Dauphine | PSL
UNIVERSITÉ PARIS

... → COMPUTER SCIENCE, DECISION-MAKING, AND DATA → FURTHER EDUCATION

Algorithmic and advanced Programming in Python

Eric Benhamou eric.benhamou@dauphine.eu
Chien-Chung.Huang chien-chung.huang@ens.fr
Sofía Vázquez sofia.Vazquez@dauphine.eu

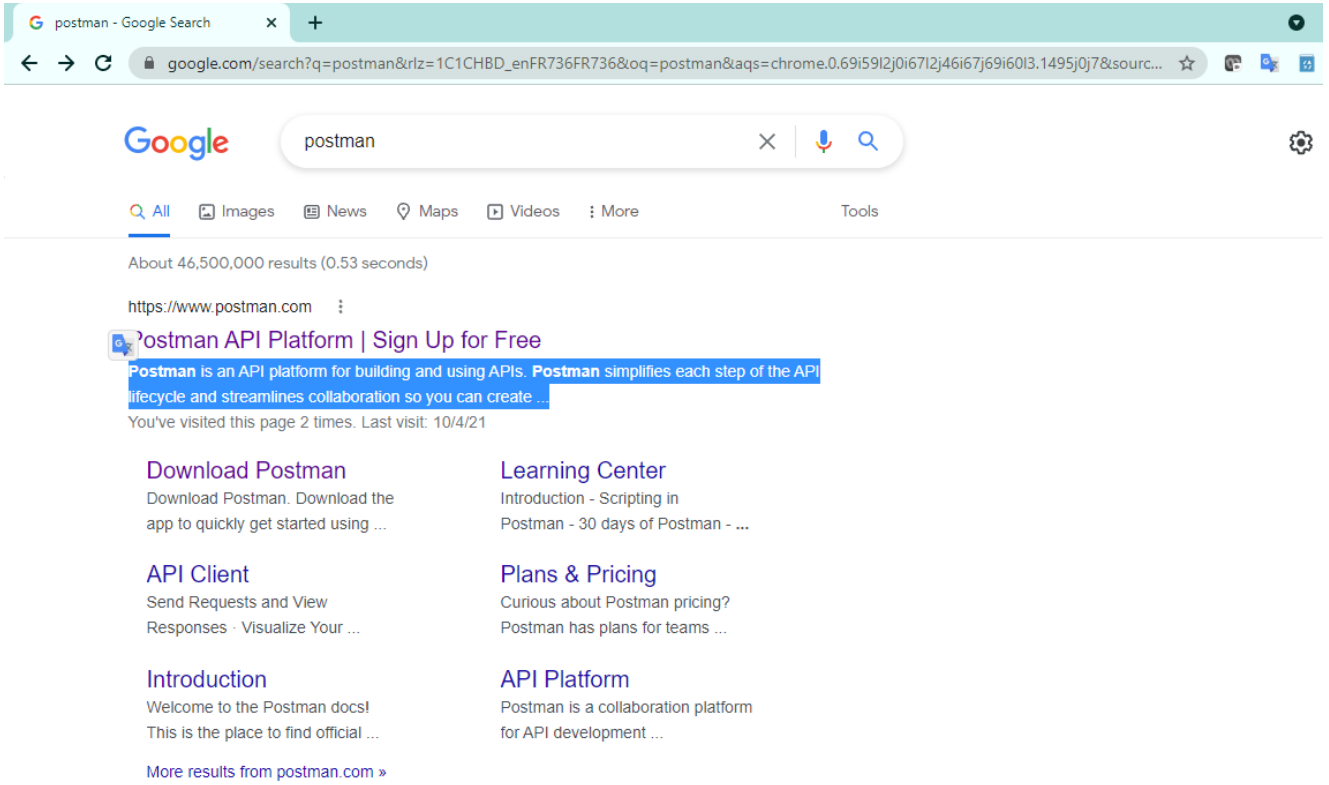


Lab 6 Part 2

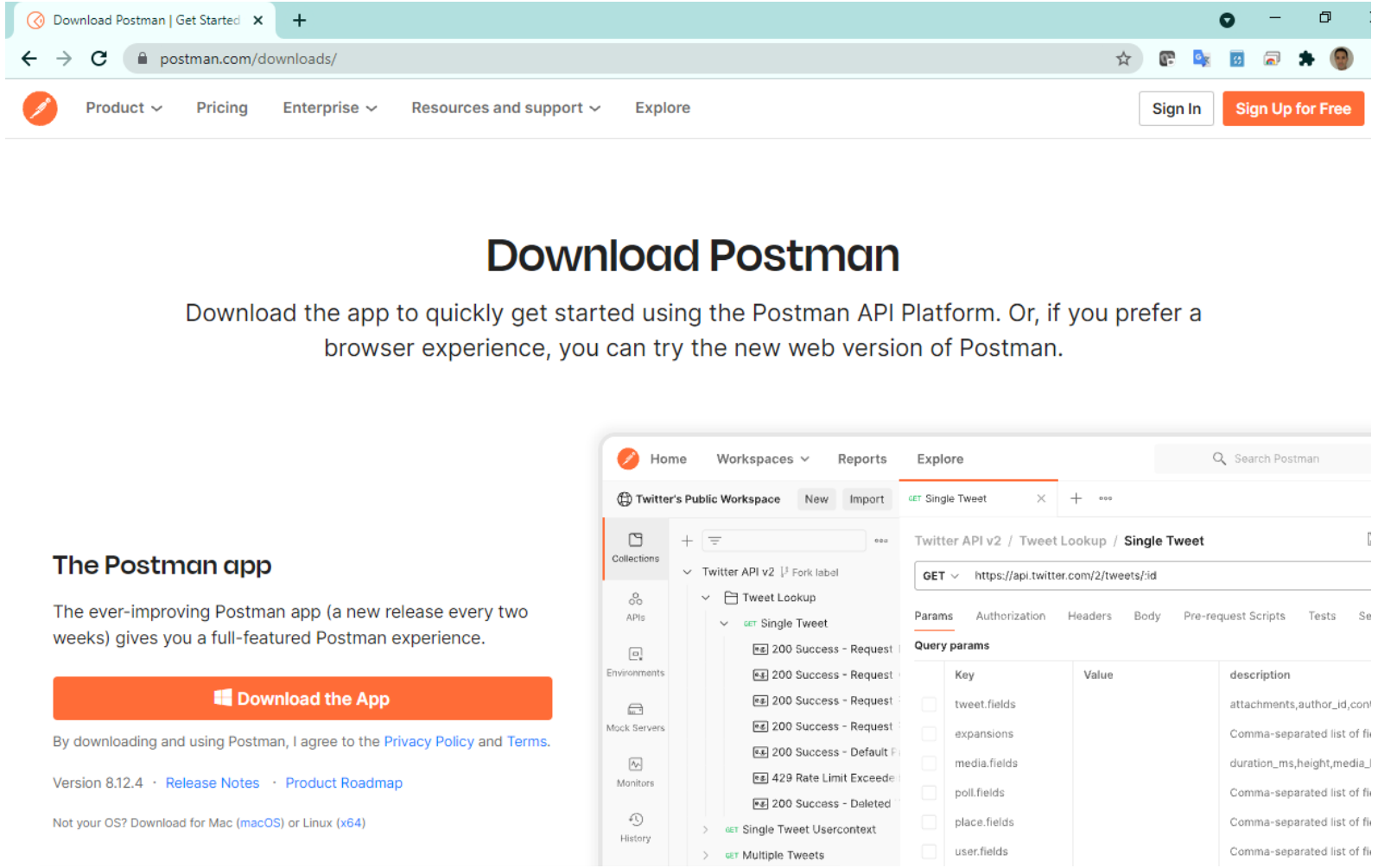
Let us play manually with postman

Postman is an API platform for building and using APIs. **Postman** simplifies each step of the API lifecycle and streamlines collaboration so you can create easily REST request

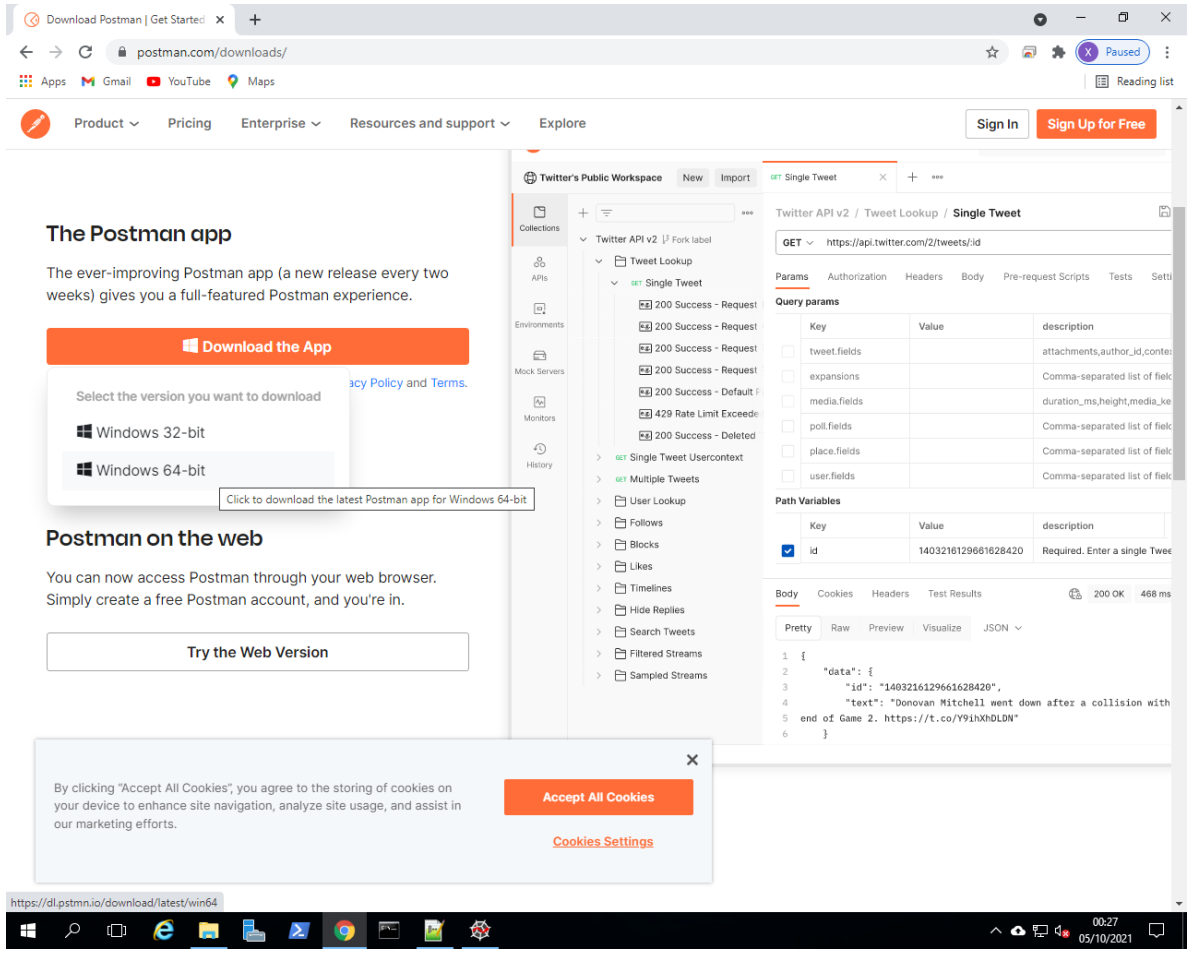
Download and install postman



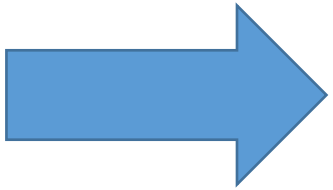
Download the app 1/2



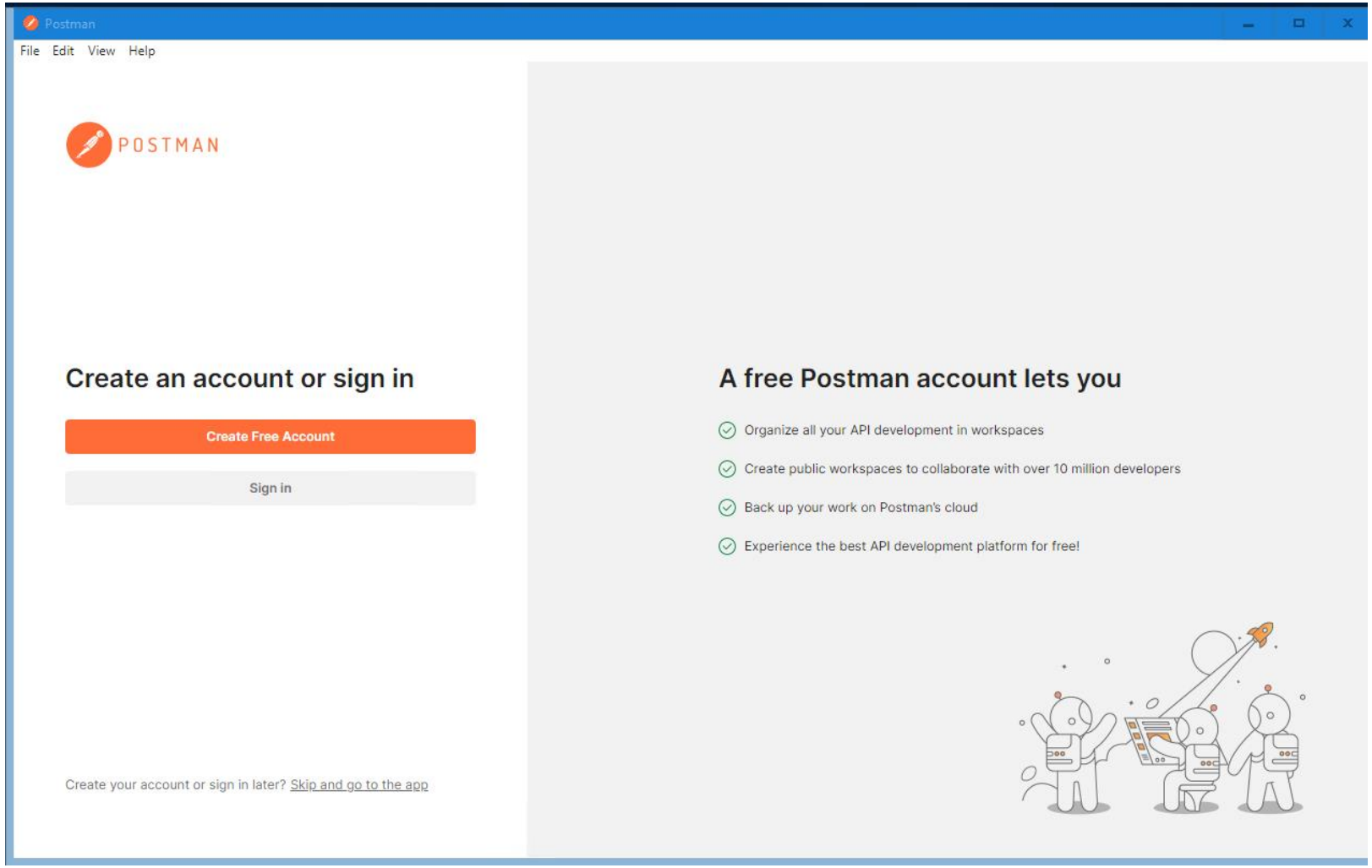
Download the app 2/2



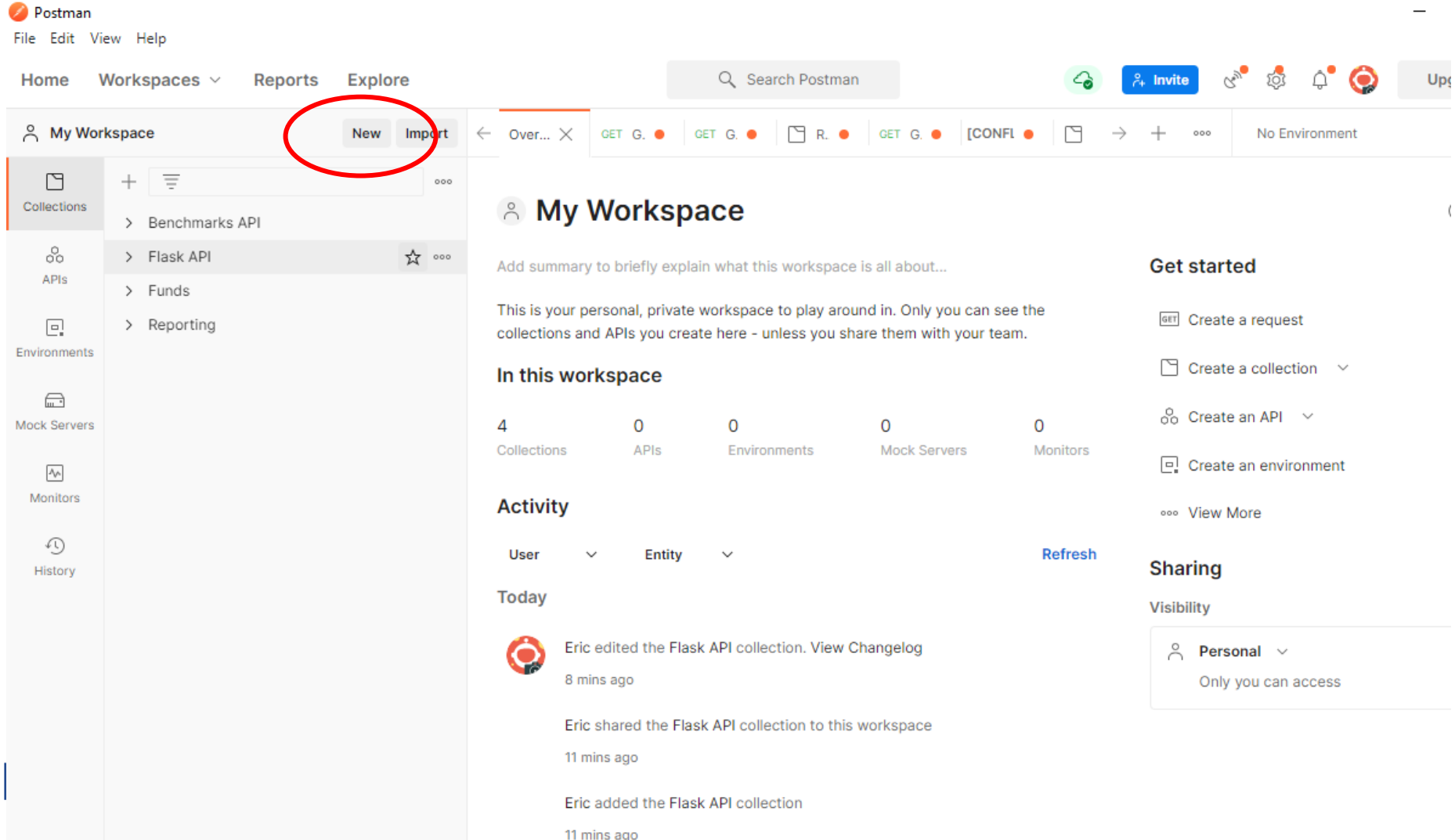
In your desktop



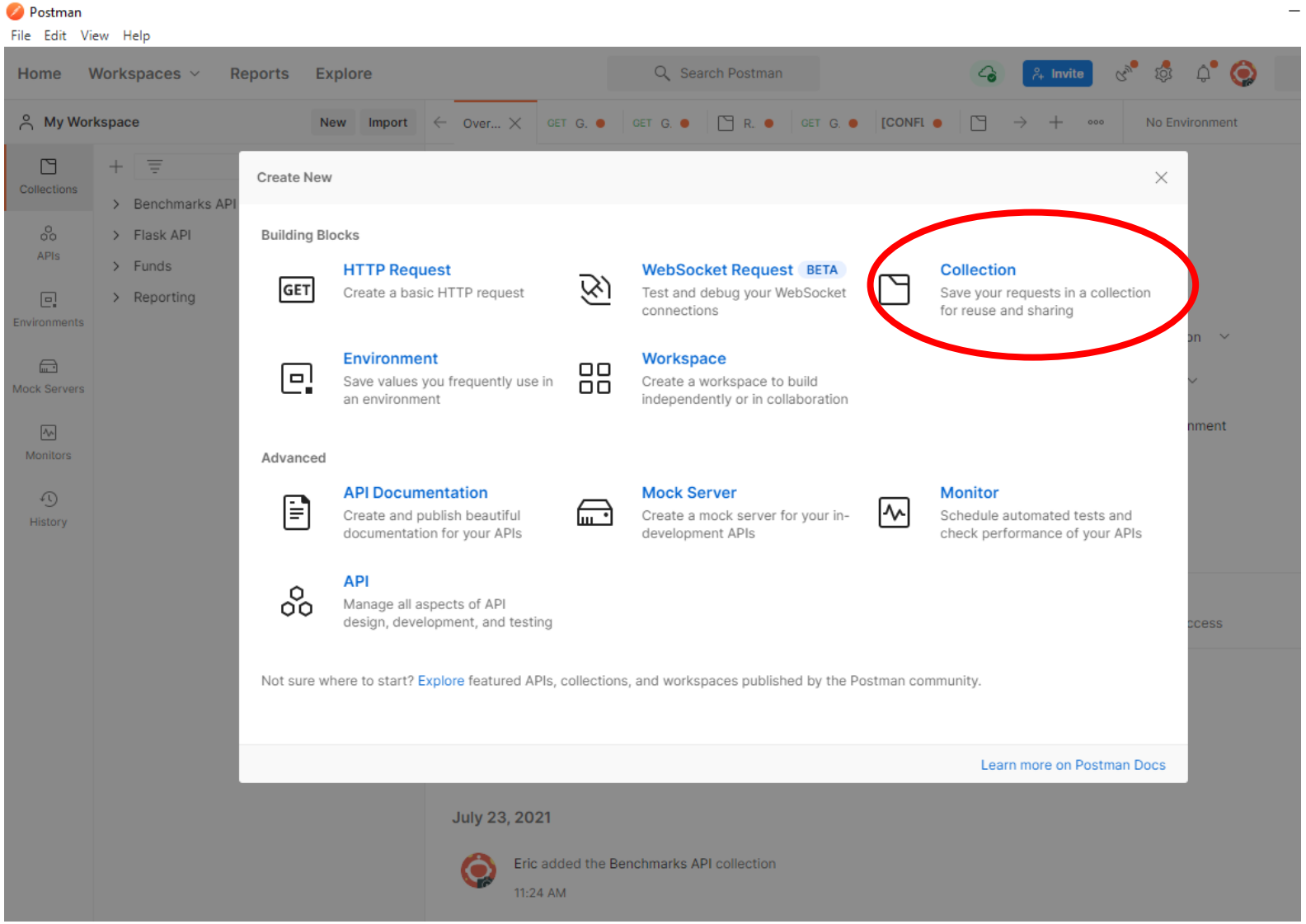
Create an account



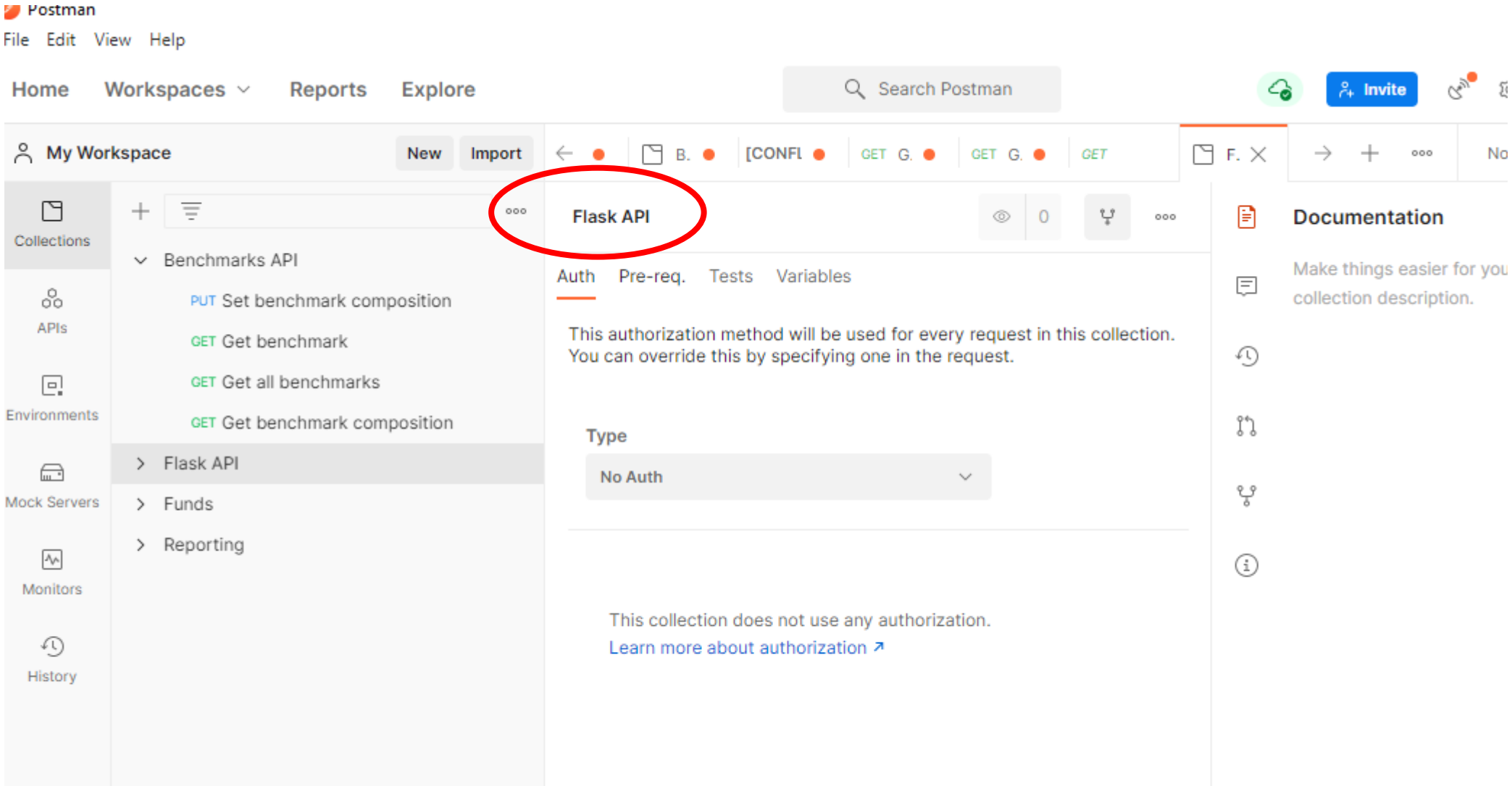
Create your first post request



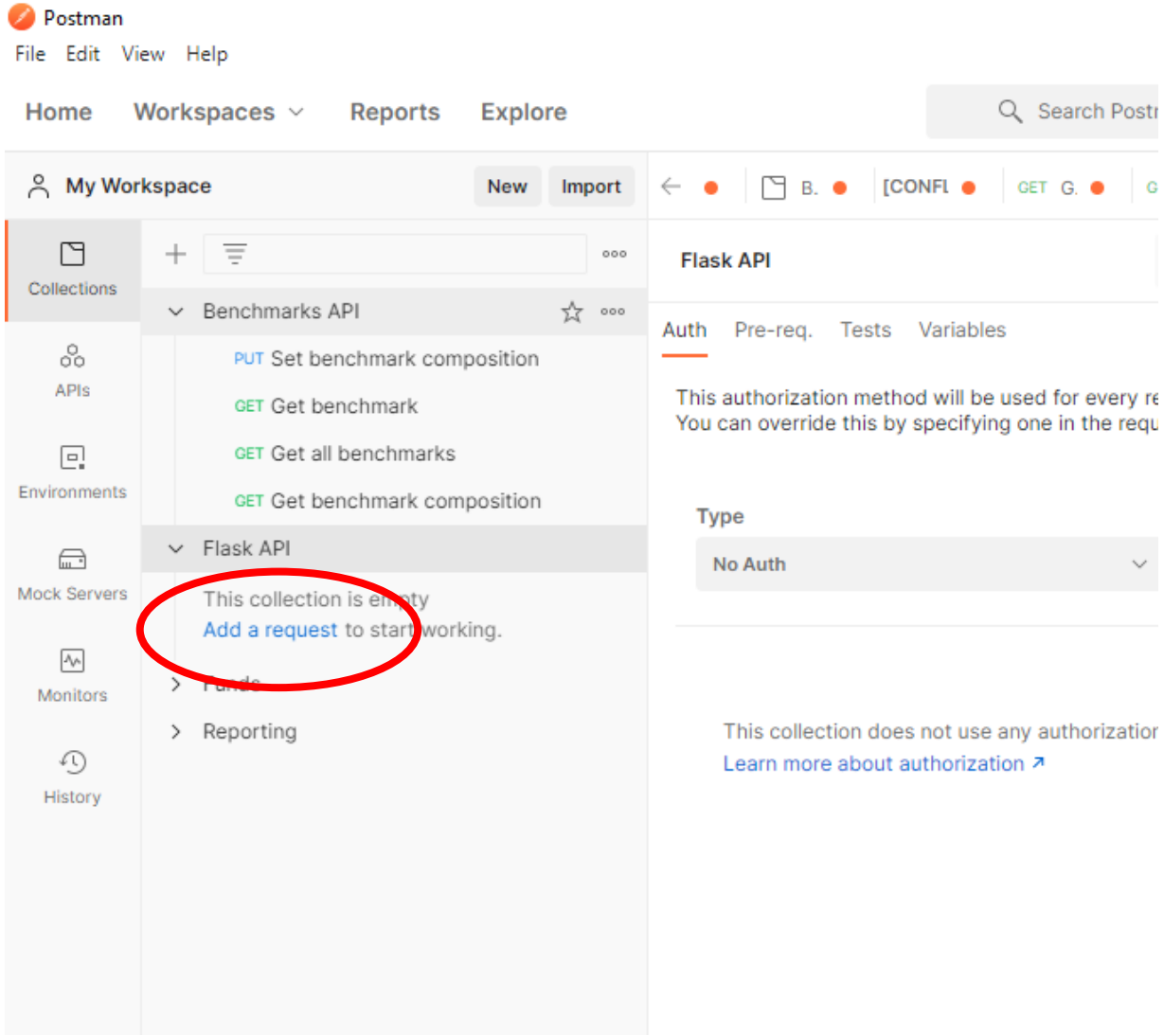
Create a collection



Give a name to your collection



Add a request



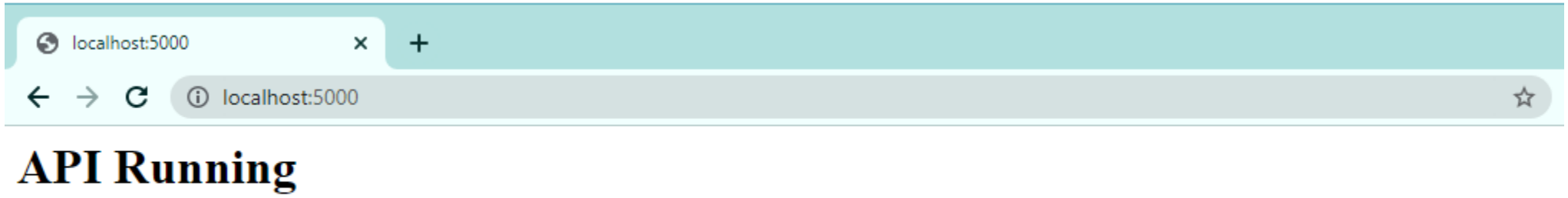
Run your server

- Run 3.4 create routes.py in python or in spyder or in pycharm
- Ensure that you have added a function to see if this is running like this

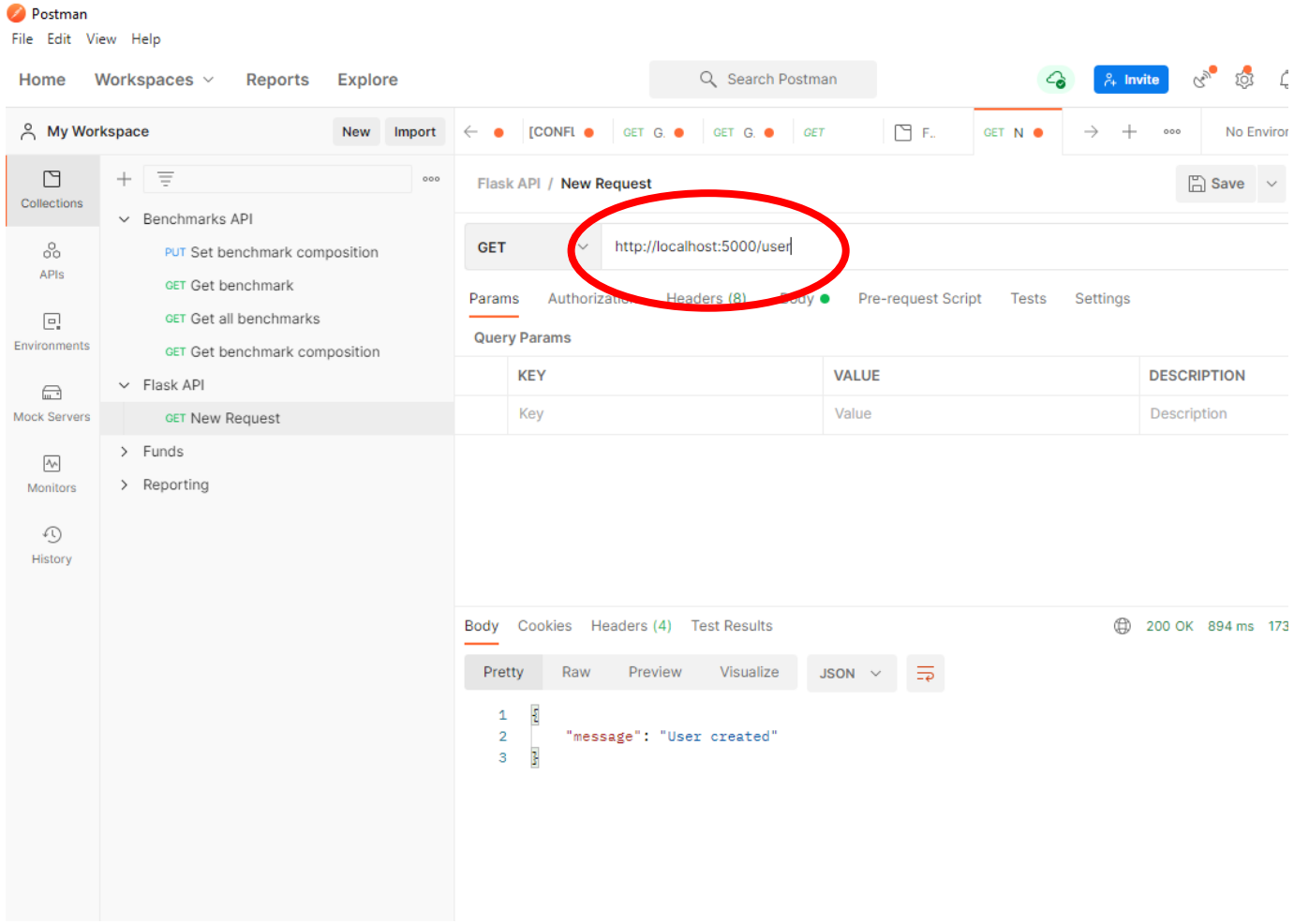
```
@app.route("/")
def home():
    return "<h1>API Running</h1>"
```

- You should see that the local host is working and get something like

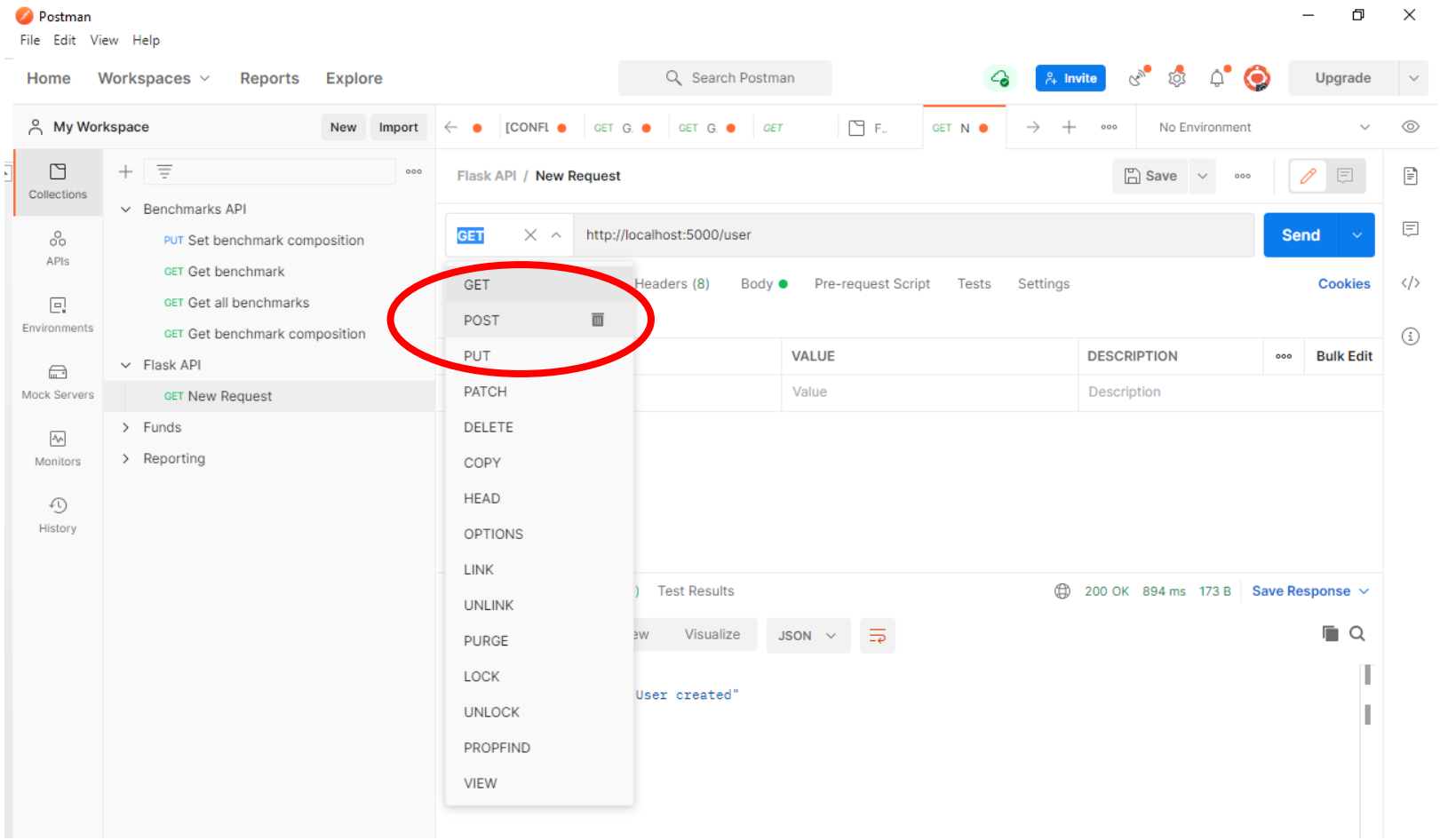
Ensure the API server is running



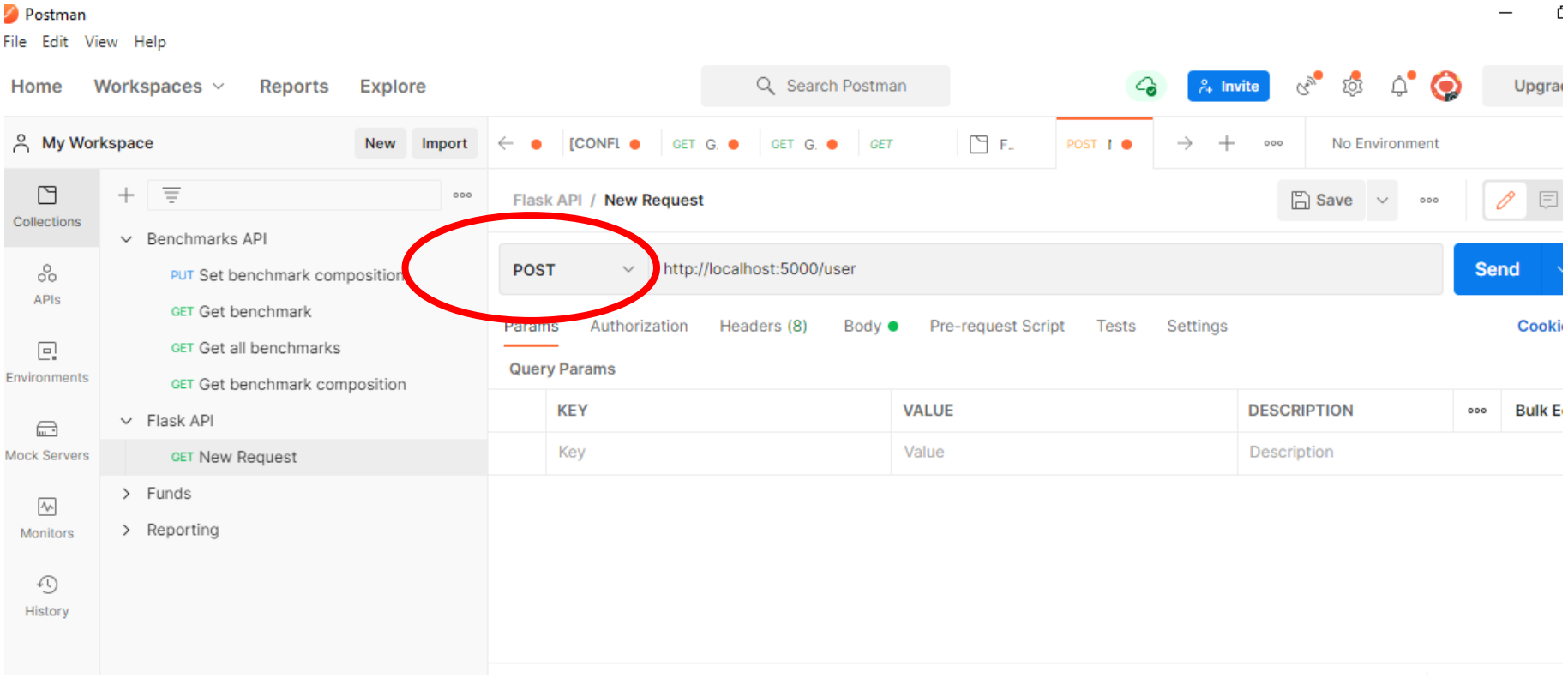
Type the correct url in postman



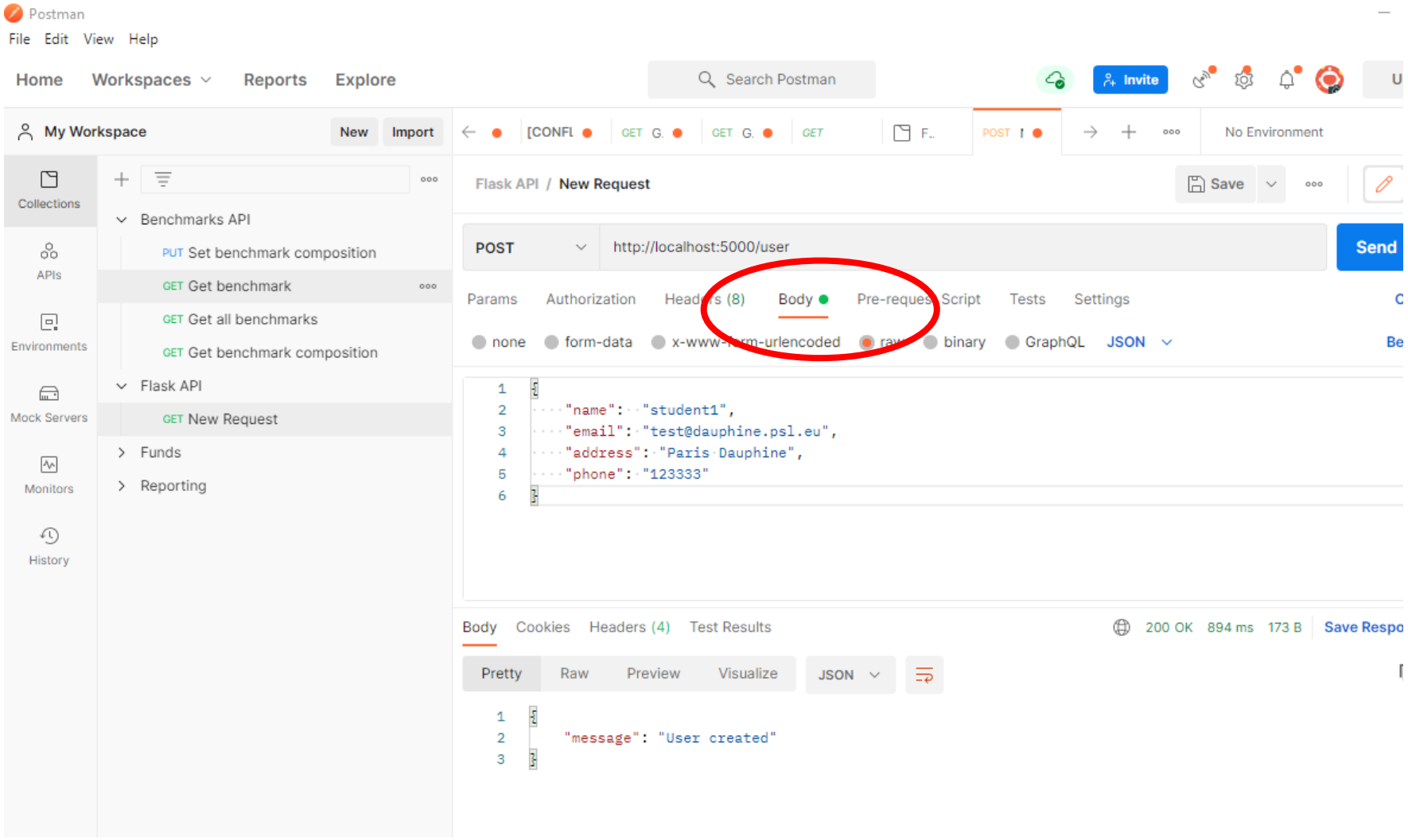
Change the GET to POST request



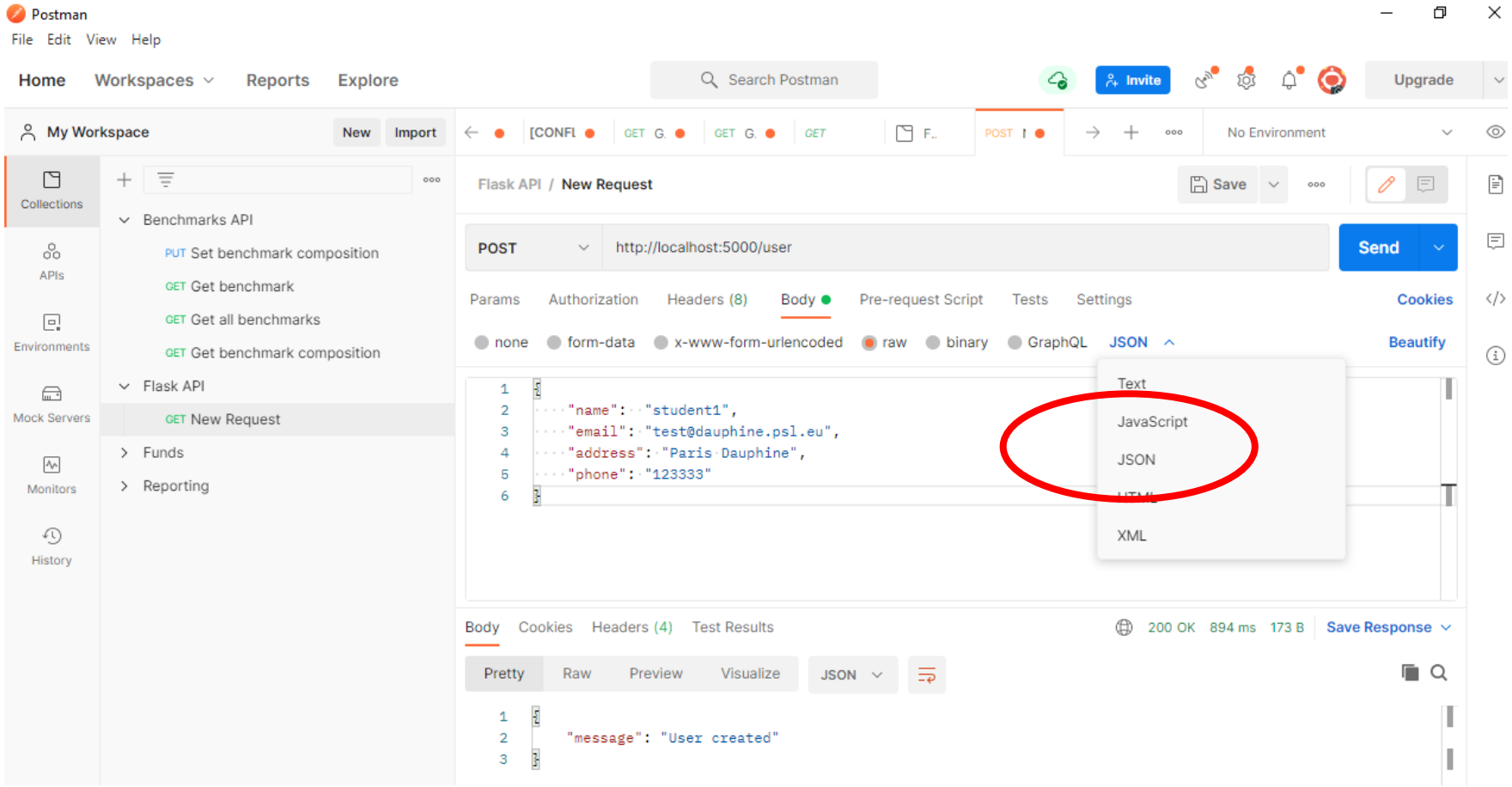
Say post


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Benchmarks API
PUT Set benchmark composition
GET Get benchmark
GET Get all benchmarks
GET Get benchmark composition
Flask API
GET New Request
Funds
Reporting
No Environment
Flask API / New Request
Save
POST
http://localhost:5000/user
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Query Params
KEY VALUE DESCRIPTION
Key Value Description

Goto body

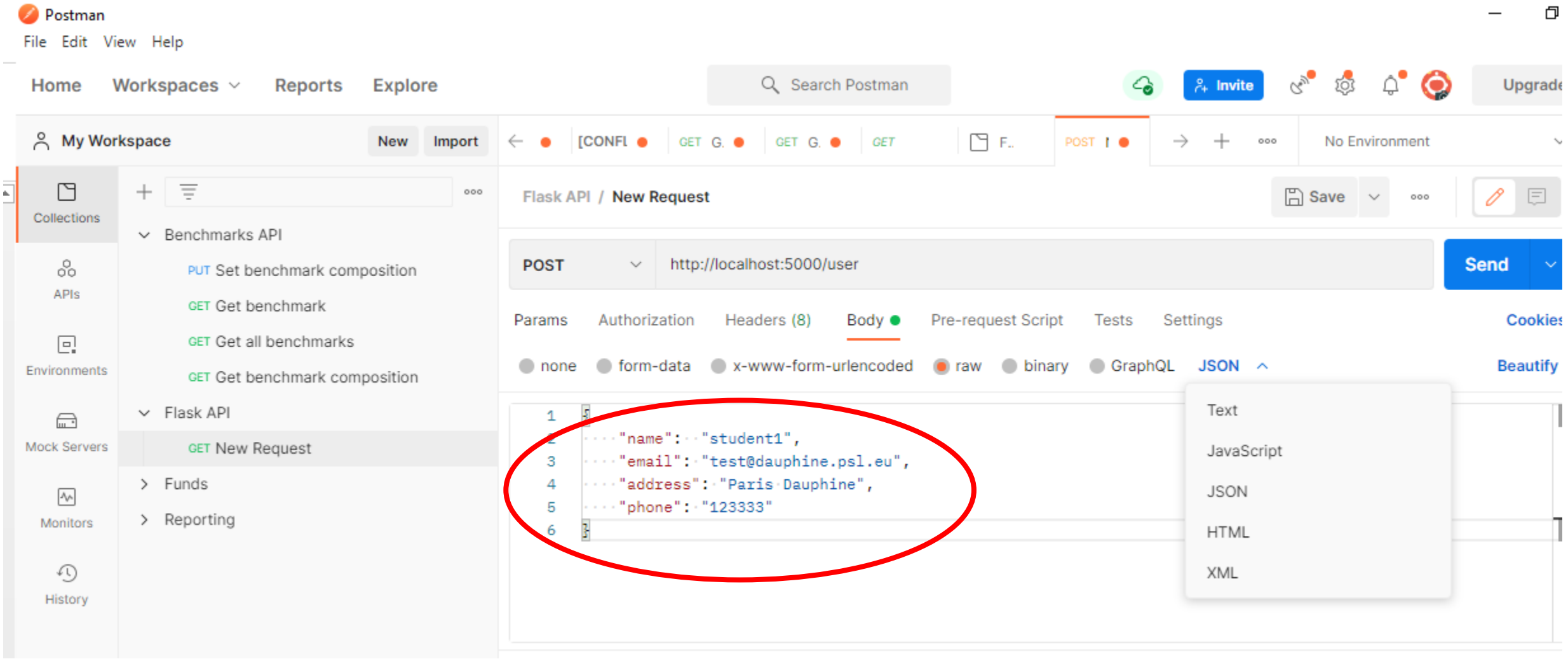

Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
My Workspace
New Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
History
Benchmarks API
PUT Set benchmark composition
GET Get benchmark
GET Get all benchmarks
GET Get benchmark composition
Flask API
GET New Request
Funds
Reporting
Flask API / New Request
Save
POST
http://localhost:5000/user
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
{
"name": "student1",
"email": "test@dauphine.psl.eu",
"address": "Paris Dauphine",
"phone": "123333"
}
Body Cookies Headers (4) Test Results
200 OK 894 ms 173 B
Pretty Raw Preview Visualize JSON
{
"message": "User created"
}

Select Json

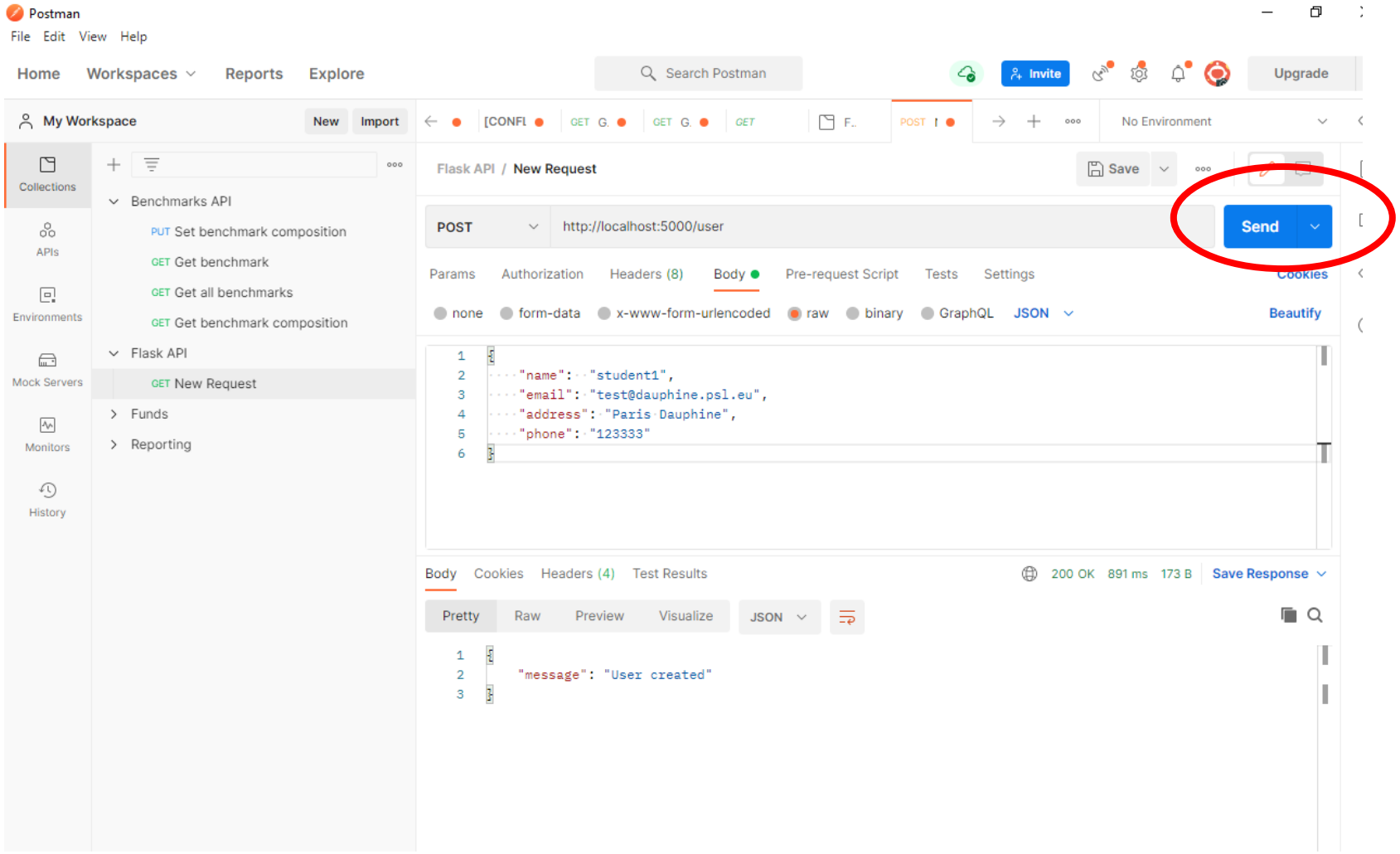

Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
History
Benchmarks API
PUT Set benchmark composition
GET Get benchmark
GET Get all benchmarks
GET Get benchmark composition
Flask API
GET New Request
Funds
Reporting
Flask API / New Request
Save
POST
http://localhost:5000/user
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"name": "student1",
"email": "test@dauphine.psl.eu",
"address": "Paris Dauphine",
"phone": "123333"
}
Text
JavaScript
JSON
XML
Body Cookies Headers (4) Test Results
200 OK 894 ms 173 B
Save Response
Pretty Raw Preview Visualize JSON
{
"message": "User created"
}

Enter the following input

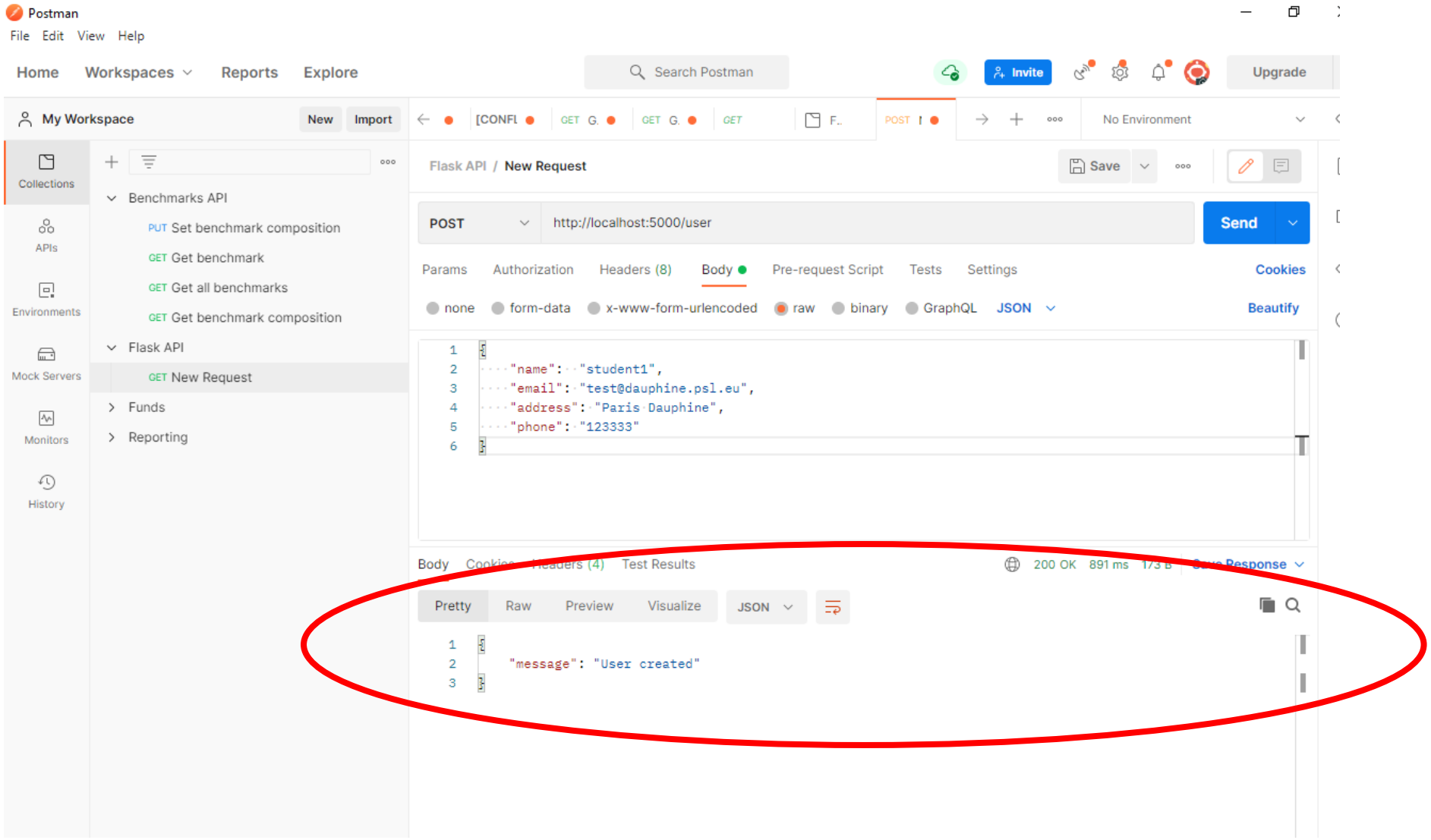

```
{
   "name":  "student1",
   "email": "test@dauphine.psl.eu",
   "address": "Paris Dauphine",
   "phone": "123333"
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Benchmarks API
PUT Set benchmark composition
GET Get benchmark
GET Get all benchmarks
GET Get benchmark composition
Flask API
GET New Request
Funds
Reporting
No Environment
Flask API / New Request
Save
POST
http://localhost:5000/user
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"name": "student1",
"email": "test@dauphine.psl.eu",
"address": "Paris Dauphine",
"phone": "123333"
}
Text
JavaScript
JSON
HTML
XML

Send the command

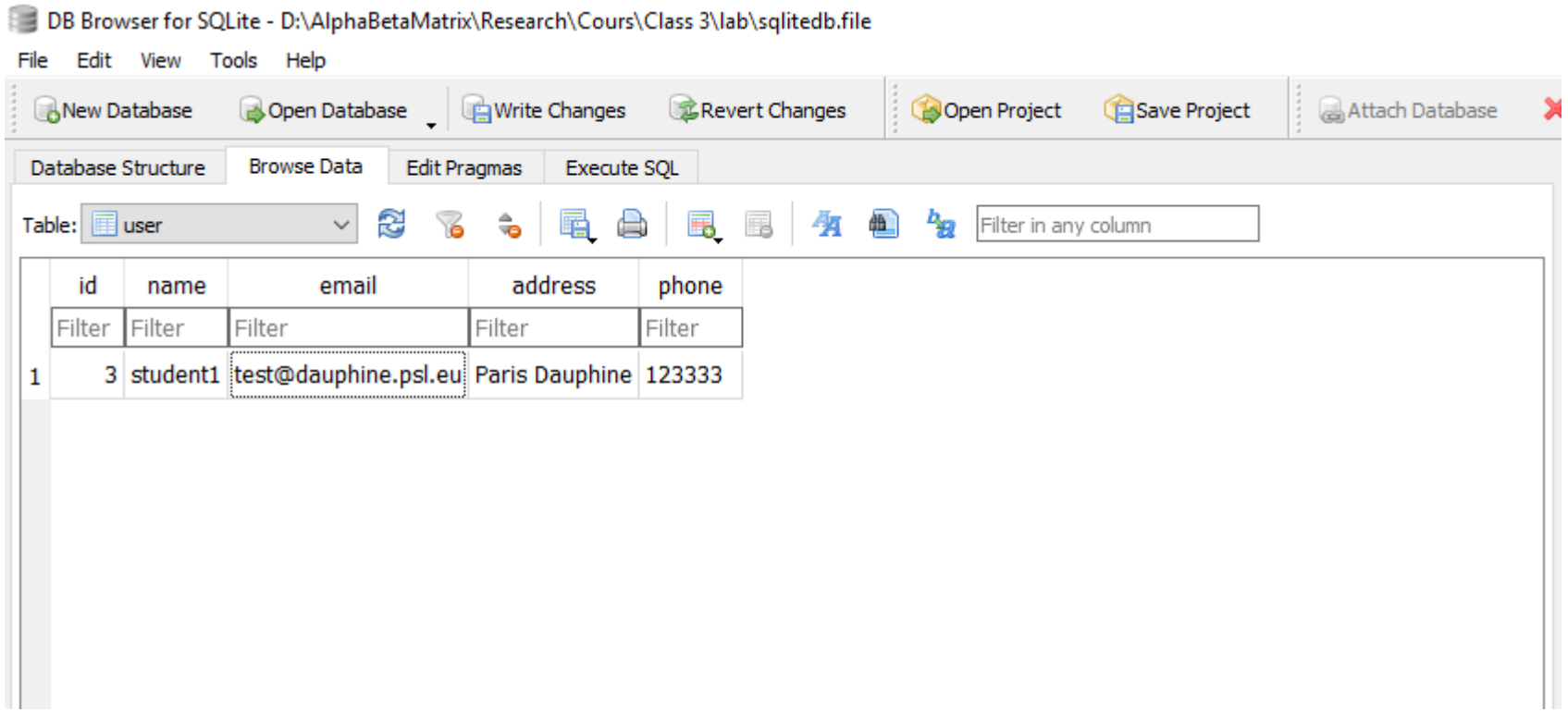


You should get something like



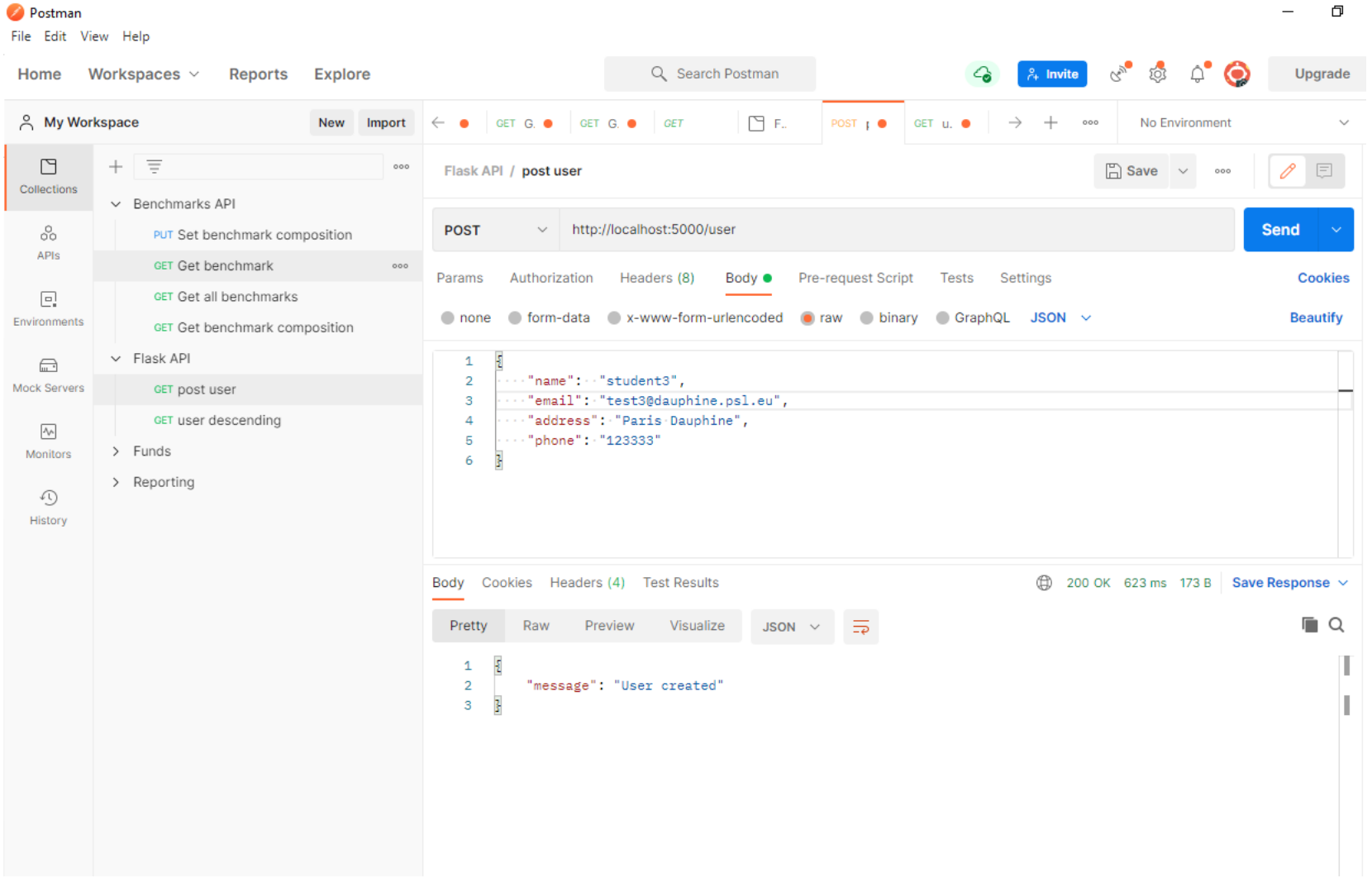
Check that there is a record in the database with DB Browser

- Open DB Browser

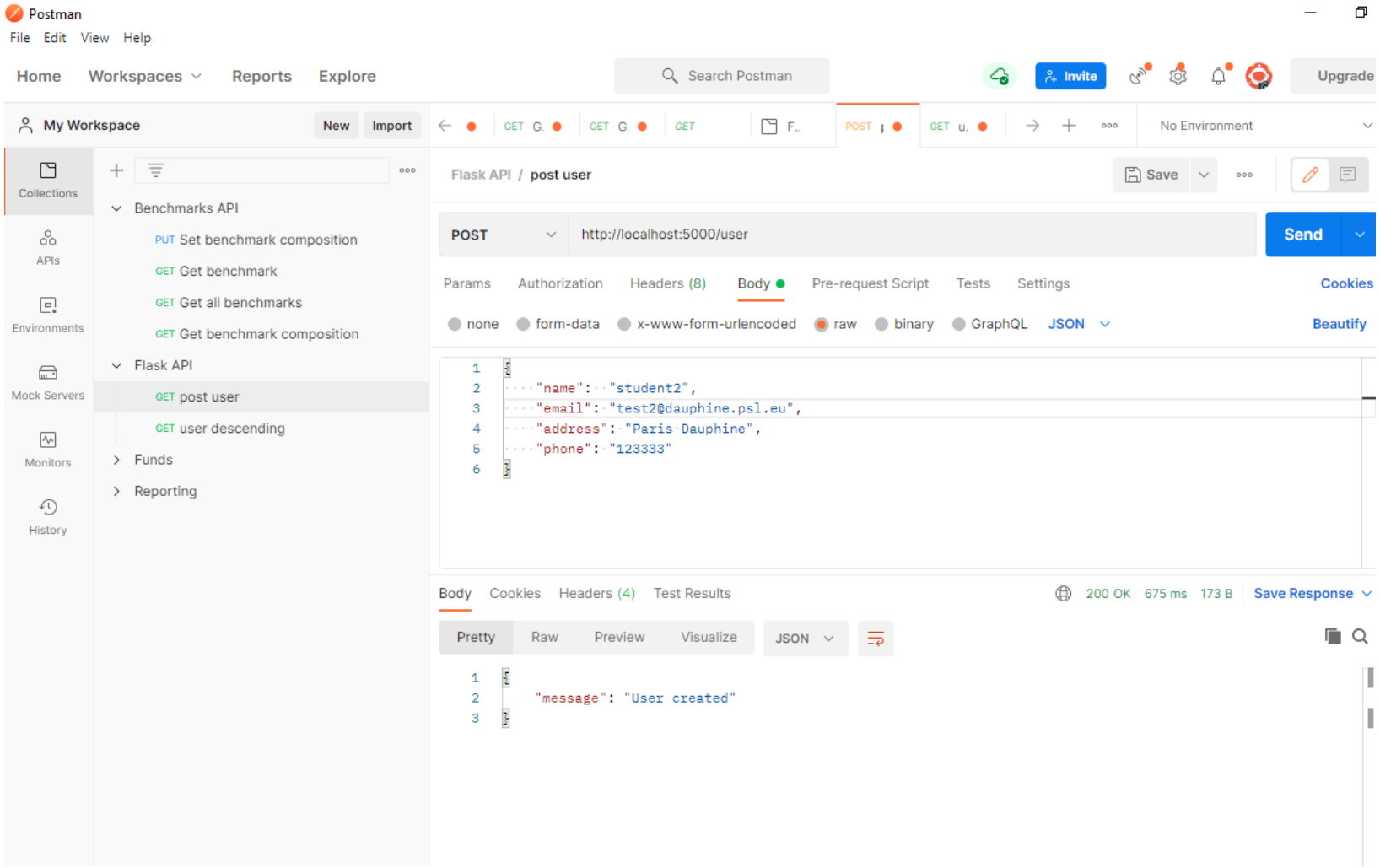


You are done!

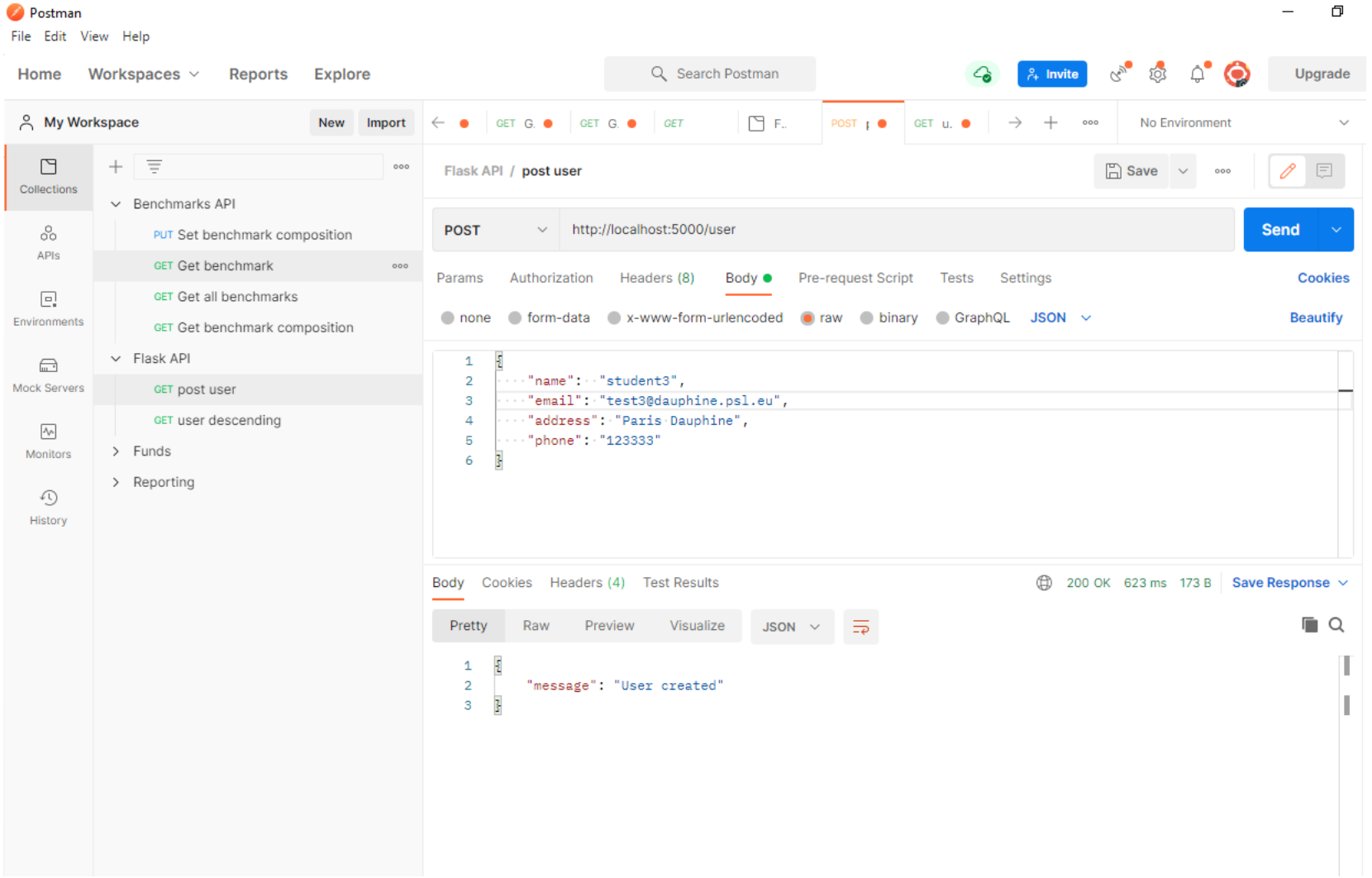
Add a few users!



Add a few users!



Add a few users!



Call the linked list to get descending and ascending users