berliCRM

REST Webservices API Reference Manual

Version 1.5.1.2
Rev. from 20.05.2016

Table Of Contents

Overview

This documentation describes the REST based application programming interface (the API) published used at the berliCRM system. It is recommended that you go through the tutorial provided in a different document for a better understanding of this documentation.

Each data set in the CRM is related to special formatted id which can be used to look up an entity's type and record. 'Create' is the only case that does not require an id.

All structural data including response from the API are represented as JSON strings.

URL Format

All REST based web service calls are http calls with the following format:

https://your_url/webservice.php?operation=[operation type]&sessionName=[session Name]&[operation specific parameters]

operation: Currently the following REST operations are supported,

- getchallenge
- login
- create
- retrieve
- update
- delete
- sync
- query
- listtypes
- describe
- logout
- extendsession

sessionName: The key that is used to identify the current session and is unique for every session. This must be part of every request.

The Response Format

Each REST call will get a response from the CRM system. All responses of the CRM have the following format.

Sucessfully processed:

```
Response{
success:Boolean=true
result:Object //the Result object
}
```

If an error occurred while processing the request the following message will be presented:

```
Response{
success:Boolean=false
error:ErrorObject
}
ErrorObject{
code:String //String representation of the error type
message:String //Error message from the api.
}
```

The error code is a string representation of the error type.

VtigerObject

A Map representing the contents of a crmentity based object. All reference fields are represented by using an Id type. A key called id of type Id represents the object's unique id. This field is present for any object fetched from the CRM's database.

Id Format

objectTypeId 'x' objectId

objectTypeId - Id of the object type. This is generated uniquely for each entity supported by the web service API and returned as the result of describe operation as idPrefix.

objectId - id of object in database. This is a unique id for all objects of the given entity.

Map

A map is an associative array of key value pairs. This is usually used for create operations.

TimeStamp

A long representation of the number of seconds since unix epoch. See https://en.wikipedia.org/wiki/Unix_time

GetChallengeResult

An object representing the response of a getchallenge operation,

```
GetChallengeResult{
token:String //Challenge token from the server.
serverTime:TimeStamp //The current server time.
expireTime:TimeStamp //The time when the token expires.
}
```

LoginResult

An object representing the response of a login operation,

LoginResult{
sessionName:String //Unique Identifier for the session
userId:String //The CRM id for the logged in user
version:String //The version of the webservices api
vtigerVersion:String //The version of the CRM system.
}

SyncResult

An object representing the response of a sync operation,

SyncResult{
updated:[Object]   //List of Objects created or modified.
deleted:[Id]   //List of *Id* of objects deleted.
lastModifiedTime:Timstamp //time of the latest change. This can be used in the next call to the Sync api to get all the latest changes that the client hasn't obtained.
}

Operations

The following describes all REST operations in detail.

Get Challenge

Get a challenge token from the server. This operation should always be executed before any login.

getchallenge(username:String):GetChallengeResult
Request Type: GET
username: A CRM username.
response: A GetChallengeResult object containing the server's challenge token.

This has to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=getchallenge&username=[username]

Login

Login to the server using the challenge token obtained in get challenge operation.

login(username:String, accessKey:String):LoginResult
Request Type:POST

username: A CRM username.
accessKey: An md5 of the concatenation of the challenge token and the user's webservice access key.
response: A LoginResult object containing the session id, the api version id and the user id.

This has to be a POST request.

Required URL format:
https://your_url/webservice.php?operation=login&username=[username]&accessKey=[accessKey]

Note: In accessKey 'K' is uppercase.

Create

Create a new data entry at the CRM server.

create(element:Map, elementType:String): VtigerObject
Request Type: POST
element: Fields of the object to populate. Values for mandatory fields must be provided.
elementType: The class name of the object.

This hast to be a POST request.

Required URL format:
https://your_url/webservice.php?operation=create&sessionName=[session id]&element=[object]&elementType=[object type]

response: A VtigerObject instance representing the new object.

Retrieve

Retrieve an existing entry from the server.

retrieve(id: Id): VtigerObject
Request Type: GET
id: The Id of the object.
response: A VtigerObject instance representing the retrieved object.

This hast to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=retrieve&session_name=[session id]&id=[object id]

Update

Update an existing entry on the CRM object.

update(object: VtigerObject): VtigerObject
Request Type: POST
object: The VtigerObject to update.
response:A VtigerObject representing the object after the update.
This must be a POST request.

Required URL format:
https://your_url/webservice.php?operation=update&sessionName=[session id]&element=[object]

Delete

Delete an entry from the server.

delete(id:Id):Nothing
Request Type: POST
id: The Id of the object to be deleted.
response: A map with one key status with value 'successful'

This has to be a POST request.

Required URL format:
https://your_url/webservice.php?operation=delete&sessionName=[session id]&id=[object id]

Query

The query operation provides a way to query the CRM for data.

query(queryString : String): [VtigerObject]
Request Type: GET
queryString:The query to process.
response: A list of Map containing the fields selected.

Queries are currently limited to a single object. Joins are not supported. Query always limits its output to 100 records. A client application may use limit operator to get different records.

The query format

select * | <column_list> | <count(*)>
from <object> [where <conditionals>]
[order by <column_list>] [limit [<m>,]<n>];
The column list in the 'order by' clause can have at most two column names.

- column_list: comma separated list of field names.
- object: type name of the object.

- conditionals: condition operations or in clauses or like clauses separated by 'and' or 'or' operators these are processed from left to right. There is no grouping supported which is done width bracket operators.
- conditional operators: <, >, <=, >=, =, !=.
- in clauses: in ().
- like clauses: like 'sqlregex'.
- value list: a comma separated list of values.
- m, n: integer values to specify the offset and limit respectively.

This has to be a GET request.

Query operation is currently supported for Entity modules only.

Required URL format:

https://your_url/webservice.php?operation=query&sessionName=[session id]&query=[query string]

Sync

Sync will return a SyncResult object containing details of changes after modifiedTime.

sync(modifiedTime: Timestamp, elementType: String):SyncResult

Request Type: GET

modifiedTime: The time of the last synced modification.
elementType: This is an optional parameter, if specified the changes for that module after the given time otherwise changes to all user accessible module are returned.

Returns a SyncResult object representing the sync data.

This has to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=sync&sessionName=[session id]&modifiedTime=[timestamp]&elementType=[elementType]

Logout

Logout from the webservices session, this leaves the webservice session invalid for further use.

logout(): Map
Request Type: GET
Returns a map containing the key 'message' with the value 'successful'.

This has to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=logout&sessionName=[session id]

List Types

List the names of all the CRM objects available through the API.

listtypes(): Map
Request Type: GET
Returns a map containing the key 'types' with the value being a list of names of CRM objects.

This has to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=listtypes&sessionName=[session id]

Describe

Get the type information about a given CRM object.

describe(elementType: String): DescribeResult
Request Type: GET
elementType: The type name of the CRM object to describe.
Returns a DescribeResult instance.

This has to be a GET request.

Required URL format:
https://your_url/webservice.php?operation=describe&sessionName=[session id]&elementType=[elementType]

Extend Session

Extends current CRM web-session to webservices and returns a webservices session id.

extendsession(username:String):LoginResult
Request Type: POST
username: A CRM username.

This has to be a POST request.

Required URL format:
https://your_url/webservice.php?operation=extendsession

CRM Entities

List of possible CRM entities exposed by the API.

Name	Description
Calendar	The Calendar Module manages Tasks and Meetings
Emails	The Emails module is an email client used to manage your emails.
Leads	The Leads module is used to track Sales leads.
HelpDesk	The HelpDesk module is used to track customer issues such as feedback, problems etc.
Accounts	The Accounts module is used to manage individual or organizations involved in your business.
Faq	The FAQ module is used to manage the frequently asked question by your customer.
Contacts	The Contacts module is used to manage individuals, who may be associated with an Account.
Vendors	The Vendors Module is used for managing manufacturers.
Potentials	Potential module is used to manage sales Opportunities.
PriceBooks	The PriceBook Module is used for managing pricing of products or services.
Products	The Products module is used to Manage the product that your organization sales.
Documents	The Documents module is used to Manage the uploaded documents and notes.
Quotes	The Quotes Module is used for managing the quotations for products or services.
PurchaseOrder	The PurchaseOrder module is used for managing the purchase orders.
SalesOrder	The SalesOrder module is used for managing the sales orders from customers.
Invoice	The Invoice module is used for creating invoices.
Campaigns	The Campaigns module is used for managing Marketing Campaigns.
Events	The Events module is used for Managing Activities such as Calls and Meetings.
Users	The Users module is used for managing the CRM users.
Groups	Manages users groups on the CRM.
Currency	Currency module lets the administrator define different currencies and set the expected conversion rate with respect to the base currency. These currencies can be used in Inventory modules to support multi-currency.
DocumentFolders	The DocumentFolders module is used to group documents..

Field Types

List of field types exposed by the API.

picklist

A field that can a hold one of a list of values, the map will contain two elements, picklistValues which is a list of possible values, and defaultValue which is the default value for the picklist.

Name	Description
picklistValues	Which is a list of possible values.
defaultValue	Specifies which value should be used as the default value for the picklist.
name	Name of the field type.

reference

A field that shows a relation to another object, the field type map will contain another element called refersTo which is an array containing the name of modules of which the field can point to.

Name	Description
refersTo	Which is an array containing the name of modules to which the field can point to.
name	Name of the field type.

datetime

A string representing the date and time, the format is base on the user's settings date format.

date

A string representing a date, the field type map will contain another element called format which is the format in which the value of this field is expected, its based on the user's settings date format.

Name	Description
format	The format in which the value of this field is expected.
name	Name of the field type.

text

A multiline text field.

time

A string of the format hh:mm, format is based on the user's settings time format.

string

A one line text field.

boolean

A boolean field, can have the values true or false.

integer

A none decimal number field.

owner

A field for defining the owner of a field which could be a group or individual user.

autogenerated

These are fields for which the values are generated automatically by the CRM, this is usually an object's id field.

email

A field for storing email ids.

phone

A field for storing phone numbers.

url

A field for storing urls.

double

A field for for floating point numbers.

file

A field for adding a file to the CRM.

Name	Description
maxUploadFileSize	Max allowed size of a file getting uploaded.
name	Name of the field type.

password

A field for storing passwords.

decimal

A field for floating point numbers.

skype

A field for storing skype ids or phone numbers.

multipicklist

A picklist field where multiple values are existent can be selected.

Known Issues

- Sync does not work on the users modul and non entity modules like Currency, Groups etc.
- Query does not work on non entity modules like Currency, Groups etc..

Acknowledgement

This tutorial was originally provided by vtiger. With the friendly permission of vtiger it has been modified to meet the specific requirements of crm-now's CRM offerings.