

Take Home Project

1. How long, roughly, did you spend working on this project? This won't affect your evaluation; it helps us norm our problems to our expected time to complete them.

I spent an estimated total of 3 hours working on the project. This includes the hours spent on reading the API documentation and Basic Authentication, brainstorming ideas to extract data, filtering JSON fields, generating unit tests, and coding the final result.

2. Give the steps needed to deploy and run your code, written so that someone else can follow and execute them.

Refer to the readme file in the repository.

3. What could you do to improve your code for this project if you had more time? Could you make it more efficient, easier to read, more maintainable? If it were your job to run this report monthly using your code, could it be made easier or more flexible? Give specifics where possible.

Following assumptions were made for the project:

- API key exists and is valid.
- The API endpoints are valid.
- There exists at least a unique emailMessageId.
- Each emailMessageId has at least one message variant with emailMessageVariantId.
- The statistics field exist and has the required fields that need extraction.
- The required data fields to be extracted exist and the values are not NULL.
- The data is in JSON format
- The data remains uniform for an endpoint.

Given the time, I could have checked and validated these assumptions so that the program could handle various edge cases and perform accordingly.

A readable, detailed, and clear documentation is equally important as the source code. It plays an important role when extending a module, fixing bugs and patches, refactoring code, and making any major changes in the codebase. Having such will help in areas of maintenance, development, and knowledge transfer to other teams.

In the source code itself, comment blocks explaining classes, parameters, methods, and possible error are also equally helpful.

We can automate the project to run every month for the purposes of generating reports. If there are no major changes to the source data, we can schedule a job using tools like Apache Airflow. This will reduce time needed to manually deploy and execute the code.

4. Outline a testing plan for this report, imagining that you are handing it off to someone else to test. What did you do to test this as you developed it? What kinds of automated testing could be helpful here?

We can create the following test plans:

- I. The NPG class has four methods that retrieve different API endpoints. We can check and see if the endpoints are being returned as needed.
- II. Build a test case to check that the API key exists and is valid.
- III. Build test case to check the API endpoints are valid.
- IV. Check to see if there exists at least a unique emailMessageId and each emailMessageId has at least one message variant with emailMessageVairantId.
- V. Check and see if the required data fields to be extracted exist and the values are not NULL.
- VI. Check if the response object is in JSON format.

To test the functions that retrieve specific API endpoints, I built four-unit tests using Pytest to validate the results of the API endpoints with the custom functions.

We can employ automated testing tools. We can integrate Continuous Integration in the project using Gitlab, Travis CI or Heroku. This will allow us to run automatic tests for every commit, push and PRs and examine the build files, logs, check errors and warning, fix bugs when needed.