1. Introduction

I read the problem, understood what was asked and solved it. However, after I was done, I felt like this was a great opportunity for me to use FastAPI and further showcase my experience with RESTful API's as I would be exposed to that kind of work if I join PollyEx. To be frank, it was also an excuse to play around with it as I've been wanting to for a while now. If you are not familiar with it, FastAPI is a modern, high-performance, web framework for building APIs with Python 3.6+ based on standard Python type hints.

2. Design and Implementation

The solution consists of a RESTful API with a single endpoint "/payment-details" that you can call with a HTTP GET request to get a loan's payment details. They are calculated from the loan's amount, interest, down payment, and term which need to be supplied through URL query parameters.

There are multiple ways to interact with the API. I didn't want to stray too far from what was asked, for that reason I implemented a client that will take standard input, parse and validate it, make a GET request, and return the JSON response to the console. Alternatively, you can use Curl or Postman or any software to make the request. More on this later.

I took into consideration the complexity I would be adding to the solution. To minimize that, everything is containerized using Docker, and you can build and run it using the provided docker-compose file. A requirements file is also provided so all dependencies will be managed for you. You won't have to log into the container either as I provided bash scripts that will let you run the client and execute the tests from outside the container.

3. Running the Application

This was developed in my home server that runs Ubuntu Server 18.04 and tested in my MacBook. It was not tested in a Windows machine as I don't have one available right now, but it shouldn't cause any problems and the steps should be similar.

1.1. Requirements

- Internet connection
- Docker
- Docker Compose

1.2. Running the API

1. Open a terminal

2. Navigate into the *pollyex-project* folder. Make sure not to change the name of the folder.

3. Execute the following commands to build the image and run the container
   ```
   sudo docker-compose up -d --build api
   ```

4. Make the client and test scripts executable
   ```
   chmod +x client.sh
   chmod +x test.sh
   ```

1.3. Interacting with the API using the client

1. Make sure the API is running and the client script is executable (Follow the steps in section *1.2. Running the API*)

2. Run the client script
   ```
   ./client.sh
   ```

3. Enter the loan details. Make sure to press Ctrl+D to signal end of input

4. Once end of input has been signaled the client will make the request and display the JSON response if successful, otherwise an error will be displayed

5. Press Ctrl+C to exit the client

1.4. Interacting with the API using Curl

1. Make sure the API is running (Follow the steps in section *1.2. Running the API*)

2. Run the following command to call the endpoint.
   ```
   curl -X GET “http://localhost:8000/payment-details?amount=100000&interest=5.5&downpayment=20000&term=30”
   ```

3. Optionally, the output can be piped to *json_pp* to make it easier to read
   ```
   curl -X GET “http://localhost:8000/payment-details?amount=100000&interest=5.5&downpayment=20000&term=30” | json_pp
   ```

4. Tests

1.5. Running the tests

1. Make sure the API is running and the test script is executable (Follow the steps in section *1.2. Running the API*)

2. Run the test script
   ```
   ./test.sh
   ```