

# Sugar Market Connector

Last Modified on 03/11/2024 4:31 am EDT

## Authentication Prerequisites

The Sugar Market connector uses the **Basic** authentication workflow. Before you can authenticate an instance of the Sugar Market connector, you would primarily require the API Username and Password, for which you will need to contact the Sugar Market team. For more details on the same, read about Authentication in [Sugar Market API Documentation](#).

## Provider sandbox

A Sugar Market account and sandbox are required. Only '**Administrator**' accounts can generate the credentials required to authenticate with the connector.

There are default values provided by Sugar for API username and password. The only parameters required by the user are:

- Sugar Market Site URL
- Username
- Password

## Required Authentication Parameters

You can authenticate a connector instance using the UI or via API calls. For detailed information on how to authenticate an instance, see our related documentation:

- [Authenticate a Connector Instance \(UI\)](#)
- [Authenticate a Connector Instance \(API\)](#)

To authenticate an instance of the Sugar Market connector, you will need the following parameters:

Parameter Name/UI Value	API Value/Key	Description	Source
Site URL	site.url	The unique URL established for your Sugar Market site.	This is the same URL that you use to sign in to your Sugar Market platform. Contact your Sugar admin for assistance.
Username	username	The username you use to sign in to your Sugar Market site.	Contact your Sugar admin for assistance.
Password	password	The password you use to sign in to your Sugar Market site.	Contact your Sugar admin for assistance.

## Sample Configuration JSON

```
"configuration": {  
  "site.url": "http://my-sugar-market-url.com",  
  "username": "MyUsername",  
  "password": "MyPassw0rd"  
}
```

## Optional Authentication Parameters

Parameter Name	API Value	Description
Default select fields	default.select.fields.map	Generic map used to specify default fields for bulk download and GET /all requests. Each key should be the canonical objectName and values may be specified as a comma-delimited string or a list of strings.

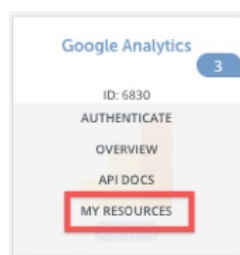
Parameter Name	API Value	Description
Instance Variables	instance.variables	States what the instance variables are.
Synchronous Bulk Notification	synchronous.bulk.notification	Enables synchronous bulk callback notification.
Filter Null Values from Response	filter.response.nulls	Used to enable/disable filtering of null values from the responses

## Connector Fact Sheet and Reference

### Authentication and Configuration Parameters

To see all authentication parameters for the Sugar Market connector, follow these steps:

1. Sign in to SAP Open Connectors and navigate to Connectors.
2. Hover over the card for a connector, and then click My Resources.



3. In the top navigation toolbar, click Setup.
4. From the **Setup** page, you can view a complete list of the connector's authentication and configuration parameters.

## Connector Resources

In order to provide users with normalized APIs across Hubs, SAP Open Connectors resources were mapped to the following Vendor resources.

Platform Placeholder	Platform Method	CE Resource	Vendor Method	Vendor Resource	Description
Accounts	GET	/accounts	GET	/accounts	List accounts
	GET/{id}	/accounts/{id}	GET/{id}	/accounts/{PK}	Retrieve account
	POST	/accounts	POST	/accounts	Create account
	PATCH	/accounts/{id}	PATCH	/accounts/{PK}	Update account
	DELETE	/accounts/{id}	DELETE	/accounts/{PK}	Delete account
Alerts	GET	/alerts	GET	/alerts	List for user
	GET/{id}	/alerts/{id}	GET/{id}	/alerts/{PK}	Retrieve alert
Campaigns	GET	/email-campaigns	GET	/campaigns/email	List campaigns
	GET/{id}	/email-campaigns{id}	GET/{id}	/campaigns/email/{id}	Retrieve email campaign
	POST	/email-campaigns	POST	/campaigns/email	Create email campaign

Platform Placeholder	Platform Method	CE Resource	Vendor Method	Vendor Resource	Description
	PATCH	/email-campaigns{id}	PATCH	/campaigns/email/{id}	Update email campaign
	DELETE	/email-campaigns{id}	DELETE	/campaigns/email/{id}	Delete email campaign
Campaign Lists	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/lists/	
	POST	/email-campaigns/{id}/{action}	POST	/campaigns/email/{PK}/lists/	
	POST	/email-campaigns/{id}/{action}	PUT	/campaigns/email/{PK}/lists/?archive=0	
Campaigns Schedule	POST	/email-campaigns/{id}/{action}	POST	/campaigns/email/{PK}/schedule/	
	POST	/email-campaigns/{id}/{action}	POST	/campaigns/email/{PK}/send/	
	POST	/email-campaigns/{id}/{action}	POST	/campaigns/email/{PK}/recall/	
Campaign Statistics	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/statistics/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/recipients/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/clicks/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/opens/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/unsubscribes/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/bounces/	
	GET	/email-campaigns/{id}/{action}	GET	/campaigns/email/{PK}/schema/	
Campaigns-Schema	GET	/email-campaigns-schema	GET	/campaigns/email/schema/	
Contacts	GET	/contacts	GET	/contacts	
	GET/{id}	/contacts/{id}	GET/{id}	/contacts/{PK}	
	POST	/contacts	POST	/contacts	
	PATCH	/contacts/{id}	PATCH	/contacts/{PK}	
	DELETE	/contacts/{id}	DELETE	/contacts/{PK}	
Opportunities	GET	/opportunities	GET	/opportunities	
	GET/{id}	/opportunities/{id}	GET/{id}	/opportunities/{PK}	
	POST	/opportunities	POST	/opportunities	
	PATCH	/opportunities/{id}	PATCH	/opportunities/{PK}	
	DELETE	/opportunities/{id}	DELETE	/opportunities/{PK}	
Distribution Lists	GET	/distribution-lists	GET	/lists/distribution/	
	POST	/distribution-lists	POST	/lists/distribution/	
	GET/{id}	/distribution-lists/{id}	GET/{id}	/lists/distribution/{PK}/	
	PATCH	/distribution-lists/{id}	PATCH	/lists/distribution/{PK}/	
	DELETE	/distribution-lists/{id}	DELETE	/lists/distribution/{PK}/	
	GET	/distribution-lists/{id}/contacts	GET	/lists/distribution/{PK}/contacts/	
	POST	/distribution-lists/{id}/contacts	POST	/lists/distribution/{PK}/contacts/	
	GET/{id}	/distribution-lists/{id}/contacts/{id}	GET/{id}	/lists/distribution/{PK}/contacts/{contact_id}	
	DELETE	/distribution-lists/{id}/contacts/{id}	DELETE	/lists/distribution/{PK}/contacts/{contact_id}	
	GET	/users	GET	/users	

Platform Placeholder	Platform Method	CE Resource	Vendor Method	Vendor Resource	Description
Users	GET / {id}	/users/{id}	GET/{id}	/users/{PK}	
	POST	/users	POST	/users	
	PATCH	/users/{id}	PATCH	/users/{PK}	
	DELETE	/users/{id}	DELETE	/users/{PK}	
Legacy landing pages	GET	/legacy-pages	GET	/landing_pages/legacy/	
	GET / {id}	/legacy-pages/{id}	GET/{id}	/landing_pages/legacy/{PK}/	
Tasks	GET	/tasks	GET	/tasks	
	GET / {id}	/tasks/{id}	GET/{id}	/tasks/{PK}	
	POST	/tasks	POST	/tasks	
	PATCH	/tasks/{id}	PATCH	/tasks/{PK}	
	DELETE	/tasks/{id}	DELETE	/tasks/{PK}	
Campaign Nurtures	GET	/campaign-nurtures	GET	/campaigns/nurtures/	
	GET / {id}	/campaign-nurtures/{id}	GET	/campaigns/nurtures/{PK}/	
	GET / {id}	/campaign-nurtures/{id}/contacts	GET	/campaigns/nurtures/{PK}/contacts/	
Campaign Nurtures Schema	GET	/campaign-nurture-schemas	GET	/campaigns/nurtures/schema/	
Campaign Nurtures Steps	GET	/campaign-nurture-steps	GET	/campaigns/nurtures/steps/	
	GET / {id}	/campaign-nurture-steps/{id}	GET/{id}	/campaigns/nurtures/steps/{PK}/	
	GET / {id}	/campaign-nurture-steps/{id}/schema	GET/{id}	/campaigns/nurtures/steps/{PK}/schema/	
Events	GET	/events	GET	/events	
	GET / {id}	/events/{id}	GET/{id}	/events/{pk}	
	GET / {id}	/events/{id}/sessions	GET/{id}	/events/{pk}/sessions/	
	GET	/events/{id}/sessions/{id}	GET/{id}	/events/{pk}/sessions/{pk}	
	GET	/events/{id}/sessions/{id}/attendees	GET	/events/{event_id}/sessions/session_id/attendees/	
	GET	/events/{id}/sessions/{id}	GET	/events/{event_id}/sessions/{pk}/	
	GET	/events/{id}/session-schema	GET	/events/{event_id}/sessions/schema/	
Event-attendees	GET	/events-attendees	GET	/events/attendees/	
	GET / {id}	/events-attendees/{id}	GET/{id}	/events/attendees/{pk}/	
	PUT	/events-attendees/{id}	PUT/{id}	/events/attendees/{pk}/	
	DELETE	/events-attendees/{id}	DEL/{id}	/events/attendees/{pk}/	
	GET / {id}	/events-attendees/{id}/schema	GET/{id}	/events/attendees/{pk}/schema/	
Event-schema	GET	/events-schemas	GET	/events/schema	
Event-attendee-schema	GET	/events-attendees-schemas	GET	/events/attendees/schema	
Listeners	GET	/listener-histories	GET	/listener/history/	
	GET	/landing-pages	GET	/landing_pages/	

Platform Placeholder	Platform Method	CE Resource	Vendor Method	Vendor Resource	Description
Landing Pages	GET /{id}	/landing-pages/{id}	GET/{id}	/landing_pages/{pk}/	
	POST	/landing-pages	POST	/landing_pages/	
	PATCH	/landing-pages/{id}	PATCH	/landing_pages/{pk}/	
	DELETE	/landing-pages/{id}	DELETE	/landing_pages/{pk}/	
Forms	GET	/forms	GET	/forms	
	GET /{id}	/forms/{id}	GET/{id}	/forms/{pk}/	
	POST	/forms	POST	/forms	
	PATCH	/forms/{id}	PATCH	/forms/{pk}/	
	DELETE	/forms/{id}	DELETE	/forms/{pk}/	
Web Activities	GET	/activities	GET	/webactivity/	
	GET /{id}	/activities/{id}	GET/{id}	/webactivity/{pk}/	

## Streamlining Patterns in some Resources

Sugar Market API has patterns across some of its resources, that is, the resources shown in the table below follow the pattern of `/campaigns/email/{PK}/{action}` where **PK** is the **primary key** or **object id** of the object. In our implementation we built the same where action is a parameter. So, we have GET and POST on `/campaigns/email/{PK}/{action}` to support all the actions as stated below. (actions=lists,schedule,recall and so on...)

GET	/campaigns/email/{PK}/lists/
POST	/campaigns/email/{PK}/lists/
PUT	/campaigns/email/{PK}/lists/?archive=0
POST	/campaigns/email/{PK}/schedule/
POST	/campaigns/email/{PK}/send/
POST	/campaigns/email/{PK}/recall/
GET	/campaigns/email/{PK}/statistics/
GET	/campaigns/email/{PK}/recipients/
GET	/campaigns/email/{PK}/clicks/
GET	/campaigns/email/{PK}/opens/
GET	/campaigns/emails/{PK}/unsubscribes/
GET	/campaigns/email/{PK}/bounces/
GET	/campaigns/email/{PK}/schema/
GET	/campaigns/email/schema/

## Events and Supported Resources

The Sugar Market connector supports events via polling. For detailed information about our Events framework and how to configure a connector instance with events, see our documentation:

- [Authenticate a Connector Instance with Events \(UI\)](#)
- [Authenticate a Connector Instance with Events \(API\)](#)

You can set up polling events for the following resources of the Sugar Market connector:

- accounts
- contacts
- legacyPages
- events

Sugar Market ✕

Add a new tag to the element instance

Event Configuration

☒ Events Enabled

Event Type

polling

Event Notification Callback URL

Event Notification Basic Password


Event Notification Basic User Name


Event Notification Callback Headers


1 Event poller refresh interval (mins)


Configure Polling

☒ Select All Resources

☒  accounts

☒  contacts

☒  legacyPages

☒  events

REVISIONS

Example data for an event created for Sugar Market connector

```
{
  "id": "Af39j449iEFidDeh1ua6",
  "status": "NOTIFIED",
  "timestamp": "Tue Nov 17 2020 06:55:22 pm",
  "accountId": 16271,
  "element": "sugarmarket",
  "elementInstance": "879101",
  "error": "",
  "providerData": {
    "accounts": [
      {
        "account": "https://developer.salesfusion.com/api/2.0/accounts/388/",
        "account_id": 388,
        "account_name": "Kamba",
        "billing_city": "Cupertino",
        "contacts": "https://developer.salesfusion.com/api/2.0/accounts/388/contacts/",
        "created_by": "https://developer.salesfusion.com/api/2.0/users/4/",
        "created_by_id": 4,
        "created_date": "2020-11-17T18:55:00Z",
        "eventType": "CREATED",
        "key_account": false,
        "objectId": "388",
        "owner": "https://developer.salesfusion.com/api/2.0/users/1/",
        "owner_id": 1,
        "phone": "",
        "updated_by": "https://developer.salesfusion.com/api/2.0/users/4/"
      }
    ]
  }
}
```

```

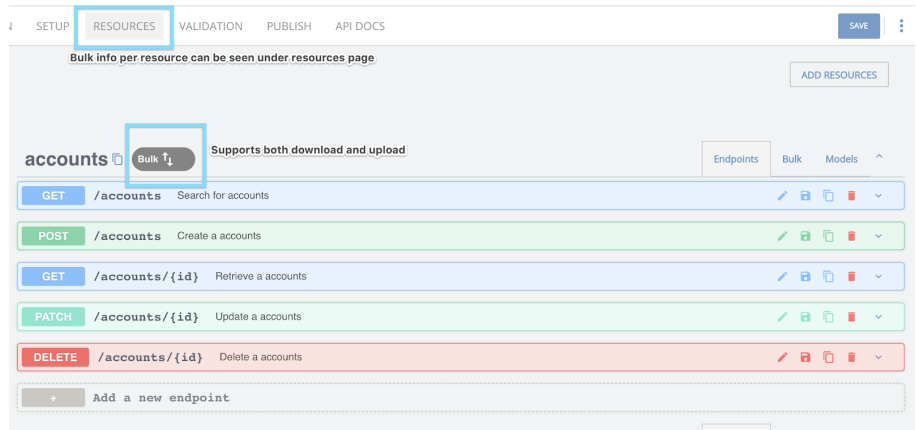
        "updated_by_id": 4,
        "updated_date": "2020-11-17T18:55:00Z"
    }
},
"instance_id": 879101,
"objectType": "accounts",
"pollDate": 1605639297882
},
"eventElementInstances": [
    {
        "eventId": "AXXXj7pFliXZYNylua6",
        "elementInstancelid": 879101,
        "userId": 18375,
        "accountid": 16271,
        "notificationId": 149215795,
        "notifiedData": {
            "accountid": 16271,
            "companyId": 6812,
            "elementKey": "sugarmarket",
            "eventId": "AXXXj7pFliXZYNylua6",
            "events": [
                {
                    "objectType": "accounts",
                    "objectId": "388",
                    "pollDate": "2020-11-17T18:54:57Z",
                    "eventType": "CREATED",
                    "elementKey": "sugarmarket",
                    "hubKey": "marketing"
                }
            ],
            "instancelid": 879101,
            "instanceName": "Demo",
            "instanceTags": [
                "Demo"
            ],
            "instance_id": 879101,
            "raw": {
                "accounts": [
                    {
                        "account": "https://developer.salesfusion.com/api/2.0/accounts/388/",
                        "account_id": 388,
                        "account_name": "Kamba",
                        "billing_city": "Cupertino",
                        "contacts": "https://developer.salesfusion.com/api/2.0/accounts/388/contacts/",
                        "created_by": "https://developer.salesfusion.com/api/2.0/users/4/",
                        "created_by_id": 4,
                        "created_date": "2020-11-17T18:55:00Z",
                        "key_account": false,
                        "owner": "https://developer.salesfusion.com/api/2.0/users/1/",
                        "owner_id": 1,
                        "phone": "",
                        "updated_by": "https://developer.salesfusion.com/api/2.0/users/4/",
                        "updated_by_id": 4,
                        "updated_date": "2020-11-17T18:55:00Z"
                    }
                ],
                "eventHeaders": {
                    "Raw-Element-Key": "sugarmarket",
                    "encodedid": "ODc5MTAx",
                    "id": "879101"
                },
                "objectType": "accounts",
                "pollDate": 1605639297882
            },
            "userId": 18375
        },
        "createdDate": "2020-11-17T18:55:22Z",
        "updatedDate": "2020-11-17T18:55:22Z",
        "objectIds": [
            {
                "value": "388"
            }
        ]
    }
}

```

```
}
}
}
}
```

Bulk

Bulk provides an option to upload a large number of resources, such as contacts, to a service provider all at once. You can use bulk to download a csv or json file from a large number of records or upload a csv or json file to add multiple records. See [Using Bulk](#) for more information on managing bulk.



**Note:** When there is no native bulk available from the provider, SAP Open Connectors provides a bulk service for uploading and downloading data from the endpoint. Sugar Market connector makes use of Bulk V3. For more information, see our [Bulk documentation](#).

Bulk Details

Bulk Information	Details
Bulk Type	SAP Open Connectors Bulk.
Upsert Support	No
Field Selection Support	Yes. You can specify fields within a resource. For example, <code>select name,id from contacts</code> .
Order By Support	No

Querying

The Open Connectors Query Language ( OCNQL ) provides a standard way to search across all of our connectors. Rather than having users research how to query each resource at a provider, we've normalized your search experience. SAP Open Connectors translates your queries to the API provider's search syntax. OCNQL represents the "where" portion of a typical query that can include many other components like field selection, sorting, and pagination. For more details on Querying, see [Querying with OCNQL](#).

To determine whether where clause is supported for an object, you can make an API call to `object/{objectName}/metadata` while providing the Connector version and check for `"filterable"="true"`. In the example shown below, we're checking it for the object - accounts.

Your browser does not support HTML5 video.

A few examples for the usage of where clause would be:

- `/accounts?where=account_id='385'`
- `/accounts?where=account_name='BulkUploadTest'`
- `/accounts?where=created_date>'2020-07-02T13:41:00Z'`
- `/accounts?where=account_name='test' and billing_city='Cupertino'`



And shown below is a short video on how to make an API call using the **where** clause.

There are several operators that are valid to be used for querying. You can find more details on them [here](#). But each object which has "**filterable**"=**true**", can also tell you what operators can be used for that particular filterable, as shown below.

Code	Details
200	<div>Response body</div> <pre>      "response": true,       "request": true     }   ],   {     "type": "boolean",     "path": "active",     "vendorPath": "active",     "vendorNativeType": "boolean",     "filterable": true,     "filterableOperators": [       "=",       "!="     ],     "createable": true,     "updateable": true,     "method": [</pre>

## Connector Extension

Connector Builder gives you the integration tools to extend the current feature set of any of our connectors by adding a resource or method. If your use case requires a resource our connector doesn't currently support, you can easily add it in yourself using Connector Builder.

Sugar Market is an extendable connector and therefore, you can customize it according to your needs. For more details, read the [Extend Connectors](#) article which gives a description in detail as to how it can be done.

## Discovery APIs

Discovery APIs are usually used in conjunction with a specific authenticated connector instance to:

- Identify data objects in a connector
- Identify data objects and their source (the vendor, a common resource, or SAP Open Connectors library)
- Find metadata, including custom fields, related to an object
- Retrieve the Open API specifications (API docs) of an object

However, Discovery APIs are **not supported** by SAP Open Connectors for the Sugar Market connector, as there is no support by the vendor for the same.

## Connector API Documentation

Read more in the [Sugar Market API Documentation](#).

## Provider Documentation

- [Sugar Market API Documentation](#)
-