Postman Collection Integration: Enhance Customer Experiences with Postman

CS4991 Capstone Report, 2022
Kedar Kulkarni
Department of Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ksk6rz@virginia.edu

ABSTRACT
A software engineering company that facilitates integration between cloud-based enterprise applications has incorporated Postman, a popular Application Programming Interface (API) platform, into the services it provides in order to reduce customer development overhead and attract a greater customer base. As an intern, I utilized a Postman Collection, a JavaScript Object Notation (JSON) format for describing API endpoints, to drive a seamless user experience for configuring endpoints Representational State Transfer (REST) endpoints. I also devised a technical design that detailed the requirements and motivation from the stakeholder, the solution, the outcome, and potential limitations. Our team employed REST, APIs, and microservice architecture, OAuth internet protocol, as well as the Java programming language, JSON and Docker. As a result, clients could model their REST endpoint configuration from Postman without having to manually create them in our services, which offered them greater flexibility and functionality. In the future, additional core features from Postman should be integrated and comprehensive testing completed to ensure the viability of the current implementation.

1 INTRODUCTION

In an area of rapid technological change, the world is moving to the cloud, a network of servers connected to the internet that manages data and resources. These servers, hosted around the world in physical data centers, allow corporations to manage their software and deploy applications without the need to oversee their own physical infrastructure [1].

Moreover, corporations employ software and services from all fronts on the cloud. The challenge is how to best integrate these services seamlessly. For instance, a company may have a workflow that sends an alarm to its customers once an outage occurs, using two different software offerings from differing companies that must be integrated.

My project dealt with enhancing the creation of these integrations and workflows by allowing the end user to utilize previous REST endpoint configurations in Postman and view them in our services. Postman, being a very lightweight tool, allows users to easily debug and test their endpoints for correctness. Previously, the customer would have to go into Postman in order to diagnose their problem and recreate those same configurations manually in our platform. This wastes time and loses productivity for both the client and our team. Therefore, I created a feature that allows the customer to import all their configurations from Postman into our services in order to configure their API and endpoints utilizing our features and thus reap the benefits of using the cloud.

2 RELATED WORKS

My work required fundamental knowledge in many technologies. Although there are many ways to communicate over the internet, my project narrowed its scope to only REST calls. As

described by Gupta (2022), REST is an architectural style that lays the foundations on how systems should talk to over the internet. Namely, REST defines a resource, which can be any piece of information. In addition, REST defines stateless calls, which ensures that no context is needed to be stored on the client or server to successfully execute a call [2].

Security is another major facet of communication over the internet. Hardt (1970) helped me to understand the configurations for one of the most common authorization protocols: OAuth 2.0. This protocol involves utilizing tokens that are granted and authorized in order to gain access to systems [3].

Although REST does not specify any format to define the data for requests, I mainly dealt with JSON for writing request payloads through HTTP and REST. JSON is a widely used and lightweight format that is human and machine readable, according to Mozilla (2022) [4].

Understanding the importance of Docker and virtualization was vital to the success of my project. I utilized Dockerfiles, images, and containers in order to package the code. As explained by IBM (2022), Docker allows for easy distribution of code in multiple systems through containerization, which ships an entire package with all its dependencies and requirements [5].

3 PROCESS DESIGN

My internship timeline consisted of three phases: research, requirements and design, and the development phase. Lastly, I will highlight the challenges that I faced in the project.

3.1 RESEARCH

Firstly, I had to understand the product of my team and its functionality, which can be best explained by a simple example. Say that I am a company that uses a messaging tool. I want to update an internal database of users and leverage the messaging tool's APIs in order to reflect that information on their end. Therefore, this customer can leverage our product in order to create a logical flow from querying their database of users, updating the information, then automatically updating the information of each user in the messaging tool, requiring very little development.

Once I attained this foundational understanding of the integration space, I had to research heavily on Postman and the possible endpoint configurations that could be made. Further, a Postman Collection is the medium in which data can be transferred to our services. A Postman Collection is a flat list of HTTP requests (every request connects to some endpoint) that is described in a JSON document. Within each request, there will be a request URL. Moreover, the user can describe path or query parameters and various headers. The body of the request can be in numerous formats like JSON, raw, binary, and more. Lastly, the authorization for the request is important and there are many protocols that can be utilized. These requests could also be bundled into folders, which creates a hierarchal structure in a Postman Collection.

3.2 REQUIREMENTS AND DESIGN

After conducting my research, I devised the design and requirements for what would be possible in a 12-week internship. Since a Postman Collection and the Postman requests therein were highly customizable, I narrowed the scope of the project to certain core aspects. I restricted the design to handle requests that were non-hierarchal, meaning they could not be within unique folders. In addition, the design initially supported two types of authentications: OAuth 2.0 and basic authentication. Lastly, I supported only JSON request bodies. It was essential that a very similar experience to configuring requests within Postman was presented, so that any customer could easily transition to our services. I therefore devised a logical flow and front-end user interface that resembled Postman.

3.3 DEVELOPMENT PHASE

During the development of this feature, I incrementally coded every aspect up to the finished product. I first ensured that I would be able to successfully load a Postman Collection into our services and deserialize it as an object so that the differing components within our product could understand it. I then worked on creating the user interface that would resemble a Postman-esque experience, displaying the flat list of requests and the aforementioned configuration options. This aspect allowed me to reuse a lot of existing back-end logic for configuring REST endpoints.

3.4 CHALLENGES

The most consequential challenge was understanding all the aspects of Postman requests and how they fit together in order to make a successful endpoint configuration. This demanded an understanding of a wide variety of technologies. Furthermore, tweaking the user interface in our services to be like Postman also brought challenges, as I had to gauge the common uses of Postman and create an importance on their user interface design to mimic in my design.

4 RESULTS

This project resulted in a successful end-to-end feature. The customer can create several requests in Postman and export them as a Postman Collection. Then, they can upload it into our services and both view and update the configurations for each request. Lastly, they can connect to the endpoint by running their requests. This feature saves a lot of time for customers that use Postman who want to transition to our services and reuse their endpoints. Customers often like using Postman's simplicity to debug their requests when they face issues, allowing them to use both products in a seamless and integrated manner.

5 CONCLUSION

Integration in the cloud is a salient tool as technology becomes a more integral part of people's lives. In this project, I leveraged the power of Postman to make a better experience for customers using our product. I added functionality that does not limit the customer to only one service, thus utilizing the strengths across multiple platforms for an optimal and efficient use case. As customers can create more robust integrations through the help of my feature, their development lifecycle becomes more efficient.

6 FUTURE WORK

This feature can be expanded to include all aspects of a Postman Collection. This includes supporting hierarchal Postman requests, non-JSON request and response bodies, authentications other than OAuth 2.0, and more. The ability to support these within our product will further enhance the development experience of integrations and allow the customer to customize their workflows to very specific use cases.

REFERENCES

[1] Cloudflare. What is the cloud? Retrieved September 23, 2022 from https://www.cloudflare.com/learning/cloud/what-is-the-cloud/

[2] Gupta, L. 2022. What is rest. (April 2022). Retrieved September 23, 2022 from https://restfulapi.net/

[3] Hardt, D. 1970. The oauth 2.0 authorization framework. (October 1970). Retrieved September 23, 2022 from https://www.rfc-editor.org/rfc/rfc6749

[4] Mozilla. 2022. Working with JSON - learn web development: MDN. (2022). Retrieved September 23, 2022 from https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON

[5] IBM. 2022. What is Docker? (June 2022). Retrieved September 23, 2022 from https://www.ibm.com/cloud/learn/docker