SnapLogic Elastic Integration Platform

Integration & ETL Beginner Lab Guide

Introduction

In these labs you will learn the core functionalities of the SnapLogic Integration Platform. After completing this lab you will have an understanding of how to create, develop and use the core Snaps and the platform to build your pipeline.

Important prerequisite!!!

Please make sure you are on the CORRECT SNAPLEX.
Check your Top Right Corner next to the SnapLogic Toolbar it should say:

If you dont have the “Cloudplex” Snaplex, please press the

 button and change your Project Path to cloudplex

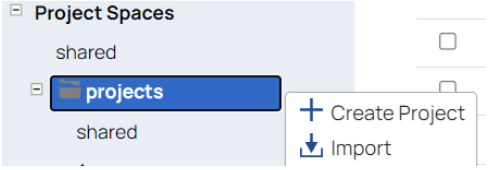
.

If this is not possible, please proceed and come back to this step AFTER creating your project folder to make sure you are on the correct Snaplex.

Press Save afterwards to make the change 

Please make sure you have this
before moving on, again if you need to create a pipeline with a different name/project, create it first and come back to this step.

Below is the example to create a project from "Manager".



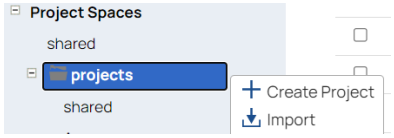
Lab 1: Collecting API Data into SnapLogic

Our first pipeline will be responsible for processing Data from an API.
The steps we will be performing are the following:

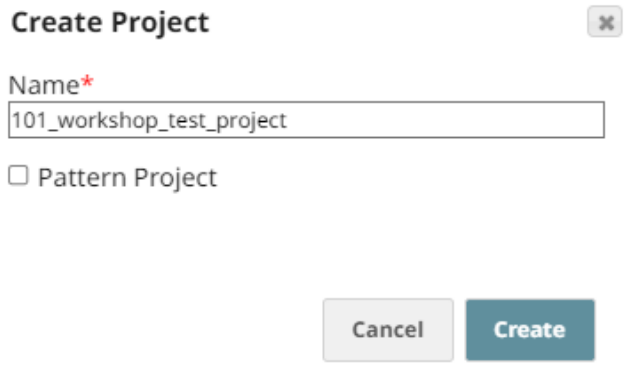
1. Retrieve a dataset of Coin Base API Data
2. Convert the dataset to an appropriate data format
3. Filter out and sort the data we're interested in
4. Transform and export data

Create New Project

1. In Snaplogic, press **Manager**. In the left navigation bar, expand **Project** under **Project Spaces**. Hover over the 101 Workshop space until the drop-down arrow appears

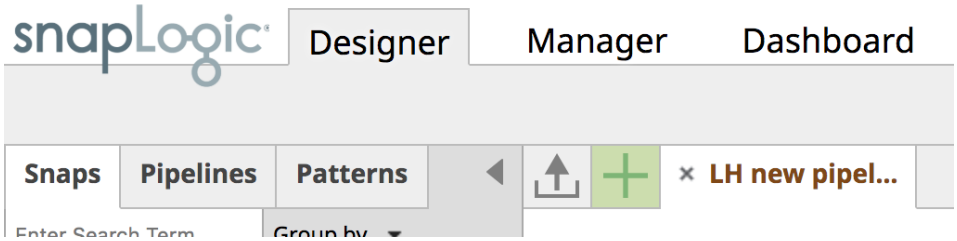


Press the **arrow** -> **Create Project**. In the dialog, enter a unique project name or initials and press **Create**.



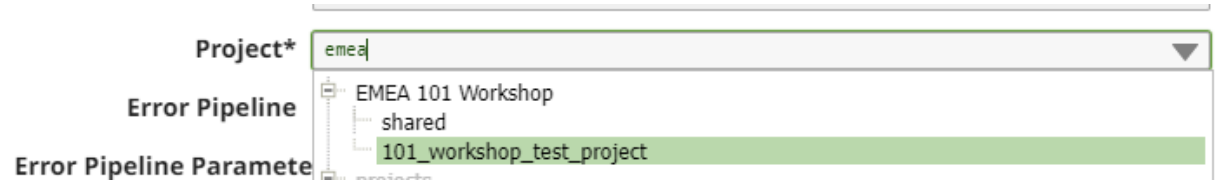
Create New Pipeline

2. Switch back to **Designer**.
3. On the top left on the canvas, press the **+** icon.



If you are not prompted for anything, press the **Pipeline Properties icon**. If you are prompted by a dialog, skip to 13.

4. Under **Label**, enter the name of our first Pipeline, e.g CreateCoinBaseModel
5. Under **Project**, press the **arrow** and select your recently created Project *(only if you were prompted by a dialog when creating the Pipeline, if you had to manually press the Pipeline Properties, you won't be able to change the project.)*



Press **Save**

REST Get Snap

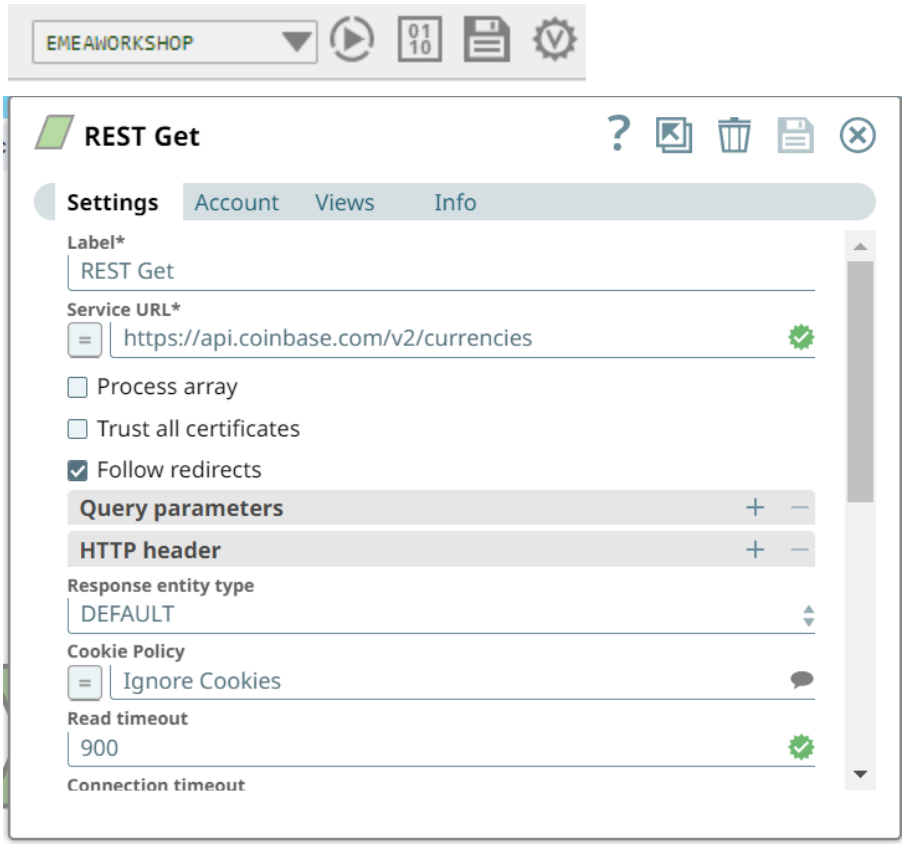
We will now retrieve our dataset of Currency codes and names from a Rest Get response.

6. Now the first step towards building our pipeline will be to drag the **REST - Get** Snap to your canvas

For most Snaps, a notification will pop up stating that we need to add an account to our Snap. This is needed in cases where you have a layer of security or an endpoint that requires authentication. SnapLogic supports a vast variety of authentications and endpoints, for example OAuth2 etc. Since the CoinBase API doesn't require any authentication we won't use that for now.

7. Press the Snap on your canvas (or **Settings**, if you already have a dialog open) Under *Service URL*, paste the URL below. Also, make sure the right plex is chosen to run this pipeline. See below. The query will return the data as seen before

   https://api.coinbase.com/v2/currencies



8. Press the Save icon (disk on top right of the Snap settings) and close the dialog
9. Depending on your user settings, SnapLogic will start a Validation whenever you save your Snaps' settings or your Pipeline. If this setting is off, you will have to manually press the Validate icon (highlighted in green below).

10. After a while, your Snap should turn green. Press the dark green document(red dot in picture below) outline on the right of the Snap to see the data returned.

As you can see, some of the returned data contains data that we're not interested in. But also it contains headers we are not interested in as well.
Press **Close**.

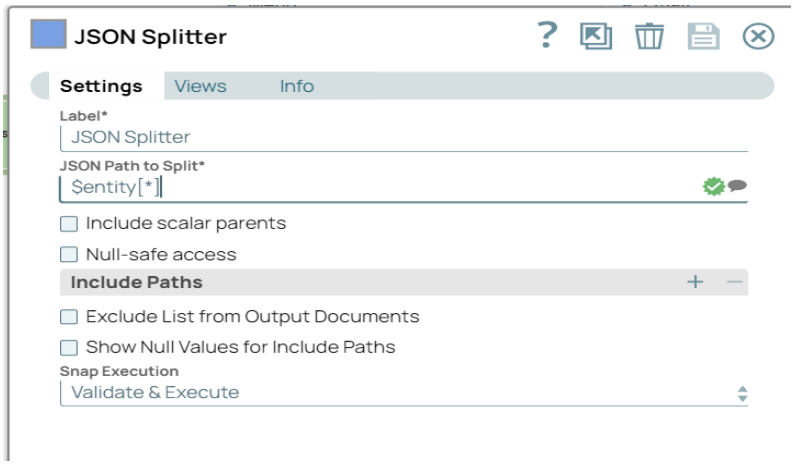
JSON Splitter Snap

The JSON Splitter Snap will allow us to specify what data in a response we're interested in using, it will "split out" the fields that we choose. As we remember from our previous data, the various headers and the data is still unstructured, therefore this is good practise to only work with the data necessary.

11. Repeat the steps outlined in the previous chapter but instead of Rest GET Snap, search and drag a *JSON Splitter Snap* onto the canvas. It should be connected to the Rest GET Snap as below(in most cases, our IRIS Artificial Intelligence will recommend which Snap has the best fit).

12. Press the **JSON Splitter** to enter the properties dialog.
13. Next to the *Json Path* setting, press the empty area or the speech balloon.

14. Select **entity[*]**, as shown on the screen below



15. Press the output document again and view the result of the **entity[*].** Press the save button.
    Now we can see that the data is more structured and it contains the required information we need to proceed. However note that there are some empty values that we are not interested in.

JSON Splitter output0

Preview Type: Table

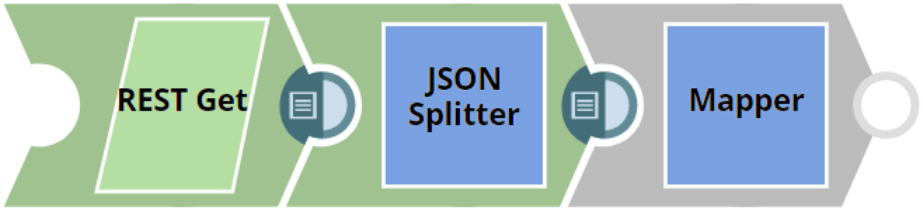
id	name	min_size
AED	United Arab Emirat...	0.01000000
AFN	Afghan Afghani	0.01000000
ALL	Albanian Lek	0.01000000
AMD	Armenian Dram	0.01000000
ANG	Netherlands Antille...	0.01000000
AOA	Angolan Kwanza	0.01000000
ARS	Argentine Peso	0.01000000
AUD	Australian Dollar	0.01000000
AWG	Aruban Florin	0.01000000
AZN	Azerbaijani Manat	0.01000000

16. Close down the property dialog.

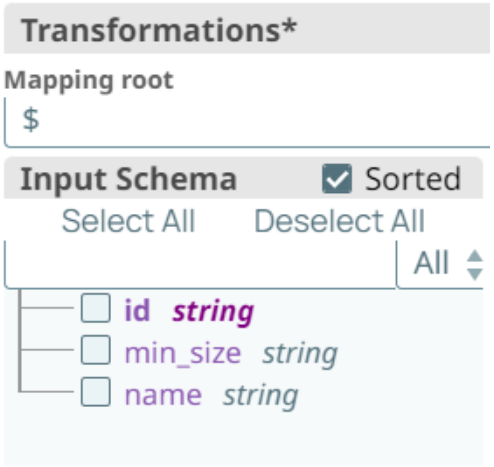
Mapper Snap

We will use the Mapper Snap in order to transform our data into relevant data. This is needed as we want our dataset to be properly presentable, but also contain the fields that we are interested in as mentioned earlier.

17. Drag a **Mapper** Snap to connect with the *JSON Splitter* as shown below



18. Open the *Mapper Snap* and you can already see SnapLogic has populated the Input Schema with categories of values which we can filter on.



19. We are going to work with 2 fields for this lab, which is *ID and Name.*
    Therefore, add 2 entries by pressing the + sign in the middle.
    Next, drag the *ID* and *Name* from the Input Schema to the empty fields under Expression.
    If you now hover the mouse over each field, you can see that SnapLogic has already presented you with a preview of how the data will look like. Here it is possible to use our JavaScript library to modify and change the data to the appropriate format.

    Next we will populate the *Target Path* with the same variables. This is for SnapLogic to know where to put this data into variables that we can work with later on. The final result will look like this:

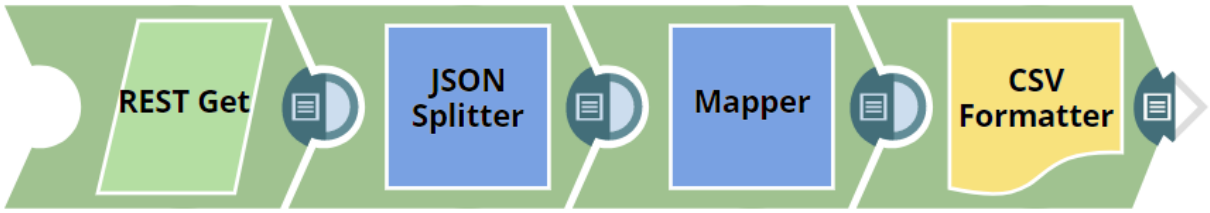
20. Now if we press the output document once again, our data will look like this:

CSV Formatter

Since our dataset is now ready to be added into our endpoint we can now prepare the data to be in the right format for the endpoint. For example, if this would have been inserted into a Database, we would use one of the Database Insert snaps (SQLServer,MySQL, Oracle etc) but for this lab we will add the dataset into a CSV.

The CSV Formatter reads a document stream at the input, formats and writes CSV data to the output.
There are many options that we can work with here, for example if we can change the Delimiter to something else than the default ",". For this lab, we can keep it as is.

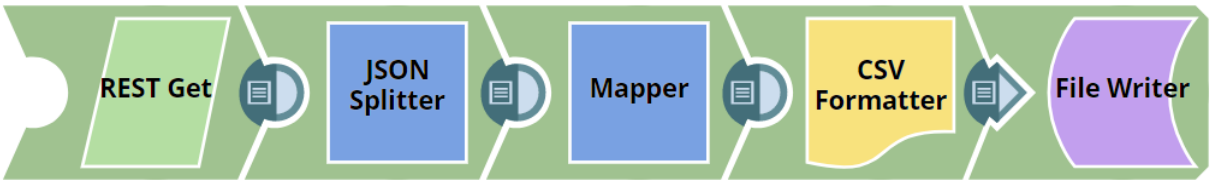
Notice that the output changes to diamond jigsaw icon rather than a circular one, the reason for this is because our data is changing to a binary format as CSV is.



File Writer Snap

The File Writer Snaps accepts a binary input (indicated by the diamond jigsaw icon rather than the circular rone). It supports writing to multiple different types of targets and protocols, such as ftp, http, Azure Blob Storage, AWS S3, Google Storage, etc.

21. Connect a **File Writer** Snap to the *CSV Formatter* Snap.
22. Open the settings for the Snap. Under *File name*, enter **<yourInitials>.csv**
23. Before you close and save the properties, note the default value of *Snap Execution* as set to *Execute only*. As for the *Snowflake - Execute* Snap, this is the default value when a Snap performs a write/execute/delete operation in order to not mistakenly edit mission critical data or systems in the validation mode.
24. Perform a final Validate to ensure all Snaps are turning green.



Well done! Our first pipeline is now ready to be executed.

Execute the Pipeline

In order to actually generate a model based on the full dataset (and save the file), we need to Execute the Pipeline.

25. Press the **Execute button (play icon)**

26. Press the **Execution Statistics button (01 10 icon) after the execution has finished**

    This will show you the live execution details of the Pipeline. For each Snap, it will tell you the time consumption, CPU/memory/network impact as well as the number of bytes and documents it processed.

27. You can now view your newly created file by going to the "*Manager"* and navigate to *Project Spaces -> 101 Workshop ->Your Folder* and then under the *Files tab* you can either download the .csv file or view it in SnapLogic file previewer.

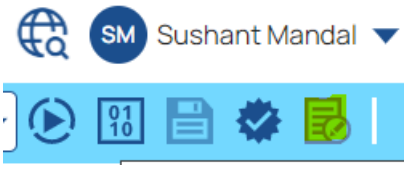
Lab 2: Modifying the Pipeline and create a Triggered Task

Our next lab will focus on pulling parameter from the pipeline properties.

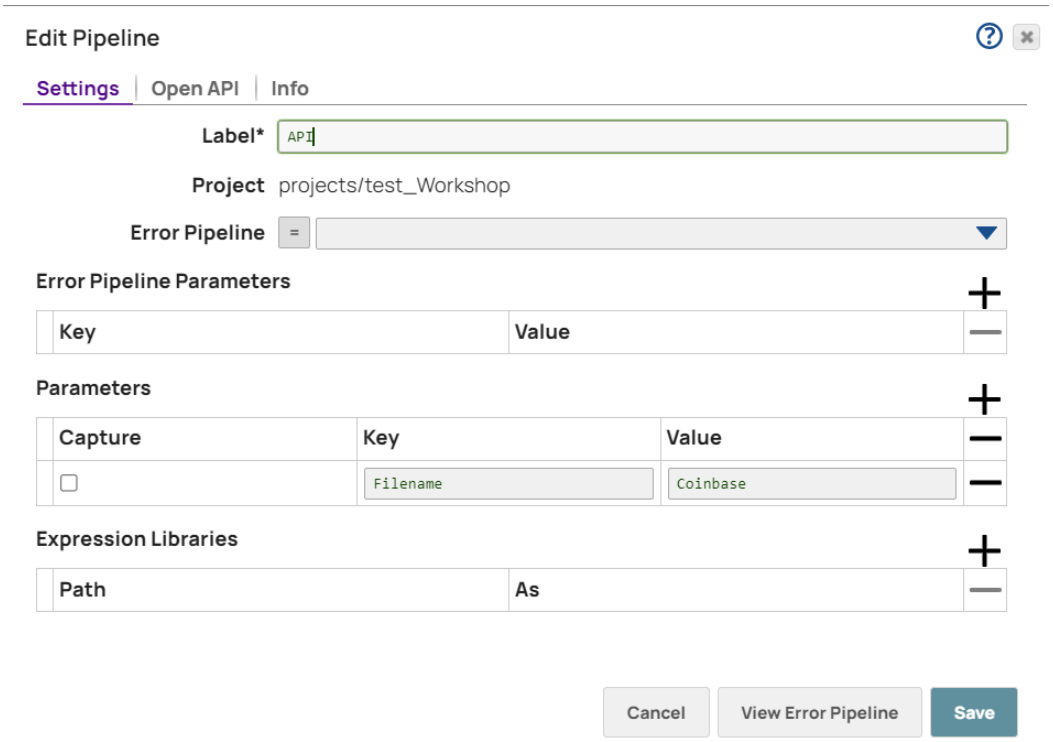
Pipeline parameters

If you remember from our previous lab, when creating a Pipeline, the first thing we do is to open the *Pipeline Properties.*

1. To add custom parameter to capture throughout our CoinBase pipeline we need to open this setting again



2. Press the “+” sign and add Capture under Parameters and add a new parameter. As Key we want the "Filename" variable to be the key variable throughout the pipeline which can be changed. In Value we add a hardcoded "Filename", we do this since we need a value in order to test the pipeline during validation and we will add “Coinbase” as the hard coded "Filename":



3. To above parameter to name the output file(csv).

Creating a Triggered Task

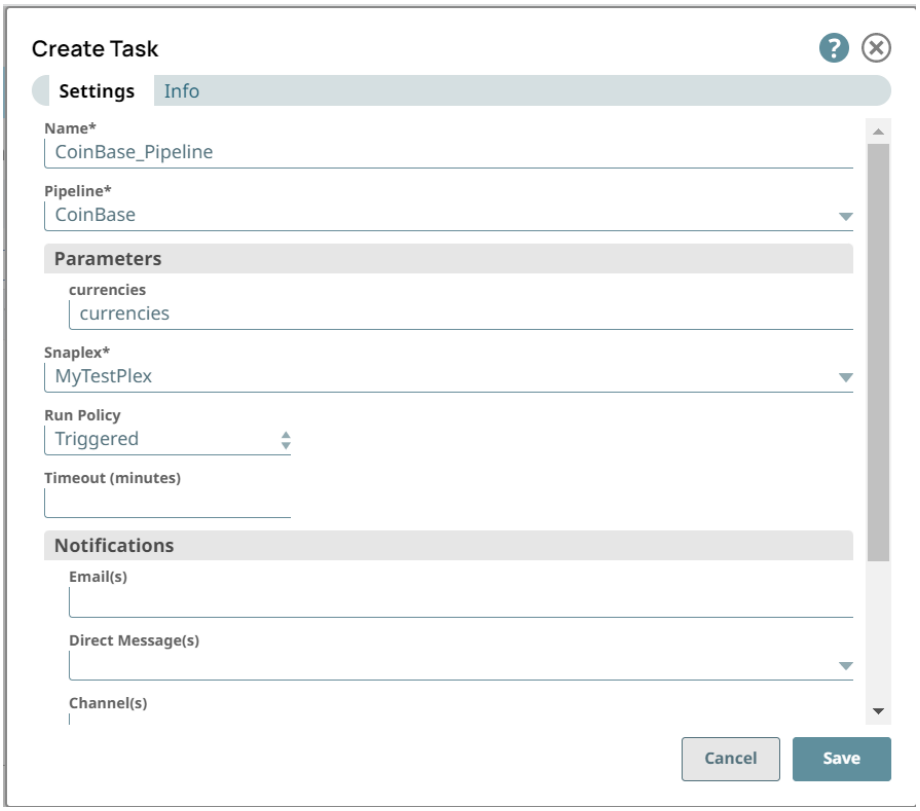
In SnapLogic, there are 3 different types of Task. In this lab, we will focus on Triggered Tasks.

Triggered Tasks offer the method of invoking your Pipelines with an HTTP endpoint. Your Pipeline can now be treated as a service or API Call, to and from which clients can invoke to send or get data. Likewise, you can invoke the Pipeline underlying a Triggered Task from an on-premise system or from the Cloud.

10. Expand the SnapLogic toolbar by pressing the arrow to the far up right corner of the canvas:

11. Press the *Create Task* button

12. Name the Task for example *CoinBase_Pipeline Task* and make sure its pointing at your pipeline and being created as a Triggered/Scheduled Task. Press Save and execute



13. Open the *Manager* and navigate to your Project folder. Open the newly created file by pressing the arrow down button ▼ next to the name and then *Download*.

14. Now, change the file name by invoking the trigger API using Postman/Powershell command.