GPT-Assisted Test Data Auto Generation in POSTMAN for Web API Testing

Haobo Liu
UNI: hl3645

Synopsis

1 Introduction/Background

1.1 RESTful API

In many current application development processes, API is an important part of the tools that many developers tend to use. Either they need to leverage some services, to access some data, or just to simplify their development, API is a great tool allowing developers to use. In detail, developers can use Web APIs that follow the RESTful behaviors (CRUD) to interact with remote services, which can enhance the performance and efficiency of their applications. However, for most Web APIs, users/developers do not have direct access to the source code of the interfaced applications. Most APIs have official API documentations which record the expected behaviors of the API. For example, for web API, it will provide details on the CRUD methods to use each API function, the endpoints (URLs) to access the API, and the request parameters passing along to get the correct responses. Overall, the documentations do not provide enough information or methods to help monitoring the status of the API. As a developer, when you use the API in your application, the common practice is to also compose test units to monitor the APIs used.

1.2 Postman API Platform

Recent years, the new tool, Postman, provides an excellent GUI for testing Web APIs. It standardized the process of testing various APIs under the same workspace. With Postman, developers can easily test their APIs by sending requests to the API endpoints and receiving responses from the server. The tool provides a variety of features such as automatic request generation, environment management, and collaboration capabilities, making it a powerful tool for API development and testing. Furthermore, Postman is particularly useful for testing RESTful APIs, which rely on the HTTP protocol to interact with client applications. It supports a variety of HTTP methods such as GET, POST, PUT, DELETE, and PATCH, as well as common data formats such as JSON and XML.

In Nov 2022, Postman published a new feature called ‘Flow’, which allows users to build on their current collections of API test suites, and to formulate the API requests in a flow-chart like manner. These visual aids allow users to test API calls and process the data inside Postman workspace.

1.3 GPT-3

OpenAI’s GPT-3 model is a powerful state-of-the-art LLM. It is capable of generating high-quality human-like text. It is designed to understand the nuances of natural language and can produce coherent and engaging text across a wide range of topics. For developers, GPT-3 is able to understand programming-specific prompts and generate data accordingly; and it also has a decent amount of knowledge of API testing. Therefore, since OpenAI has granted access to the GPT-3 model through API, many developers have been using GPT-3 to generate codes and test data.

2 Overview of the Project

In this project, I plan to leverage GPT-3 in the API testing workflow in the Postman Flow feature. I intend to develop a reusable flow map that can integrate GPT-3 into testing usage of Postman. Although Postman provides a great platform for RESTful API testing, users still need to generate their own test data, as well as format it to send the right requests. However, automatic test data generation was already a popular concept[1]. So the intention of this project is to help users of Postman achieve similar automated test data generation inside Postman without dealing with the complex coding.

2.1 Novelty

Web API testing is an important way to monitor the service provided by the interfaced server. While Postman provides users with visual ways to test their Web APIs, the test data generation is still mostly manual. In this project, by leveraging the knowledge base of API testing embedded in GPT-3, the implemented Flow chart created can help users to integrate GPT-3 inside Postman to generate test data more efficiently. Although there are some Postman Workspace touched on testing OpenAI APIs in their collections, from my research, most of them did not have ways to use the responses from OpenAI API within the Workspace. Since the new *Flow* feature allows users to connect different API tests as blocks, and gives methods to process the returned data, the developed Flows can help users to put OpenAI's GPT-3 and other models into their Postman usage.

2.2 Value to the community

The value of this project shows in demonstrating how users of Postman can leverage GPT-3 without dealing with the complex coding scripts. For most users on Postman, they want to enjoy the nice UI provided by Postman, and to avoid as much coding as they can. Therefore, the developed Flow in this project can further help them achieve more in their API development/testing process.

Since this project is mostly focusing on developing an extension-like Postman Flow that allows users to leverage GPT-3 prompt completion inside Postman. By reading this project paper and testing the Flow themselves, users can learn about how this template Flow works, and by studying the example Flow charts in the workspace, users can learn about how to implement this template in their API testing.

Research Questions

To measure how this extension of Postman Flow can help API Testing in the field, and specifically, how GPT-3 integrated Postman Flow can help users to test their API more efficiently, in this project, I started the extension development with the following research questions. Additionally, since Postman Flow is a relatively new feature and extending on GPT-3 so far cannot bring some quantitative impact. The following research questions have been focusing on the characteristics of this development.

3 Research Questions

RQ1: Is it possible to make Postman Desktop Application use GPT-3 prompt completion function from OpenAI? And how can Postman process the data returned and use it for other API's requests?

RQ2: Can GPT-3 data generation be used in automated testing to improve efficiency of API testing?

RQ3: Compared to traditional script testing using coding language, what are the benefits of Postman Flow?

To answer the above questions, I conducted a user survey study on the use of suchPostman Flow compared to typical manual API test writing. The aim of this survey was to ask people who had similar experience in Web API testing about their views on the developed Postman Flow after using it to their imagined usage.

3.1 Methodology

Since this development in the project is still a relatively new feature in Postman, there is not a lot of dataset or implementation I can compare to in this project. Therefore, I formulated a user study survey to collect qualitative data on how people thought of this product after interacting with it. I intended to survey people who had experiences with RESTful API testing in the industry, and preferably, people who are familiar with automated testing or Postman. After collection certain size (6) of survey responses, there are some interesting findings related to the research questions:

3.1.1 Participants:

In the study, I reached out to several co-workers who are from my past internships, as well as some peers from school who claimed to have Web API testing experiences. The participants all have 4+ years of experience software engineering, and most (⅚) of them have experiences in Web API testing. Also, one of the 6 participants have used Postman in the past. The reason for choosing this group of participants is because this project helps developers dealing with Web API testing to have better tools to automate their testing process.

3.1.2 Before Survey

Before the participants did the survey, I reached out again for people who got back to me. In the request, I asked them to try the Postman Workspace I had created for this project on API Test Data generation. I also gave them guidance on how to use and interact with each element in the Postman Flow. I also gave them extra time to test the workspace themselves to collect their thoughts on this project. After the demo and testing, I then asked them to fill out a survey about this project.

3.1.3 Survey Questions

The survey questions consist of four major parts: 1) familiarity of Web API testing, 2) familiarity of Postman tool, 3) Effectiveness of GPT-3 generated data, and 4) views on the developed workspace Flow after testing it themselves.

Familiarity of Web API testing: In this section, participants were asked to answer questions about how familiar they were with Web API, and how often they use API testing in their development process. All of the questions are scalar questions.

Familiarity of Postman tool: In this section, participants were asked to answer questions about their acquaintance with Postman as a Web API testing tool. And the survey also asked about their likeness to use Postman before today. Moreover, it also asked about their familiarity with the Postman Flow feature.

Effectiveness of GPT-3: In this section, the participants were asked to rate the effectiveness of data generation in the Postman Flow using GPT-3. And they were also asked about what aspects of data they think GPT-3 was able to generate. Also, participants were asked to rate their preference on automated generated data.

Views on the Developed Workspace Flow: This section of survey focused on collecting participants' view on the project after they used it for some time. First, they were asked to rate the effectiveness of Test Data generation in the GPT-3 Flow; then, participants were asked to rate their intention to use such Postman in their future development, as well as the intention to use the developed Flow in this project demo.

The goal of this survey was to collect participants' user feedback and views on the Flow developed in this project. By collecting the responses and summarizing them into qualitative viewpoints, I can understand the usefulness of this project in the Web API testing field.

3.1.4 Results and Findings

Since there are only 6 participants who responded to the survey, the findings may not be conclusive about this development. However, because all participants are from related fields and had decent understanding of Web API testing, the survey was able to produce some meaningful and interesting findings.

RQ1: Is it possible to make Postman Desktop Application use GPT-3 prompt completion function from OpenAI? And how can Postman process the data returned and use it for other API's requests?

Based on the demo session to participants and their own effort before submitting the survey, most (4 / 6) users wrote that they liked the idea of using the GPT-3 prompt completion feature to generate test data. According to the survey response, participants think GPT-3 was quite effective when generating specific Test data for Web APIs. Combined with the project development process, I found GPT-3 was able to generate meaningful and useful data that can be used in API testing.

RQ2: Can GPT-3 data generation be used in automated testing to improve efficiency of API testing?

Based on the survey responses, participants had both positive and negative views on this Flow tool. One participant wrote that she would just like to use GPT-3 in coding instead of dealing with Postman UI. Another participant wrote the GPT-3 extension in Postman is too simple for any future usage. On the other hand, 3 of other participants expressed the Postman Flow with GPT-3 can help them test their API more efficiently.

RQ3: Compared to traditional script testing using coding language, what are the benefits of Postman Flow?

According to the responses, for most (5 / 6) of the participants who are still testing Web API using coding scripts, they all express strong intention to incorporate Postman and the project Flow in their future projects. According to participants' views, the Postman Flow feature and the implemented Flow in this project provided visual aid for Web API testing. And the API testing native in Postman is simple enough for them to use right away.

Deliverables

On Github: (https://github.com/chrisliuuuu/auto_API_testGen)

This repo includes a link to the Postman Workspace created in this project, it is a public link that people can fork the workspace with a Postman account. Additionally, the repository also contains readme on how to navigate the developed Flow in Postman, along with demo slides and project report.

Postman Workspace:
(https://www.postman.com/mission-technologist-21481539/workspace/auto-flow-test-data-gpt-3/overview)

This workspace contains a template Flow designed for generating test data for Youtube Comment API. However, the green-color blocks represent reusable blocks in the flow that can be used to generate other data based on the prompt input.

Self-Evaluation

Although this project produced some interesting and useful tool extensions for Web API testing, I feel like more can be done if I can spend more time researching the API specification documentation processing.

To start with, I think the main contribution of this project is showing developers using or not using Postman on how they can integrate this new LLM feature inside Postman, which leads to smarter and more efficient testing inside Postman without the help of other coding scripts. Moreover, this project provided a reusable Flow template that can be used to generate data based on the prompt engineering. In my view, Postman is a very user friendly tool that, if more features were added, will become very popular in API testing. So I hope the outcome of this project can help peers to start considering using Postman when testing their Web APIs

On the other hand, definitely more can be done to make this project more applicable to wider usage. For example, the initial plan of this project was to let LLM like gpt-3 to study API specification documents to generate test data and cases for targeted APIs. However, after spending quite some time researching online, processing the API documentation can be tedious work, since the format of many popular APIs' documentation can be various. Therefore, using LLM lib like SpaCy can be really hard. Therefore, in the limited amount of time remained for this project, the focus of this project was changed to test data generation.

Overall, this project also helps me to explore the world of Web API testing and how AI and LLM can help to simplify the process.

Reference

[1] Alberto Martin-Lopez, Sergio Segura, and Antonio Ruiz-Cortés. 2022. Online testing of RESTful APIs: promises and challenges. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). Association for Computing Machinery, New York, NY, USA, 408–420. https://doi.org/10.1145/3540250.3549144

[2] Despa, V. (2023, March 1). Getting Started with the OpenAI API with Postman. Medium. https://medium.com/apis-with-valentine/getting-started-with-the-openai-api-with-postman-52fea7f8131a

Appendix

How to use the workspace (demo video link)
https://drive.google.com/file/d/1Y_a-whbKh0aqoxgwvc-6Ydxi1tdvhDxV/view?usp=sharing