**Running Jobs with Informatica Cloud REST API**

Informatica Cloud offers REST API for us to interact with the platform programmatically. At this point in time, the latest official reference is found [here](https://kb.informatica.com/proddocs/Product%20Documentation/6/CDI_Spring2018April_RESTAPIReference_en.pdf). In this post, we are going to use Python to trigger jobs through API.

Triggering a job is a 2-step process. We need to authenticate with a username and password to retrieve a temporary session id. Then, we can make an API call by using the id. Informatica has an example of running jobs through API with curl [here](https://kb.informatica.com/howto/6/Pages/18/495902.aspx).

Once you understand this process, all the APIs work the same. If you are looking for something specific, you can always look for it in the documentation.

**Loading Modules**

We are using the [requests](http://docs.python-requests.org/en/master/) module for REST API calls. We also need the json module to convert dictionary to string format of JSON.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

**Authentication**

The base URL is the same one as you use to log in through the browser. You do not need to specify org as the user name is unique. The user information needs to be passed as a string, rather than as a dictionary.

On a successful attempt, it returns a session id and server URL which are to be used for subsequent API calls.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

**Triggering Job**

The end point for starting a job is ‘api/v2/job’. This comes after the server URL from the previous call. The temporary session Id is passed in the header. This is a POST request. In the body, we pass the id of the job and the job type. The Id can be found at the end of the url of the job through the UI.

The value for ‘Accept’ in the header needs to be ‘application/xml’ instead of json.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

**Execution**

Now that we have functions to authenticate and trigger jobs, we can execute the entire code. You need to define your user name, password, task id and task type (for example, MTT for Mapping Configuration Task). Then you are ready to go!

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]