Core Developer Guide

Deployment

Core is currently deployed on Heroku at http://corecloudapp.herokuapp.com
Administrative functions can be accessed at https://dashboard.heroku.com/apps/corecloudapp/

Rails Controllers and Models

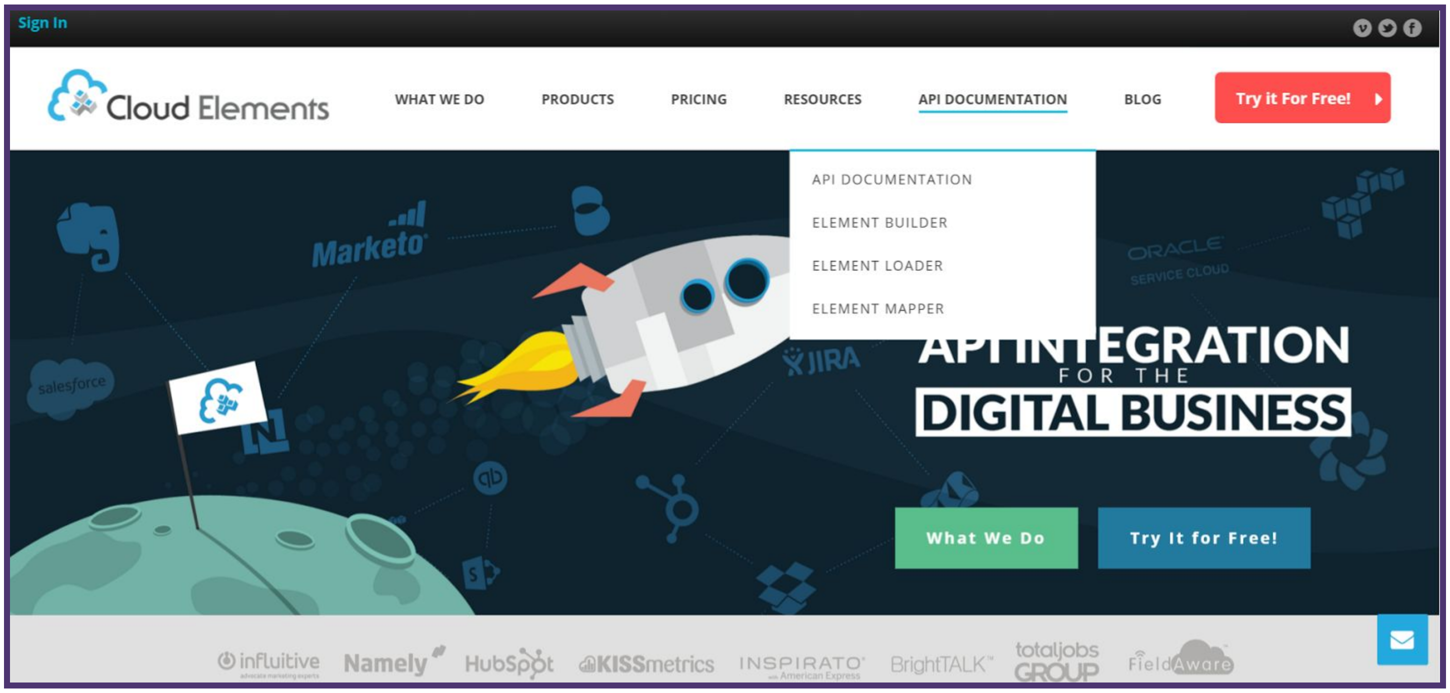
Below is a list of the various controllers and models in the Ruby on Rails architecture as well as their basic function to the overall program.

- Controllers
  - **CallbackController** - Receives POST data from Cloud Elements when events occur and sends the information to the appropriate database writing method
  - **CoreController** - Provides various methods for passing information from the models to the view of Core
  - **ElementsController** - Contains methods for creating instances in Cloud Elements and providing information for various API calls
  - **SessionsController** - Manages user sessions on Core
  - **WelcomeController** - Controls the flow of information at login as well as the new-org flow
- Models
  - **CloudElements** - Provides additional methods for managing communications with Cloud Elements
  - **Database** - Methods for writing and modifying the various data models in the database
  - **Org** - Represents a single organization belonging to Core
  - **User** - Represents an individual Core user - belongs to an Org
  - **QuickbooksCustomer** - A single customer in Quickbooks - belongs to an Org
  - **QuickbooksInvoice** - An invoice in Quickbooks - belongs to a QuickbooksCustomer
  - **QuickbooksPayment** - A payment made through Stripe to Quickbooks - belongs to a QuickbooksCustomer
  - **QuickbooksReport** - A report available through Quickbooks - belongs to a QuickbooksCustomer
  - **SalesforceAccount** - A single account in Salesforce - belongs to an Org
  - **SalesforceContact** - A contact in Salesforce - belongs to a SalesforceAccount
  - **SalesforceLead** - A potential lead in Salesforce - belongs to an Org
  - **SalesforceOpportunity** - An opportunity in Salesforce - belongs to a SalesforceAccount
  - **SalesforceReport** - A report available through Salesforce - belongs to a SalesforceAccount
  - **StripeCustomer** - Provides information on the API keys to catch Stripe events

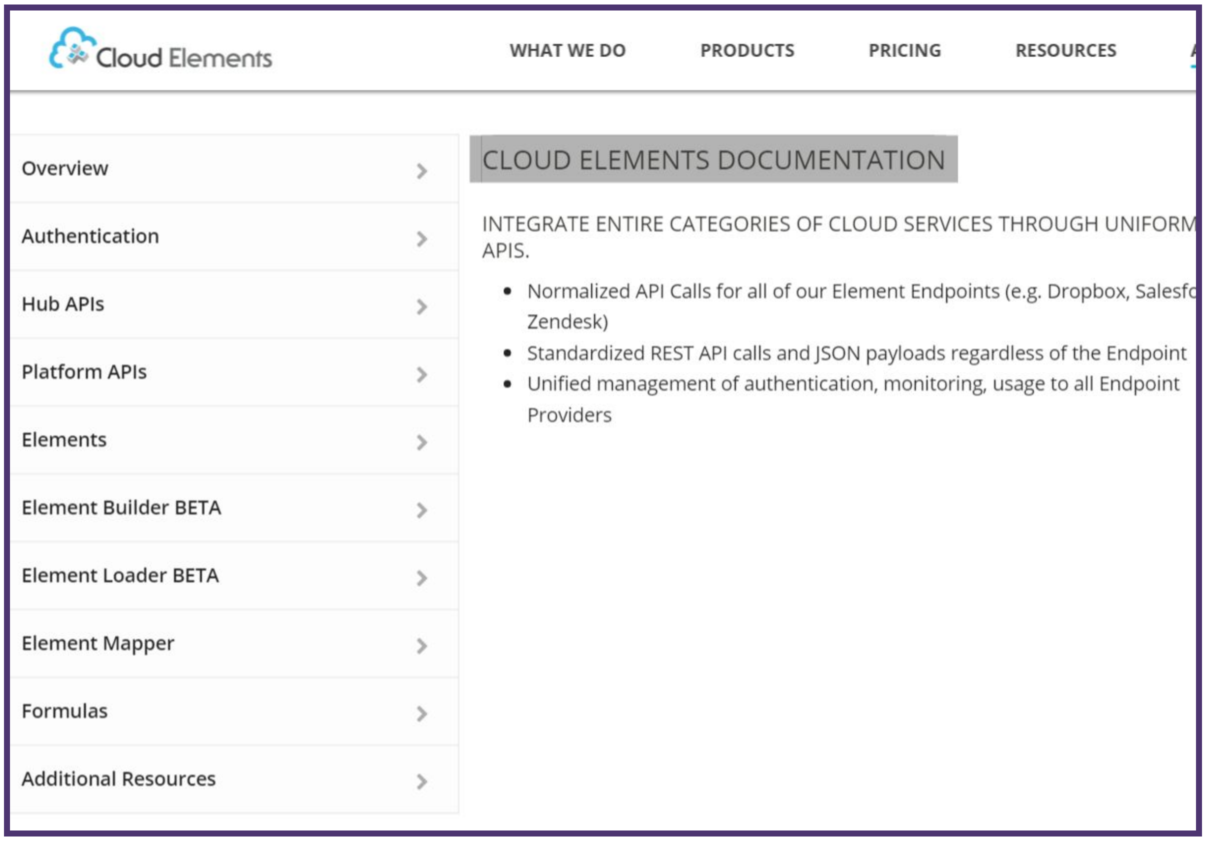
Cloud Elements

Please visit http://cloud-elements.com/ for the documentation walkthrough.

Once at the website you will click on “API DOCUMENTATION” in the tab bar at the top.



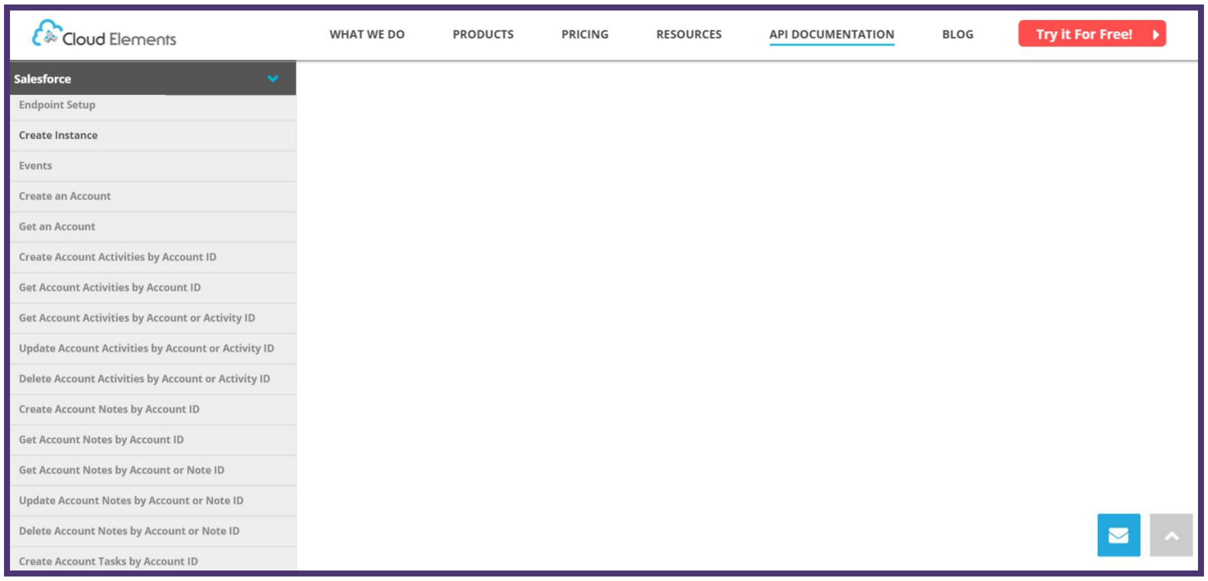
Once on the new page, you can look at the tabs on the left for more information on what you may be doing but but in order to create the correct endpoints you will click on “Elements” on the left hand side and find the correct app that you want to implement. Then, you will click on the correct tab and follow along.



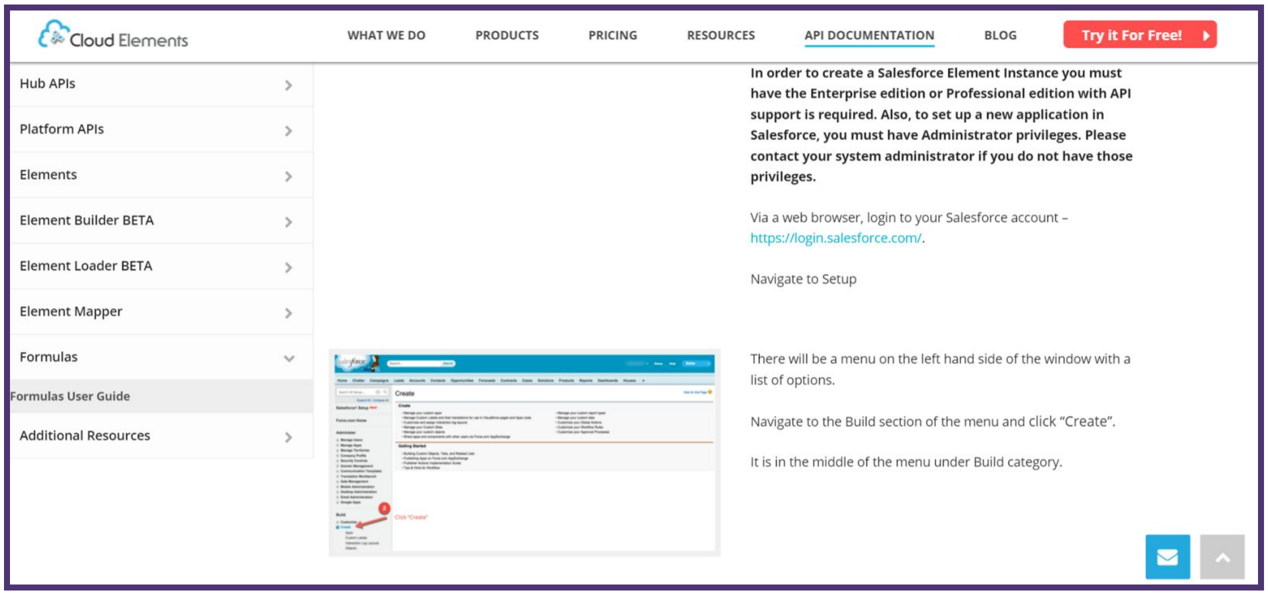
CLOUD ELEMENTS DOCUMENTATION

INTEGRATE ENTIRE CATEGORIES OF CLOUD SERVICES THROUGH UNIFORM APIS.

- Normalized API Calls for all of our Element Endpoints (e.g. Dropbox, Salesfo Zendesk)
- Standardized REST API calls and JSON payloads regardless of the Endpoint
- Unified management of authentication, monitoring, usage to all Endpoint Providers


Cloud Elements
WHAT WE DO
PRODUCTS
PRICING
RESOURCES
API DOCUMENTATION
BLOG
Try it For Free!
Salesforce
Endpoint Setup
Create Instance
Events
Create an Account
Get an Account
Create Account Activities by Account ID
Get Account Activities by Account ID
Get Account Activities by Account or Activity ID
Update Account Activities by Account or Activity ID
Delete Account Activities by Account or Activity ID
Create Account Notes by Account ID
Get Account Notes by Account ID
Get Account Notes by Account or Note ID
Update Account Notes by Account or Note ID
Delete Account Notes by Account or Note ID
Create Account Tasks by Account ID

In order to set up your formulas you can follow the documentation by again clicking on the “Formulas” tab on the left and then follow the “Formulas User Guide”.



In order to create a Salesforce Element Instance you must have the Enterprise edition or Professional edition with API support is required. Also, 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/.

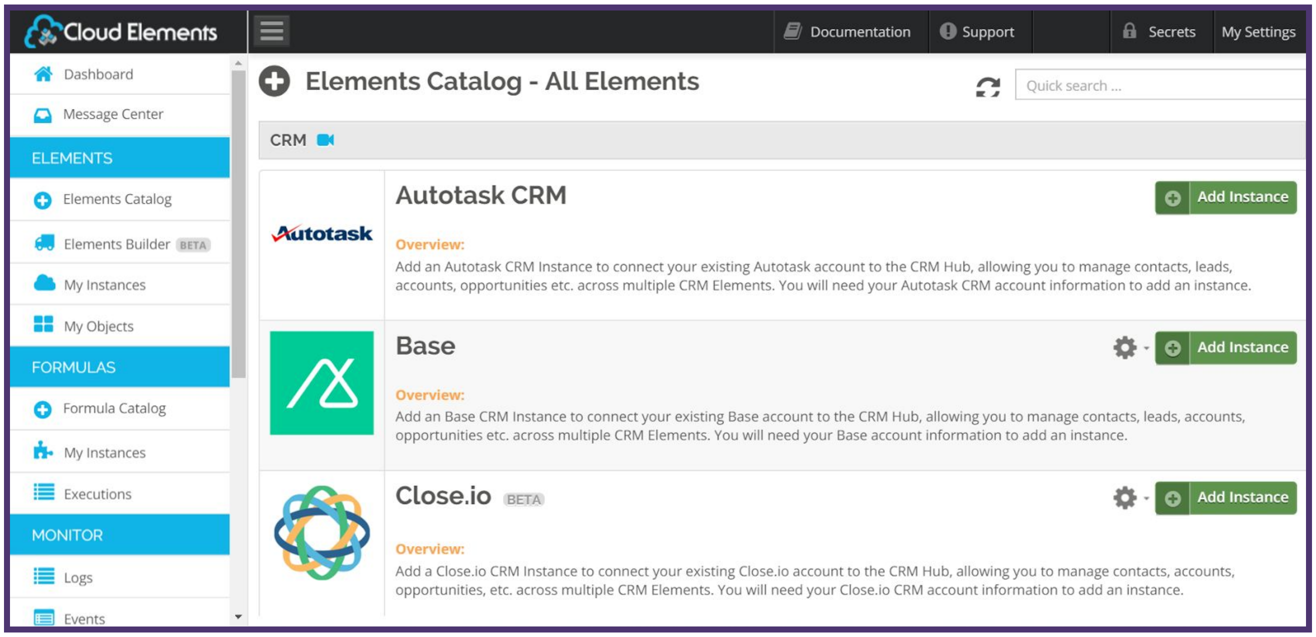
Navigate to Setup

There will be a menu on the left hand side of the window with a list of options.

Navigate to the Build section of the menu and click “Create”.

It is in the middle of the menu under Build category.

To edit formulas using the UI visit either:
Testing: https://staging.cloud-elements.com
Production: https://console.cloud-elements.com



Elements Catalog - All Elements

CRM

Autotask CRM — Add Instance

Overview:
Add an Autotask CRM Instance to connect your existing Autotask account to the CRM Hub, allowing you to manage contacts, leads, accounts, opportunities etc. across multiple CRM Elements. You will need your Autotask CRM account information to add an instance.

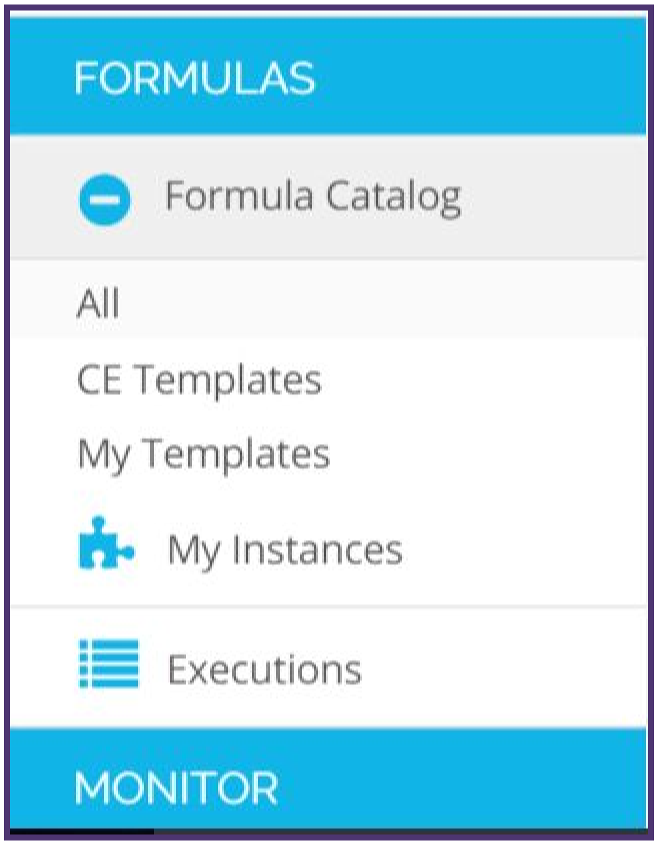
Base — Add Instance

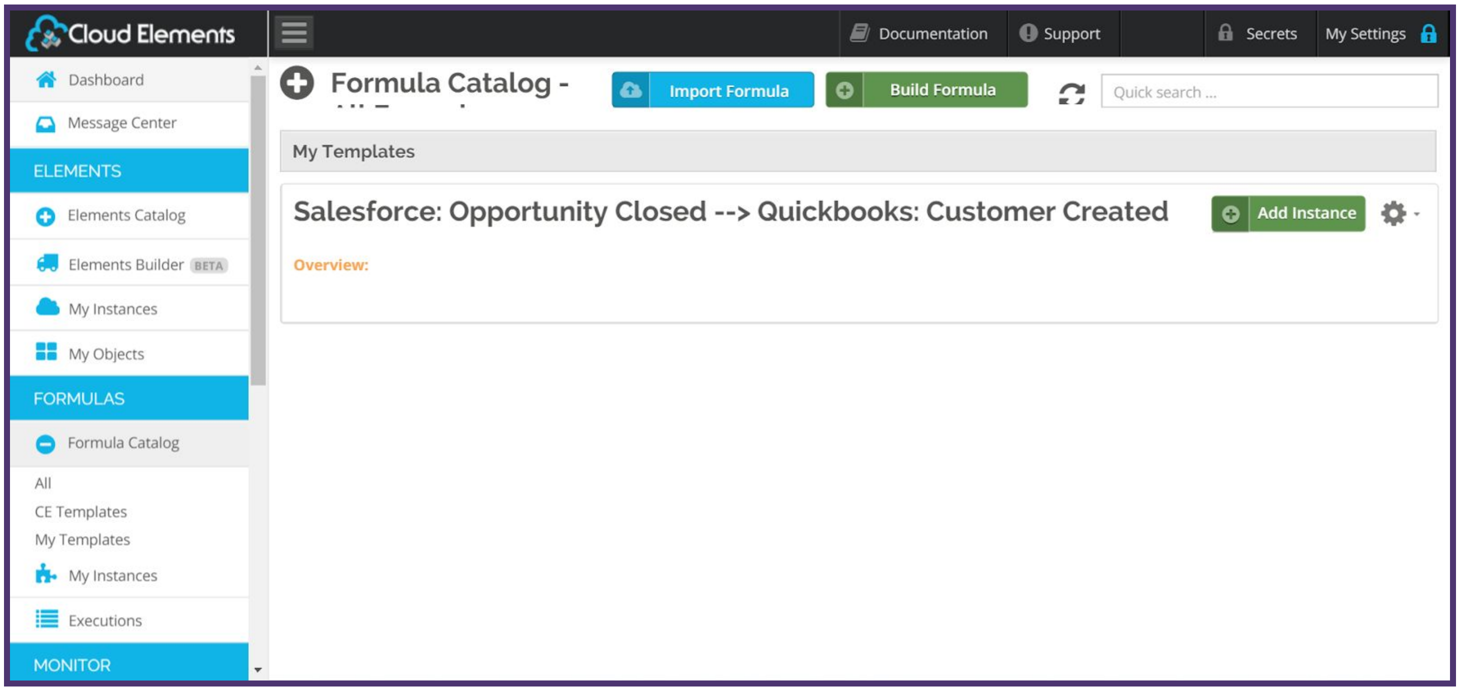
Overview:
Add an Base CRM Instance to connect your existing Base account to the CRM Hub, allowing you to manage contacts, leads, accounts, opportunities etc. across multiple CRM Elements. You will need your Base account information to add an instance.

Close.io BETA — Add Instance

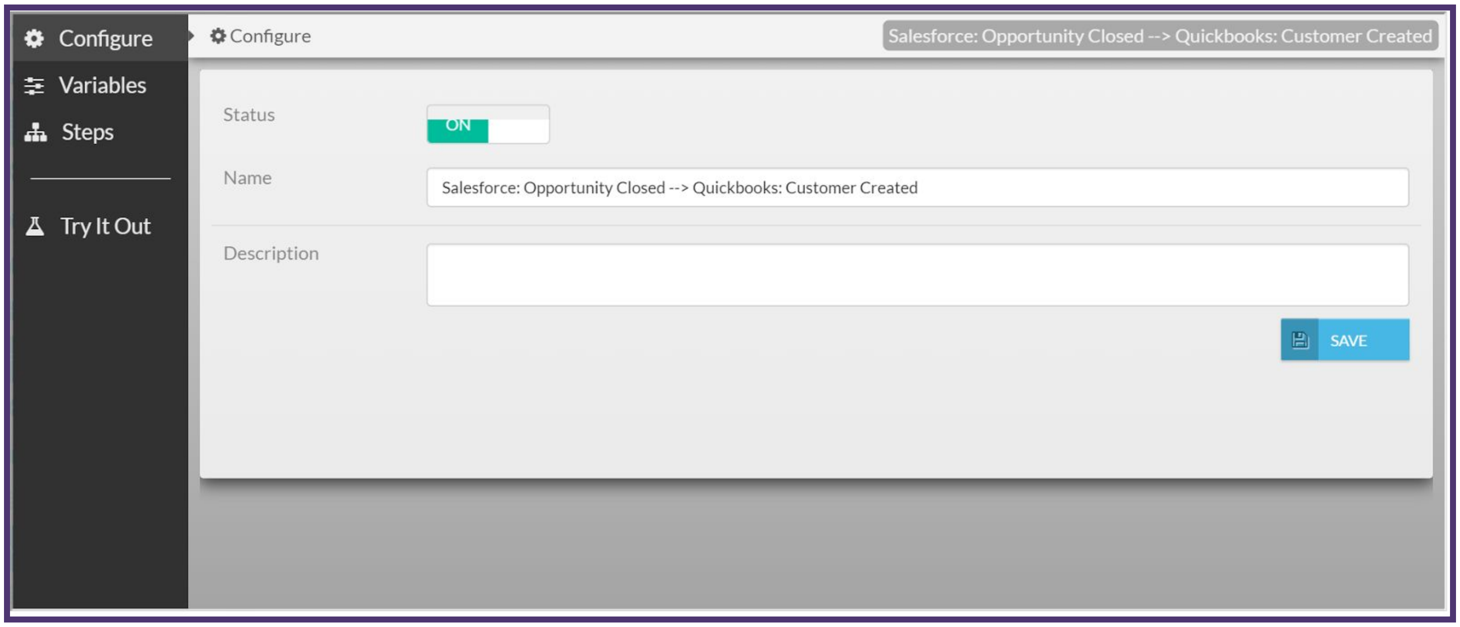
Overview:
Add a Close.io CRM Instance to connect your existing Close.io account to the CRM Hub, allowing you to manage contacts, accounts, opportunities, etc. across multiple CRM Elements. You will need your Close.io CRM account information to add an instance.

On the left hand side of the screen, you will select "Formula Catalog".



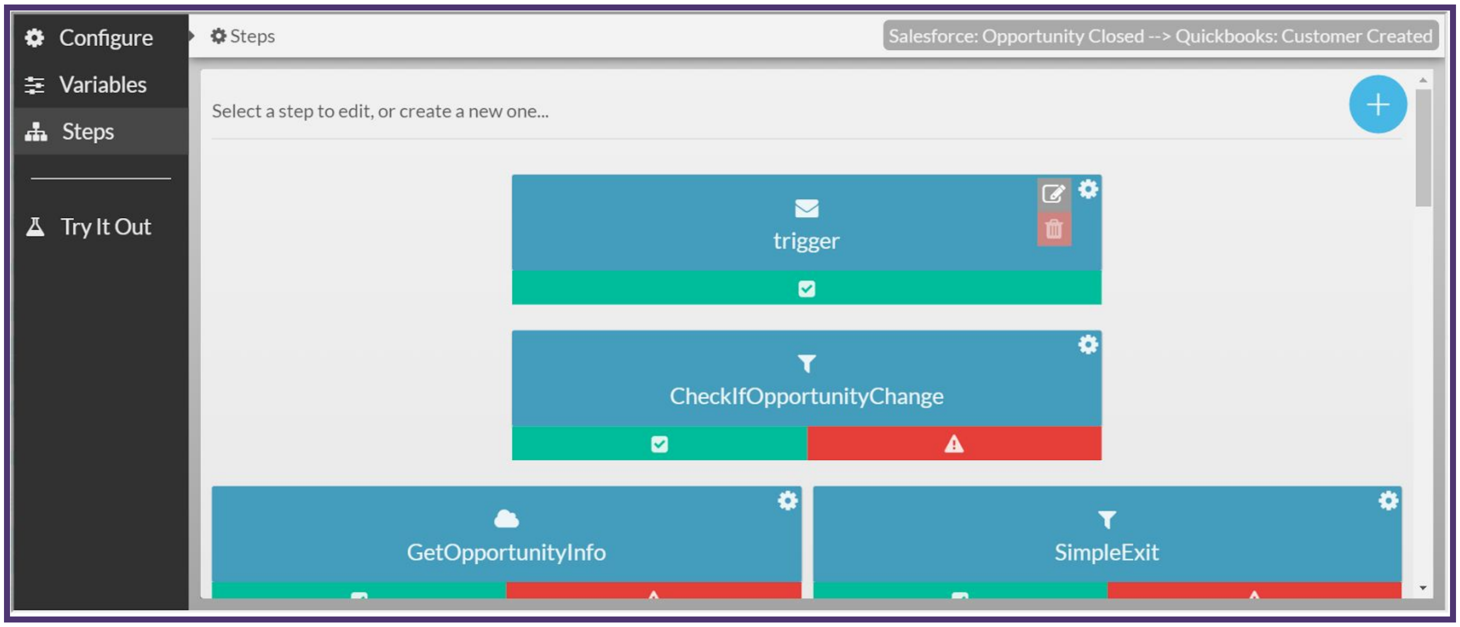


Once on this page, you can edit the formula by clicking on the gear icon next to the button labeled “Add Instance” and click on the edit drop-down.


Configure
Variables
Steps
Try It Out
Configure
Salesforce: Opportunity Closed --> Quickbooks: Customer Created
SAVE

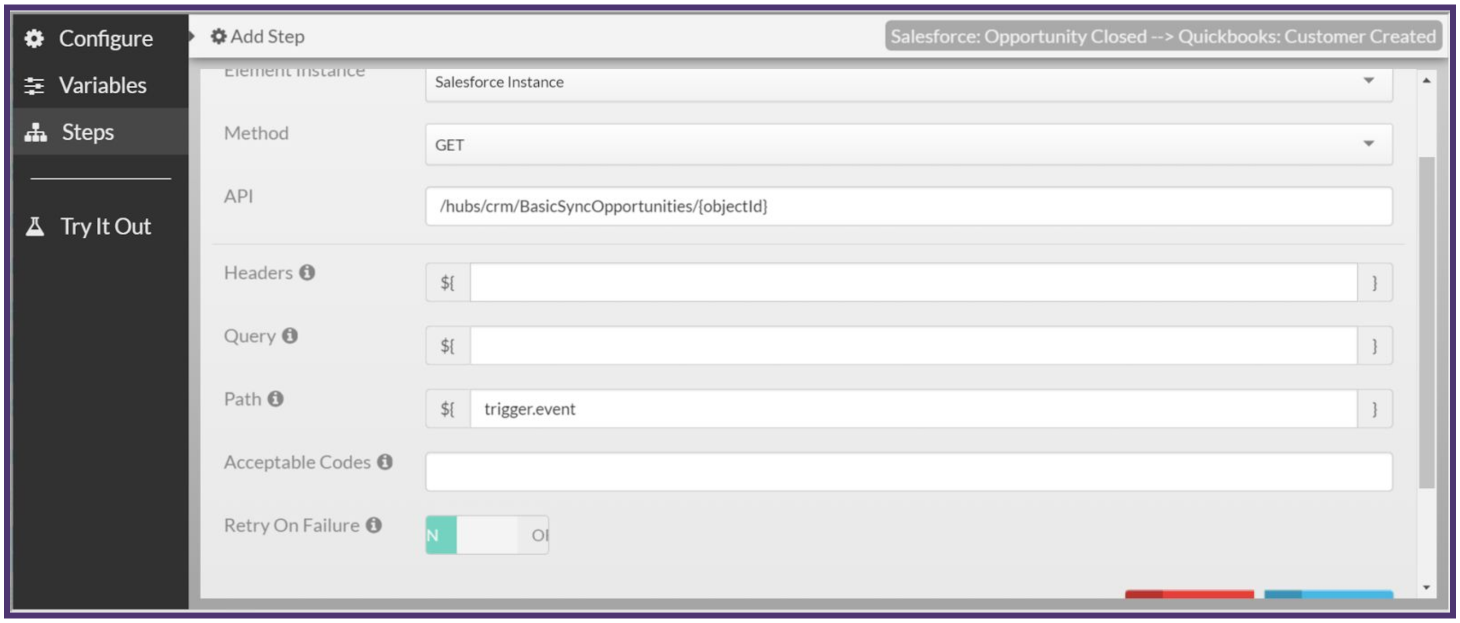
Status: ON
Name: Salesforce: Opportunity Closed --> Quickbooks: Customer Created
Description:

This page will pop up and on the left-hand side you will click on “Steps” on the left-hand side.



Select a step to edit, or create a new one...

To edit each step of the formula, click on the gear in the top right of each step and select the pencil looking icon. (To delete, click on the trash can).


Configure
Variables
Steps
Try It Out
Add Step
Salesforce: Opportunity Closed --> Quickbooks: Customer Created

Element Instance: Salesforce Instance
Method: GET
API: /hubs/crm/BasicSyncOpportunities/{objectId}
Headers: ${ }
Query: ${ }
Path: ${ trigger.event }
Acceptable Codes:
Retry On Failure: ON | OFF

Then, edit each step and make sure to scroll to the bottom and click “Save”. Then you can test out your formula or the steps you are using.

Once you have a formula created, you must create an instance of that formula for your specific app instances. You can see an example of this in ***CloudElements.create_salesforce_to_quickbooks_formula_instance***.

If you need help, make sure to reach out to the Cloud Elements support team at: support@cloudelements.zendesk.com

Core **Developer Guide**

Style Guide

Colors:
Primary purple: **#4d3571**
Secondary purple: **#d5c8e5**
Light grey for button backgrounds, text on dark background, accents/borders: **#F0F0F0**
Subtext on white background: **#A0A0A0**
Single-number metrics text: **#606060**

All element dimensions set in % or em/rem EXCEPT:
header height = 55px
sidebar width = 120px
sidebar button height = 40px

All fonts set in em or rem, buttons sized with padding in em

Button borders: radius=2px,thickness=1px

Program Extension

Needed functionality: Signup confirmation email, forgot password email, security for inviting users (right now just sends a URL with org name in plaintext), pulling full reports in when clicked on.

Adding New Apps: To add a new app to Core, there are a few steps that need to be taken:

1. Setup the endpoint in the app. The way you do this varies per app. The instructions on how to do so can be found in the documentation on Cloud Elements here: http://cloud-elements.com/developer-documentation/ under the Elements category. Find the app you wish to add and look at the "Endpoint Setup". When setting up the endpoint, you will be given an API Key and an API Secret. You will need these for creating each instance of the app, so add them as configuration variables in heroku and environmental variable in the .env file.

2. Once the endpoint is setup, you must look at "Create Instance" under the same documentation. There are three parts for the oauth authentication and instance creation. All apps require you to generate an oauth URL where the user will be redirected to sign-in, but some apps like Quickbooks, require further API calls. Next the user will be redirected to the Callback URL. When this is done, there will be at least one oauth variable that must be pulled out of the URL depending on the app. To create the instance, you must send a JSON file to Cloud Elements through an API call. The content of the JSON file depends upon the app. You can find the

necessary content in the Cloud Elements documentation. The JSON file will always require the variable(s) that you pull from the callback URL after the redirect.

3. Once you find the specificities for the app you are trying to integrate, you must add the code to the project. There are two files that you will be adding code to: **elements_controller.rb** and **cloud_elements.rb**. In **cloud_elements.rb**, you will add the method to generate the oauth URL and the method to create and send the JSON file to Cloud Elements to create the instance. In **elements_controller.rb**, you will add a call and redirect to the oauth URL method in **run** a call to the instance creation method in **callback**.

4. After adding instance creation you must then set up polling. If you do not do this, your instance will not send an event to Core when something is added or changed in the account. To do this, make a call to ***CloudElements.setup_polling(instance_id, app)*** from your instance creation method after the instance is created. You may want to check the instance in Cloud Elements to see if you need to set the event type (the list of things you want to be updated on). If this is the case, you may need to add a condition in ***setup_polling*** method such as there is for Salesforce.

5. If you have a formula that you wrote for a connection with this app and another, be sure to write a method to create that formula instance if they both are connected to Core. An example of this is ***CloudElements.create_salesforce_to_quickbooks_formula_instance***.

**When testing the instance creation you can go into the Cloud Elements to see if the instance is being created. Be careful about creating too many instances of the same app account, you can over call the API. To prevent this, delete old test formula instances and app instances that you do not need.

Database Extension

Extending the database should include not only adding additional data models, but adding or modifying existing methods in the Database model as well as adding the correct handler in the CallbackController if an app is added through Cloud Elements. Needed dependencies should be added to the appropriate data models.