POSTMAN

Overview

In this project, we learn the usage of POSTMAN. This project consists of learning the different options POSTMAN has when creating requests for connecting with an API. The API in question is http://dummy.restapiexample.com/

Objectives of project

- Understand what is a POSTMAN
- Improve the knowledge of POSTMAN
- Improve the knowledge of various functionalities of POSTMAN

Table of Contents

8. Git Workflow
9. Technologies used
10. Incidents

Project requirements

After installing POSTMAN, review the menus provided by the interface and add in the documentation a section explaining what it is for and how each of the following elements is used:

- Request
- Collection
- Environment
- API Documentation
- Mock server
- Monitor
- API

It is possible that to have access to any of the above elements you must create a Postman account.Finally, you must implement the following Postman functionalities using an example online API to make the requests:

- Create a new POSTMAN Collection (research previously about it)
- Create a request for each of the following methods as shown in the image above:
    - GET
    - POST
    - PUT
    - DELETE
- Create a test for each previous request where you will verify that the response meets the following conditions:
    - The state code is as expected
    - The name of the state code is the corresponding one (OK, Created, ...)
    - The answer has body
    - The answer is in JSON
    - The header has “Content-Type”
    - The response time is less than 200ms
- Create a monitor that runs the previously created collection every 5 min
- Create an API document from the previous collection with the Postman application
- Export the created collection and import it back into Postman (you must incorporate the generated file in the project repository.

You must document all the actions performed previously and take screenshots of all the processes.

Risk Management

Every project has risks. These risks must be taken into account to improve the workflow of the project. Managing these risks is important for due completion of the project. Unchecked risks could not only lead to hamper of project deliverance but also a badly executed project. The risks hence associated with the project are documented as follows:

Risk	Risk level
Unfamiliarity with POSTMAN	Low
Unfamiliarity with designing Requests	Medium to Low
Health issues	High
Delivering in time	Medium
Completing all the project requirements	Medium

Task Management

The following are the tasks which needed to be accomplished for the completion of the project. The tasks have been divided according to the project specifics. Each task has been clearly defined and their priority as well as their difficulty and time needed. Difficulty level is explained on a scale of 1 to 5 (5 being the most difficult). Priority is explained on the level of 1 to 5 (again 5 the highest parameter being the most prioritized work).

Task	Priority	Description	Difficulty	Time
Read the description of the project	5	This task involves complete understanding of requirements for the project	1	30 min
Create git repo	5	Creating of git repository for project execution and delivery	1	2 mins

Create files for the project	4	Creating the necessary files, folders and subfolders for this project. This is important especially for this project of PhP unit	1	5 mins
Download POSTMAN	3	Downloading the POSTMAN which is required for this project	2	10-20 mins
Install POSTMAN	4	Installing the POSTMAN	3	30 mins
Create all the requests	2	Create all the requests as required by the project	4	15 mins
Run the tests	2	Run the tests	5	30 mins
Documentation	1	Write the documentation and push it to github	2	1 hr

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.

Chronogram

This shows the time-line it took for the project to finish and also the tasks that were achieved during those timeline.

The tasks

Tasks	Tue 1000 hrs	Tue 1100 hrs	Tue 1200 hrs
Read project descriptio-n	**X**		
Git Repo	**X**		

Create files	**X**		
Download POSITMAN		**X**	
Install POSTMAN		**X**	
Create the requests		**X**	
Run the tests		**X**	
Documentation			**X**

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 the admin and other internet user can follow through the processes.
For more information go to :
https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Incidents

- None

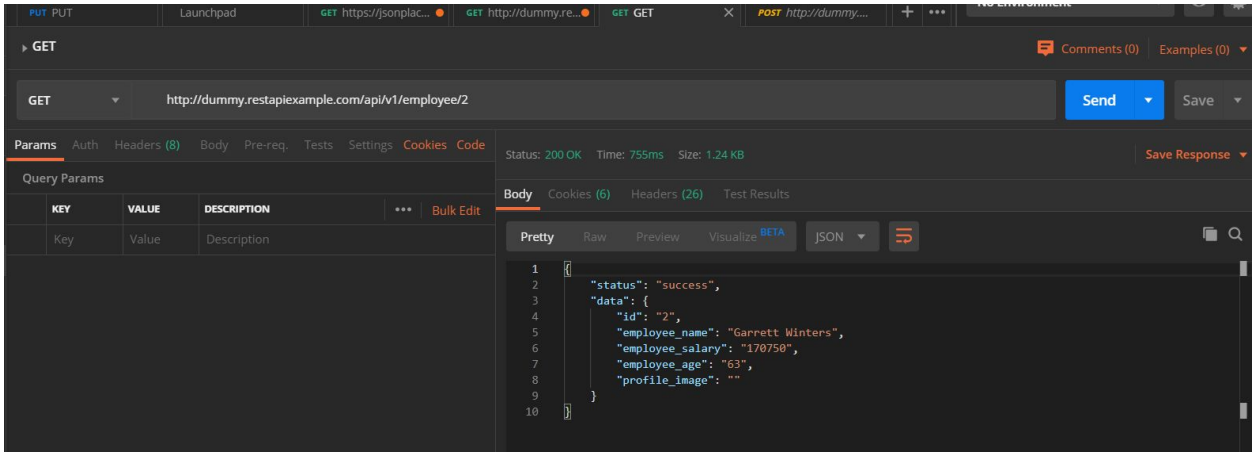
Technologies used

For this project, we will use the following technologies:

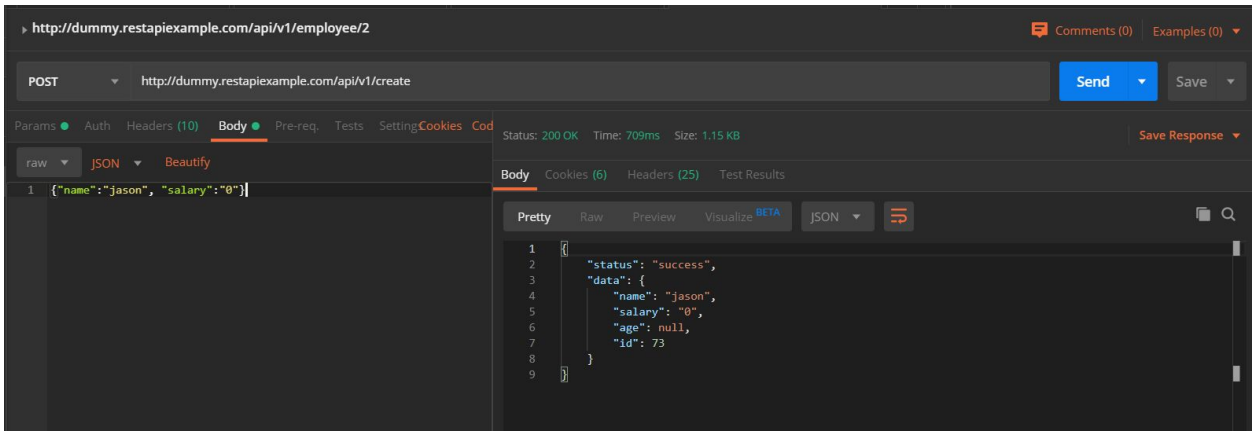
- POSTMAN
- GIT

ScreenShots

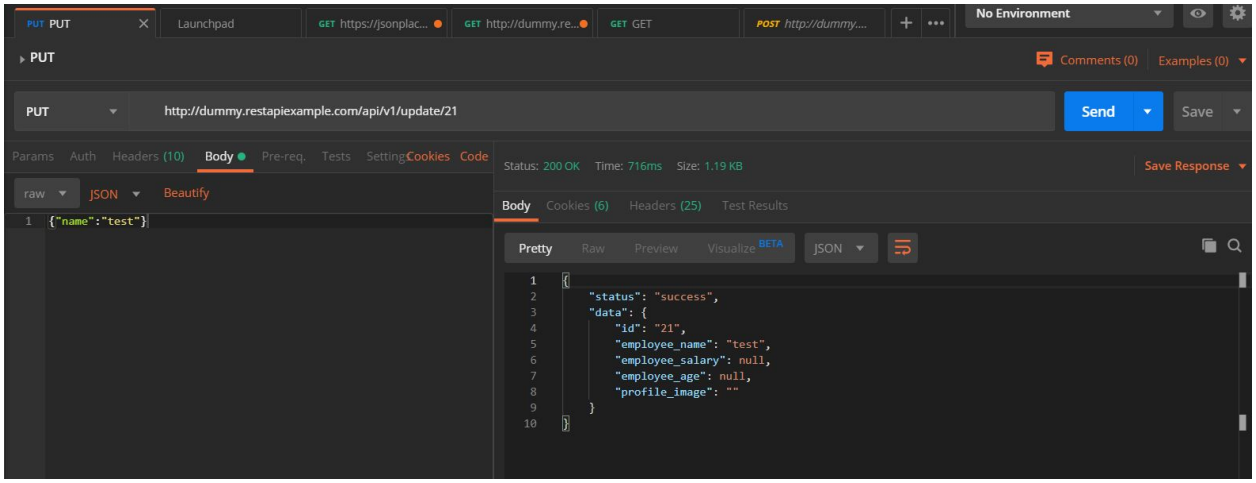
GET



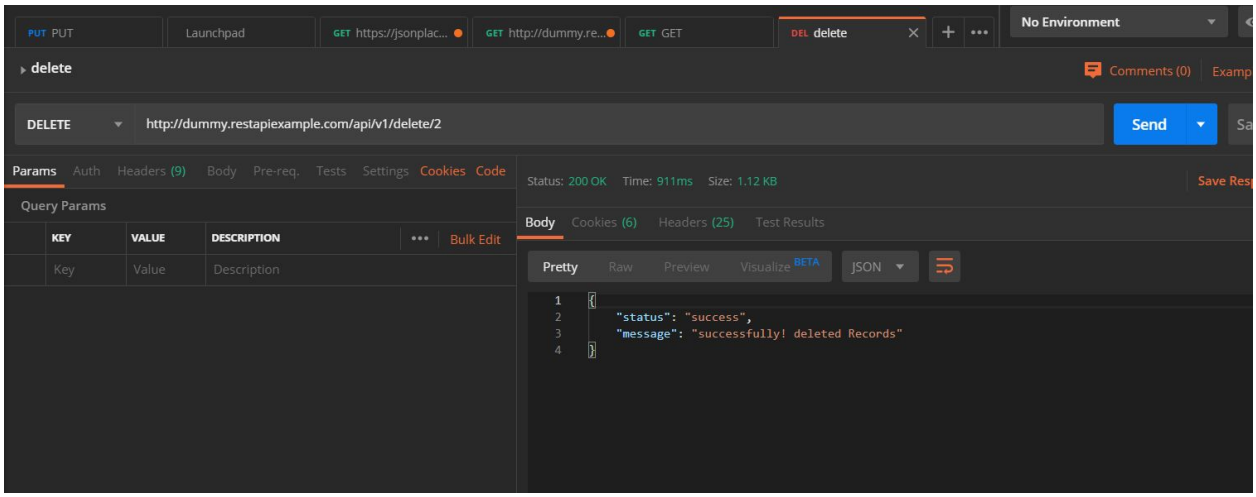
POST



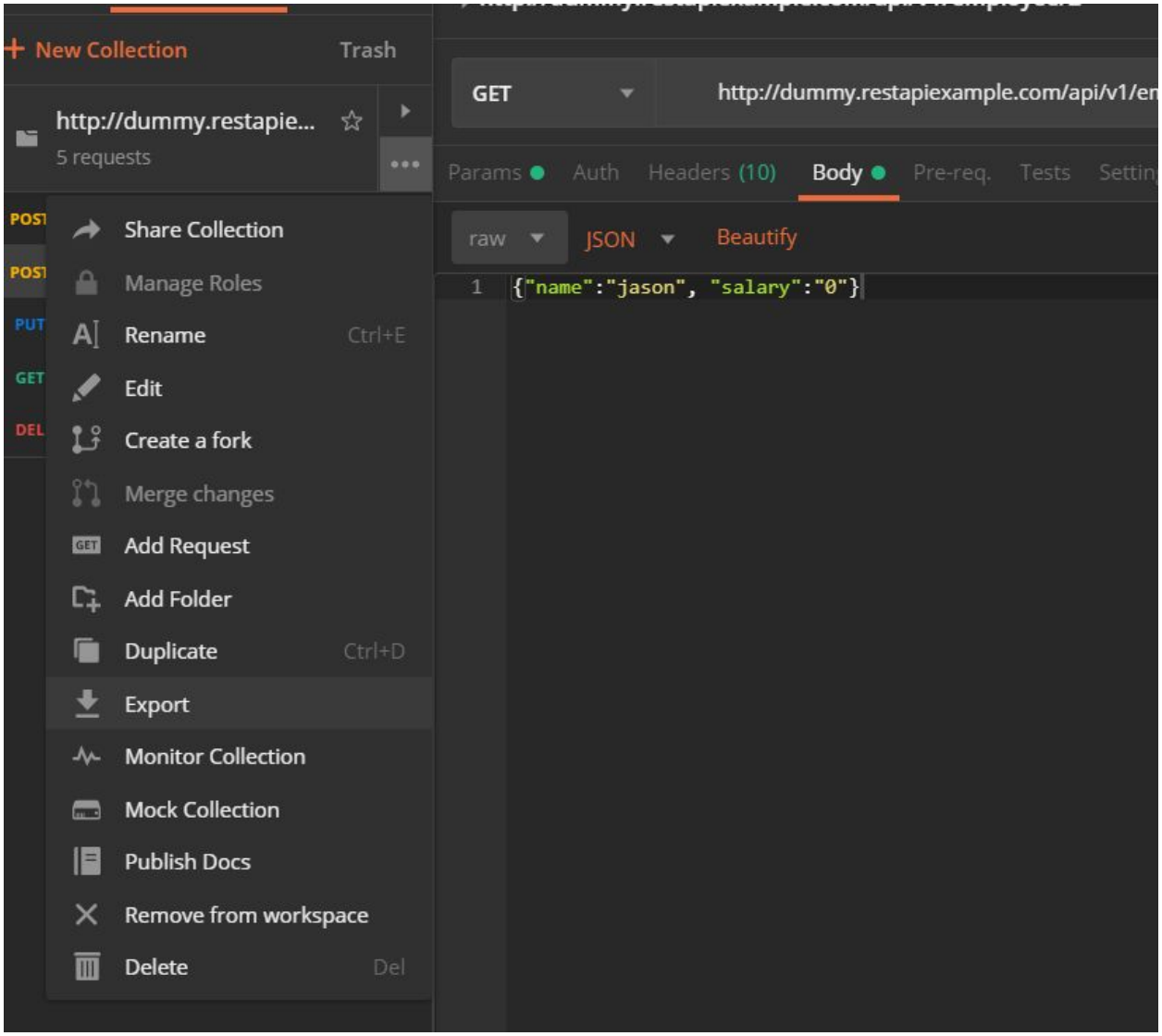
PUT



DELETE



EXPORT Collections


New Collection
Trash
http://dummy.restapie...
5 requests
Share Collection
Manage Roles
Rename Ctrl+E
Edit
Create a fork
Merge changes
Add Request
Add Folder
Duplicate Ctrl+D
Export
Monitor Collection
Mock Collection
Publish Docs
Remove from workspace
Delete Del
GET
http://dummy.restapiexample.com/api/v1/en
Params
Auth
Headers (10)
Body
Pre-req.
Tests
raw
JSON
Beautify
1 {"name":"jason", "salary":"0"}