

Cloud Elements

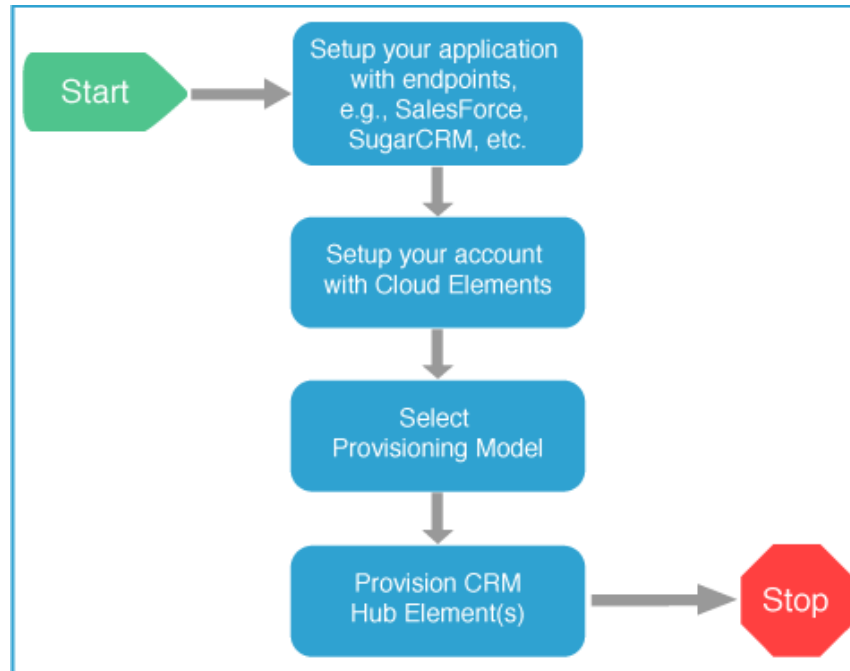
CRM Hub Provisioning and Usage Guide

API Version 2.0



Introduction

The CRM Hub provides a uniform API that allows applications to use various endpoints such as Salesforce, Zoho, SugarCRM, and AutoTaskCRM. The following illustration shows the overall flow of the CRM Hub element provisioning process.



In order to use one or more of the the CRM Hub Elements via the uniform API in your application, you have to first follow three steps -

1. Setting up your application with the CRM Hub Element, e.g., Salesforce, SugarCRM, etc., endpoint
2. Sign up for the Cloud Elements Services.
3. Provisioning the CRM Hub Element

The following sections guide you through this process. This guide will use Salesforce as the example Element. Directions on how to setup a Zoho, SugarCRM, and AutoTaskCRM Instance can be found in the Appendix of this document.

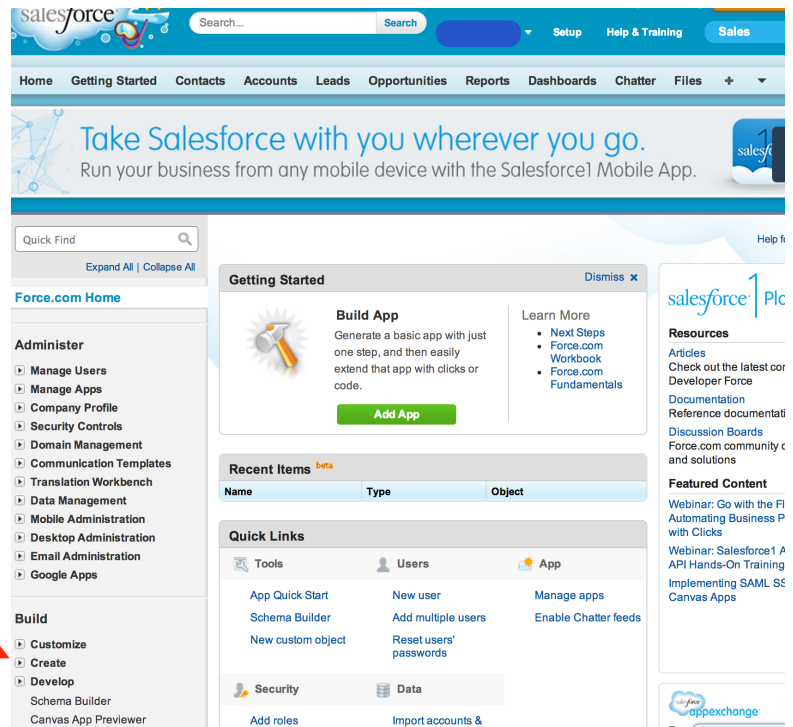
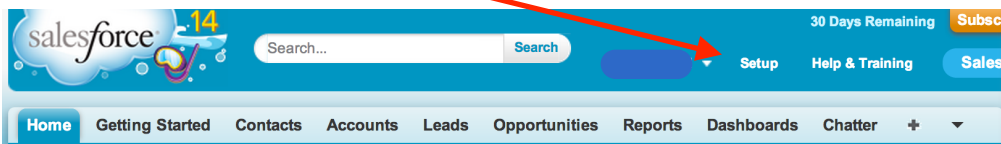
Setup your application with the Endpoint

Create a new Salesforce Connected App

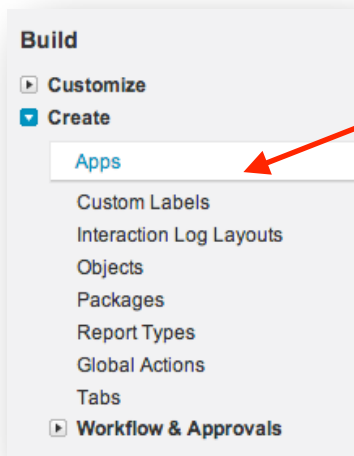
For this guide, we will use Salesforce as the example to setup your application with the endpoint. Instructions on setting up SugarCRM and Zoho are located in the Appendix. For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>. **In order to set up a new application in Salesforce, you must have Administrator privileges. Please contact your system administrator if you do not have those privileges.**

Via a web browser, login to your Salesforce account - <https://login.salesforce.com/>.

Navigate to Setup

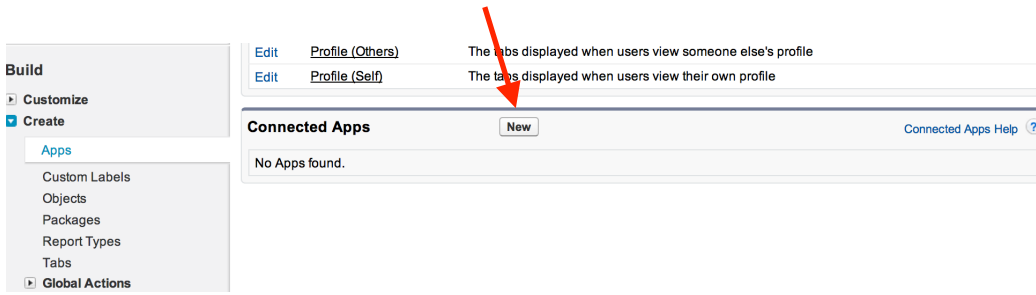


From the menu on the left hand side, click "Create".

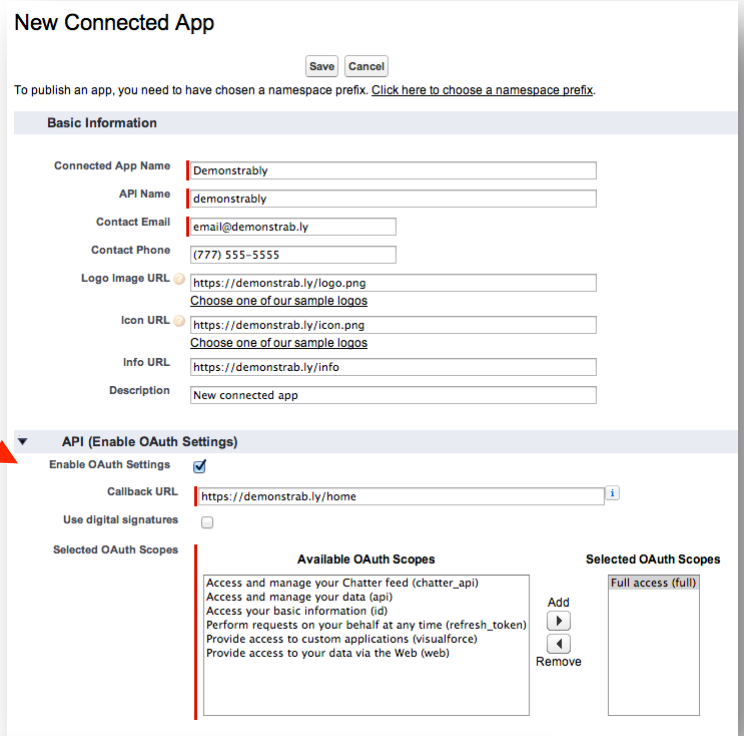


Click "Apps".

Click “New”

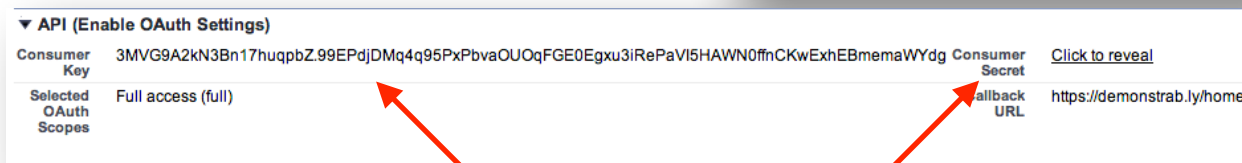


*You'll be required to enter a callback URL from the endpoint. This URL will be in your application's address space, and you will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback is further down in this document. For our example, we'll use a callback URL of <https://demonstrab.ly/home>. *Make sure to select “Enable OAuth Settings”.



Add “Available OAuth Scopes”.

Click “Save” at the bottom of the window.

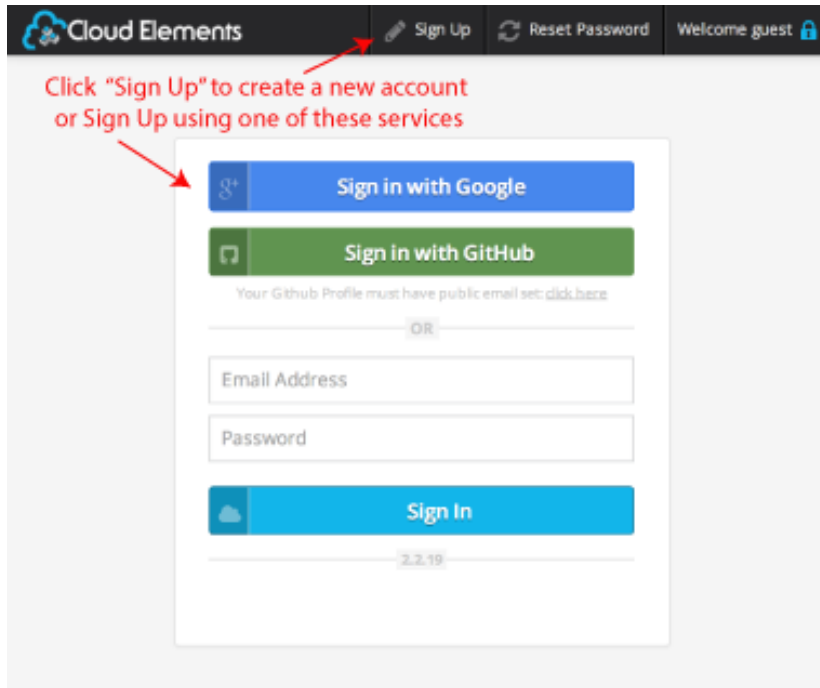


Please make a note of the “Consumer Key” and “Consumer Secret” assigned by Salesforce for the demonstrab.ly application. The next step in the process is to sign up for the Cloud Elements service.

Signup for the Cloud Elements Service

To sign up for the Cloud Elements service, using a web browser, go to:

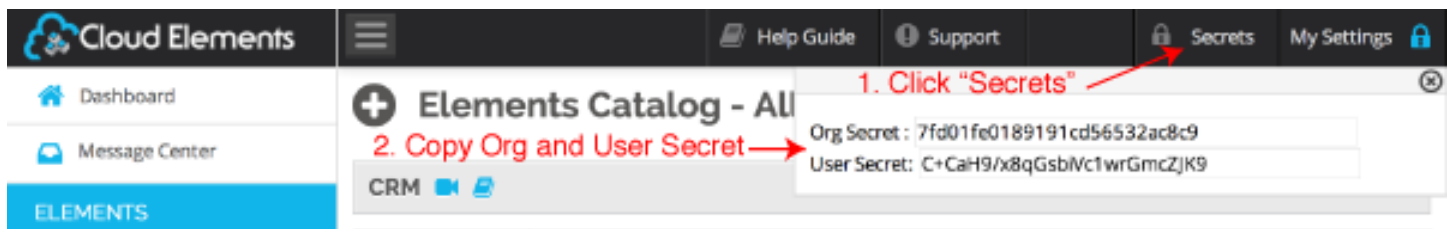
<https://console.cloud-elements.com>.



You can create a new account with Cloud Elements using the **“Sign Up”** link shown here, or sign up for Cloud Elements servicing your **Google** or **GitHub account**. When signing up via GitHub, you must set a public email address in GitHub profile. Screen shots on how to set a public email are on the next page.

If you choose to not use one of these services to sign up, you will be required to validate your new account, and will be then required to reset your password.

After completing this process, click "Secrets" in the top right corner. Make note of the Org and User secrets as they needed to provision an Element Instance. The Secrets option is available from the top ribbon at all times.



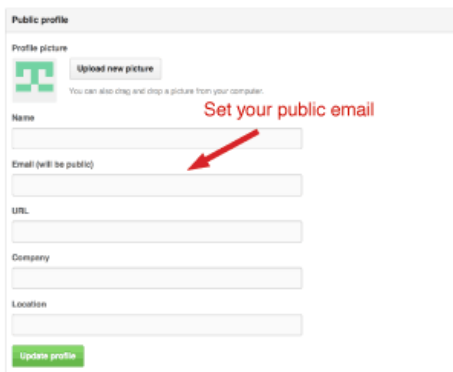
You're now ready to start provisioning the CRM Hub Elements via the Cloud Elements Provisioning APIs.

Setting a Public Email in GitHub

Navigate to your settings in GitHub.



Type in your public email in the designated field and click "Update Profile".

A screenshot of the GitHub 'Public profile' settings page. The page has a title 'Public profile' and a sub-section 'Profile picture' with an 'Upload new picture' button. Below this are several input fields: 'Name', 'Email (will be public)', 'URL', 'Company', and 'Location'. A red arrow points from the text 'Set your public email' to the 'Email (will be public)' field. At the bottom left, there is a green 'Update profile' button.

Documentation continued on next page.

Provisioning the CRM Hub Elements

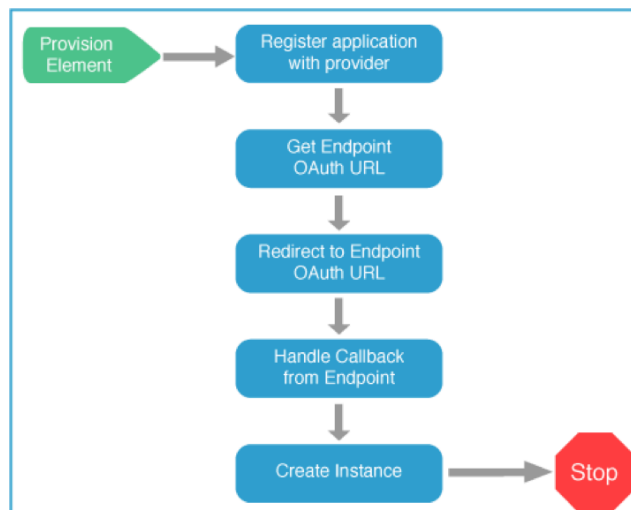
Before we start provisioning Elements, we need to understand how your application will use the CRM Hub Elements. There are currently two provisioning models for all Elements:

1. Single Application User
2. Multiple Accounts and Users

A diagram of the workflow for provisioning an element for a single user will be shown first. However, in the demonstration documentation, the multiple users provisioning flow will be shown.

Single Application User

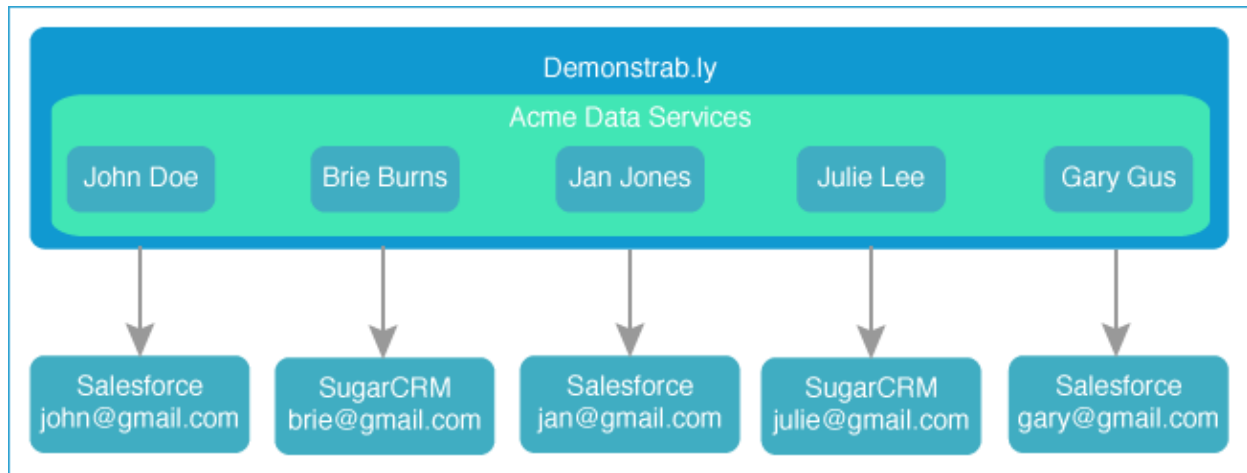
The workflow for a single application user is shown below.



Multiple Accounts and Users

This model allows you as the administrator to provision Elements on behalf of your customer's (account) users. The provisioning workflow for multiple users is the same as the single application user workflow.

The following illustration shows how a fictitious application, demonstrab.ly, might access the various endpoints via the CRM Hub, via multiple users of the application.



Before provisioning an element, the account and the account users need to be set up. The API to set up an account is POST /accounts. The full API documentation can be found at <https://api.cloud-elements.com/elements/api-v2/accounts>.

Below are examples of how to create a new account, as well as a user for the newly created account.

Create a new Account within your Organization

POST /accounts

Create an account within your organization (identified by your organization secret). The provided user secret must be that of the default admin user for the organization.

This API call requires a JSON file with the required data to create your account. An example JSON file can be found below along with an example response from the API call.

`create-account.json` - these fields are required in your JSON file.

```
{
  "name": "<Name of the Account>",
  "description": "<Description of the Account>",
  "externalId": "<ID of the Account>"
}
```

Example of Successful Response - data returned in JSON format

```
{
  "companyId": <The ID of the organization that owns the account>,
  "name": "<The name of the account>",
  "description": "<The description of the account>",
  "externalId": "<The external ID of the Account>",
  "id": <The CE assigned ID of the Account>
}
```

Below is an example cURL command demonstrating the API call and response with example data.

POST /accounts

POST /accounts

```
curl -X POST \
-H 'Authorization: User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>' \
-H 'Content-Type: application/json' \
-d @create-account.json \
'https://api.cloud-elements.com/elements/api-v2/accounts'
```

create-account.json

```
{
  "name": "Acme Data Services",
  "description": "The Acme Data Services Account",
  "externalId": "brie@acmedataservices.com"
}
```

Example of Successful Response

```
{
  "companyId": 6523,
  "name": "Acme Data Services",
  "description": "The Acme Data Services Account",
  "externalId": "acmedataservices.com",
  "id": 8341
}
```

After successful creation of the account. You may now add users to that account.

Below are examples of how to create a new user for the newly created account.

POST /accounts/{id}/users

Create a new user for the newly created account.

This API call requires a JSON file with the required data to create your user. An example JSON file can be found below along with an example response from the API call.

create-account-user.json - JSON file needed to create user.

```
{
  "email": "<The user's email address>",
  "firstName": "<The user's first name>",
  "lastName": "<The user's last name>"
}
```

Example of Successful Response - data returned from the API call

```
{
  "id": <The CE ID for the user>,
  "email": "<The user's email address>",
  "firstName": "<The user's first name>",
  "lastName": "<The user's last name>",
  "secret": "<The user's secret>"
}
```

Below is an example cURL command demonstrating the API call and response with example data.

POST /accounts/{id}/users

```
curl -X POST \
-H 'Authorization: User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>' \
-H 'Content-Type: application/json' \
-d @create-account-user.json \
'https://api.cloud-elements.com/elements/api-v2/accounts/{id}/users'
```

create-account-user.json

```
{
  "email": "brie@acmedataservices.com",
  "firstName": "Brie",
  "lastName": "Burns"
}
```

Example of Successful Response - data returned from the API call

```
{
  "id": 12802,
  "email": "brie@acmedataservices.com",
  "firstName": "Brie",
  "lastName": "Burns",
  "secret": "KHh2387n2hk34/979283u012="
}
```

There may be other fields in the Response JSON.

To provision Elements, use your Organization secret and the User Secret of the newly created account user. The provisioning flow will be the same using the POST /instances API. Element Provisioning is covered in the Create Instance section further down in this document.

The Multiple Users and Accounts workflow gives you the flexibility to create instances for specific accounts. For example, if you wanted to create an instance for one of your customers, then you would create an account for that customer first. After the account has been successfully created for that customer, you may create individual users, and Element instances for those users under the newly created account, using your own Organization secret and your customer's new User secret.

REST API Documentation

Note: The complete interactive documentation for the CRM Hub API can be found at:

<https://api.cloud-elements.com/elements/api-docs/#!/crm>

The documentation for the Provisioning API can be found at:

<https://api.cloud-elements.com/elements/api-docs/#!/instances>

and

<https://api.cloud-elements.com/elements/api-docs/#!/accounts>

Provisioning with REST APIs

The following sections describe which APIs to use for each step in the above flowchart. Provisioning is done through a REST API. Each request is made using an HTTP verb and the requested URL.

There are 5 parts to an HTTP request that are required by the Cloud Elements API.

1. HTTP Headers: headers that are required for each request
2. HTTP Verb: the verb that must be used in the request
3. Request URL: the form of the URL where to make the request
4. Request Body: the text of the request body in JSON format
5. Query Parameters: a query string that is appended to the URL

Get Elements OAuth Information

HTTP Header: None

HTTP Verb: GET

Request URL: /elements/{key}/oauth/url

Request Body: None

Query Parameters:

- **apiKey**- the key obtained from registering your app with the provider
- **apiSecret** - the secret obtained from registering your app with the provider
- **callbackUrl** - the URL that you supplied to the provider when registering your app, state - any custom value that you want passed to the callback handler listening at the provided callback URL.

Description: The result of this API invocation is an OAuth redirect URL from the endpoint. Your application should now redirect to this URL, which in turn will present the OAuth authentication and authorization page to the user. When the provided callback URL is executed, a code value will be returned, which is required for the Create Instance API.

A sample request illustrating the API is shown below.

Input Query Parameters

Note: the state can be any custom value that you decide to pass in. The returned OAuth URL will contain the “state” query parameter set to your custom value that was passed in.

Example cURL Request

```
curl -X GET
-H 'Content-Type: application/json'
'https://api.cloud-elements.com/elements/api-v2/elements/sfdc/oauth/url?
apiKey=fake_salesforce_api_key&apiSecret=fake_salesforce_api_secret&callbackUrl=https
://www.demonstrab.ly/home&state=sfdc'
```

Successful Example Response

```
{
  "element": "sfdc",
  "oauthUrl": "https://login.salesforce.com/services/oauth2/authorize?
response_type=code&client_id=fake_salesforce_api_key&client_secret=xyz789&scope=full
%20refresh_token&redirect_uri=https://www.demonstrab.ly/home&state=sfdc"
}
```

Handle Callback from the Endpoint

Upon successful authentication and authorization by the user, the endpoint will redirect to the callback URL you provided when you setup your application with the endpoint, in our example, <https://www.demonstrab.ly/home>.

The endpoint will also provide two query string parameters: "state" and "code". The value for the "state" parameter will be the name of the endpoint, e.g., "sfdc" in our example, or whatever value you sent initially. The value for the "code" parameter is the code required by Cloud Elements to retrieve the OAuth access and refresh tokens from the endpoint.

If the user denies authentication and/or authorization, there will be a query string parameter called "error" instead of the "code" parameter. In this case, your application can handle the error gracefully.

Once you've gotten the "code" from the callback, your application can call the API described below to create the instance of the element.

Create Instance

To provision your Salesforce.com element, use the /instances API. You can find the documentation for this at <https://api.cloud-elements.com/elements/api-docs/#/instances>. Below is an example of the provisioning API call.

An Element token is returned upon successful execution of this API. This token needs to be retained for use in subsequent API calls.

account.json - file needs to be included in your instance API call with your account information. Example below

```
{
  "element": {
    "key": "sfdc"
  },
  "providerData": {
    "code": "<Code_On_The_Return_URL>"
  },
  "configuration": {
    "oauth.callback.url": "https://www.demonstrab.ly/home",
    "oauth.api.key": "<Insert_Client_ID>",
    "oauth.api.secret": "<Insert_Client_Secret>"
  },
  "tags": [
    "<Add_Your_Tag>"
  ],
  "name": "<Insert_Instance_Name>"
}
```

A sample request illustrating the POST /instances API is shown below. **NOTE: the “key” into Cloud Elements for Salesforce is “sfdc”. The “key” assigned to Sugar CRM is “sugar”.**

HTTP Headers

```
Authorization: User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>
```

Successful Response JSON *continued on next page

```
{
  "id": <INSTANCE_ID>,
  "name": "<YOUR_INSTANCE_NAME>",
  "token": "<YOUR_TOKEN>",
  "element": {
    "id": <ID>,
    "name": "<ELEMENT_NAME>",
    "key": "<KEY_INTO_CLOUD_ELEMENTS_SYSTEM>",
    "description": "<DESCRIPTION OF APP SERVICE>",
    "active": true,
    "deleted": false,
    "typeOauth": true,
    "trialAccount": false,
    "existingAccountDescription": "<EXISTING ACCOUNT DESCRIPTION>",
    "configDescription": "If you do not have a Salesforce.com account, you can create
one at <a href=\"http://www.salesforce.com\" target=\"_blank\">Salesforce.com
Signup</a>",
    "signupURL": "http://www.salesforce.com"
  },
  "provisionInteractions": [],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "cachingEnabled": false
}
```

Example

Request

```
curl -X POST \  
-H 'Authorization: User <INSERT_USER_SECRET>, Organization  
<INSERT_ORGANIZATION_SECRET>' \  
-H 'Content-Type: application/json' \  
-d @create-instance-template.json \  
'https://api.cloud-elements.com/elements/api-v2/instances'
```

create-instance-template.json *Continued on next page

```
{  
  "element": {  
    "key": "sfdc"  
  },  
  "providerData": {  
    "code": "<Code_On_The_Return_URL>"  
  },  
  "configuration": {  
    "oauth.callback.url": "https://www.demonstrab.ly/home",  
    "oauth.api.key": "za6jhr3hychnadajfldjafdsaiidadief0",  
    "oauth.api.secret": "eghxt9akud990a72uv"  
  },  
  "tags": [  
    "<Add_Your_Tag>"  
  ],  
  "name": "<Insert_Instance_Name>"  
}
```

Response

```
{
  "id": <INSTANCE_ID>,
  "name": "<YOUR_INSTANCE_NAME>",
  "token": "<YOUR_TOKEN>",
  "element": {
    "id": <ID>,
    "name": "<ELEMENT_NAME>",
    "key": "<KEY_INTO_CLOUD_ELEMENTS_SYSTEM>",
    "description": "<DESCRIPTION OF APP SERVICE>",
    "active": true,
    "deleted": false,
    "typeOauth": true,
    "trialAccount": false,
    "existingAccountDescription": "<EXISTING ACCOUNT DESCRIPTION>",
    "configDescription": "If you do not have a Salesforce.com account, you can create
one at <a href=\"http://www.salesforce.com\" target=\"_blank\">Salesforce.com
Signup</a>",
    "signupURL": "http://www.salesforce.com"
  },
  "provisionInteractions": [],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "cachingEnabled": false
}
```


Use The CRM Hub In Your Application

Now that you have finished provisioning your Salesforce element, you are ready to use the CRM APIs to access Accounts, Contacts, Leads and more in your Salesforce account.

Each object within the CRM Hub can be accessed and manipulated using the following HTTP verbs:

POST, GET, PATCH, and DELETE

The documentation examples will show the Account object being used with each of the HTTP verbs. The workflow is similar for the other objects available in the CRM Hub.

Create Account

```
curl -X POST \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
-H 'Content-Type: application/json' \  
-d @account.json \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/accounts'
```

account.json

```
{  
  "phone": "7775551234",  
  "fax": "7775551234",  
  "type": "Prospect",  
  "annualRevenue": 100000,  
  "numberOfEmployees": 50,  
  "name": "My Awesome Account"  
}
```

Example of Success Response: the account object will be returned.

```
{  
  "id": "001i0000010i6jMAAQ",  
  "phone": "7775551234",  
  "fax": "7775551234",  
  "annualRevenue": 100000,  
  "name": "My Awesome Account",  
  "numberOfEmployees": 50,  
  "type": "Prospect"  
}
```

Retrieve Account - Continued on Next Page.

```
curl -X GET \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/accounts?where=name='Foo''
```

Example of Successful Response

```
[  
  {  
    "SicDesc": null,  
    "phone": null,  
    "TickerSymbol": null,  
    "type": null,  
    "NumberOfEmployees": null,  
    "ShippingAddress": null,  
    "industry": null,  
    "createdDate": "2014-05-06T13:17:21.000+0000",  
    "lastViewedDate": "2014-07-15T05:38:42.000+0000",  
    "ShippingLatitude": null,  
    "BillingLongitude": null,  
    "Jigsaw": null,  
    "fax": null,  
    "Site": null,  
    "lastActivityDate": null,  
    "number": null,  
    "BillingAddress": null,  
    "deleted": false,  
    "ShippingLongitude": null,  
    "Rating": null,  
    "MasterRecordId": null,  
    "lastModifiedDate": "2014-05-06T13:17:21.000+0000",  
    "billingAddress": {  
      "street1": null,  
      "postalCode": null,  
      "stateOrProvince": null,  
      "country": null,  
      "city": null  
    },  
    "JigsawCompanyId": null,  
    "Ownership": null  
  }  
]
```

Update Account

```
curl -X PATCH \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
-H 'Content-Type: application/json' \  
-d @account.json \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/accounts/701i00000016M2s'
```

account.json

```
account.json  
{  
  "phone": "(777) 555-5555",  
  "fax": "(777) 555-5555",  
  "annualRevenue": "100000",  
  "name": "Test Account",  
  "numberOfEmployees": "50",  
  "shippingAddress": {  
    "street2": "102 street 2",  
    "street1": "1234 Street drive",  
    "postalCode": "80808",  
    "stateOrProvince": "CO",  
    "country": "USA",  
    "city": "Denver"  
  },  
  "number": "101",  
  "industry": "Software",  
  "type": "Customer - Direct"  
}
```

Example Successful Response:

```
{  
  "phone": "(777) 555-5555",  
  "fax": "(777) 555-5555",  
  "number": "101",  
  "type": "Customer - Direct",  
  "id": "001i00000010i6jM",  
  "annualRevenue": "100000",  
  "name": "Test Account",  
  "numberOfEmployees": "50",  
  "industry": "Software",  
  "shippingAddress": {  
    "street1": "1234 Street drive",  
    "postalCode": "80808",  
    "stateOrProvince": "CO",  
    "country": "USA",  
    "city": "Denver"  
  }  
}
```

Delete Account

```
curl -X DELETE \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/accounts/001i000000oMV0UAAW'
```

Example Successful Response:

Response Code: 200

Find Accounts by ID.

```
curl -X GET \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/accounts/701i00000016M2s'
```

Example Successful Response

```
{
  "SicDesc": null,
  "phone": "7775551234",
  "TickerSymbol": null,
  "type": "Prospect",
  "NumberOfEmployees": 50,
  "ShippingAddress": null,
  "industry": null,
  "createdDate": "2014-08-05T18:17:37.000+0000",
  "lastViewedDate": "2014-08-05T18:17:37.000+0000",
  "ShippingLatitude": null,
  "BillingLongitude": null,
  "Jigsaw": null,
  "fax": "7775551234",
  "Site": null,
  "lastActivityDate": null,
  "number": "2",
  "BillingAddress": null,
  "deleted": false,
  "ShippingLongitude": null,
  "Rating": null,
  "CreatedById": "005i0000001VNdeAAG",
  "AccountSource": null,
  "OwnerId": "005i0000001VNdeAAG",
  "BillingLatitude": null,
  "AnnualRevenue": 100000,
  "id": "701i00000016M2s",
  "parentId": null,
  "SystemModstamp": "2014-08-05T18:17:37.000+0000",
  "name": "My Awesome Account",
  "PhotoUrl": "/services/images/photo/001i0000010i6jMAAQ",
  "websiteUrl": null,
  "shippingAddress": {
    "street1": null,
    "postalCode": null,
    "stateOrProvince": null,
    "country": null,
    "city": null
  },
  "sic": null,
  "Description": null,
  "lastReferencedDate": "2014-08-05T18:17:37.000+0000",
  "LastModifiedById": "005i0000001VNdeAAG",
  "MasterRecordId": null,
  "lastModifiedDate": "2014-08-05T18:17:37.000+0000",
  "billingAddress": {
    "street1": null,
    "postalCode": null,
    "stateOrProvince": null,
    "country": null,
    "city": null
  }
}
```

Ping Element

```
curl -X GET \  
-H 'Authorization: Element <INSERT_ELEMENT_TOKEN>, User <INSERT_USER_SECRET>' \  
'https://api.cloud-elements.com/elements/api-v2/hubs/crm/ping'
```

Successful Example Response

```
{  
  "dateTime": "2014-08-05T17:02:01-05:00",  
  "endpoint": "sfdc"  
}
```

The other objects supported in the CRM Hub are Contacts, Leads, Objects (custom), Opportunities, and Users.

If you would like to try out the other APIs in the CRM Hub, then please visit: <https://api.cloud-elements.com/elements/api-docs/#!/crm>

Error Codes

Errors are never fun. To save you time we took a list of error codes from our service providers and normalized them into one list. If an error message is returned that includes a specific detail about the method call, we will send our normalized error code along with any additional data that comes from the provider in the body of the message. Below is a sample message from a service provider that would be included with our normalized error code.

```
{  
  "message": "Failed to upload file. File with the same name already exists.  
Change the name of the file you're uploading or set the overwrite query parameter  
to true.",  
  "requestId": "53dbca9430045c016b46566c",  
  "providerMessage": "Item with the same name already exists"  
}
```

A complete list of our normalized error codes can be found on the next page.

Response Code	Error
200	OK – Everything worked as expected.
400	Bad Request – Often due to a missing request parameter.
401	Unauthorized – An invalid element token, user secret and/or org secret provided.
403	Forbidden – Access to the resource by the provider is forbidden.
405	Method Not Allowed – Incorrect HTTP verb used, e.g., GET used when POST expected.
406	Not Acceptable – The response content type does not match the “Accept” header value.
409	Conflict – A resource being created already exists.
415	Unsupported Media Type – The server cannot handle the requested Content-Type.
500	Server Error – Something went wrong on the Cloud Elements server.
502	Failed Request – Endpoint provider was not able to perform a CRUD request.

Appendix

Zoho Application setup

Using a web browser, navigate to: <https://www.zoho.com/crm/> and sign up for service. You must have Administrator privileges in order to provision an Element Instance. Please contact your system administrator if you do not have those privileges.

Your username/email and password are required to provision an instance. An example cURL command demonstrating the call is below.

POST /instances

The create-instance-zoho.json file needs to be completed with your credentials. This file is sent with your POST /instances call. An example cURL command demonstrating the call is below.

Request

```
curl -X POST \  
-H 'Authorization: User <INSERT_USER_SECRET>, Organization  
<INSERT_ORGANIZATION_SECRET>' \  
-H 'Content-Type: application/json' \  
-d @create-instance-zoho.json \  
'https://api.cloud-elements.com/elements/api-v2/instances'
```

create-instance-zoho.json *Continued on next page

```
{  
  "name": "zohotest",  
  "element": {  
    "key": "zohocrm"  
  },  
  "configuration": {  
    "crm.zohocrm.username": "jon@acmedata.com",  
    "crm.zohocrm.password": "password1"  
  }  
}
```

Example of Successful Response


```

{
  "id": 51,
  "name": "zohotest",
  "token": "bi53fmAY1YBaW9tRnf6CuGdJXvAgBIMfTlraFy",
  "element": {
    "id": 44,
    "name": "Zoho CRM",
    "key": "zohocrm",
    "description": "Attract, Retain and Delight More Customers. Grow Your Business
Now With Zoho CRM.",
    "image": "elements/provider_zohocrm.png",
    "active": true,
    "deleted": false,
    "typeOauth": false,
    "trialAccount": false,
    "configDescription": "If you do not have a Zoho CRM account, you can create one
at <a href='\"https://www.zoho.com/crm/zohocrm-pricing.html\" target='\"_blank\">Zoho
Signup</a>",
    "signupURL": "https://www.zoho.com/crm/"
  },
  "provisionInteractions": [],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "transformation": {
    "objectNames": [
      {
        "name": "account",
        "vendorName": "Accounts"
      },
      {
        "name": "contact",
        "vendorName": "Contacts"
      },
      {
        "name": "lead",
        "vendorName": "Leads"
      },
      {
        "name": "product",
        "vendorName": "Products"
      },
      {
        "name": "opportunity",
        "vendorName": "Potentials"
      },
      {
        "name": "user",
        "vendorName": "Users"
      }
    ]
  },
  "cachingEnabled": false,
  "defaultTransformation": false
}

```

Sugar CRM Application Setup

Using a web browser, navigate to: <http://www.sugarcrm.com/> and sign up for service.

Your username/email, password, and callback URL are required to provision an instance.

Below is an example cURL command documenting the JSON file needed to create an instance. This file will need your sugar credentials.

Request

```
curl -X POST \  
-H 'Authorization: User <INSERT_USER_SECRET>, Organization  
<INSERT_ORGANIZATION_SECRET>' \  
-H 'Content-Type: application/json' \  
-d @create-instance-sugar.json \  
'https://api.cloud-elements.com/elements/api-v2/instances'
```

create-instance-sugar.json *Continued on next page

```
{  
  "name": "sugar-test",  
  "element": {  
    "key": "sugar"  
  },  
  "configuration": {  
    "sugarcrm.server.url": "<your_callback_url>",  
    "sugarcrm.username": "<your_sugar_username>",  
    "sugarcrm.password": "<your_sugar_password>",  
    "sugarcrm.user.display.name": "<your_sugar_display_name>"  
  }  
}
```

Example of Successful Response

```

{
  "id": 50,
  "name": "sugar-test",
  "token": "p0M0hKm03AHb/Z1EZehMhwJaTXw0io3Hllo",
  "element": {
    "id": 24,
    "name": "SugarCRM",
    "key": "sugar",
    "description": "Business and customer management, together. with the SugarCRM
Element, integrate a leading customer management service into your application in a
fraction of the time, with a fraction of the code. Your campaigns, projects, CRM,
payments and documents are all in one place, all integrated seamlessly into your
app.",
    "image": "elements/provider_sugarcrm.png",
    "active": true,
    "deleted": false,
    "typeOauth": false,
    "trialAccount": false,
    "existingAccountDescription": "Use Your SugarCRM Account <br /><span class=
\"buttonDescription\">Enter your credentials and feed details for <b>SugarCRM
Account</b></span>",
    "configDescription": "If you do not have a SugarCRM account, you can create one
at <a href=\"http://www.sugarcrm.com\" target=\"_blank\">SugarCRM Signup</a>",
    "signupURL": "http://www.sugarcrm.com"
  },
  "provisionInteractions": [],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "transformation": {
    "objectNames": [
      {
        "name": "account",
        "vendorName": "Accounts"
      },
      {
        "name": "contact",
        "vendorName": "Contacts"
      },
      {
        "name": "opportunity",
        "vendorName": "opportunities"
      },
      {
        "name": "lead",
        "vendorName": "Leads"
      },
      {
        "name": "note",
        "vendorName": "Notes"
      }
    ]
  },
  "cachingEnabled": false,
  "defaultTransformation": false
}

```

AutoTask CRM Application Setup

Using a web browser, navigate to: <https://www.autotask.com/> and sign up for service.

If you already have an account, navigate to: <https://www.autotask.net/> and login.

Your username/email, password, and callback URL are required to provision an instance.

Below is an example cURL command documenting the JSON file needed to create an instance. This file will need your sugar credentials.

Request

```
curl -X POST \
-H 'Authorization: User <INSERT_USER_SECRET>, Organization
<INSERT_ORGANIZATION_SECRET>' \
-H 'Content-Type: application/json' \
-d @create-instance-autotask.json \
'https://api.cloud-elements.com/elements/api-v2/instances'
```

create-instance-sugar.json *Continued on next page

```
{
  "name": "autotasktestWP",
  "element": {
    "key": "autotask"
  },
  "configuration": {
    "psa.autotask.username": "YOUR_AUTOTASK_USERNAME",
    "psa.autotask.password": "YOUR_AUTOTASK_PASSWORD",
    "psa.autotask.server.url": "https://webservices2.autotask.net/
atservices/1.5/atws.asmx"
  }
}
```

Example of Successful Response

```
{
  "id": 20,
  "name": "autotasktestWP",
  "token": "1VY01PafYe3T+AjlTDPnZ8ZjTcFcb9A3jCeF3A=",
  "element": {
    "id": 27,
    "name": "Autotask.com",
    "key": "autotask",
    "description": "Autotask.com A Complete Business Management Platform for Technology Solutions Providers",
    "image": "elements/provider_autotask.png",
    "active": false,
    "deleted": false,
    "typeOAuth": false,
    "trialAccount": false,
    "configDescription": "If you do not have an Autotask.com account, you can create one at <a href=\"http://
www.autotask.com\" target=\"_blank\">Autotask.com Signup</a>",
    "signupURL": "http://www.autotask.com"
  },
  "provisionInteractions": [],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "transformation": {
    "objectNames": [
      {
        "name": "account",
        "vendorName": "Account"
      },
      {
        "name": "contact",
        "vendorName": "Contact"
      }
    ]
  },
  "cachingEnabled": false,
  "defaultTransformation": false
}
```

End Appendix

Now what? Now you are ready to start integrating our CRM Hub Elements into your application. Give your customers access to their Salesforce, Zoho, Sugar, and AutoTask CRM accounts right from you app.

This concludes the documentation for the CRM Hub. If you have any questions, please contact us at support@cloud-elements.com

We will do our best to get back to you within 24 hours. Your success is our success.

Thanks for reading.

The Cloud Elements Team