Postman

Overview

Postman is a tool that is used, above all, for API REST testing, although it also supports other features that come out of what is included in the testing of this type of systems.

Thanks to this tool, in addition to testing, consuming and debugging REST APIs, we can monitor them, write automated tests for them, document them, mock them, simulate them, etc.

Table of Contents

Project Requirements	1
Risk Management	1-2
Project Tasks	2
Analyzing the Interface	3
Practical Section	4-7
Git Workflow	7
Technologies Used	8
Lessons	8
Incidents	8

Project Requirements

- Create a clear and orderly directory structure
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use online styles
- In the case of using different programming languages always define the implementation in separate terms
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project

Risk Management Plan

Every project has risks. This risks must be taken into account to improve the workflow of the project. I've listed the risks for the project, along with the impact they might have, and the priority of them.

	Risk	Consequence	Prob. (1-5)	imp. (1-5)	Pri. (1-25)	Mitigation approach
1	Breaking my computer	Can't do anything	1	5	5	Keep my repo work to date, look for other computers
2	Getting sick	Wouldn't be as productive	2	3	6	Eat and sleep well
3	Not concentrating enough	Won't com	2	5	10	Focus on the Minimum Viable Product.
4	Unrealistic deadlines	Deadlines wouldn't be met +	2	3	6	Be more organized with the tasks, set new

		development shortcuts would have been taken affecting the robustness of the code				deadlines.
5	Being unfocused	Loss of control over the development flow of the project	4	5	20	Focus on finishing the most important tasks only.

Project Tasks

Defining this part is crucial to the development of the project. It is important to make a good analysis of the situation to organize the project in a good way:

#	Task	Priority (1-5)	Description	Difficulty (1-5)	Estimation
1	Reading the description	4	Reading the description of the project	1	30 min
2	Create Repo	3	Creating git repo for the project	1	2 min
3	Analyze the interface	5	Analyzing the interface of postman and answering the theoretical questions	4	1 hr
4	Put into practice	5	Putting into practice the concepts learned.	3	1 hr
5	Review	2	Review Project	2	30 min

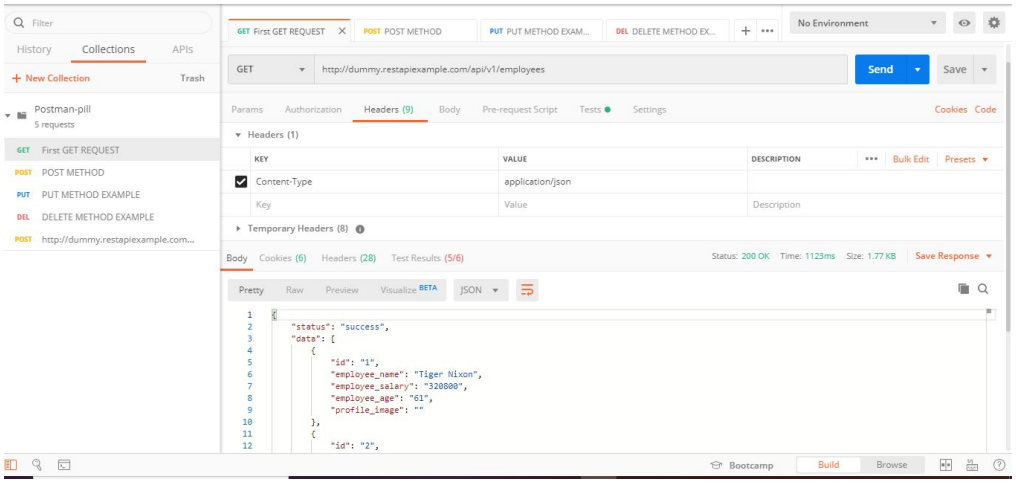
Analyzing the Interface

In this section, we will review the menus provided and add explain what they are for and how they are used:

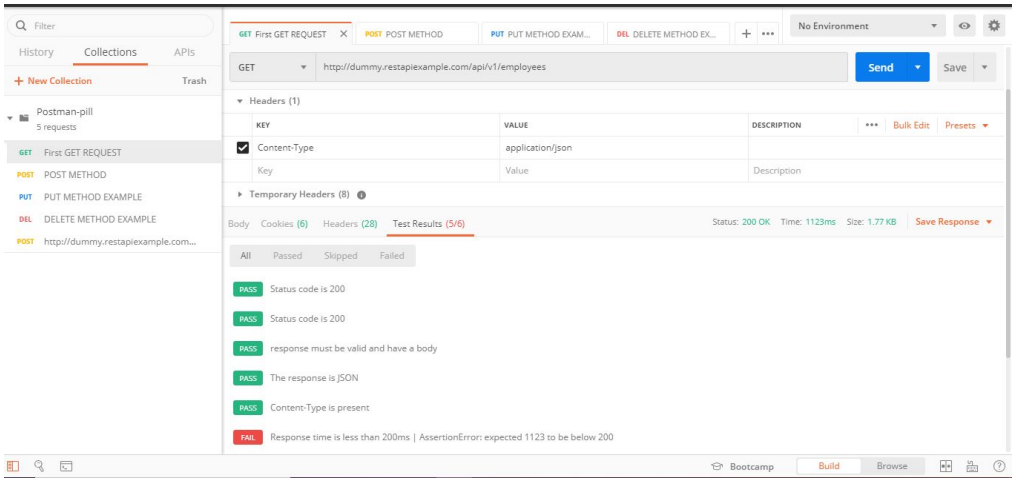
- **Request**: Request in this case stands for HTTP requests. You can send requests in Postman to connect to APIs you are working with. Your requests can retrieve, add, delete and update data.
- **Collection**: A Postman Collection lets you group individual requests together. You can organize these requests into folders.
- **Environment**: Environments allow you to run requests and collections against different data sets. For example, you could have an environment for development, one for testing, and another for production. You can use variables to pass data between requests and tests, for example if you are chaining requests using a collection.
- **API Documentation:** You can automatically generate documentation for your Postman APIs. You can share your documentation privately or publish it on the web. Postman generates and hosts documentation based on collections, synced in realtime and accessible via the browser. You can use documentation to collaborate with team members and partners, or to support developer adoption for your public APIs.
- **Mock server:** Mock Servers in Postman let you simulate APIs. You can create mock servers from the Postman app, from the web dashboard, and using the Postman API. You will need a Postman account to set up a mock server.
- **Monitor:** Postman lets you monitor shared or private collections. If you choose to monitor a shared collection, your team can see the monitor. If you create a monitor on an unshared collection, the monitor is private and only visible to you.
- **API:** This allows you to create APIs directly within the Postman App. You can add new collections, update existing collections, update environments, or add and run monitors directly through the API. This API enables you to programmatically access your data stored in your Postman account with ease.

Practical Section

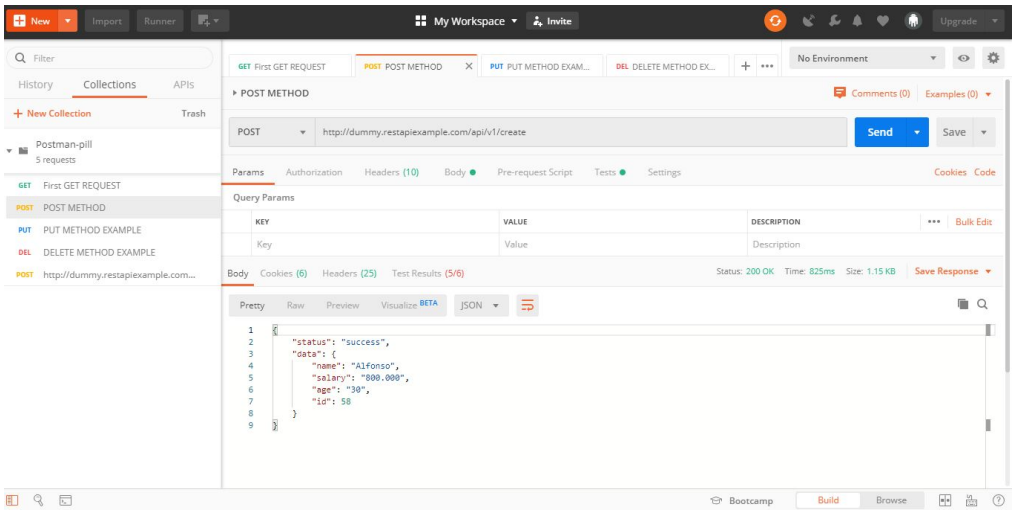
Using the Get Method:



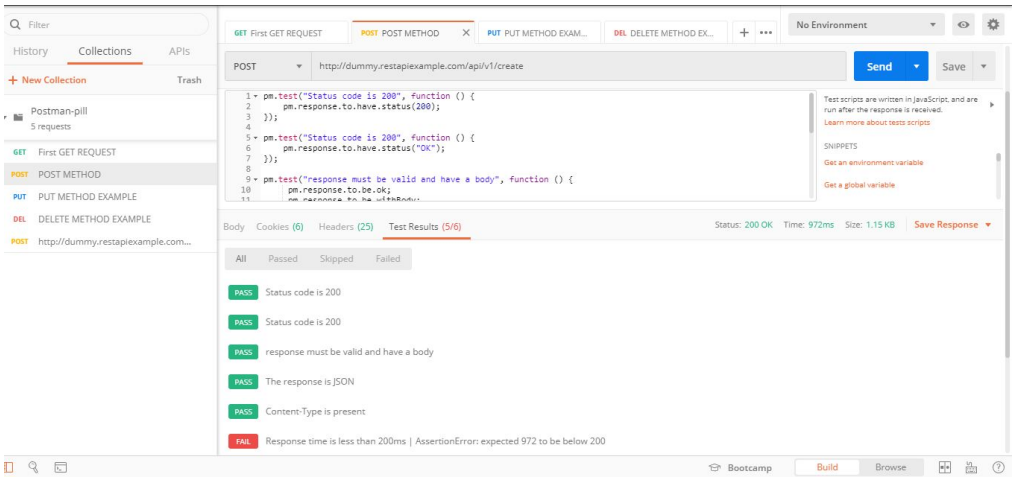
Test for the Get Method



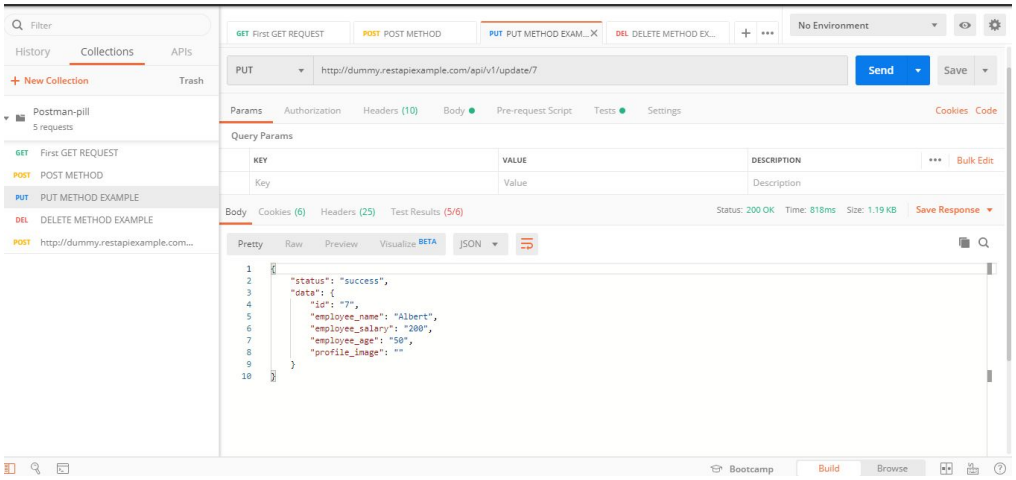
Using the Post Method:



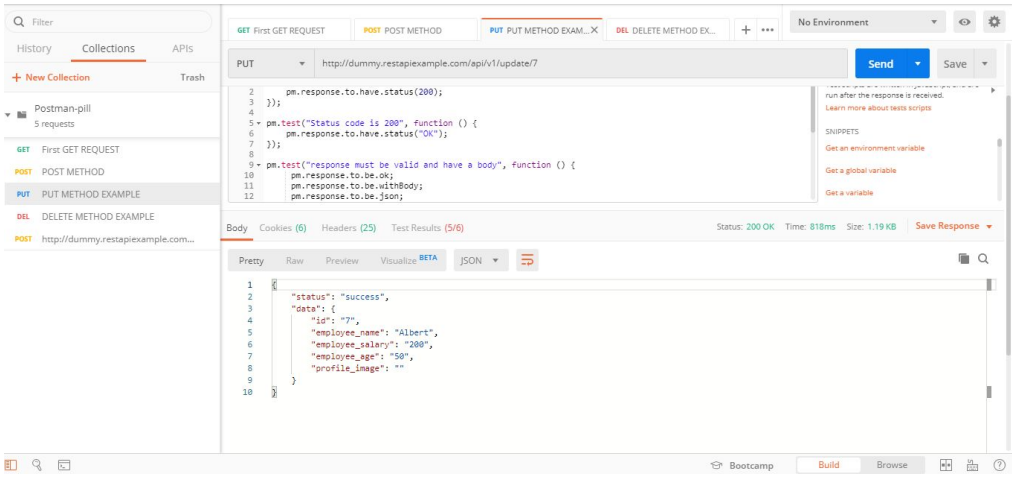
Test for the Post Method



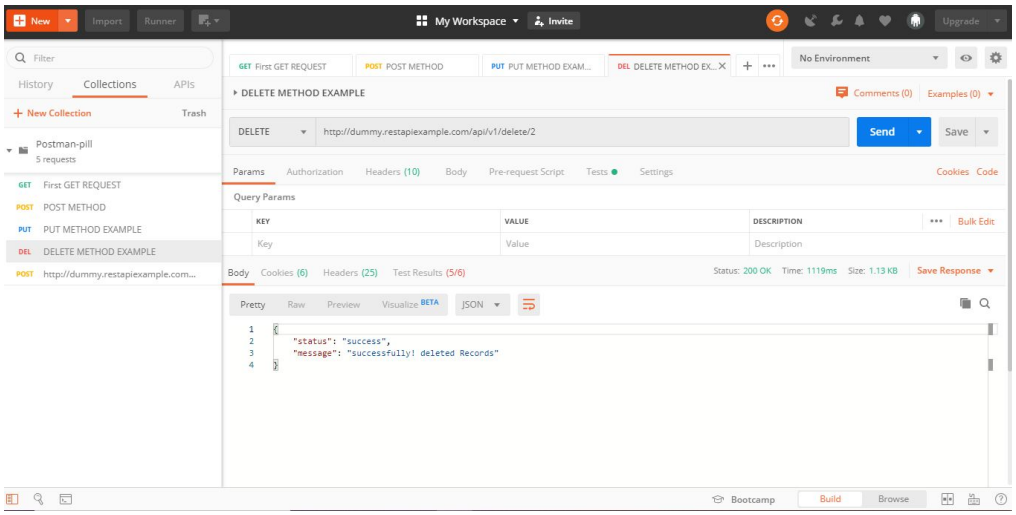
Using the Put Method:



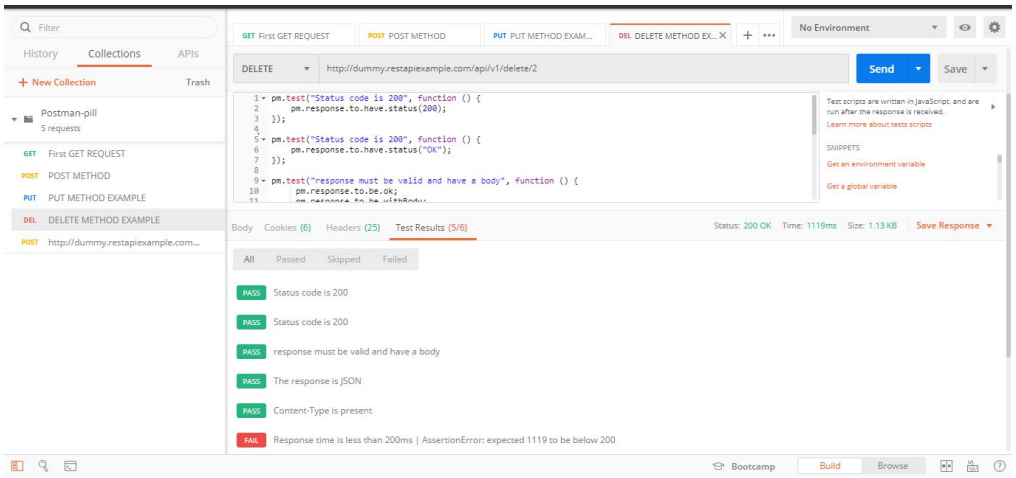
Test for the Put Method



Using the Delete Method:



Test for Delete method:



Git Workflow

For this project, all commits we'll be pushed directly to the Master branch. All commits will use a descriptive message, so that myself or other users can easily go to the Git version that they need to. This is very important for working in teams as it increases communication and efficiency between all members.

Technologies used

For this project, we will use the following technologies:

- Postman
- Google Docs
- Snipping Tool

Lessons learned

- Postman allows you to do API testing. API testing helps you determine if the APIs meet expectations for functionality, reliability, performance and security.
- Postman has many options for displaying data for you. These are named: “Pretty”, “Raw”, “Preview” and “Visualize”.

Incidents

- At first, I lost some time trying to find the definitions of the first section.
- At first, I didn’t know how to do the tests.