**How To Get Survey Response Data From Qualtrics With Python**

In the [previous post](https://www.mydatahack.com/moving-data-in-s3-and-redshift-with-python/), we had a look at Python code examples of basic data engineering with AWS infrastructure. By using Qualtrics API, I would like to present a coding example of API data ingestion into S3 and Redshift. This code can be scheduled hourly, daily or weekly in a server or AWS [Data Pipeline](http://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/what-is-datapipeline.html).

[Qualtrics](https://www.qualtrics.com/) is an online survey software which allows you to send surveys via email or SMS, receive responses and generate reports. The aim of the ingestion is to get the survey response data into Redshift.

**API Reference**

Qualtrics API is a simple REST-based API. Once you generate an API token, you are pretty much ready to go. They have comprehensive API documentation. In a nutshell, you can use the requests module to make a POST requests with the token in the header to get the data as a csv file. Check out Python API reference at [Using the API Code Samples](https://api.qualtrics.com/docs/using-the-api-code-samples) and survey response export code sample at [Getting Survey Responses](https://api.qualtrics.com/docs/response-exports).

I found Qualtrics API was unreliable hard way. My code initially failed randomly because I didn’t have the for loop to keep the request repeating until it connects. In the code example, I set the maximum to 200 (see the bulk\_export method). It usually works within 20 times.

**Key Points**

In this example, we are using truncate & load because the data comes in one csv file with all the responses. We cannot obtain data incrementally. But, this is ok. We can leverage the power of Redshift copy command from S3, which is extremely fast. Truncate & load should be fine unless you have massive volume or other business requirements. If you want to do an incremental load, you can pick the record in the exported file according to the last updated time for insertion, which can be done relatively easily.

The program exports response data in a csv format into a local directory, push it to a specified S3 bucket, and execute copy command after truncating the table. It can ingest responses from multiple surveys. The argument for survey project names has to be concatenated by ‘,’.

The get\_project\_id method will return a list of survey ids based on the survey project name, which in turn uses to get the survey-specific response data.

The format\_colnames method takes care of formatting the column as the exported file has multiple rows for the column names. I am using the pandas package to do the data manipulation.

Note that you also need AWS Access Key Id & Secret Access Key for the copy command.

OK, here comes the code.

Enjoy!

**Code**

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

**[/cc]**