

**Sage CRM**

**Sage CRM 2016 R1**  
**Guide des services**  
**Web**



© Copyright 2015 Sage Technologies Limited, éditeur de ce produit. Tous droits réservés.

Il est interdit de copier, photocopier, reproduire, traduire, copier sur microfilm, et de reproduire de quelque manière que ce soit des parties de cette documentation sans l'accord écrit préalable de Sage Technologies Limited.

L'utilisation des programmes de logiciel décrite ci-dessus et cette documentation sont soumis à l'accord de licence de l'utilisateur final disponible dans le package du logiciel, ou qui a été acceptée lors de l'inscription au système.

Sage et le logo Sage sont des marques déposées ou des marques commerciales de The Sage Group PLC. Toutes les autres marques sont des marques commerciales ou des marques déposées de leurs propriétaires respectifs.

# Table des matières

---

<b>Chapitre 1 : Introduction aux services Web</b>	<b>1-1</b>
Présentation générale de la technologie des services Web .....	1-1
Fonctionnalités des services Web Sage CRM .....	1-2
<b>Chapitre 2 : Configuration des services Web Sage CRM</b>	<b>2-1</b>
Conditions préalables .....	2-1
Etapas d'utilisation des services Web .....	2-1
Configuration de l'utilisateur des services Web .....	2-1
Spécification des paramètres de configuration des services Web .....	2-2
Paramètres de configuration recommandés .....	2-3
Accès au fichier WSDL .....	2-3
<b>Chapitre 3 : Présentation des objets et fonctions</b>	<b>3-1</b>
Manipulation des enregistrements .....	3-1
Fonctions .....	3-1
Objets .....	3-1
<b>Chapitre 4 : Liste des fonctions des services Web</b>	<b>4-1</b>
<b>Chapitre 5 : Liste des objets des services Web</b>	<b>5-1</b>
Objets abstraits .....	5-1
Objets standard .....	5-2
Insertion et mise à jour des articles de devis et de commande .....	5-2
<b>Chapitre 6 : L'objet RecordType Sage CRM</b>	<b>6-1</b>
<b>Chapitre 7 : Champs de sélection dans les services Web</b>	<b>7-1</b>
Liste des champs de sélection .....	7-1
Champs de sélection Opportunité .....	7-2
Champs de sélection Ticket .....	7-2
Champs de sélection Adresse et Produit .....	7-3
Utilisation de GetDropDownValues .....	7-3
<b>Chapitre 8 : Exemples de requêtes SOAP</b>	<b>8-1</b>
Récupération dynamique des détails WSDL et exemple de requête d'authentification ...	8-1
Exemple de requête Soap de connexion .....	8-3
Exemple de requête Soap de suppression .....	8-4

Exemple de requête Soap de mise à jour .....	8-4
Exemple de requête Soap pour QueryEntity .....	8-5
Exemple Soap XML représentant une société .....	8-5

# Chapitre 1 : Introduction aux services Web

L'API (Application Programming Interface) des services Web Sage CRM permet aux développeurs de manipuler à distance les enregistrements Sages CRM à l'aide du protocole SOAP (Simple Object Access Protocol) via HTTP et XML (Extensible Markup Language). Il est possible d'accéder à un serveur Sage CRM ou à un système hébergé à partir d'une machine client spécifiée (généralement un autre serveur) afin de lire, créer, mettre à jour ou supprimer des enregistrements pour chaque entité exposée, par exemple, les sociétés, les contacts, les opportunités, les tickets, les devis et les commandes.

Pour plus d'informations sur l'insertion et la mise à jour des champs Article de devis et de commande, reportez-vous à la section [Liste des objets des services Web \(page 5-1\)](#).

**Remarque** : l'API des services Web SOAP n'est pas disponible dans l'édition Sage CRM Essentials. Veuillez consulter le Tableau des versions dans l'aide produit pour plus d'informations.

Les principales étapes impliquées dans la communication avec les services Web Sage CRM sont les suivantes :

1. Le fichier WSDL (Web Services Description Language) est généré sur le serveur Sage CRM.
2. L'utilisateur accède alors au fichier WSDL à partir du client et prépare la requête.
3. La machine client transmet la requête avec ses paramètres au service Web.
4. Le service Web traite la requête et envoie une réponse au client.
5. Le client reçoit la réponse de manière synchrone et il traite les données retournées ou s'occupe des erreurs.

## Présentation générale de la technologie des services Web

Les services Web constituent une méthode standard pour l'intégration des applications Web à l'aide de XML, SOAP et WSDL via une dorsale IP. Les composants des services Web fonctionnent comme suit :

- Le langage XML balise les données.
- Le langage SOAP transfère les données. Pour plus d'informations sur le langage SOAP, consultez le site <http://www.w3.org/TR/SOAP>.
- Le langage WSDL décrit les services disponibles.

Cette technologie permet aux organisations d'échanger des données sans requérir une connaissance approfondie des systèmes informatiques derrière le pare-feu de chaque partie impliquée. Elle n'offre pas d'interface graphique aux utilisateurs, contrairement aux modèles client/serveur traditionnels. Au lieu de cela, les services Web partagent la logique, les données et les processus via une interface de programmation sur un réseau. Les développeurs peuvent ajouter le service Web à une interface graphique, telle qu'une page Web ou un programme exécutable, afin de fournir aux utilisateurs les fonctionnalités requises.

La technologie permet à des applications différentes de sources différentes de communiquer entre elles sans codage personnalisé. Comme toutes les communications utilisent le langage XML, les services Web ne limitent pas l'utilisateur à un langage de programmation.

## **Fonctionnalités des services Web Sage CRM**

Dans Sage CRM, la possibilité de manipuler les enregistrements à distance permet de profiter des fonctionnalités suivantes :

- Modification des données. Possibilité d'ajouter, de mettre à jour et de supprimer des enregistrements dans la base de données Sage CRM.
- Intégration à des applications tierces. L'accès à l'API des services Web Sage CRM permet d'intégrer les applications tierces utilisées au sein de votre organisation (par exemple, logiciels de comptabilité ou systèmes ERP), avec le serveur Sage CRM ou sur les versions Cloud.
- Environnements Cloud. Outre la manipulation des enregistrements sur un serveur Sage CRM standard, les services Web Sage CRM sont compatibles avec les versions Cloud. Par conséquent, les clients du Cloud peuvent tirer profit de la technologie et ses fonctionnalités.

**Remarque :** Sage CRM Cloud a été conçu pour optimiser les performances des clients qui utilisent activement leur compte Sage CRM, c'est-à-dire ceux qui se connectent et se déconnectent, utilisent le plug-in Outlook, Sage CRM SData ou encore l'API des services Web. Le déploiement des instances Sage CRM des clients qui ne sont pas utilisées est annulé, afin de ne pas consommer de ressources inutiles.

# Chapitre 2 : Configuration des services Web Sage CRM

## Conditions préalables

Tous les environnements de développement à jour compatibles avec Soap 1.1 sont compatibles avec les services Web Sage CRM. Les environnements pris en charge incluent Microsoft Visual Studio 2003 et versions ultérieures (C#, J#, VB.NET) ainsi que Microsoft Visual C# 2005 Express Edition.

## Étapes d'utilisation des services Web

L'utilisation des services Web inclut les étapes suivantes :

1. configuration d'un utilisateur des services Web
2. spécification des paramètres de configuration des services Web
3. accès au fichier WSDL
4. préparation et soumission de la requête aux services Web
5. traitement de la réponse, valeurs retournées ou des messages d'erreur.

Les étapes 1 et 2 sont décrites ci-dessous. Pour savoir comment préparer la requête et gérer la réponse, reportez-vous aux rubriques [Présentation des objets et fonctions \(page 3-1\)](#) et Exemples de services Web.

## Configuration de l'utilisateur des services Web

Avant de pouvoir accéder aux services Web, un compte utilisateur doit être défini pour les services Web .

Pour configurer un utilisateur pour les services Web :

1. Sélectionnez **Administration | Utilisateurs | Utilisateurs** et recherchez l'utilisateur pour lequel vous souhaitez activer l'accès aux services Web.
2. Sélectionnez le lien hypertexte pour l'utilisateur, puis cliquez sur le bouton **Modifier**.
3. Faites défiler le volet Profil de sécurité, définissez le champ **Autoriser l'accès aux services Web** sur Vrai.
4. Cliquez sur le bouton **Sauvegarder**.

Un seul utilisateur des services Web peut se connecter avec le même code à la fois. Si un utilisateur tente de se connecter en tant qu'autre application, une erreur est affichée invitant celui-ci à se déconnecter. Il est toutefois possible de se connecter au bureau ou à partir d'un périphérique associé au même code alors qu'une application des services Web est en cours d'exécution.

La fonction Sécurité des champs détermine quels champs peuvent être consultés ou mis à jour avec les méthodes des services Web. Par exemple, la fonctionnalité de sécurité des champs

bloque l'accès en lecture d'un utilisateur à un champ, les méthodes appelées par une session des services Web utilisant les informations de connexion de cet utilisateur ne peuvent pas retourner, mettre à jour ou supprimer les valeurs de ce champ. Pour plus d'informations sur la sécurité des champs, reportez-vous à l'Guide de l'administrateur système.

## Spécification des paramètres de configuration des services Web

Pour accéder aux paramètres de configuration du service Web, sélectionnez **Administration | Système | Services Web**.

Le tableau suivant décrit les champs de la page Paramètres des services Web.

Champ	Description
Nombre maximum d'enregistrements à retourner	Nombre maximum d'enregistrements que la page Services Web peut afficher simultanément à l'écran. Elle s'utilise conjointement avec les méthodes query et queryrecord. Le nombre entré ici correspond au nombre d'enregistrements qui seront retournés dans un lot en réponse à une requête. A mesure que chaque lot est retourné, vous êtes invité à appeler le lot suivant, jusqu'à ce que tous les enregistrements correspondant à la requête soient retournés. Si le champ est défini sur 0, tous les enregistrements correspondant à la requête sont retournés dans un lot unique.
Format maximum de la requête	Nombre maximum de caractères que les utilisateurs peuvent envoyer sur la page Services Web.
Rendre WSDL disponible pour tous	Lorsque vous choisissez Oui, L'URL à laquelle le fichier WSDL est accessible se trouve dans <b>Administration   Mon compte   Accès externe   Informations sur l'accès externe</b> .
Activer les services Web	Lorsque ce champ est défini sur Oui, la fonction Services Web est activée. Si le champ est défini sur Non, la fonction Services Web est désactivée. Pour activer ou désactiver les services Web pour une table ou une entité individuelle, allez dans <b>Administration   Personnalisation   Nom d'entité/de table   Accès externe</b> et définissez le champ <b>Services Web</b> sur Oui pour activer et sur Non pour désactiver.
Champs déroulants comme chaînes dans le fichier WSDL	La valeur par défaut est Oui. Les listes déroulantes s'affichent dans le fichier WSDL sous la forme de types énumérés, par exemple comp_status comme énumération avec des valeurs à sélectionner. Pour plus d'informations, reportez-vous à la section Objets et fonctions. Lorsque vous choisissez Oui, les types énumérés deviennent des « chaînes ». Il s'agit du paramètre recommandé. Cela signifie, par exemple, que dans le contexte Société, le champ comp_status a désormais le type « Chaîne ».
Envoyer et retourner toutes les dates et heures en temps universel	Lorsque cette option est sélectionnée, toutes les dates provenant du serveur sont définies en temps universel. Par ailleurs, toutes les dates provenant du serveur Web sont décalées par rapport au temps universel. Cela représente un intérêt pour les migrations effectuées vers la version Cloud à partir de fuseaux horaires différents.

Champ	Description
Accepter la requête Web à partir de l'adresse IP	Indique l'adresse IP unique à partir de laquelle vous souhaitez que le fichier WSDL soit accessible. Dans ce cas, le champ « Rendre l'accès aux services Web disponible pour tous » doit être défini sur Non.
Forcer la connexion du service Web	Si la connexion entre le client du service Web et le service est interrompue de façon inattendue, le client demeure connecté au serveur qui héberge le service. Par conséquent, une nouvelle instance du client ne pourra pas se connecter au serveur. Cependant, si l'option « Forcer la connexion du service Web » est définie sur Oui, l'ancienne instance du client est automatiquement déconnectée lorsqu'une nouvelle instance tente de se connecter. En forçant les nouvelles connexions, ce champ évite à l'utilisateur d'être « banni » d'un service Web suite à une connexion ou déconnexion ayant échoué.

### Paramètres de configuration recommandés

Voici les paramètres recommandés pour permettre à votre client d'accéder au service Web pendant le développement :

1. Définissez le champ **Activer les services Web** sur **Oui**.
2. Sélectionnez **Oui** dans le champ **Rendre l'accès WSDL disponible pour tous**.
3. Définissez le champ **Forcer la connexion du service Web** sur **Oui**.

Une fois le test du client des services Web terminé, il est recommandé de définir le paramètre **Rendre l'accès WSDL disponible pour tous** sur **Non** pour renforcer la sécurité.

### Accès au fichier WSDL

Comme dans le cas des services Web SOAP standard, Sage CRM fournit un fichier WSDL (Web Services Description Language).

Pour avoir accès à ce fichier depuis l'application cliente, ouvrez le fichier CRMWebService.WSDL depuis votre adresse d'installation.

Sur SageCRM.com, L'URL à laquelle le fichier WSDL est accessible se trouve dans **Administration | Mon compte | Accès externe | Informations sur l'accès externe**. L'URL ressemble à ceci :

`https://cloud.sagecrm.com/[accountname123]/eware.dll/webservices/CRMwebservice.wsdl`

Le fichier WSDL de Sage CRM décrit toutes les API que Sage CRM expose, ainsi que les types XML que les API attendent. Le fichier décrit aussi le serveur et l'emplacement où ces services spécifiques peuvent être trouvés. Une fois que le client a lu et analysé le fichier WSDL, il peut appeler les API de la même manière que n'importe quelle fonction standard. Dans la mesure où ces données sont transmises et retournées en langage XML, elles peuvent être facilement interprétées et manipulées par le client.

Par exemple, si vous utilisez Microsoft Visual Studio pour créer une application client, votre projet Visual Studio doit contenir une référence Web à, par exemple

`https://cloud.sagecrm.com/[accountname123]/eware.dll/webservices/CRMwebservice.wsdl`

Lorsque vous ajoutez la référence dans Visual Studio, le volet principal répertorie les méthodes disponibles à partir du service Web.

Si vous nommez le service **CRMWebServices** alors un dossier appelé CRMWebServices est ajouté à votre projet qui contient les fichiers webservice.discomap et webservice.wsdl. Le « proxy de service Web » version C# du fichier WSDL qui traite le transfert des données au format SOAP vers le service Web est créé automatiquement.



**Remarque** : Dans Visual Studio 2008, pour ajouter une référence Web, vous devez sélectionner **Ajouter une référence de service | Avancé | Ajouter une référence Web**.

# Chapitre 3 : Présentation des objets et fonctions

## Manipulation des enregistrements

Avant de commencer à travailler avec les services Web Sage CRM, vous devez être familiarisé avec toutes les fonctions que vous pouvez invoquer pour manipuler des enregistrements, ainsi que les objets exposés dans l'API sur lesquels les fonctions sont invoquées.

## Fonctions

Les fonctions sont des actions invoquées depuis l'ordinateur client pour effectuer certaines tâches (ajout, mise à jour ou suppression d'informations, etc.) sur le serveur. Les fonctions Sage CRM sont des requêtes synchrones effectuées automatiquement. Une fois exécutés, les services Web Sage CRM traitent la requête et retournent une réponse. L'application client traite ensuite la réponse en conséquence.



**Remarque** : généralement, les insertions doivent être effectuées par entité. Vous pouvez toutefois mettre à jour une société (ou personne) en même temps que les informations d'adresse, de téléphone et d'e-mail. Cela vise à faciliter l'intégration. Dans de nombreux systèmes, un enregistrement de contact représente les informations d'une société, d'une personne, de téléphone, d'e-mail et d'adresse.

Reportez-vous à la rubrique [Liste des fonctions des services Web \(page 4-1\)](#) pour une liste complète.

## Objets

Les objets sont des représentations obtenues par programmation des données dans le système. Dans Sage CRM, les objets représentent des entités principales telles que des sociétés et des personnes, ainsi que des entités secondaires telles que des adresses et des produits. Les données sont manipulées quand l'API des services Web interagit avec les propriétés d'objet, qui représente des champs dans les entités.

Reportez-vous à la section [Liste des objets des services Web \(page 5-1\)](#) pour une liste complète. Voir aussi [L'objet RecordType Sage CRM \(page 6-1\)](#) et [Champs de sélection dans les services Web \(page 7-1\)](#).



# Chapitre 4 : Liste des fonctions des services Web

Tous les objets exposés suivants sont définis dans le fichier WSDL.

Fonction	Description
logon	Se connecte au serveur et commence une session.
logoff	Se déconnecte du serveur et termine la session.
query	<p>Exécute une requête sur un objet spécifié selon une clause Where et retourne un enregistrement ou un ensemble d'enregistrements correspondant à la requête.</p> <p>Renvoie des résultats par lots (dont la taille est définie par le champ <b>Nombre maximum d'enregistrements à retourner à Administration   Système   Services Web</b>).</p> <p>Chaque lot est associé à un indicateur appelé « More » Si More est <b>Vrai</b>, alors il existe d'autres enregistrements en attente sur le serveur pour cette requête. Appelez la fonction <b>Next</b> pour accéder au lot de données suivant. Si une autre fonction est appelée, la requête est fermée.</p>
next	Retourne le lot suivant d'enregistrements correspondant à une requête. Chaque lot est associé à un indicateur appelé « More » Tant que More est <b>Vrai</b> , vous pouvez continuer à appeler la fonction <b>Next</b> jusqu'à ce que tous les lots soient retournés (c'est-à-dire jusqu'à ce que More soit <b>Faux</b> ).
queryentity	Retourne un enregistrement si vous indiquez un objet (par exemple, Company) et un code. Par exemple, queryentity (company, 42).
queryid	Retourne un objet de type <i>aisid</i> (reportez-vous à la rubrique <a href="#">Liste des objets des services Web (page 5-1)</a> ). Interroge la base de données à l'aide d'une clause Where. Une date et un nombre de codes sont retournés avec une série d'indicateurs sur chacun pour indiquer si l'enregistrement a été créé, mis à jour ou supprimé depuis cette date. Cette fonction s'avère utile pour la synchronisation des données.
queryidnodate	Retourne un objet de type <i>aisid</i> (reportez-vous à la rubrique <a href="#">Liste des objets des services Web (page 5-1)</a> ). Interroge la base de données à l'aide d'une clause Where. Cette fonction s'avère utile lorsque vous avez besoin, par exemple, d'un

Fonction	Description
	ensemble de codes de société sans les données associées.
getmetadata	Lorsque vous indiquez un nom de table, cette fonction retourne la liste des types de champ Sage CRM pour obtenir les métadonnées (par exemple, fieldname, type) relatives à la table faisant l'objet de la requête.
getdropdownvalues	Lorsque vous indiquez une table, cette fonction retourne la liste des champs déroulants de cette table et la liste des valeurs attendues par Sage CRM pour ce champ. Il s'agit d'un aspect important car Sage CRM attend un ensemble donné de valeurs pour les champs déroulants. Vous devez donc être en mesure d'obtenir ces valeurs par programmation.
add	Ajoute des enregistrements ou des listes d'enregistrements à un objet spécifié (par exemple, Company). Par exemple, add ("company", NewCompany1, New Company2, New Company3).
addresource	Ajoute un utilisateur comme ressource. Cet utilisateur n'est pas totalement activé. Cette fonctionnalité sert uniquement à faciliter la migration des données.
update	Met à jour des enregistrements ou des listes d'enregistrements pour un objet spécifié (par exemple, Company).
altercolumnwidth	Permet de redimensionner la largeur d'une colonne pour assurer la compatibilité avec les bases de données tierces (par exemple ACT!).
delete	Supprime des enregistrements ou des listes d'enregistrements pour un objet spécifié (par exemple, Company). Notez que vous ne pouvez pas supprimer des enregistrements des tables suivantes car elles contiennent des données historiques : newproduct, uomfamily, productfamily, pricing, pricinglist.
addrecord	Identique à la fonction add sauf qu'elle dispose d'une signature différente et utilise les listes de champs dans le type crmrecord. Reportez-vous à la rubrique <a href="#">L'objet RecordType Sage CRM (page 6-1)</a> .
queryrecord	Identique à la fonction query sauf qu'elle dispose d'une signature différente et utilise les listes de champs dans le type crmrecord. Reportez-vous à la rubrique <a href="#">L'objet RecordType Sage CRM (page 6-1)</a> .

Fonction	Description
nextqueryrecord	Retourne le lot suivant d'enregistrements correspondant à une requête queryrecord. Chaque lot est associé à un indicateur appelé « More » Tant que More est <b>Vrai</b> , vous pouvez continuer à appeler la fonction <b>Next</b> jusqu'à ce que tous les lots soient retournés (c'est-à-dire jusqu'à ce que More soit <b>Faux</b> ).
updaterecord	Identique à la fonction update sauf qu'elle dispose d'une signature différente et utilise les listes de champs dans le type cmrecord. Reportez-vous à la rubrique <a href="#">L'objet RecordType Sage CRM (page 6-1)</a> .
getallmetadata	Retourne une liste de champs associés aux tables avec des informations relatives au type.
getversionstring	Retourne la version de Sage CRM.



# Chapitre 5 : Liste des objets des services Web

Les objets suivants sont représentatifs des entités Sage CRM (principales et secondaires). Si des entités personnalisées sont ajoutées au système Sage CRM, ces entités sont également disponibles. Due to the fact that the WSDL is generated dynamically, any customizations made to the system—such as adding a new entity—are picked up each time the WSDL is refreshed at the client side.

## Objets abstraits

Nom de l'objet	Description
ewarebase abstract	Déclaration abstraite héritée par tous les autres objets Sage CRM.
ibase abstract	Déclaration abstraite héritée par tous les types de code.
ewarebaselist	Représente la liste des objets abstraits ci-dessus.
crmrecordtype	Énumération représentant les types d'un champ Sage CRM, c'est-à-dire chaîne, date/heure, entier, décimal.  La valeur multiselectfield indique un ensemble imbriqué de chaînes représentant les valeurs d'un champ à sélection multiple. La dernière option est <code>crmrecord</code> . Ceci indique un type de champ contenant d'autres champs. Pour plus d'informations, reportez-vous à <a href="#">L'objet RecordType Sage CRM (page 6-1)</a> .
crmrecord	Contient un nom d'entité et une liste d'objets de type <code>recordfield</code> représentant un enregistrement dans la base de données Sage CRM.
aisid	Contient le code de l'enregistrement, la date de création et de mise à jour et un indicateur de l'ajout, la mise à jour ou la suppression de l'enregistrement depuis que le jeton a été transmis à <code>queryid</code> .
multiselectfield	Ce type représente un champ à sélection multiple de Sage CRM. Il contient un nom de champ et un ensemble de chaînes représentant les valeurs du champ dans Sage CRM. Comme pour les autres champs, ces valeurs sont des traductions.
recordfield	Représente un champ dans un enregistrement de base de données. Il dispose d'une valeur de nom et d'un type de <code>crmrecordtype</code> . Il peut également représenter une structure imbriquée. Par exemple, le nom du champ <code>recordfield</code> dans l'enregistrement <code>crmrecord</code> d'une société peut être <code>Contact</code> . The type would be <code>crmrecord</code> and the

Nom de l'objet	Description
	record property would contain a list of crmrecords – one for each person in the company.

### Objets standard

Nom de l'objet	Description
company	Représente l'entité Société dans Sage CRM.
person	Représente l'entité Contact dans Sage CRM.
lead	Représente l'entité Prospect dans Sage CRM.
communication	Représente l'entité Communication dans Sage CRM.
opportunity	Représente l'entité Opportunité dans Sage CRM.
cases	Représente l'entité Tickets dans Sage CRM.
users	Représente l'entité Utilisateurs dans Sage CRM.
quotes	Représente l'entité Devis dans Sage CRM.
orders	Représente l'entité Commandes dans Sage CRM.
quoteitem	Représente l'entité Quoteltems dans Sage CRM.
orderitem	Représente l'entité lineitems des commandes dans Sage CRM.
opportunityitem	Représente l'entité Elément d'opportunité dans Sage CRM.
currency	Représente l'entité Devise dans Sage CRM.
address	Représente l'entité Adresse dans Sage CRM.
phone	Représente l'entité Téléphone dans Sage CRM.
email	Représente l'entité E-mail dans Sage CRM.
newproduct	Représente l'entité Nouveau produit dans Sage CRM.
uom	Représente l'entité Unité de mesure dans Sage CRM.
uomfamily	Représente l'entité Fourchette d'unités de mesure dans Sage CRM.
pricing	Représente l'entité Prix dans Sage CRM.
pricinglist	Représente l'entité Liste de prix dans Sage CRM.
productfamily	Représente l'entité Famille de produits dans Sage CRM.

### Insertion et mise à jour des articles de devis et de commande

Lors de l'insertion et la mise à jour des champs pour les articles de devis et de commande, notez que différents types articles de la gamme imposent certains champs. Le service Web crée une exception s'ils sont introuvables.

Lors de l'insertion d'un article de gamme standard, les champs suivants sont obligatoires :

- orderquoteid
- opportunityid
- lineitemtype (peut être 'i' pour un élément simple, 'f' pour un élément de texte libre ou 'c' pour une ligne de commentaire)
- productid
- uomid
- quantity
- quotedprice

Lors de l'insertion d'un article de gamme en texte libre, les champs suivants sont obligatoires :

- orderquoteid
- opportunityid
- lineitemtype ('i', 'f' ou 'c')
- description
- quantity
- quotedprice

Lors de l'insertion d'un commentaire de ligne article, les champs suivants sont obligatoires :

- orderquoteid
- opportunityid
- lineitemtype ('i', 'f' ou 'c')
- description

Lors de la mise à jour d'une ligne article standard, les champs suivants imposent une valeur :

- quantity
- quotedprice
- uomid

Lors de la mise à jour d'une ligne article en texte libre, les champs suivants imposent une valeur :

- description
- quantity

Lors de la mise à jour d'un commentaire de ligne article standard, les champs suivants imposent une valeur :

- description

Les deux champs suivants ne peuvent pas être mis à jour, et créent une exception :

- linetype
- orderquoteid

En outre, certains champs sont calculés ou remplacés par Sage CRM dans le code de service Web. Les valeurs que l'utilisateur leur transmet seront ignorées. Ces champs sont les suivants :

- quotedpricetotal
- listprice
- discount
- discountsum

# Chapitre 6 : L'objet RecordType Sage CRM

L'objet `crmrecordtype` (et ses fonctions associées : `add`, `update` et `delete`) offre un environnement de programmation dynamique et flexible. Plutôt que d'interroger une entité (par exemple, une société) et de recevoir un objet fortement typé (`company`), la flexibilité de l'objet `crmrecordtype` permet d'interroger une entité et de recevoir une liste de champs dans laquelle vous pouvez effectuer une itération. Ainsi, il est possible de spécifier les champs que vous désirez recevoir dans votre requête.

La possibilité d'effectuer une itération dans les enregistrements offre aux programmeurs une interface puissante et flexible. Il permet l'ajout dynamique de champs aux entités des services Web, et supprime le besoin d'objets fortement typés dans les applications client. Les exemples de code doivent être fidèlement suivis lors de l'exécution de ces tâches.

L'exemple de requête suivant spécifie une liste de champs et un nom d'entité, une clause `Where` et un tri par. Si vous entrez un astérisque (\*) ou laissez la liste de champs vide, vous obtiendrez tous les champs.

```
Private static void CallQueryRecordOnCompanyEntity()
{
    String companyid = ReadUserInput("Veuillez entrer un nom de société : ");
    Queryrecordresult arestult = Binding.queryrecord("comp_companyid,address","comp_
name='compol'", "company", "comp_companyid");
}
```



# Chapitre 7 : Champs de sélection dans les services Web

Si des champs déroulants sont définis comme chaînes, ces champs n'apparaîtront pas dans le fichier WSDL. Comme les chaînes sont l'option par défaut, ces champs n'apparaîtront pas dans une configuration standard.

Les tableaux ci-dessous répertorient les champs de sélection Sage CRM. Dans le fichier WSDL, un type énuméré représente ces valeurs pour chaque champ contenant des valeurs. Il existe plusieurs champs tel que celui-ci pour chaque entité.



**Remarque** : Les valeurs énumérées sont retournées dans le langage système par défaut.

```
<s:simpleType name="case_problemtyp">
<s:restriction base="s:string">
<s:enumeration value="Logiciel supplémentaire requis" />
<s:enumeration value="Bug logiciel" />
<s:enumeration value="Configuration/Installation" />
<s:enumeration value="Connaissances du client" />
</s:restriction>
</s:simpleType>
```

## Liste des champs de sélection

### Champs de sélection Société

- comp\_employees
- comp\_indcode
- comp\_mailrestriction
- comp\_revenue
- comp\_sector
- comp\_source
- comp\_status
- comp\_territory
- comp\_type

### Champs de sélection Contact

- pers\_gender
- pers\_salutation
- pers\_source
- pers\_status

- pers\_territory
- pers\_titlecode

#### **Champs de sélection Prospect**

- lead\_decisiontimeframe
- lead\_priority
- lead\_rating
- lead\_source
- lead\_stage
- lead\_status

#### **Champs de sélection Communication**

- comm\_action
- comm\_hasattachments
- comm\_notifydelta
- comm\_outcome
- comm\_priority
- comm\_status
- comm\_type

#### **Champs de sélection Opportunité**

- oppo\_priority
- oppo\_product
- oppo\_scenario
- oppo\_source
- oppo\_stage
- oppo\_status
- oppo\_type

#### **Champs de sélection Ticket**

- case\_foundver
- case\_problemtyp
- case\_productarea
- case\_solutiontype
- case\_source
- case\_stage
- case\_status
- case\_targetver

## Champs de sélection Adresse et Produit

- addr\_country
- prod\_uomcategory

## Utilisation de GetDropDownValues

Utilisez la fonction *getdropdownvalues*. Reportez-vous à la section [Liste des fonctions des services Web \(page 4-1\)](#) pour obtenir la liste des champs déroulants dans un tableau et la liste des valeurs attendues par Sage CRM pour ces champs.

Ceci est un exemple dans C# d'une fonction permettant de renseigner une zone de liste déroulante avec des valeurs de sélection d'un champ donné.

```
private void LoadDropDowns(string entity, string fieldname, ComboBox controlname,
WebService WS)
{
    dropdownvalues[] DropDowns;
    DropDowns = WS.getdropdownvalues(entity);
    controlname.Items.Clear();
    for (int i = 0; i < DropDowns.Length; i++)
    {
        if (DropDowns[i].fieldname == fieldname)
        {
            for (int x = 0; x < DropDowns[i].records.Length; x++)
            {
                controlname.Items.Add(DropDowns[i].records[x].ToString());
            }
        }
    }
    controlname.SelectedIndex = 0;
}
```

Pour utiliser la fonction permettant d'afficher les valeurs de sélection comp\_sector dans une zone de liste déroulante appelée « comboSector » (où l'objet du service Web est appelé oWebService) :

```
LoadDropDowns("company", "sector", comboSector, oWebService);
```



# Chapitre 8 : Exemples de requêtes SOAP

Les sections suivantes fournissent un certain nombre d'exemples de requête Soap. Certains exemples de requête sont en langage C# et d'autres en langage XML.

## Récupération dynamique des détails WSDL et exemple de requête d'authentification

Vous pouvez utiliser le point de terminaison de GetCRMURL pour récupérer dynamiquement l'URL personnalisée pour le fichier WSDL et la chaîne de connexion du service Web pour un compte Sage CRM Cloud. Ce point de terminaison renverra également les URL personnalisées de l'API SDATA et de l'URL de Dossier Prospect Web sur le compte.

Pour récupérer ces renseignements, effectuez simplement une requête HTTP GET sur un des points de terminaison suivants :

- Point de terminaison du datacenter AN : <https://cloud.na.sagecrm.com/getcrmurls>
- Point de terminaison du datacenter UE : <https://cloud.eu.sagecrm.com/getcrmurls>

Vous devrez vous authentifier avec ce point de terminaison en envoyant des informations de connexion CRM valides aux en-têtes suivants. Vérifiez que l'accès aux Services Web des informations de connexion que vous utilisez est défini sur Vrai dans CRM.

Les sessions Sage CRM Cloud expirent après 30 minutes d'inactivité.

Une authentification réussie au point de terminaison de GetCRMURL démarrera votre système Sage CRM Cloud si celui-ci n'est pas encore en cours d'exécution sur la plateforme Cloud. Si votre instance Sage CRM n'est pas développée, le point de terminaison déclenche un déploiement et récupère votre Service Web ou votre URL SDATA pour toute installation Sage CRM Cloud, en s'authentifiant avec votre nom d'utilisateur et votre mot de passe Sage CRM Cloud. Vous ne pouvez pas utiliser ces points de terminaison dans votre navigateur, car vous devez transmettre votre nom d'utilisateur et votre mot de passe dans un en-tête d'autorisation.

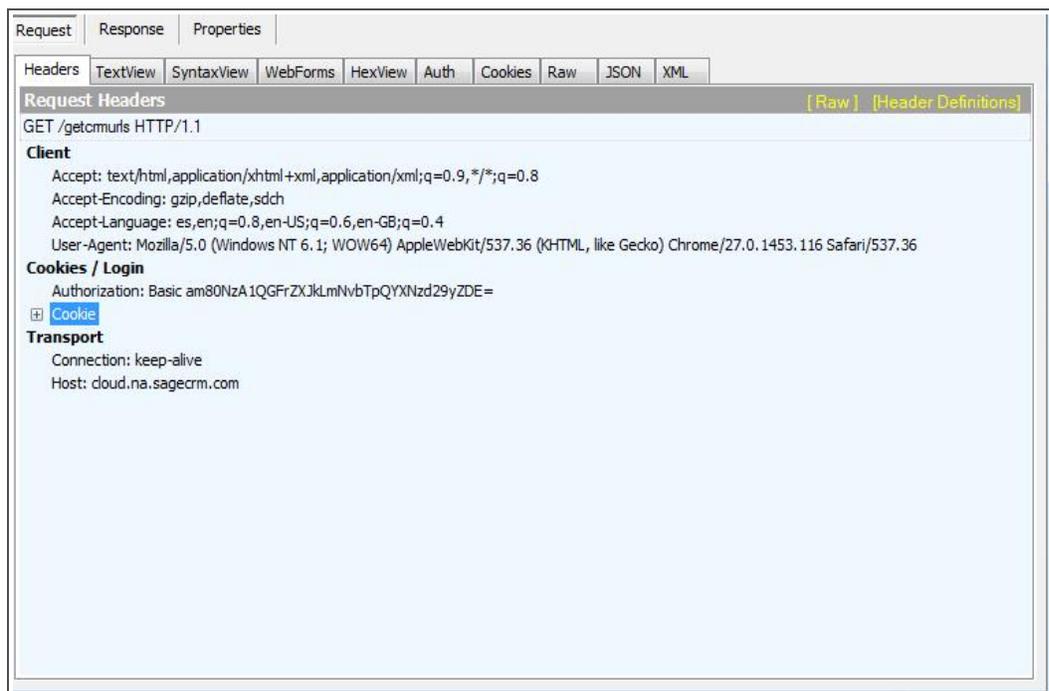
Nous recommandons aux développeurs d'utiliser cette API pour récupérer leur adresse WSDL des services Web pendant le processus de connexion afin de s'assurer que leur compte Sage CRM Cloud est prêt à accepter les requêtes de services Web.

Les éléments suivants expliquent comment effectuer une requête d'authentification :

### Remarque :

- Vous devez vous authentifier avec le point de terminaison à l'aide d'un en-tête d'autorisation de base.
- L'autorisation pour ce point de terminaison est fournie dans les en-têtes de requête.
- Votre nom d'utilisateur et votre mot de passe doivent être au format `#NomUtilisateur#:#MotdePasse#`.

- Le type d'autorisation est basique et doit être codé en base 64. Le format est le suivant :  
Authorization: Basic QWxhZGRpbjpvYVUyIHNlc2FtZQ==



En-tête de requête pour une requête d'authentification HTTP

Cet exemple de réponse est fourni après une authentification réussie pour le point de terminaison de GetCRMURL. La réponse est au format JSON :

```
{ "installName": "<myCustomerID>", "wsdlUrl": "https://cloud.na.sagecrm.com/<myCustomerID>/eware.dll/webservices/CRMwebservice.wsdl", "wsUrlConnection": "https://cloud.na.sagecrm.com/<myCustomerID>/eware.dll/webservices/", "web2LeadUrl": "https://cloud.na.sagecrm.com/<myCustomerID>/eware.dll/SubmitLead", "sDataUrl": "https://cloud.na.sagecrm.com/sdata/<myCustomerID>j/sagecrm/", "edition": "professional", "domain": "cloud.na.sagecrm.com", "userId": 1, "teamId": 1, "found": true, "userAuthenticated": true, "userWebServicesEnabled": true, "userDisabled": false }
```

**Remarque** : si votre compte Cloud se trouve dans le datacenter européen, vous devez utiliser le point de terminaison UE et, de la même façon, si votre compte se trouve dans notre datacenter d'Amérique du Nord, veuillez utiliser le point de terminaison AN. Si vous envoyez la requête vers un point de terminaison incorrect, elle sera quand même redirigée vers le datacenter approprié dans lequel se trouve votre compte Sage CRM Cloud. Cependant, utiliser le point de terminaison

approprié vous garantira l'obtention d'une réponse rapide de la part de la plateforme Sage CRM et permettra à votre application de fonctionner correctement.

## Exemple de requête Soap de connexion

Lorsque vous créez une application qui utilise l'API des services Web Sage CRM Cloud, il est important de régler le paramètre « PermettreRedirectAuto » sur Vrai afin de s'assurer que votre application peut gérer les redirections HTTP internes qui pourraient avoir lieu sur la plateforme Sage CRM Cloud.

Cet exemple de code C# illustre comment se connecter au serveur :

```
//Une Instance du service Web.
private static WebService binding = null;
//Persistant pour la durée du programme, conserver les résultats de connexion
private static logonresult SID = null;
private static void LogonToCRMSystem()
{
    try
    {
        HttpWebRequest request = (HttpWebRequest)
        WebRequest.Create
        ("http://cloud.sagecrm.com/myCustomerID/eware.dll/webservices/CRMwebservice.wsd
        1");
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();

        //redirection automatique des demandes de connexion aux services Web vers
        l'instance Sage CRM des clients
        binding.AllowAutoRedirect = true;
        SID = binding.logon("admin", "");
        binding.SessionHeaderValue = new SessionHeader();
        binding.SessionHeaderValue.sessionId = SID.sessionid; //SID persistant
        return true;
    }
    catch (SoapException e)
    {
        Write(e.Message);
    }
    catch (Exception e)
    {
        Write(e.Message + "\n" + e.StackTrace);
    }
}
```

Voici la requête XML traitée par les services Web :

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
        <logon xmlns="http://tempuri.org/type">
            <username>admin</username>
            <password />
        </logon>
    </soap:Body>
</soap:Envelope>
```

### Exemple de requête Soap de déconnexion

Cet exemple XML illustre comment se déconnecter :

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <SessionHeader xmlns="http://tempuri.org/type">
      <sessionId>57240080053832</sessionId>
    </SessionHeader>
  </soap:Header>
  <soap:Body>
    <logoff xmlns="http://tempuri.org/type">
      <sessionId>57240080053832</sessionId>
    </logoff>
  </soap:Body>
</soap:Envelope>
```

### Exemple de requête Soap de suppression

Cet exemple C# montre comment supprimer une société dont le code est 66 :

```
ewarebase[] idList = new ewarebase[1];
companyid aCompanyId = new companyid();
aCompanyId.companyid1 = 66; //66 est le code de la société à supprimer
idList[0] = aCompanyId;
deleteresult aResult = binding.delete("company",idList);

if(aResult.deletesuccess == true)
    Console.WriteLine("Numéro supprimé avec succès : " + aResult.numberdeleted);
```

Voici la requête XML traitée par les services Web :

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <SessionHeader xmlns="http://tempuri.org/type">
      <sessionId>127169567253830</sessionId>
    </SessionHeader>
  </soap:Header>
  <soap:Body>
    <delete xmlns="http://tempuri.org/type">
      <entityname>société</entityname>
      <records xsi:type="companyid">
        <companyid>66</companyid>
      </records>
    </delete>
  </soap:Body>
</soap:Envelope>
```

### Exemple de requête Soap de mise à jour

Cet exemple C# montre comment modifier le nom d'une société dont le code est 66 :

```
private static void UpdateACompany()
{
    String idString = "66";
    String newName = "nouveau nom";
    ewarehouse[] companyList = new ewarehouse[1]; //peut mettre à jour un certain nombre
de sociétés
    company aCompany = new company();
    aCompany.companyid = Convert.ToInt16(idString);
    aCompany.companyidSpecified = true;

    aCompany.name = newName;
    companyList[0] = aCompany;
    updateresult aresult = binding.update("company", companyList);

    if(aresult.updatesuccess == true)
    {}
else
    {}
}
```

Voici la requête XML traitée par les services Web :

```
<?xml version="1.0" encoding="utf-8" ?>
  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Header>
  <SessionHeader xmlns="http://tempuri.org/type">
    <sessionId>12663708753831</sessionId>
  </SessionHeader>
</soap:Header>
<soap:Body>
  <update xmlns="http://tempuri.org/type">
    <entityname>société</entityname>
    <records xsi:type="company">
      <people xsi:nil="true" />
      <address xsi:nil="true" />
      <email xsi:nil="true" />
      <phone xsi:nil="true" />
      <companyid>933</companyid>
      <name>Design Wrong Inc</name>
    </records>
  </update>
</soap:Body>
</soap:Envelope>
```

## Exemple de requête Soap pour QueryEntity

Cet exemple interroge un enregistrement de société dont le code est 66 :

```
company aCompany = (company) binding.queryentity( 66, "company").records;
```

## Exemple Soap XML représentant une société

Le code XML suivant représente une société dont le code est 65 :

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```

    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<queryentityresponse xmlns="http://tempuri.org/type">
<result>
<records xsi:type="typens:company" xmlns:typens="http://tempuri.org/type">
    <typens:companyid>65</typens:companyid>
    <typens:primarypersonid>79</typens:primarypersonid>
    <typens:primaryaddressid>77</typens:primaryaddressid>
    <typens:primaryuserid>9</typens:primaryuserid>
    <typens:name>AFN Interactive</typens:name>
    <typens:website>http://www.AFNInteractive.co.uk</typens:website>
    <typens:createdby>1</typens:createdby>
    <typens:createddate>2004-08-30T18:10:00</typens:createddate>
    <typens:updatedby>1</typens:updatedby>
    <typens:updateddate>2004-08-30T18:10:00</typens:updateddate>
    <typens:timestamp>2004-08-30T18:10:00</typens:timestamp>
    <typens:librarydir>A\AFN Interactive (65)</typens:librarydir>
    <typens:secterr>-1845493753</typens:secterr>
    <email>
        <entityname>e-mail</entityname>
        <records xsi:type="typens:email" xmlns:typens="http://tempuri.org/type">
            <typens:emailid>120</typens:emailid>
            <typens:companyid>65</typens:companyid>
            <typens:type>Sales</typens:type>
            <typens:emailaddress>sales@AFNInteractive.co.uk</typens:emailaddress>
            <typens:createdby>1</typens:createdby>
            <typens:createddate>2004-08-30T18:10:00</typens:createddate>
            <typens:updatedby>1</typens:updatedby>
            <typens:updateddate>2004-08-30T18:10:00</typens:updateddate>
            <typens:timestamp>2004-08-30T18:10:00</typens:timestamp>
        </records>
        </email>
    <phone>
        <entityname>téléphone</entityname>
        <records xsi:type="typens:phone" xmlns:typens="http://tempuri.org/type">
            <typens:phoneid>211</typens:phoneid>
            <typens:companyid>65</typens:companyid>
            <typens:type>Bureau</typens:type>
            <typens:countrycode>44</typens:countrycode>
            <typens:areacode>208</typens:areacode>
            <typens:number>848 1051</typens:number>
            <typens:createdby>1</typens:createdby>
            <typens:createddate>2004-08-30T18:10:00</typens:createddate>
            <typens:updatedby>1</typens:updatedby>
            <typens:updateddate>2004-08-30T18:10:00</typens:updateddate>
            <typens:timestamp>2004-08-30T18:10:00</typens:timestamp>
        </records>
        </phone>
    <address>
        <entityname>adresse</entityname>
        <records xsi:type="typens:address" xmlns:typens="http://tempuri.org/type">
            <typens:addressid>77</typens:addressid>
            <typens:address1>Greenside House</typens:address1>
            <typens:address2>50 Station Road</typens:address2>
            <typens:address3>Wood Grn</typens:address3>
            <typens:city>LONDRES</typens:city>
            <typens:postcode>N22 7TP</typens:postcode>
            <typens:createdby>1</typens:createdby>
            <typens:createddate>2004-08-30T18:10:00</typens:createddate>
        </records>
    </address>
</result>
</queryentityresponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```
        <typens:updatedby>1</typens:updatedby>
        <typens:updateddate>2004-08-30T18:10:00</typens:updateddate>
        <typens:timestamp>2004-08-30T18:10:00</typens:timestamp>
    </records>
</address>
</records>
</result>
</queryentityresponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Code version du document CRM : CLO-WES-FRA-161-1.0