodes et les types de données échangés par les méthodes du service. Le fichier `custom-logic-service.wsdl` des services Web externes se trouve à l'emplacement du kit de ressources suivant : `C:\<répertoire d'installation d'infamdm>\hub\resourcekit\samples\BESEExternalCall\source\resources\webapp\WEB-INF\wsdl\`

Le kit de ressources comprend un exemple de code qui implémente la logique personnalisée et la validation. Lorsque vous installez le kit de ressources, le fichier `bes-external-call.ear` de l'exemple de logique personnalisée et de validation est déployé sur le serveur d'applications.

Événements pris en charge

Un service d'entité d'entreprise comprend des étapes de service. Vous pouvez ajouter une logique personnalisée et une validation à n'importe laquelle de ces étapes.

Vous pouvez effectuer des appels externes pour les événements suivants :

- `WriteCO.BeforeEverything`
- `WriteCO.BeforeValidate`
- `WriteCO.AfterValidate`
- `WriteCO.AfterEverything`
- `WriteView.BeforeEverything`
- `WriteView.BeforeValidate`
- `WriteView.AfterValidate`
- `WriteView.AfterEverything`
- `MergeCO.BeforeEverything`
- `MergeCO.AfterEverything`
- `PreviewMergeCO.BeforeEverything`
- `PreviewMergeCO.AfterEverything`
- `ReadCO.BeforeEverything`
- `ReadCO.AfterEverything`
- `ReadView.BeforeEverything`
- `ReadView.AfterEverythingEvents`

Configuration des appels externes

Un service d'entité d'entreprise comprend des étapes de service. Les demandes entrantes suivent chaque étape de service. Vous pouvez configurer les appels vers les services externes pour certaines étapes de la logique d'exécution du service d'entité d'entreprise.

Effectuez les étapes suivantes pour configurer les appels externes :

1. Générez et déployez le fichier `bes-external-call.ear`.
2. Dans l'outil d'approvisionnement, effectuez les tâches suivantes :
 - a. Chargez le fichier WSDL du service externe.
 - b. Enregistrez le service Web en tant que service SOAP.
 - c. Configurez un appel externe.

Pour plus d'informations sur le chargement du fichier WSDL, l'enregistrement du service SOAP et la configuration des appels externes, consultez le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

Pour plus d'informations sur la génération et le déploiement du fichier EAR, consultez le *Guide du kit de ressources de MDM Multidomain*.

Exemple : validation et logique personnalisées des services d'entité d'entreprise

Vous pouvez tester la validation et la logique personnalisées lorsque vous ajoutez et fusionnez les enregistrements Personne. La validation personnalisée vérifie si l'enregistrement Personne dispose d'une adresse. La logique personnalisée ne vous autorise pas à fusionner deux numéros de téléphone. Utilisez les API REST pour créer et fusionner les enregistrements Personne.

1. Pour vérifier la validation lorsque vous créez un enregistrement Personne, procédez comme suit :
 - a. Utilisez l'API Créer pour créer un enregistrement Personne sans adresse. Vous obtenez une erreur de validation.
 - b. Utilisez l'API Créer pour créer un enregistrement Personne avec une adresse. L'opération a réussi.
2. Pour vérifier la logique personnalisée lorsque vous fusionnez des enregistrements, procédez comme suit :
 - a. Utilisez l'API Créer pour créer deux enregistrements Personne contenant une adresse et un numéro de téléphone chacun.
 - b. Utilisez l'API Aperçu de fusion pour fusionner les deux enregistrements Personne. Ajoutez `overrides` à la demande d'aperçu de fusion pour fusionner les adresses et les numéros de téléphone. La réponse montre une seule adresse, mais deux numéros de téléphone. La logique personnalisée empêche la fusion des numéros de téléphone.

Prérequis

Pour vérifier la logique personnalisée et la validation, vous devez déployer le fichier WSDL dans l'outil d'approvisionnement. Vous devez enregistrer le service SOAP et l'opération. Liez le service aux entités d'entreprise et aux événements pour lesquels vous voulez utiliser la logique personnalisée et la validation. Vous pouvez tester la logique et la validation des entités d'entreprise et des événements spécifiés.

Étape 1. Tester la validation personnalisée

Utilisez l'API Créer pour créer l'enregistrement Personne suivant sans adresse :

```
POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_Sample/Person?systemName=Admin
{
    firstName: "John"
}
```

Vous obtenez une erreur de validation.

Utilisez l'API Créer pour créer l'enregistrement Personne suivant avec une adresse :

```
POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_Sample/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
```

```

        item: [
            {
                cityName: "Toronto"
            }
        ]
    }
}

```

La demande crée un enregistrement Personne.

Étape 2. Tester la logique personnalisée

Effectuez les étapes suivantes pour tester la logique personnalisée :

1. Utilisez l'API Créer pour créer deux enregistrements Personne contenant une adresse et un numéro de téléphone chacun :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-mdm_sample/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
        item: [
            {
                cityName: "Toronto"
            }
        ]
    },
    TelephoneNumbers: {
        item: [
            {
                phoneNum: "111-11-11"
            }
        ]
    }
}

POST http://localhost:8080/cmx/cs/localhost-orcl-mdm_sample/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
        item: [
            {
                cityName: "Ottawa"
            }
        ]
    },
    TelephoneNumbers: {
        item: [
            {
                phoneNum: "222-22-22"
            }
        ]
    }
}

```

La réponse contient les rowids suivants :

- Personne : 161923, 161924
- Adresses : 2123, 2124
- Numéros de téléphone : 101723, 101724

2. Exécutez l'API PreviewMerge pour fusionner des deux enregistrements Personne :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-mdm_sample/Person/161923?
action=previewMerge&depth=2
{
    keys: [
        {

```



```

        rowid: "161924"
    }
}
}

```

La réponse est un enregistrement Personne contenant deux adresses et deux numéros de téléphone.

```

{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person: , Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 2,
      "pageSize": 2,
      "item": [
        {
          "rowidObject": "101723",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:35.904-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "111-1111",
          "phoneCountryCd": "1"
        },
        {
          "rowidObject": "101724",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:54.768-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "222-2222",
          "phoneCountryCd": "1"
        }
      ]
    }
  },
  "Addresses": {
    "link": [],
    "firstRecord": 1,
    "recordCount": 2,
    "pageSize": 2,
    "item": [
      {
        "rowidObject": "2123",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:37.956-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Addresses",

```

```

      "Address": {
        "rowidObject": "2121",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:36.922-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:37.923-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Address",
        "cityName": "Toronto"
      }
    },
    {
      "rowidObject": "2124",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:54.790-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:54.790-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "Addresses",
      "Address": {
        "rowidObject": "2122",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:54.777-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:54.777-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Address",
        "cityName": "Ottawa"
      }
    }
  ]
}

```

3. Exécutez l'API PreviewMerge pour fusionner les deux enregistrements Personne avec le paramètre « overrides » pour fusionner les adresses et les numéros de téléphone :

```

POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person/161923?
action=previewMerge&depth=3
{
  keys: [
    { rowid: "161923" }
  ],
  overrides: {
    Person: {
      Addresses: {
        item:[
          {
            rowidObject: "2123",
            MERGE: {
              item:[{key:{rowid: "2124"}}],
              $original: {
                item:[null]
              }
            }
          }
        ]
      },
      TelephoneNumbers: {
        item:[
          {
            rowidObject: "101723",
            MERGE: {
              item:[{key:{rowid: "101724"}}],

```

```
{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person: Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 2,
      "pageSize": 2,
      "item": [
        {
          "rowidObject": "101723",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:35.904-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "111-1111",
          "phoneCountryCd": "1"
        },
        {
          "rowidObject": "101724",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:54.768-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "222-2222",
          "phoneCountryCd": "1"
        }
      ]
    }
  },
  "Addresses": {
    "link": [],
    "firstRecord": 1,
    "recordCount": 1,
    "pageSize": 1,
    "item": [
      {
        "rowidObject": "2123",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:37.956-04:00",
```

```

        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Addresses"
    }
}
],
},
"PersonDetails": {
    "link": [],
    "recordCount": 0,
    "pageSize": 0,
    "item": []
}
}
}

```

ANNEXE A

Utilisation d'API REST pour ajouter des enregistrements

Cette annexe comprend les rubriques suivantes :

- [Présentation de l'utilisation d'API REST pour ajouter des enregistrements, 285](#)
- [Structure de l'entité commerciale Person, 286](#)
- [Étape 1. Obtenir des informations sur le schéma, 286](#)
- [Étape 2. Création d'un enregistrement, 292](#)
- [Étape 3. Lire l'enregistrement, 294](#)

Présentation de l'utilisation d'API REST pour ajouter des enregistrements

Après avoir créé un modèle d'entité d'entreprise et configuré la structure de l'entité d'entreprise, vous pouvez utiliser les API REST pour ajouter les enregistrements.

Les sections suivantes utilisent l'exemple de l'entité d'entreprise Person pour vous montrer comment ajouter des enregistrements à l'aide d'API REST. Cette entité contient des données concernant les employés de votre société.

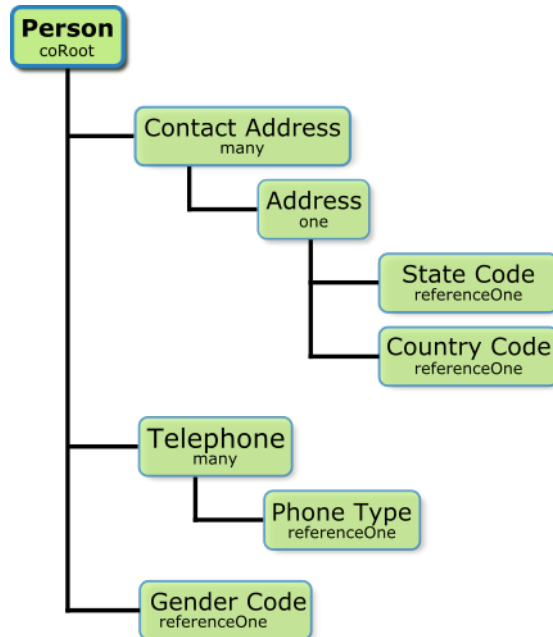
Utilisez les API suivantes pour ajouter les détails de vos employés :

1. Obtenir des informations sur le schéma. Utilisez l'API REST Obtenir les métadonnées pour obtenir des informations sur la structure de données d'une entité d'entreprise, y compris la structure, la liste de champs, les types de champs et les détails des champs de recherche. Vous pouvez aussi accéder aux fichiers XSD (XML Schema Definition) qui décrivent les éléments et attributs que vous pouvez utiliser. Les fichiers XSD sont enregistrés sous l'emplacement `http://<host>:<port>/cmx/csfiles`.
2. Créez un enregistrement. Utilisez l'API REST Créer un enregistrement pour créer l'enregistrement.
3. Lisez les données de l'enregistrement que vous avez ajouté. Utilisez l'API REST Lire l'enregistrement pour récupérer les données de l'enregistrement.

Structure de l'entité commerciale Person

Nous allons ajouter un enregistrement Person à l'aide d'API REST. Le nœud racine Person est le nœud supérieur de la structure de l'entité commerciale Person. Les nœuds inférieurs comportent les détails de l'employé comme le sexe, l'adresse et le numéro de téléphone.

L'image suivante montre la structure de l'entité commerciale Person :



Person est le nœud racine de l'entité commerciale Person. Le type de nœud, affiché sous le nom du nœud, indique la relation entre le nœud parent et le nœud enfant. Il existe une relation de type un-à-un entre Contact Address et Address, qui indique que chaque adresse de contact ne peut être associée qu'à une seule adresse. Il existe une relation de type un-à-plusieurs entre Person et Telephone, qui indique qu'un enregistrement Person peut être associé à plusieurs enregistrements de numéros de téléphone. Il existe une relation de type un-à-un entre Person et Gender, qui indique qu'un enregistrement Person ne peut comporter qu'une seule valeur de sexe. Les valeurs de sexe figurent dans une table de recherche. De même, les valeurs de code d'État et de pays figurent dans des tables de recherche.

Étape 1. Obtenir des informations sur le schéma

Utilisez l'API REST Obtenir les métadonnées pour obtenir des informations sur un schéma. L'API Obtenir les métadonnées renvoie la structure de données d'une entité commerciale. Les métadonnées répertorient les champs d'entité commerciale, les types de champs et les détails des champs de recherche.

L'URL Obtenir les métadonnées est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

L'exemple de demande suivant récupère les informations de métadonnées pour l'entité commerciale Person :

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

Obtenir la réponse des métadonnées

L'exemple suivant montre des extraits de la structure de données de l'entité commerciale Person :

```
{
  "object": {
    "field": [
      {
        "allowedValues": [
          "Person"
        ],
        "name": "partyType",
        "label": "Party Type",
        "dataType": "String",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "imageUrl",
        "label": "Image URL",
        "dataType": "ImageURL",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "statusCd",
        "label": "Status Cd",
        "dataType": "String",
        "length": 2,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "displayName",
        "label": "Display Name",
        "dataType": "String",
        "length": 200,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "birthdate",
        "label": "Birthdate",
        "dataType": "Date",
        "length": 0,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "firstName",
        "label": "First Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
```

```

        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "lastName",
        "label": "Last Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "middleName",
        "label": "Middle Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "dirtyIndicator",
        "label": "Dirty Indicator",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "hubStateInd",
        "label": "Hub State Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "lastRowidSystem",
        "label": "Last Rowid System",
        "dataType": "String",
        "length": 14,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },

```



```

-----
    {
      "name": "genderCd",
      "label": "Gender Cd",
      "dataType": "lookup",
      "readOnly": false,
      "required": false,
      "system": false,
      "lookup": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_uil/LUGender?action=list&idlabel=genderCode%3AgenderDisp",
            "rel": "lookup"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_uil/LUGender?action=list",
            "rel": "list"
          }
        ],
        "object": "LUGender",
        "key": "genderCode",
        "value": "genderDisp"
      }
    }
  ],

```

```

-----
    "child": [
      {
        "field": [
          {
            "name": "cityName",
            "label": "City Name",
            "dataType": "String",
            "length": 100,
            "totalDigits": 0,
            "fractionDigits": 0,
            "readOnly": false,
            "required": false,
            "system": false
          },
          {
            "name": "addressLine2",
            "label": "Address Line2",
            "dataType": "String",
            "length": 100,
            "totalDigits": 0,
            "fractionDigits": 0,
            "readOnly": false,
            "required": false,
            "system": false
          },
          {
            "name": "addressLine1",
            "label": "Address Line1",
            "dataType": "String",
            "length": 100,
            "totalDigits": 0,
            "fractionDigits": 0,
            "readOnly": false,
            "required": false,
            "system": false
          },
          {
            "name": "isValidInd",
            "label": "Is Valid Ind",

```

```

        "dataType": "String",
        "length": 1,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "postalCd",
        "label": "Postal Cd",
        "dataType": "String",
        "length": 10,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    }
},

-----

{
    "name": "countryCode",
    "label": "Country Code",
    "dataType": "lookup",
    "readOnly": false,
    "required": false,
    "system": false,
    "dependents": [
        "Person.Address.Address.stateCd"
    ],
    "lookup": {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_uil/LUCountry?action=list",
                "rel": "list"
            },
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_uil/LUCountry?action=list&idlabel=countryCode%3AcountryNameDisp",
                "rel": "lookup"
            }
        ],
        "object": "LUCountry",
        "key": "countryCode",
        "value": "countryNameDisp"
    }
},
{
    "name": "stateCd",
    "label": "State Cd",
    "dataType": "lookup",
    "readOnly": false,
    "required": false,
    "system": false,
    "parents": [
        "Person.Address.Address.countryCode"
    ],
    "lookup": {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_uil/LUCountry/{Person.Address.Address.countryCode}/LUState?action=list",
                "rel": "list"
            },
            {
                "href": "http://localhost:8080/cmx/cs/

```

```
localhost-hub101-ds_ui1/LUCountry/{Person.Address.Address.countryCode}/LUState?
action=list&idlabel=stateAbbreviation%3AstateNameDisp",
    "rel": "lookup"
  }
},
    ],
    "object": "LUCountry.LUState",
    "key": "stateAbbreviation",
    "value": "stateNameDisp"
  }
}
},
    ],
    "name": "Address",
    "label": "Address",
    "many": false
  }
},
    ],
    "name": "Address",
    "label": "Contact Address",
    "many": true
  },
  {
    "field": [
      {
        "name": "phoneNum",
        "label": "Phone Number",
        "dataType": "String",
        "length": 13,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },

```

```

      {
        "name": "phoneTypeCd",
        "label": "Phone Type",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "lookup": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/cs/localhost-
hub101-ds_ui1/LUPhoneType?action=list&idlabel=phoneType%3AphoneTypeDisp",
              "rel": "lookup"
            },
            {
              "href": "http://localhost:8080/cmx/cs/localhost-
hub101-ds_ui1/LUPhoneType?action=list",
              "rel": "list"
            }
          ],
          "object": "LUPhoneType",
          "key": "phoneType",
          "value": "phoneTypeDisp"
        }
      }
    ],
    "name": "Telephone",
    "label": "Telephone",
    "many": true
  }
},
    ],
    "name": "Person",
    "label": "Person",
    "many": false
  }
}

```

```
}  
}
```

Étape 2. Création d'un enregistrement

Utilisez l'API REST Créer un enregistrement pour créer l'enregistrement. Le nom de l'entité d'entreprise et le nom du système source sont des paramètres requis. Envoyez les données pour l'enregistrement dans le corps de demande.

L'URL Créer un enregistrement est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the  
source system>
```

Le paramètre `systemName` est un paramètre requis et spécifie le nom du système source.

L'entité d'entreprise `Person` contient le nœud racine `Person` et les nœuds d'adresse, de sexe et de numéro de téléphone de deuxième niveau.

L'exemple de demande suivant crée un enregistrement `Personne` :

```
POST http://localhost:8080/cmx/cs/localhost-hub101-ds_uil/Person?systemName=Admin  
{  
  "firstName": "Boris",  
  "lastName": "Isaac",  
  "genderCd": {  
    "genderCode": "M"  
  },  
  "Address": {  
    "item": [  
      {  
        "Address": {  
          "addressLine1": "B-203, 101 Avenue, New York",  
          "stateCd": {  
            "stateAbbreviation": "NY"  
          },  
          "countryCode": {  
            "countryCode": "US"  
          }  
        }  
      }  
    ]  
  },  
  "Telephone": {  
    "item": [  
      {  
        "phoneNum": "1234567",  
        "phoneTypeCd": {  
          "phoneType": "HOM"  
        }  
      },  
      {  
        "phoneNum": "7654321",  
        "phoneTypeCd": {  
          "phoneType": "MOB"  
        }  
      }  
    ]  
  }  
}
```

Le corps de demande spécifie les détails suivants de l'enregistrement `Personne` :

- Prénom.

- Nom.
- Sexe.
- Adresse avec le code d'État et le code de pays.
- Numéros de téléphone et type de téléphone, par exemple téléphone du domicile et téléphone portable.

Créer une réponse de l'enregistrement

L'exemple suivant affiche la réponse qui s'affiche après la création réussie d'un enregistrement Personne :

```
{
  "Person": {
    "key": {
      "rowid": "658248",
      "sourceKey": "66240000025000"
    },
    "rowidObject": "658248",
    "genderCd": {
      "key": {
        "rowid": "2"
      },
      "rowidObject": "2"
    },
    "Address": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "101526",
            "sourceKey": "66240000028000"
          },
          "rowidObject": "101526",
          "Address": {
            "key": {
              "rowid": "121506",
              "sourceKey": "66240000027000"
            },
            "rowidObject": "121506",
            "countryCode": {
              "key": {
                "rowid": "233"
              },
              "rowidObject": "233"
            },
            "stateCd": {
              "key": {
                "rowid": "52"
              },
              "rowidObject": "52"
            }
          }
        }
      ]
    },
    "Telephone": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "20967",
            "sourceKey": "66240000029000"
          },
          "rowidObject": "20967",
          "phoneTypeCd": {
            "key": {
              "rowid": "8"
            }
          }
        }
      ]
    }
  }
}
```

```

    },
    "rowidObject": "8"
  },
  {
    "key": {
      "rowid": "20968",
      "sourceKey": "66240000030000"
    },
    "rowidObject": "20968",
    "phoneTypeCd": {
      "key": {
        "rowid": "6"
      },
      "rowidObject": "6"
    }
  }
]
}
}

```

Remarque: Le corps de réponse contient l'enregistrement avec les ID de ligne générés.

Si vous configurez un processus de flux de travail afin qu'il démarre lors de la création d'un enregistrement, voici ce qu'il se passe :

- L'enregistrement est créé dans l'état En attente.
- Le processus de flux de travail est démarré.
- L'ID du processus de flux de travail est renvoyé dans l'en-tête de réponse.

Si vous ne configurez pas de processus de flux de travail, l'enregistrement est créé dans l'état Actif, par défaut.

L'API renvoie un ID d'interaction dans l'en-tête de réponse si vous traitez la demande en utilisant un ID d'interaction.

Étape 3. Lire l'enregistrement

Utilisez l'API REST Lire l'enregistrement pour récupérer les détails d'un enregistrement racine que vous avez ajouté. Vous pouvez utiliser l'API pour récupérer les détails des enregistrements enfants d'un enregistrement racine.

L'URL Lire l'enregistrement est au format suivant :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

Utilisez le paramètre `depth` pour spécifier le nombre de niveaux enfants à renvoyer. Indiquez 2 pour renvoyer le nœud racine et ses enfants directs et 3 pour renvoyer le nœud racine, ses enfants directs et ses petits-enfants. Utilisez l'URL suivante pour renvoyer les détails des enregistrements enfants :

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>?depth=n
```

L'exemple de demande suivant renvoie les détails du nœud racine, des enfants directs et des petits-enfants.

```
GET http://localhost:8080/cmx/cs/localhost-hub101-ds_uil/Person/658248?depth=3
```

La demande renvoie les détails d'un enregistrement actif.

Remarque: Si un flux de travail est lancé lorsque vous créez un enregistrement, l'enregistrement créé est en attente. Par défaut, la demande Lire l'enregistrement lit les enregistrements actifs. Utilisez le paramètre `recordStates` pour spécifier l'état en attente de l'enregistrement.

L'exemple de demande suivant lit les détails d'un enregistrement en attente :

```
GET http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=3&recordStates=PENDING
```

Lire la réponse de l'enregistrement

L'exemple de réponse suivant montre les détails de l'enregistrement que vous avez ajouté :

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "658248",
  "label": "Person",
  "partyType": "Person",
  "displayName": "BORIS ISAAC",
  "firstName": "BORIS",
  "lastName": "ISAAC",
  "genderCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2?depth=2",
        "rel": "children"
      }
    ]
  },
  "rowidObject": "2",
  "label": "LU Gender",
  "genderCode": "M",
  "genderDisp": "MALE"
},
  "Address": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/Address",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
        "rel": "parent"
      }
    ]
  },
  "firstRecord": 1,
  "pageSize": 10,
```

```

        "searchToken": "SVR1.PCWJ",
        "item": [
            {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address",
                        "rel": "parent"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526?depth=2",
                        "rel": "children"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526",
                        "rel": "self"
                    }
                ],
                "rowidObject": "101526",
                "label": "Contact Address",
                "Address": {
                    "link": [
                        {
                            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506?depth=2",
                            "rel": "children"
                        },
                        {
                            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526",
                            "rel": "parent"
                        },
                        {
                            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506",
                            "rel": "self"
                        }
                    ],
                    "rowidObject": "121506",
                    "label": "Address",
                    "addressLine1": "B-203, 101 Avenue, New York",
                    "countryCode": {
                        "link": [
                            {
                                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506/countryCode",
                                "rel": "self"
                            },
                            {
                                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506",
                                "rel": "parent"
                            },
                            {
                                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506/countryCode?depth=2",
                                "rel": "children"
                            }
                        ],
                        "countryCode": "US"
                    },
                    "stateCd": {
                        "link": [
                            {
                                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Address/101526/Address/121506",
                                "rel": "parent"
                            },
                            {

```



```

        "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd?depth=2",
        "rel": "children"
    },
    {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd",
        "rel": "self"
    }
],
"stateAbbreviation": "NY"
}
}
]
},
"Telephone": {
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
            "rel": "parent"
        },
        {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/Telephone",
            "rel": "self"
        }
    ],
    "firstRecord": 1,
    "pageSize": 10,
    "searchToken": "SVR1.PCWK",
    "item": [
        {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone",
                    "rel": "parent"
                },
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967?depth=2",
                    "rel": "children"
                }
            ],
            "rowidObject": "20967",
            "label": "Telephone",
            "phoneNum": "1234567",
            "phoneTypeCd": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967",
                        "rel": "parent"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8",
                        "rel": "self"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8?depth=2",
                        "rel": "children"
                    }
                ]
            }
        }
    ]
}

```

```

        ],
        "rowidObject": "8",
        "label": "LU Phone Type",
        "phoneTypeDisp": "HOME",
        "phoneType": "HOM"
    }
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
            "rel": "self"
        },
        {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968?depth=2",
            "rel": "children"
        },
        {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone",
            "rel": "parent"
        }
    ],
    "rowidObject": "20968",
    "label": "Telephone",
    "phoneNum": "7654321",
    "phoneTypeCd": {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6",
                "rel": "self"
            },
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
                "rel": "parent"
            },
            {
                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6?depth=2",
                "rel": "children"
            }
        ]
    },
    "rowidObject": "6",
    "label": "LU Phone Type",
    "phoneTypeDisp": "MOBILE",
    "phoneType": "MOB"
}
    ]
}
}

```

ANNEXE B

Utilisation d'API REST pour le chargement de fichiers

Cette annexe comprend les rubriques suivantes :

- [Présentation de l'utilisation d'API REST pour le chargement de fichiers, 299](#)
- [API REST pour les fichiers, 300](#)
- [Composants de fichier, 300](#)
- [Types de stockage, 301](#)
- [Joindre des fichiers à des enregistrements, 301](#)
- [Joindre des fichiers à des tâches, 303](#)
- [Chargement de fichiers de groupement de ressources, 306](#)

Présentation de l'utilisation d'API REST pour le chargement de fichiers

Les API REST vous permettent de charger des fichiers vers un type de stockage. Après avoir chargé un fichier, vous pouvez l'ajouter en pièce jointe à un enregistrement ou une tâche, ou l'utiliser pour localiser l'interface utilisateur de Data Director.

Selon la façon dont vous souhaitez exploiter les fichiers, la combinaison des API REST, les composants de fichier et le type de stockage que vous utilisez peuvent varier. Par exemple, pour joindre des fichiers à des enregistrements ou des tâches, vous pouvez créer les métadonnées du fichier et charger ce dernier vers un stockage temporaire. Après avoir chargé le fichier, vous pouvez l'ajouter en pièce jointe à un enregistrement dans une base de données ou à une tâche dans le stockage BPM. Pour localiser l'interface utilisateur de Data Director, vous téléchargez le fichier ZIP, modifiez les fichiers compressés, puis chargez le groupement ZIP modifié vers le stockage du groupement.

API REST pour les fichiers

Vous pouvez exploiter un ensemble d'API REST génériques pour charger et gérer des fichiers à ajouter en pièces jointes ou à utiliser pour la localisation.

Le tableau suivant répertorie les API REST pour les fichiers :

API REST	Description	Type de stockage pris en charge
Lister les métadonnées des fichiers	Renvoie la liste des métadonnées de fichier dans un stockage.	BPM ou TEMP
Créer les métadonnées du fichier	Crée les métadonnées d'un fichier dans un stockage.	DB ou TEMP
Obtenir les métadonnées du fichier	Renvoie les métadonnées d'un fichier.	BPM, BUNDLE, DB ou TEMP
Mettre à jour les métadonnées du fichier	Met à jour les métadonnées d'un fichier.	DB ou TEMP
Charger le contenu du fichier	Charge le contenu d'un fichier vers un stockage.	BUNDLE, DB ou TEMP
Obtenir le contenu du fichier	Télécharge le contenu d'un fichier.	BPM, BUNDLE, DB ou TEMP
Supprimer un fichier	Supprime un fichier dans un stockage, y compris les composants associés tels que les métadonnées et le contenu.	BUNDLE, DB ou TEMP

Composants de fichier

Pour joindre des fichiers à des enregistrements ou des tâches, créez les métadonnées du fichier, puis chargez le contenu du fichier. Pour localiser l'interface utilisateur de Data Director, téléchargez le fichier de groupement de ressources, puis chargez le fichier de groupement de ressources modifié.

Métadonnées du fichier

Informations sur le fichier telles que le nom de fichier, le type de fichier et le type de contenu. Selon le type de stockage utilisé, vous devrez peut-être inclure d'autres paramètres tels que le créateur, l'heure de création et la date de chargement.

Contenu du fichier

Contenu du fichier. Par exemple, le texte, l'image, le document ou le groupement de ressources.

Types de stockage

Chargez et stockez des fichiers dans une implémentation de stockage prise en charge. Le type de stockage utilisé n'est pas le même selon que vous souhaitez localiser l'interface utilisateur de Data Director ou joindre des fichiers à des enregistrements ou des tâches.

La liste suivante décrit les types de stockage pris en charge :

BPM

Stocke les fichiers joints aux tâches avec les données de tâche. Lorsque vous joignez un fichier à une tâche, le processus stocke le fichier dans un stockage BPM à partir du stockage TEMP.

Les fichiers conservés dans le stockage BPM utilisent le format d'ID de fichier suivant :

`taskId::filename.`

Remarque: Pour joindre un fichier à des tâches déclenchées ou des tâches existantes, dans l'outil Provisionnement, activez les pièces jointes pour les déclencheurs de tâche, les types de tâche et les actions de tâche. Pour plus d'informations, consultez le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

BUNDLE

Stocke les fichiers de groupement de ressources qui localisent l'interface utilisateur de Data Director.

Les fichiers conservés dans le stockage BUNDLE utilisent le format d'ID de fichier suivant : `besMetadata.`

DB

Stocke les pièces jointes de fichier des enregistrements dans la table C_REPOS_ATTACHMENTS. Lorsque vous joignez un fichier à un enregistrement, le processus stocke le fichier dans un stockage DB à partir du stockage TEMP.

Les fichiers conservés dans le stockage DB utilisent le format d'ID de fichier suivant : `DB_<RowID>.`

Remarque: Pour joindre un fichier à un enregistrement, dans l'outil Provisionnement, configurez un champ avec le type de données `FileAttachment`. Pour plus d'informations sur la configuration du type de données, consultez le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

TEMP

Stocke les fichiers de manière temporaire dans la table C_REPOS_ATTACHMENTS et leur appose l'indicateur TEMP. Les fichiers sont supprimés du stockage TEMP une fois qu'ils ont été chargés avec succès vers le stockage BPM ou DB ou après le délai d'expiration pré-configuré.

Les fichiers conservés dans le stockage TEMP utilisent le format d'ID de fichier suivant :

`TEMP_<ROWID_ATTACHMENT>.`

Pour plus d'informations sur la configuration du délai d'expiration, consultez le *Guide de configuration de MDM Multidomain*.

Joindre des fichiers à des enregistrements

Pour joindre un fichier à un enregistrement, vous devez d'abord créer les métadonnées du fichier, puis charger le fichier vers le stockage temporaire.

1. Pour créer les métadonnées d'un fichier, utilisez l'API REST Créer les métadonnées du fichier avec le type de stockage TEMP.

Par exemple, la demande suivante crée les métadonnées du fichier `Document_3.pdf` :

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
Content-Type: application/json
{
  "fileName": "Document_3.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Remarque: Créez toujours les métadonnées du fichier dans le stockage TEMP.

L'API REST Créer les métadonnées du fichier renvoie un ID pour le fichier. L'ID de fichier est au format suivant : `<Storage Type>_<RowID>`. RowID désigne l'ID de ligne du fichier que vous chargez vers le stockage.

Dans l'exemple, l'appel de l'API renvoie l'ID suivant pour le fichier `Document_3.pdf` : `TEMP_SVR1.OJU3`

Vous pouvez utiliser l'ID de fichier pour charger, joindre, mettre à jour, télécharger et supprimer le fichier.

2. Pour charger le fichier, utilisez l'API REST Charger le contenu du fichier avec le type de stockage TEMP. Par exemple, la demande suivante charge le fichier vers le stockage TEMP :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.OJU3/
content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

Remarque: Une fois le fichier chargé, il est conservé dans le stockage TEMP pendant une période pré-configurée de 60 minutes. Vous devez joindre le fichier à un enregistrement avant l'expiration de la période pré-configurée.

3. Pour créer un enregistrement et joindre le fichier au nouvel enregistrement, utilisez l'API REST Créer un enregistrement.

Par exemple, la demande suivante crée un enregistrement et joint le fichier avec l'ID de fichier,

`TEMP_SVR1.OJU3` :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person?systemName=Admin
Content-Type: application/json
{
  "frstNm": "John",
  "lstNm": "Smith",
  "addrLn1": "2100 Breverly Road",
  "addrTyp": {
    "addrTyp": "Billing",
    "addrTypDesc": "Billing"
  },
  "cntryCd": {
    "cntryCd": "AX",
    "cntryDesc": "Aland"
  },
  "attachments": {
    "item": [
      {
        "fileId": "TEMP_SVR1.OJU3"
      }
    ]
  }
}
```

Remarque: Lorsque vous joignez un fichier à un enregistrement, le processus stocke le fichier dans la base de données. L'ID du fichier est remplacé par `DB_<RowID>`, DB indiquant que le fichier est stocké dans la base de données.

4. Pour remplacer un fichier joint à un enregistrement, utilisez l'API REST Charger le contenu du fichier avec le type de stockage DB.

Par exemple, la demande suivante remplace le fichier joint par l'ID de fichier, DB_SVR1.OJU3, dans la base de données :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

Remarque: Le type de stockage indiqué dans l'URL de la demande est DB.

5. Pour modifier les métadonnées d'un fichier après avoir joint le fichier à un enregistrement, utilisez l'API REST Mettre à jour les métadonnées du fichier avec le type de stockage DB.

Par exemple, la demande suivante met à jour les métadonnées du fichier associé à l'ID de fichier, DB_SVR1.OJU3, dans le stockage DB :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3
Content-Type: application/json
{
  "fileName": "Document_4.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

6. Pour télécharger un fichier joint à un enregistrement, utilisez l'API REST Obtenir le contenu du fichier avec le type de stockage DB.

Par exemple, la demande suivante télécharge un fichier associé à l'ID de fichier, DB_SVR1.OJU3, à partir du stockage DB :

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content
```

7. Pour supprimer un fichier joint à un enregistrement, utilisez l'API REST Supprimer un fichier avec le type de stockage DB.

Par exemple, la demande suivante supprime un fichier associé à l'ID de fichier, DB_SVR1.OJU3, à partir du stockage DB :

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3
```

Joindre des fichiers à des tâches

Créez les métadonnées d'un fichier, puis chargez le contenu du fichier vers un stockage temporaire. Après avoir chargé le fichier, joignez-le à une tâche déclenchée ou à une tâche existante.

Remarque: Pour joindre un fichier à des tâches déclenchées ou des tâches existantes, dans l'outil Provisionnement, activez les pièces jointes pour les déclencheurs de tâche, les types de tâche et les actions de tâche. Pour plus d'informations, consultez le *Guide de l'outil d'approvisionnement de MDM Multidomain*.

1. Pour créer les métadonnées d'un fichier, utilisez l'API REST Créer les métadonnées du fichier avec le type de stockage TEMP.

Par exemple, la demande suivante crée les métadonnées du fichier file1.txt :

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

Remarque: Créez toujours les métadonnées du fichier dans le stockage TEMP.

L'API REST Créer les métadonnées du fichier renvoie un ID pour le fichier. L'ID de fichier est au format suivant : <Storage Type>_<RowID>. RowID désigne l'ID de ligne du fichier que vous chargez vers le stockage.

Dans l'exemple, l'appel de l'API renvoie l'ID suivant pour `file1.txt` : `TEMP_SVR1.1VDVS`

Vous pouvez utiliser l'ID de fichier pour charger, joindre, mettre à jour et supprimer le fichier.

2. Pour charger le fichier, utilisez l'API REST Charger le contenu du fichier avec le type de stockage TEMP. Par exemple, la demande suivante charge le fichier vers le stockage TEMP :

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/
content
```

```
Test attachment content: file 1
```

Remarque: Une fois le fichier chargé, il est conservé dans le stockage TEMP pendant une période pré-configurée de 60 minutes. Vous devez joindre le fichier à une tâche avant l'expiration de la période pré-configurée.

3. Joignez le fichier à la tâche qui est déclenchée lorsque vous gérez les enregistrements.
 - Pour joindre le fichier à la tâche qui est déclenchée lorsque vous créez un enregistrement, utilisez l'API REST Créer une entité d'entreprise avec le paramètre `taskattachments`.

Par exemple, la demande suivante crée un enregistrement et joint le fichier avec l'ID de fichier

`TEMP_SVR1.1VDVS` :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/json
```

```
{
  firstName: "John",
  lastName: "Smith",
  Phone: {
    item: [
      {
        phoneNumber: "111-11-11"
      }
    ]
  }
}
```

- Pour joindre le fichier à la tâche qui est déclenchée lorsque vous mettez à jour un enregistrement, utilisez l'API REST Mettre à jour l'entité d'entreprise avec le paramètre `taskattachments`.

Par exemple, la demande suivante met à jour un enregistrement et joint le fichier avec l'ID de fichier

`TEMP_SVR1.1VDVS` :

```
PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
```

```
{
  rowidObject: "233",
  firstName: "BOB",
  lastName: "LLOYD",
  Phone: {
    item: [
      {
        rowidObject: "164",
        phoneNumber: "777-77-77",
        $original: {
          phoneNumber: "(336) 366-4936"
        }
      }
    ]
  },
  $original: {
    firstName: "DUNN"
  }
}
```

- Pour joindre le fichier à la tâche qui est déclenchée lorsque vous fusionnez un enregistrement, utilisez l'API REST Fusionner l'entité d'entreprise avec le paramètre `taskattachments`.

Par exemple, la demande suivante fusionne un enregistrement et joint le fichier avec l'ID de fichier

TEMP_SVR1.1VDVS :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/json+xml
{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}
```

- Pour joindre le fichier à la tâche qui est déclenchée lorsque vous annulez la fusion d'un enregistrement, utilisez l'API REST Annuler la fusion de l'entité d'entreprise avec le paramètre `taskAttachments`.

Par exemple, la demande suivante annule la fusion d'un enregistrement et joint le fichier avec l'ID de fichier TEMP_SVR1.1VDVS :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskAttachments=TEMP_SVR1.1VDVS
{
  rowid: "4880369"
}
```

4. Joignez le fichier à une tâche existante.

- Pour joindre le fichier lorsque vous mettez à jour une tâche, utilisez l'API REST Mettre à jour la tâche avec le paramètre `attachments` dans le corps de demande.

Par exemple, la demande suivante met à jour une tâche et joint le fichier avec l'ID de fichier

TEMP_SVR1.1VDVS :

```
PUT http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934
{
  taskType: {
    name: "UpdateWithApprovalWorkflow"
  },
  taskId: "urn:b4p2:15934",
  owner: "John",
  title: "Smoke test task - updated",
  comments: "Smoke testing - updated",
  "attachments": [
    {
      "id": "TEMP_SVR1.1VDVS"
    }
  ],
  ...
}
```

- Pour joindre le fichier lorsque vous exécutez une action de tâche, utilisez l'API REST Exécuter une action de tâche avec le paramètre `attachments` dans le corps de demande.

Par exemple, la demande suivante exécute une action de tâche et joint le fichier avec l'ID de fichier

TEMP_SVR1.1VDVS :

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934?
taskAction=Cancel
{
  taskType: {
    name: "UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
}
```

```

        taskId: "urn:b4p2:15934",
        owner: "manager",
        title: "Smoke test task 222",
        comments: "Smoke testing",
        "attachments": [
            {
                "id": "TEMP_SVR1.1VDVS"
            }
        ],
        ...
    }

```

Une fois que vous avez joint un fichier à une tâche, le processus déplace le fichier à partir du stockage TEMP et le stocke avec les données de la tâche dans le stockage BPM. L'ID du fichier passe à `taskId::filename`.

Chargement de fichiers de groupement de ressources

Pour localiser l'interface utilisateur de Data Director, téléchargez le fichier ZIP de groupement de ressources, modifiez les fichiers du fichier ZIP, puis chargez le fichier ZIP modifié vers le stockage du groupement.

1. Pour télécharger le fichier ZIP de groupement de ressources, utilisez l'API REST Obtenir le contenu du fichier avec le type de stockage BUNDLE.

Par exemple, la demande suivante télécharge le fichier ZIP de groupement de ressources :

```

GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content

```

2. Modifiez le fichier ZIP en ajoutant les fichiers de groupement propres aux langues.
Par exemple, pour localiser des noms de champs, des libellés et des noms de tables en russe, ajoutez le fichier `besMetadata_ru.properties`.

3. Pour charger le fichier ZIP de groupement de ressources modifié, utilisez l'API REST Charger le contenu du fichier avec le type de stockage BUNDLE.

Par exemple, la demande suivante charge le fichier ZIP de groupement de ressources :

```

PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle

```

ANNEXE C

Utilisation des API REST pour gérer les rapports

Cette annexe comprend les rubriques suivantes :

- [Présentation de l'utilisation des API REST pour gérer les rapports, 307](#)
- [API REST pour les rapports, 308](#)
- [Configuration du rapport, 308](#)
- [Données du rapport, 309](#)
- [Rapports d'exploration, 310](#)
- [Rapports prêts à l'emploi, 311](#)
- [Rapports personnalisés, 315](#)
- [Dépannage des API de rapports, 320](#)

Présentation de l'utilisation des API REST pour gérer les rapports

Vous pouvez utiliser des API REST pour gérer les rapports prêts à l'emploi et les rapports personnalisés qui collectent des informations sur vos données principales. Dans l'outil de provisionnement, vous pouvez renseigner ultérieurement les composants de Graphe avec les données de rapports. Puis dans Data Director, les utilisateurs peuvent afficher les graphes afin d'analyser les informations sur les données principales.

Les rapports prêts à l'emploi collectent des informations sur les critères prédéfinis. Pour collecter d'autres informations sur vos données principales, utilisez des rapports personnalisés.

API REST pour les rapports

Vous pouvez gérer les rapports à l'aide d'un ensemble d'API REST.

Le tableau suivant répertorie les API REST pour les rapports :

API REST	Description
Répertorier les rapports	Renvoie une liste des rapports enregistrés et la configuration des rapports.
Obtenir la configuration et les données du rapport	Renvoie la configuration et les données du rapport.
Obtenir la configuration du rapport et les rapports d'exploration	Renvoie la configuration du rapport et les rapports d'exploration associés.
Enregistrer le rapport	Enregistre les rapports personnalisés. Renvoie l'ID du rapport que vous pouvez utiliser dans d'autres API.
Mettre à jour la configuration du rapport	Met à jour la configuration d'un rapport.
Ajouter ou mettre à jour les données du rapport	Ajoute ou met à jour les entrées de données dans un rapport.
Supprimer le rapport	Supprime un rapport.
Exécuter le travail de mise à jour du rapport	Démarré les travaux de lots associés aux rapports prêts à l'emploi.
Obtenir le statut du travail de mise à jour du rapport	Renvoie le statut d'un travail de mise à jour du rapport.

Configuration du rapport

La configuration du rapport comporte plusieurs paramètres, tels que le nom, la description, les dimensions et le nom de la métrologie du rapport. La configuration des rapports prêts à l'emploi est prédéfinie. Pour utiliser les rapports personnalisés, vous devez spécifier leur configuration.

Le tableau suivant décrit les paramètres dans la configuration d'un rapport :

Paramètre	Description
ROWID_RPT_CONFIG	ID de rapport.
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.

Paramètre	Description
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
RPT_NAME	Nom du rapport.
METRIC_NAME	Libellé du type de données collecté par le rapport. Par défaut, la valeur spécifiée pour ce paramètre s'affiche en tant que nom de l'axe des Y dans le graphe. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Vous pouvez utiliser le nom de la métrologie <code>Number of closed tasks</code> .
RPT_DESC	Description du rapport.
RPT_TYPE	Indique s'il s'agit d'un rapport d'exploration. Si la valeur est <code>Null</code> , il s'agit d'un rapport racine. Si la valeur n'est pas <code>Null</code> , il s'agit d'un rapport d'exploration.

Données du rapport

Ajoutez des données à un rapport pour pouvoir renseigner les graphes avec les données du rapport. Pour les rapports prêts à l'emploi, utilisez les travaux de mise à jour du rapport associés au rapport afin de collecter et d'ajouter les données du rapport. Si vous utilisez des rapports personnalisés, vous devez leur ajouter manuellement les entrées de données.

Le tableau suivant décrit les paramètres que vous devez configurer pour chaque entrée de données d'un rapport :

Paramètre	Description
DIMENSION_NAME_1	Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
DIMENSION_NAME_2	Facultatif. Nom d'une dimension de données dans le rapport. Par exemple, une dimension peut représenter les éléments suivants : Nom d'utilisateur, Type de tâche de vérification ou Type d'entité d'entreprise.
TIMEPERIOD_NAME	Facultatif. Nom de la période à laquelle les données s'appliquent. Par exemple, vous pouvez spécifier <code>Month</code> .
METRIC_VALUE	Nombre qui représente les dimensions des données. Par exemple, un rapport peut collecter des informations sur les tâches fermées par l'utilisateur. Une dimension du rapport est l'utilisateur John Smith et une autre est le type de tâche de vérification. La valeur de la métrologie peut être 5. Cela implique que John Smith a fermé cinq tâches du type de tâche de vérification.
DRILLDOWN_RPT_ID	ID d'un rapport d'exploration. Si la valeur est <code>Null</code> , aucun rapport d'exploration n'a été configuré.

L'exemple suivant présente trois entrées de données dans un rapport :

```
[
  {
```

```

    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "3",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeUpdate",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  }
  ...
]

```

Rapports d'exploration

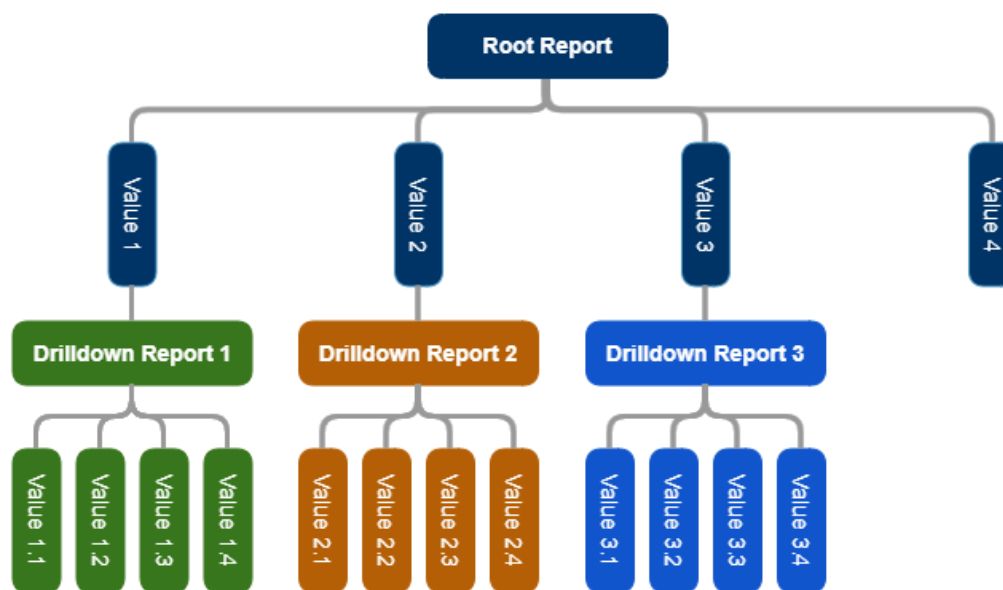
Un rapport d'exploration contient une autre couche de données sur une entrée de données d'un rapport. Vous pouvez utiliser les rapports d'exploration associés aux rapports prêts à l'emploi ou configurer des rapports d'exploration pour vos rapports personnalisés.

La création de rapports personnalisés permet d'enregistrer un rapport racine. Vous pouvez créer des rapports d'exploration et les associer à des entrées de données dans le rapport racine ou dans d'autres rapports d'exploration.

Par exemple, un rapport peut collecter des informations sur le nombre de tâches par statut. Vous pouvez associer un rapport d'exploration au nombre de tâches fermées. Le rapport d'exploration contient le nombre de tâches fermées pour chaque utilisateur. Vous pouvez également associer un rapport d'exploration au nombre de tâches approuvées. Le rapport d'exploration contient le nombre de tâches approuvées pour chaque utilisateur.

Dans Data Director, les utilisateurs peuvent afficher un graphe parent qui présente le nombre de tâches par statut. Lorsqu'ils sélectionnent le nombre de tâches fermées, ils accèdent alors à un graphe d'exploration qui présente le nombre de tâches fermées pour chaque utilisateur.

L'image suivante montre la structure des rapports d'exploration :



Pour plus d'informations sur les rapports d'exploration associés aux rapports prêts à l'emploi, consultez la rubrique ["Rapports prêts à l'emploi" à la page 311](#).

Rapports prêts à l'emploi

Les rapports prêts à l'emploi collectent des informations sur vos données principales. Certains rapports prêts à l'emploi possèdent également des rapports d'exploration. Ces derniers contiennent une autre couche de données sur les données de rapports.

Par exemple, le rapport Présentation des tâches par statut collecte des informations sur le nombre de tâches par statut. Les rapports d'exploration associés au rapport présentent les tâches attribuées à l'utilisateur pour chaque statut.

Pour utiliser les rapports prêts à l'emploi, vous devez exécuter le travail de mise à jour du rapport associé au rapport à utiliser. Le travail de mise à jour du rapport enregistre le rapport et collecte des informations sur vos données principales selon les critères du rapport. Dans l'outil de provisionnement, vous pouvez alors renseigner les composants de Graphe avec les données de rapports. Si des rapports d'exploration sont associés au rapport prêt à l'emploi, vous pouvez lier votre graphe à un graphe d'exploration.

Le tableau suivant décrit les rapports prêts à l'emploi et répertorie l'ID de chaque rapport :

ID de rapport	Rapport	Description
MDM.RPT.1	Présentation des tâches par statut	Affiche le nombre total de tâches et répartit les tâches par statut. Vous pouvez explorer les utilisateurs attribués pour le statut de chaque tâche.
MDM.RPT.2	Présentation des tâches par priorité	Affiche le nombre total de tâches et répartit les tâches par priorité. Vous pouvez explorer les utilisateurs attribués pour la priorité de chaque tâche.

ID de rapport	Rapport	Description
MDM.RPT.4	Entités d'entreprise ajoutées par an	Affiche le nombre d'enregistrements ajoutés pour chaque type d'entité d'entreprise et répartit les enregistrements par an.
MDM.RPT.5	Délai d'intégration du client	Affiche la durée de qualification des enregistrements.
MDM.RPT.6	Entités d'entreprise par systèmes sources	Affiche le nombre total d'enregistrements pour chaque type d'entité d'entreprise provenant de chaque système source. Le rapport détaillé répartit les enregistrements pour chaque paire de type d'entité d'entreprise et de système source par an.
MDM.RPT.7	Tâches attribuées par utilisateurs	Affiche le nombre total de tâches attribuées à chaque utilisateur.
MDM.RPT.8	Tâches ouvertes par rôles utilisateur	Affiche le nombre total de tâches ouvertes par rôles utilisateur.
MDM.RPT.10	Tâches fermées par les utilisateurs	Affiche le nombre total de tâches fermées par chaque utilisateur.

Gestion des rapports prêts à l'emploi

Pour utiliser les rapports prêts à l'emploi afin qu'ils collectent des informations sur vos données principales, exécutez les travaux de mise à jour associés à ces rapports. Dans l'outil de provisionnement, vous pouvez renseigner les composants de Graphe avec les données de rapports après l'exécution des travaux de mise à jour.

Avant de commencer, vérifiez les rapports prêts à l'emploi et déterminez ceux que vous souhaitez utiliser. Pour plus d'informations, consultez la rubrique ["Rapports prêts à l'emploi" à la page 311](#).

1. Pour exécuter le travail de mise à jour associé au rapport prêt à l'emploi, entrez l'URL REST dans la barre d'adresses du navigateur depuis l'API REST Exécuter le travail de mise à jour du rapport.
Par exemple, l'URL REST suivante exécute le travail de mise à jour associé au rapport Présentation des tâches par statut :

```
http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/MDM.RPT.1
```

Vous avez enregistré le rapport et commencé la collecte de données selon les critères du rapport. L'exemple de réponse suivant présente le statut du travail de mise à jour du rapport :

```
{
  "status": "PROCESSING"
}
```

2. Pour récupérer le statut du travail de mise à jour du rapport, entrez l'URL REST dans la barre d'adresses du navigateur depuis l'API REST Obtenir le statut du travail de mise à jour du rapport.
Par exemple, l'URL REST suivante récupère le statut du travail de mise à jour :

```
http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/MDM.RPT.1/status
```

L'exemple de réponse suivant présente le statut du travail de mise à jour du rapport :

```
{
  "status": "COMPLETED_SUCCESSFULLY",
  "jobId": "SVR1.2X9N1",
  "lastUpdateDate": "2019-10-31T14:37:11.846-04:00",
  "startRunDate": "2019-10-31T14:37:09.120-04:00"
}
```


3. Pour récupérer la configuration et les données du rapport, entrez L'URL REST dans la barre d'adresses du navigateur depuis l'API REST Obtenir la configuration et les données du rapport.

Par exemple, l'URL REST suivante récupère la configuration et les données du rapport :

`http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/MDM.RPT.1`

L'exemple de réponse suivant présente la configuration et les données du rapport :

```
{
  "metadata":{
    "fieldsMetadata":[
      "DIMENSION_VALUE_1",
      "DIMENSION_VALUE_2",
      "TIMEPERIOD_VALUE",
      "METRIC_VALUE",
      "DRILLDOWN_RPT_ID"
    ],
    "ROWID_RPT_CONFIG":"MDM.RPT.1",
    "DIMENSION_NAME_1":"Task Status",
    "METRIC_NAME":"Number of tasks",
    "DIMENSION_NAME_2":"Task Type",
    "TIMEPERIOD_NAME":null,
    "RPT_NAME":"Task Status/Type Report",
    "RPT_DESC":"Metrics for task status/type",
    "RPT_TYPE":null
  },
  "data":[
    [
      "Overdue",
      "AVOSBeMerge",
      null,
      "36",
      "SVR1.4FCA2"
    ],
    [
      "Overdue",
      "AVOSBeReviewNoApprove",
      null,
      "189",
      "SVR1.4FC8E"
    ],
    [
      "Overdue",
      "AVOSBeUpdate",
      null,
      "1",
      "SVR1.4FCAQ"
    ],
    [
      "Overdue",
      "AVOSBeFinalReview",
      null,
      "1",
      "SVR1.4FC9W"
    ]
  ]
}
```

4. Pour récupérer la configuration du rapport et les rapports d'exploration, utilisez l'API REST Obtenir la configuration du rapport et les rapports d'exploration.

Par exemple, l'URL REST suivante récupère la configuration du rapport et les rapports d'exploration :

`http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/meta/MDM.RPT.1`

L'exemple de réponse suivant présente la configuration du rapport et les rapports d'exploration :

```
{
  "ROWID_RPT_CONFIG":"MDM.RPT.1",
  "DIMENSION_NAME_1":"Task Status",
  "METRIC_NAME":"Number of tasks",
```

```

"DIMENSION_NAME_2":"Task Type",
"TIMEPERIOD_NAME":null,
"RPT_NAME":"Task Status/Type Report",
"RPT_DESC":"Metrics for task status/type",
"RPT_TYPE":null,
"DRILLDOWN":[
  {
    "RPT_NAME":"Task's Owner per Task Status and Task Type",
    "DETAILED_RPT_IDS":[
      "SVR1.4FCA3",
      "SVR1.4FCA4",
      "SVR1.4FCA5",
      "SVR1.4FCA6",
      "SVR1.4FCA7",
      "SVR1.4FC8F",
      "SVR1.4FC8G",
      "SVR1.4FC8H",
      "SVR1.4FC8I",
      "SVR1.4FC8J",
      "SVR1.4FCAR",
      "SVR1.4FCAS",
      "SVR1.4FCAT",
      "SVR1.4FCAU",
      "SVR1.4FCAV",
      "SVR1.4FC9X",
      "SVR1.4FC9Y",
      "SVR1.4FC9Z",
      "SVR1.4FCA0",
      "SVR1.4FCA1",
      "SVR1.4FC9L",
      "SVR1.4FC9M",
      "SVR1.4FC9N",
      "SVR1.4FC9O",
      "SVR1.4FC9P",
      "SVR1.4FC8L",
      "SVR1.4FC8M",
      "SVR1.4FC8N",
      "SVR1.4FC8O"
    ],
    "CONFIG_RPT_IDS":[
      "SVR1.4FCA2",
      "SVR1.4FC8E",
      "SVR1.4FCAQ",
      "SVR1.4FC9W",
      "SVR1.4FC9K",
      "SVR1.4FC8K",
      "SVR1.4FCAW",
      "SVR1.4FC9E",
      "SVR1.4FC88",
      "SVR1.4FC8W",
      "SVR1.4FC82",
      "SVR1.4FC92",
      "SVR1.4FC98",
      "SVR1.4FCAE",
      "SVR1.4FCA8",
      "SVR1.4FCB8",
      "SVR1.4FC9Q",
      "SVR1.4FCAK",
      "SVR1.4FC8Q",
      "SVR1.4FCB2"
    ]
  }
]
}

```

Rapports personnalisés

Si les rapports prêts à l'emploi ne répondent pas aux exigences d'entreprise, vous pouvez configurer des rapports personnalisés pour collecter d'autres informations sur vos données principales. Par exemple, vous souhaitez peut-être collecter des informations sur le nombre de tâches fermées selon la priorité.

Pour utiliser des rapports personnalisés, vous devez configurer et enregistrer les rapports, puis ajouter des entrées au rapport. Pour associer des rapports d'exploration aux données de rapports, configurez et enregistrez les rapports d'exploration. Vous pouvez ensuite associer le rapport d'exploration à une entrée de données dans le rapport personnalisé.

Gestion des rapports personnalisés

Enregistrez et configurez les rapports personnalisés à utiliser pour collecter des informations sur vos données principales. Dans l'outil de provisionnement, vous pouvez alors renseigner le composant Graphe avec les données de rapports.

Avant de commencer, déterminez les données à collecter.

1. Pour vérifier les rapports enregistrés, utilisez l'API REST Répertoire des rapports.
Par exemple, la demande suivante répertorie les rapports racine enregistrés :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
```

2. Pour enregistrer un rapport personnalisé, utilisez l'API REST Enregistrer le rapport.
Par exemple, la demande suivante enregistre un rapport personnalisé :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks"
}
```

L'exemple de réponse suivant présente le rapport enregistré :

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks",
  "RPT_TYPE": "null"
}
```

La demande renvoie l'ID de rapport dans le paramètre ROWID_RPT_CONFIG.

3. Pour ajouter des entrées de données dans un rapport, utilisez l'API REST Ajouter ou mettre à jour les données du rapport.

Par exemple, la demande suivante ajoute des entrées de données dans un rapport :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "3",
    "DRILLDOWN_RPT_ID": "null"
  },
  {

```

```

        "DIMENSION_VALUE_1": "High",
        "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    },
    {
        "DIMENSION_VALUE_1": "High",
        "DIMENSION_VALUE_2": "AVOSBeUpdate",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    }
    ...
]

```

4. Pour récupérer la configuration et les données du rapport, utilisez l'API REST Obtenir la configuration et les données du rapport.

Par exemple, la demande suivante récupère la configuration et les données du rapport :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

L'exemple de réponse suivant présente la configuration et les données du rapport :

```

{
  "metadata":{
    "fieldsMetadata":[
      "DIMENSION_VALUE_1",
      "DIMENSION_VALUE_2",
      "TIMEPERIOD_VALUE",
      "METRIC_VALUE",
      "DRILLDOWN_RPT_ID"
    ],
    "ROWID_RPT_CONFIG": "MDM.RPT.2",
    "DIMENSION_NAME_1": "Task Priority",
    "METRIC_NAME": "Number of tasks",
    "DIMENSION_NAME_2": "Task Type",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task Priority/Type Report",
    "RPT_DESC": "Metrics for task status/type",
    "RPT_TYPE": "null"
  },
  "data":[
    [
      "High",
      "AVOSBeMerge",
      "null",
      "3",
      "SVR1.48P5G"
    ],
    [
      "High",
      "AVOSBeReviewNoApprove",
      "null",
      "0",
      "null"
    ],
    [
      "High",
      "AVOSBeUpdate",
      null,
      "0",
      "null"
    ],
    [
      "High",
      "AVOSBeFinalReview",
      null,
      "0",
      "null"
    ]
  ]
}

```

```

    ]
    ...
  ]
}

```

5. Pour supprimer un rapport, utilisez l'API REST Supprimer le rapport.

Par exemple, la demande suivante supprime un rapport :

```
DELETE http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Gestion des rapports personnalisés avec des rapports d'exploration

Enregistrez les rapports personnalisés à utiliser et ajoutez des entrées à votre rapport. Puis, enregistrez les rapports d'exploration et ajoutez-leur les entrées de données. Lorsque vous ajoutez ou mettez à jour des entrées de données dans vos rapports racine, vous pouvez associer vos rapports d'exploration aux entrées de données. Dans l'outil de provisionnement, vous pouvez ensuite renseigner le composant Graphe avec les données de rapport et lier le graphe racine à un graphe d'exploration.

Avant de commencer, déterminez les données à collecter. Déterminez également les rapports d'exploration à configurer et à associer à vos entrées de données dans votre rapport personnalisé.

1. Pour vérifier les rapports enregistrés, utilisez l'API REST Répertoire les rapports.
Par exemple, la demande suivante répertorie les rapports racine et les rapports d'exploration enregistrés :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list?show=all
```

2. Pour enregistrer un rapport personnalisé, utilisez l'API REST Enregistrer le rapport.
Par exemple, la demande suivante enregistre un rapport personnalisé :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks"
}
```

L'exemple de réponse suivant présente le rapport enregistré :

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks",
  "RPT_TYPE": "null"
}
```

La demande renvoie l'ID de rapport dans le paramètre ROWID_RPT_CONFIG.

3. Pour ajouter des entrées de données dans un rapport racine, utilisez l'API REST Ajouter ou mettre à jour les données du rapport.

Par la suite, vous pouvez associer un rapport d'exploration à une entrée de données.

Par exemple, la demande suivante ajoute des entrées de données dans un rapport :

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",

```

```

        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "3",
        "DRILLDOWN_RPT_ID": "null"
    },
    {
        "DIMENSION_VALUE_1": "High",
        "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    },
    {
        "DIMENSION_VALUE_1": "High",
        "DIMENSION_VALUE_2": "AVOSBeUpdate",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    }
    ...
]

```

4. Pour enregistrer un rapport d'exploration, utilisez l'API REST Enregistrer le rapport.
Par exemple, la demande suivante enregistre le rapport d'exploration :

```

POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
    "DIMENSION_NAME_1": "Task's Owner",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task's Owner per Task Priority and Task Type",
    "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
    "METRIC_NAME": "Number of Tasks"
}

```

L'exemple de réponse suivant présente le rapport d'exploration enregistré :

```

{
    "ROWID_RPT_CONFIG": "SVR1.48P5G",
    "DIMENSION_NAME_1": "Task's Owner",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task's Owner per Task Priority and Task Type",
    "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
    "METRIC_NAME": "Number of Tasks",
    "RPT_TYPE": "High/AVOSBeMerge"
}

```

La demande renvoie l'ID de rapport d'exploration dans le paramètre ROWID_RPT_CONFIG.

5. Pour ajouter des entrées de données dans un rapport d'exploration, utilisez l'API REST Ajouter ou mettre à jour les données du rapport.

Par exemple, la demande suivante ajoute des entrées de données dans un rapport d'exploration :

```

POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.48P5G
[
    {
        "DIMENSION_VALUE_1": "admin",
        "DIMENSION_VALUE_2": "null",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    },
    {
        "DIMENSION_VALUE_1": "srmgr2",
        "DIMENSION_VALUE_2": "null",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    },
    {
        "DIMENSION_VALUE_1": "mgr2",

```

```

        "DIMENSION_VALUE_2": "null",
        "TIMEPERIOD_VALUE": "null",
        "METRIC_VALUE": "0",
        "DRILLDOWN_RPT_ID": "null"
    }
}

```

6. Pour associer un rapport d'exploration à une entrée de données dans un rapport, utilisez l'API REST Ajouter ou mettre à jour les données du rapport. Spécifiez l'ID de rapport d'exploration dans le paramètre DRILLDOWN_RPT_ID.

Par exemple, la demande suivante associe un rapport d'exploration à l'entrée de données :

```

POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "5",
    "DRILLDOWN_RPT_ID": "SVR1.48P5G"
  }
]

```

7. Pour récupérer la configuration et les données du rapport, utilisez l'API REST Obtenir la configuration et les données du rapport.

Par exemple, la demande suivante récupère la configuration et les données du rapport :

```

GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0

```

L'exemple de réponse suivant présente la configuration et les données du rapport :

```

{
  "metadata":{
    "fieldsMetadata":[
      "DIMENSION_VALUE_1",
      "DIMENSION_VALUE_2",
      "TIMEPERIOD_VALUE",
      "METRIC_VALUE",
      "DRILLDOWN_RPT_ID"
    ],
    "ROWID_RPT_CONFIG": "SVR1.2X9N0",
    "DIMENSION_NAME_1": "Task Priority",
    "METRIC_NAME": "Number of tasks",
    "DIMENSION_NAME_2": "Task Type",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task Priority/Type Report",
    "RPT_DESC": "Metrics for task status/type",
    "RPT_TYPE": "null"
  },
  "data":[
    [
      "High",
      "AVOSBeMerge",
      "null",
      "3",
      "SVR1.48P5G"
    ],
    [
      "High",
      "AVOSBeReviewNoApprove",
      "null",
      "0",
      "null"
    ],
    [
      "High",
      "AVOSBeUpdate",
      "null",
      "0",
      "null"
    ]
  ]
}

```

```

    ],
    [
        "High",
        "AVOSBeFinalReview",
        "null",
        "0",
        "null"
    ]
    ...
]
}

```

8. Pour récupérer les rapports enregistrés, y compris vos rapports personnalisés et rapports d'exploration, utilisez l'API REST Répertoire des rapports.

Par exemple, la demande suivante répertorie les rapports racine et les rapports d'exploration enregistrés :

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list?show=all
```

L'exemple de réponse suivant présente les rapports racine et les rapports d'exploration enregistrés :

```

[
  {
    "ROWID_RPT_CONFIG": "SVR1.2X9N0",
    "DIMENSION_NAME_1": "Task Priority",
    "METRIC_NAME": "Number of tasks",
    "DIMENSION_NAME_2": "Task Type",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task Priority/Type Report",
    "RPT_DESC": "Metrics for task status/type",
    "RPT_TYPE": "null"
  },
  {
    "ROWID_RPT_CONFIG": "SVR1.48P5G",
    "DIMENSION_NAME_1": "Task's Owner",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task's Owner per Task Priority and Task Type",
    "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
    "METRIC_NAME": "Number of Tasks",
    "RPT_TYPE": "High/AVOSBeMerge"
  }
]

```

9. Pour supprimer un rapport, utilisez l'API REST Supprimer le rapport.

Par exemple, la demande suivante supprime un rapport :

```
DELETE http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Dépannage des API de rapports

Si vous rencontrez des problèmes avec les API de rapports, utilisez les informations suivantes pour les résoudre.

Échec de l'exécution des travaux de mise à jour associés aux rapports prêts à l'emploi.

Les services de rapports peuvent ne pas être connectés à ActiveVOS. Si vous avez modifié les fichiers EAR d'ActiveVOS afin de personnaliser la chaîne de recherche JNDI, vous devez ajouter la propriété `activevos.jndi` au fichier `cmxserver.properties` et spécifier la chaîne de recherche JNDI personnalisée. Pour plus d'informations, consultez le *Guide de configuration de Multidomain MDM*.

INDEX

A

API REST

- action de tâches en bloc [105](#)
- ajouter ou mettre à jour les détails du rapport [232](#)
- annuler la fusion des enregistrements [149](#)
- attribuer des tâches en bloc [101](#)
- corps [30](#)
- corps de demande [31](#)
- créer les métadonnées du fichier [112](#)
- créer un enregistrement [54](#)
- créer une relation [163](#)
- créer une tâche [87](#)
- en-tête [30](#)
- en-tête de demande [31](#)
- enregistrer le rapport [230](#)
- exécuter l'action de tâche [95](#)
- exécuter le travail de mise à jour du rapport [234](#)
- exporter la hiérarchie [189](#)
- exporter les enfants et les parents directs [190](#)
- fusion en attente [142](#)
- fusionner les enregistrements [147](#)
- générateur de suggestions [68](#)
- importation DaaS [248](#)
- lecture DaaS [245](#)
- libérer des tâches en bloc [100](#)
- lire l'enregistrement [48](#)
- lire la relation [161](#)
- lire la tâche [85](#)
- lister l'enregistrement [61](#)
- lister les métadonnées des fichiers [111](#)
- lister les tâches [79](#)
- lister les utilisateurs pouvant être attribués [97](#)
- mettre à jour l'enregistrement [56](#)
- mettre à jour la configuration du rapport [231](#)
- mettre à jour la tâche [90](#)
- mettre à jour les enregistrements correspondants [212](#)
- mettre à jour les métadonnées du fichier [114](#)
- mettre à jour une relation [165](#)
- mise à jour DaaS [251](#)
- modifications des relations en bloc [198](#)
- modifier des tâches en bloc [103](#)
- obtenir la configuration du rapport et les rapports d'exploration [228](#)
- obtenir la configuration et les données du rapport [226](#)
- obtenir le chemin de la hiérarchie [177](#)
- obtenir le contenu du fichier [116](#)
- obtenir le statut du travail de mise à jour du rapport [235](#)
- obtenir les actions de tâches [106](#)
- obtenir les détails de l'événement [223](#)
- obtenir les enfants [182](#)
- obtenir les enregistrements associés [167](#)
- obtenir les enregistrements correspondants [211](#)
- obtenir les événements d'historique de l'enregistrement [220](#)
- obtenir les métadonnées [38](#)
- obtenir les métadonnées BPM [78](#)
- obtenir les métadonnées DaaS [239](#)

API REST (a continué)

- obtenir les métadonnées de la hiérarchie [172](#)
- obtenir les métadonnées du fichier [113](#)
- obtenir les modifications de la hiérarchie [192](#)
- obtenir les parents [180](#)
- prévisualiser la fusion [136](#)
- prévisualiser la promotion [133](#)
- promouvoir en bloc [200](#)
- promouvoir l'enregistrement [135](#)
- promouvoir la fusion [143](#)
- Recherche DaaS [240](#)
- rechercher un enregistrement [63](#)
- réclamer des tâches en bloc [98](#)
- rejeter en bloc [206](#)
- répertorier des métadonnées [42](#)
- répertorier les colonnes de correspondance [46](#)
- répertorier les hiérarchies [171](#)
- répertorier les propriétaires potentiels d'une tâche [110](#)
- répertorier les propriétaires potentiels des tâches [108](#)
- répertorier les rapports [225](#)
- SearchMatch [73](#)
- SearchQuery [69](#)
- supprimer le rapport [234](#)
- supprimer les enregistrements correspondants [213](#)
- supprimer les modifications en attente [136](#)
- supprimer un enregistrement [60](#)
- supprimer un fichier [116](#)
- supprimer une relation [166](#)
- tâche terminée [93](#)

API SOAP

- authentification [255](#)
- demande [258](#)
- réponse [258](#)
- WSDL [256](#)
- appels externes
 - présentation [277](#)
- approbation
 - WriteBE [12](#)
- authentification
 - HTTP de base [27](#)
 - méthode [27](#)
 - utilisation de cookies [28](#), [255](#)

C

- configuration
 - appels externes [278](#)
- Corps REST
 - Format JSON [32](#)
 - Format XML [31](#)
- créer un enregistrement
 - corps de réponse [55](#)
 - en-tête de réponse [55](#)
 - Paramètres d'URL [55](#)
 - URL de demande [54](#)

créer une tâche
URL de demande [87](#)

D

données de liaison
application personnalisée [276](#)
fractionnement [276](#)

E

enregistrement supprimé
restaurer [12](#)
enregistrements
ajout [285](#)
utilisation d'API REST [285](#)
enregistrements racines
identification [15](#)
étapes de service d'entité d'entreprise
SearchBE [13](#)
WriteBE [12](#)
événements pris en charge
liste [278](#)
exécuter l'action de tâche
corps de demande [95](#)
exemple
appels externes [279](#)
logique personnalisées [279](#)

F

format de date
à propos de [33](#)
fuseau horaire
à propos de [33](#)

L

liaison d'entreprise
prise en charge [275](#)
lire l'enregistrement
paramètres de la requête [49](#)
lire la tâche
URL de demande [85](#)
lister la tâche
URL de demande [80](#)
lister les tâches
paramètres de la requête [80](#)
paramètres de tri [81](#)

M

Méthodes REST
DELETE [27](#)
GET [27](#)
PATCH [27](#)
POST [27](#)
pris en charge [27](#)
PUT [27](#)
mettre à jour l'enregistrement
corps de réponse [59](#)
en-tête de réponse [59](#)
Paramètres d'URL [57](#)

O

obtenir les détails de l'événement
paramètres de la requête [223](#)
obtenir les événements d'historique de l'enregistrement
paramètres de la requête [221](#)

P

paramètres de la requête
depth [33](#)
firstRecord [33](#)
recordsToReturn [33](#)
returnTotal [33](#)
searchToken [33](#)
périodes effectives
WriteBE [12](#)
préface [9](#)

R

ReadBE
service d'entité d'entreprise [12](#)
restaurer
enregistrement supprimé de manière réversible [12](#)

S

SearchBE
à propos de [13](#)
SearchMatch
exportation des résultats [78](#)
SearchQuery
exportation des résultats [73](#)
service d'entité d'entreprise
ReadBE [12](#)
service de liaison
configuration [276](#)
services d'entité commerciale
Point d'extrémité EJB [13](#)
Point d'extrémité REST [14](#)
Point d'extrémité SOAP [14](#)
points d'extrémité [13](#)
Points d'extrémité SOAP [254](#)
services d'entité d'entreprise
importation DaaS [276](#)
mise à jour DaaS [276](#)
Points d'extrémité REST [26](#)
Référence d'API REST [38](#)
supprimer un enregistrement
Paramètre d'URL [60](#)
URL de demande [60](#)

T

test des appels externes
prérequis [279](#)

U

UTC
à propos de [33](#)

W

WriteBE

étape de service d'entité d'entreprise [12](#)