

Postman – For testing REST API

23.4.2024



Haaga-Helia

Postman hints, tested with Feb 2023 version

- Create **Collection**, and there folders and subfolders.
 - E.g. category_single, category_multiple, idea_single, idea_multiple folders – So you have less test cases in each folder
- Create **Environment**, and there **Variables**, like BACKEND_SERVER_PORT
 - Double-click on hard coded parts of the URL, e.g. :8777 and turn it into {{BACKEND_SERVER_PORT}}. Then by changing the current value of the variable, all test cases using that variable get the new value
- While creating a new test case variation, **start** with “Save **as**”. Otherwise risk losing the old variation
 - While saving as new, select the collection and folder that suits the best
 - I used the naming like /api/category/1 (OK) for my test cases. (FAIL) for those that are supposed to fail
- While creating a POST or PUT test case, you can first use a singleton GET to get the correct JSON format example.
 - Then edit it so that it will not have duplicate names, and so that the POST does NOT have the id
 - The JSON is put to the **Body > Raw > Text => JSON**. If you suspect a JSON format problem, use e.g. jslint.com.

Postman hints, tested with Feb 2023 version

- You can right-click the Collection and scroll list down until you find: Export. You can save your test collection as JSON file e.g. to your git repo, let's say Postman folder
 - => you will get an **off-line** copy of the test cases, yet somehow sharable by git
- Limitation (Feb 2024 situation): Sharing Postman Team **online** is free for max 3 members.
- VS Code has nowadays the REST Client addition.
 - Does it now support all the same that is mentioned here?
- What PostMan has better than VS Code REST client (April 2024) is dynamic creation of correct JWT auth/auth tokens. You will need to tell PostMan these, and without login it can create the login/authentication token:
 - Username/email/userid/password and other encrypted information
 - The exact algorithm used in encryption
 - The backend SECRET hash
- In the following SCREENSHOT you can see some basic concepts and ideas of PostMan testing. E.g. the idea of creating multiple test cases, some OK, some failing on purpose. And e.g. creating folders for single item tests, and multiple item tests. Just for clarity and convenience.

Postman, tested with Feb 2023 version

The screenshot displays the Postman application interface. At the top, there is a navigation bar with links for Home, Workspaces, API Network, and Explore, along with a search bar and an Upgrade button. A yellow notification banner at the top states: "Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#)." The main workspace is titled "task-signature-workspace" and contains a collection named "idea-case-collection22S". This collection has a sub-collection "category_single" which includes several API requests. The selected request is a POST request to "http://{{BACKEND_SERVER_IP}}:{{BACKEND_SERVER_PORT}}/api/category/" with a JSON body. The body contains the following JSON data:

```
{
  "name": "India wedding",
  "description": "Outdoor activities eg cycling through the hills",
  "budgetLimit": 1000,
  "isActive": 1
}
```

The interface also shows a sidebar with navigation options like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The right sidebar contains options for Save, Send, Cookies, Beautify, and a help icon.