CRM.es

Funciones Webservice

Introducción

Este documento explica las diferentes funciones disponibles en el webservice del crm. El webservice del crm funciona a través de peticiones POST y GET (según el tipo de operación que se realice). Cada petición tiene la siguiente estructura.

https://url_crm/webservice.php?operation=nombre_operacion&parámetro1=valor1&parámetro2=valor2...

Url_crm: Es el dominio del crm

Nombre_operacion = Es el nombre de la función.

Parámetros = Parámetros de la función.

Login Webservice

Para utilizar el webservice del crm es necesario iniciar sesión para ello debemos realizar dos acciones getchallenge y login. La primera consiste en obtener un token y la segunda en iniciar sesión utilizando el token obtenido anteriormente.

Getchallenge

Esta función obtiene un token del crm.

Función: getchallenge

Tipo de petición: GET

Parámetros:

- username: El nombre del usuario del crm

Parámetros de ejemplo:

array('username' => 'demo');

Login

Iniciar sesión en el crm utilizando el token obtenido por la función getchallenge. Esta devolverá un **id de sesión** que lo utilizaremos como parámetro en todas las funciones descritas a continuación.

Función: login

Tipo de petición: POST

Parámetros:

- username: El nombre del usuario del crm

- accessKey: Concatenar el token y la clave de acceso del usuario (se encuentra en la ficha del usuario del crm en el bloque de opciones avanzadas de usuario) y aplicar la encriptación md5.

Parámetros de ejemplo:

array('username' => 'demo' , 'accessKey' => 'ad438ff7af820261ec54505');

Nota: Las sesiones duran 5 horas. Si tiene un proceso que tarde más debe extender la sesión o realizar un nuevo login.

Funciones básicas

El webservice del crm dispone de las siguientes funciones básicas:

- Create: Sirve para crear un registro nuevo.
- Retrieve: Sirve para un registro existente.
- Update: Sirve para actualizar un registro entero.
- Revise: Sirve para actualizar campos específicos de un registro.
- Delete: Sirve para eliminar un registro del crm.

Todas estas acciones se rigen por los permisos configurados en el crm. Por ejemplo, si utilizan un usuario que no puede eliminar cuentas a pesar podrá utilizar la función delete pero esta no eliminará si no que devolverá un error.

Create

Esta función sirve para crear un registro en el crm.

Función: create

Tipo de petición: GET

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- element: Los campos del elemento a crear. Los campos obligatorios deben estar informados.
- elementType: El nombre del módulo del registro.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'* , 'element': {'lastname' => 'demo', 'assigned_user_id' => '19x1', 'account_id' => '3x12'}, 'elementType' => 'Contacts');

Retrieve

Esta función sirve para recuperar un registro existente del crm.

Función: retrieve

Tipo de petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- id: El id del registro. El id del registro es modulo_idxregistro_id. Por ejemplo, una cuenta con id del crm 12 y el id del módulo de Cuentas es 3 sería 3x12.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'* ,'id' => '3x12');

Nota: Para conocer el id del crm de un registro es muy sencillo simplemente hay que acceder al crm, abrir la vista de detalle de un registro y ver el valor del parámetro record de la url. En cambio, para conocer el id del módulo hay que realizar una select a la table vtiger_ws_entity. Esta tabla tiene la columna id que contiene el valor del id del módulo y la columna name que contiene el nombre del módulo.

Update

Esta función sirve para actualizar un registro entero.

Función: update

Tipo de petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- element: Los campos del elemento a actualizar. Los campos obligatorios deben estar informados. Además de los campos hay que añadir el id del registro a actualizar.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'*, 'element' : {'id'=>'12x20','lastname' => 'demo', 'assigned_user_id' => '19x1', 'account_id' => '11x12'});

Revise

Esta función es similar a Update, pero permite actualizar campos específicos de un registro.

Función: revise

Tipo de petición: POST

Parámetros

- sessionName: Es el id de la sesión obtenida por la función login.
- element: Los campos a actualizar y el id del registro a actualizar.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'*, ´element´ : {'id' => '12x20' , 'lastname' => 'Nueva demo'});

Delete

Esta función elimina un registro del crm.

Función: delete

Tipo de petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- id: Id del registro a eliminar.

Parámetros de ejemplo:

- array('*sessionName' => '39da795658d89dc19b408'*, 'id' => '3x12');

Funciones para elementos relacionados (sólo relación EntityRel)

Las siguientes funciones se refieren a acciones permitidas en el Webservice sobre relaciones llamadas EntityRel. Las EntityRel son relaciones que no dependen de un campo en particular.

Por ejemplo, un Contacto tiene un campo propio llamado "Cuenta" y que permite enlazar ese Contacto a una Cuenta en particular. Ese Contacto no se podrá enlazar a más de una Cuenta. Este tipo de relaciones no son EntityRel.

Sin embargo, para asociar una Cuenta y un Producto no existe ningún campo llamado "Cuenta" en los Productos. Un mismo Producto se puede enlazar a N Cuentas. Este tipo de relaciones son las conocidas como EntityRel.

Las siguientes funciones no realizarán ninguna acción si los parámetros enviados no se refieren a módulos relacionados mediante relaciones de tipo EntityRel.

Created_related

Permite enlazar 2 entidades. Por ejemplo, permite coger un Producto y enlazarlo a una Cuenta.

Función: create_related

Tipo de Petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- module: Nombre de Entidad del módulo al que le queremos enlazar un registro.
- record: Identificador del registro al que le queremos enlazar otro registro
- relmodule: Nombre de la Entidad que queremos enlazar a *module*
- relrecord: Identificador del registro que quieres enlazar a *record*

Parámetros de ejemplo:

array('sessionName' => '39da795658d89dc19b408', 'module' => 'Accounts, 'record' => 1772, 'relmodule' => 'Products', 'relrecord' => 1665);

Retrieve_related

Permite consultar qué entidades están relacionadas con una entidad. Por ejemplo, permite consultar qué Productos están asociados a una Cuenta.

Función: retrieve_related

Tipo de Petición: GET

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- module: Nombre de la Entidad del módulo del que se quieren obtener sus relaciones
- record: Identificador del registro del que queremos obtener la información
- relmodule: Nombre de la Entidad a la que se refiere los registros asociados que queremos obtener

Parámetros de ejemplo:

array('sessionName' => '39da795658d89dc19b408', 'module' => 'Accounts', 'record' => 1772, 'relmodule' => 'Products');

Delete_related

Permite eliminar la relación entre 2 entidades.

Función: delete_related

Tipo de Petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- module: Entidad principal a la que le queremos eliminar un registro asociado (no el registro en sí mismo, sino la relación)
- record: Identificador del registro principal
- relmodule: Entidad que vamos a querer desenlazar de la Entidad principal
- relrecord: Identificar del registro que queremos desenlazar

Parámetros de ejemplo:

array('sessionName'=>'39da795658d89dc19b408', 'module'=>'Accounts', 'record'=>1772, 'relmodule' => 'Products', 'relrecord' => 1665);

Funciones Adicionales

El webservice del crm a parte de las funciones descritas anteriormente dispone de otras funciones.

Logout

Salir de la sesión del webservice.

Función: logout

Tipo de Petición: POST

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.

Describe

Esta función se utiliza para obtener una descripcíon genérica de un tipo de módulo. Por ejemplo, queremos saber que campos contiene el módulo de oportunidades y si podemos crear una oportunidad. Con el objeto resultante de esta función lo podemos saber.

Función: describe

Tipo de petición: GET

Parámetros:

- sessionName: Es el id de la sesión obtenida por la función login.
- elementType: El nombre del módulo que queremos obtener

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'*, 'elementType' => 'Potentials');

Extendsession

Extender la sesión del webservice.

Función: extendsession

Tipo de petición: POST

Parámetros:

- username: El nombre del usuario.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'*, 'username' => 'demo');

Query

Esta función sirve para hacer una *select* a la base de datos.

Función: query

Tipo de petición: GET

Parámetros:

- sessionName: Es el id del sesión obtenida por la función login.
- query: La consulta a la base de datos que queremos realizar.

Parámetros de ejemplo:

array('*sessionName' => '39da795658d89dc19b408'*, 'query' => "SELECT accountname, id from accounts WHERE accountname = 'Prueba'");

Las consultas están limitadas a un único módulo y no los *Joins* no funcionan. Todas las consultas pueden obtener un máximo de 200 registros y la consulta puede utilizar el operador limit para obtener diferentes registros.

Estructura de una consulta:

*SELECT * | <lista de columnas > | <count(*)>*

FROM <Nombre del módulo>

WHERE <condiciones>

ORDER BY <columna a ordenar> LIMIT <m>,<n>