

mavQ Devops Internship Practical Assignment

Problem 1

Objective

Create a JSON API Server that will receive GET and POST requests from any external client. The API server will connect with any SQL database (of your choice) and the server will be able to write the data received from APIs to the server and return the result based on the query. The read query will be able to support filters as well as defined by the data model. The entire stack (API Server & database) should be containerized in a docker container and the ecosystem should be able to run with a simple docker-compose up command.

[GitHub repository](#)

Specification

Data Model

1. Teacher

- a. teacher_id
- b. name
- c. is_active
- d. designation

2. Course

- a. course_id
- b. course_mentor
- c. name
- d. start_date
- e. end_date
- f. description
- g. is_active

Technologies used

1. Programming languages/framework for API: Django for Python
2. Containerization: Docker
3. Database: SQLite3
4. IDE: VSCode
5. OS: Linux (Fedora Linux)
6. Version Control: Git, GitHub
7. Testing API workflow: Postman API

Docker images used for containerization

1. [keinos/sqlite3](#)
2. [python](#)

Running the API and Docker images:

For proper documentation refer to [README.md](#) file in the GitHub repository

Copy and run these commands on your terminal

```
git clone https://github.com/vishu-25/mavQ-assignment-problem-1.git
cd mavQ-assignment-problem-1/src/
chmod +x setup.sh
./setup.sh
```

Screenshots of the process.

```
[vishu@fedora test]$ git clone https://github.com/vishu-25/mavQ-assignment-problem-1.git
cd mavQ-assignment-problem-1/src/
chmod +x setup.sh
./setup.sh
Cloning into 'mavQ-assignment-problem-1'...
remote: Enumerating objects: 74, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 74 (delta 10), reused 54 (delta 5), pack-reused 0
Receiving objects: 100% (74/74), 23.13 KiB | 215.00 KiB/s, done.
Resolving deltas: 100% (10/10), done.
Building the Docker images using Dockerfile and docker-compose
[+] Building 0.3s (2/2) FINISHED
=> CANCELED [internal] load build definition from Dockerfile
=> CANCELED [internal] load .dockerignore
canceled
Creating Docker containers using docker-compose up
[+] Running 2/2
# Container assignment-database Started
# Container src-assignment-api-server-1 Started
```

```
[vishu@fedora src]$ docker container ls --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
29777edcaeb0	src-assignment-api-server	"python manage.py ru..."	2 minutes ago	Up 2 minutes	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp	src-assignment-api-server-1
5543f452be87	keinos/sqlite3:latest	"/bin/sh -c /usr/bin..."	2 minutes ago	Exited (0) 2 minutes ago		assignment-database

```
[vishu@fedora src]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
src-assignment-api-server	latest	e469d72ccd12	About a minute ago	1.04GB
keinos/sqlite3	latest	226a1123d74d	2 months ago	14.8MB

```
[vishu@fedora ~]$ cd mavQ-assignment-problem-1/
[vishu@fedora mavQ-assignment-problem-1]$ cd src/
[vishu@fedora src]$ ./setup.sh
Building the Docker images using Dockerfile and docker-compose
[+] Building 9.5s (10/10) FINISHED
=> [internal] load docker image
=> => transferring context: 28
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load metadata for docker.io/library/python:3.9
=> [1/5] FROM docker.io/library/python:3.9@sha256:10f220f410b87604e204d05dad51765753e5067cc497370007136474008054d
=> [internal] load build context
=> => transferring context: 2.74kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt /app
=> CACHED [4/5] RUN pip install -r requirements.txt
=> [5/5] COPY
=> exporting to image
=> => exporting layers
=> => writing image sha256:e469d72ccd121258cae7441063290dc106d0b1422045f9ee413b07123b07b40e
=> => naming to docker.io/library/src-assignment-api-server
Creating Docker containers using docker-compose up
[+] Running 6/6
# assignment-database Pulled
# 544f08030ba5 Pull complete
# 6ae78021a8ff Pull complete
# 170051020005 Pull complete
# 002c180a2a67 Pull complete
# 4f40700aef54 Pull complete
[+] Running 2/2
# Container assignment-database Started
# Container src-assignment-api-server-1 Started
```

```
[vishu@fedora ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2e77c8f377de	project_assignment-api-server	"python manage.py ru..."	About a minute ago	Up About a minute	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp	project_assignment-api-server_1

```
[vishu@fedora ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2e77c8f377de	project_assignment-api-server	"python manage.py ru..."	About a minute ago	Up About a minute	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp	project_assignment-api-server_1
ed3c3cf6f082	keinos/sqlite3:latest	"/bin/sh -c /usr/bin..."	34 minutes ago	Exited (0) About a minute ago		assignment-database

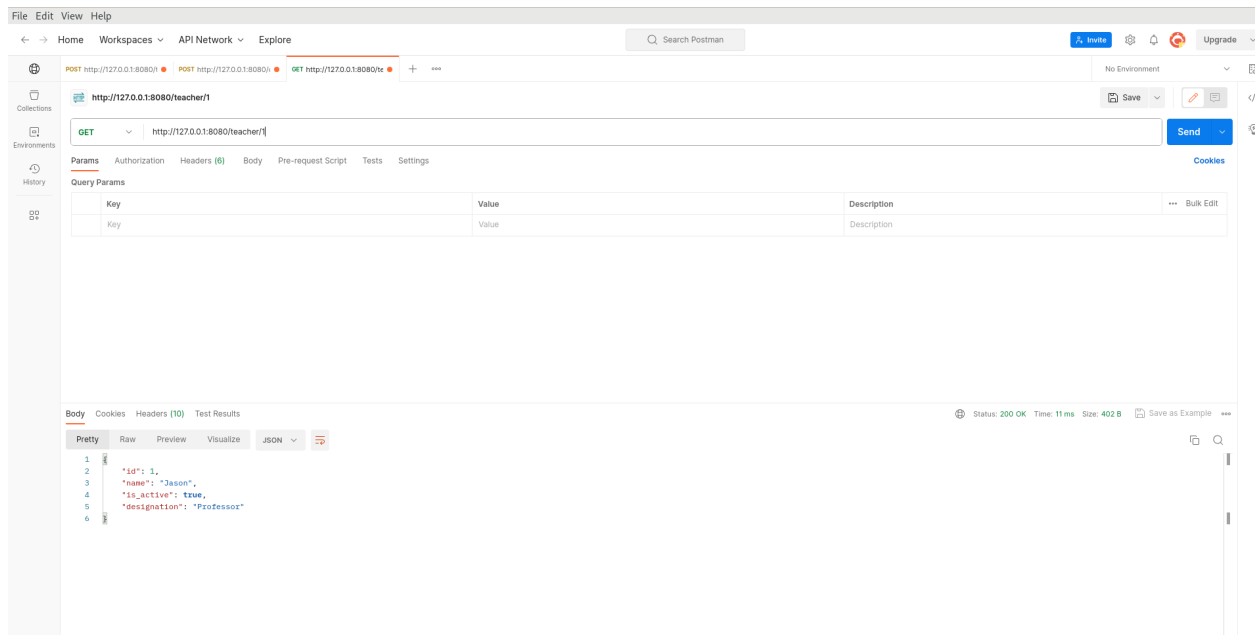
API Specification

1. GET - /teacher/{id} - Get a specific teacher record by ID.

- a. Example Request: /teacher/1

Example Response:

```
{  
  "teacher_id":1,  
  "name":"Jason",  
  "is_active:true,  
  "designation":"Professor"  
}
```

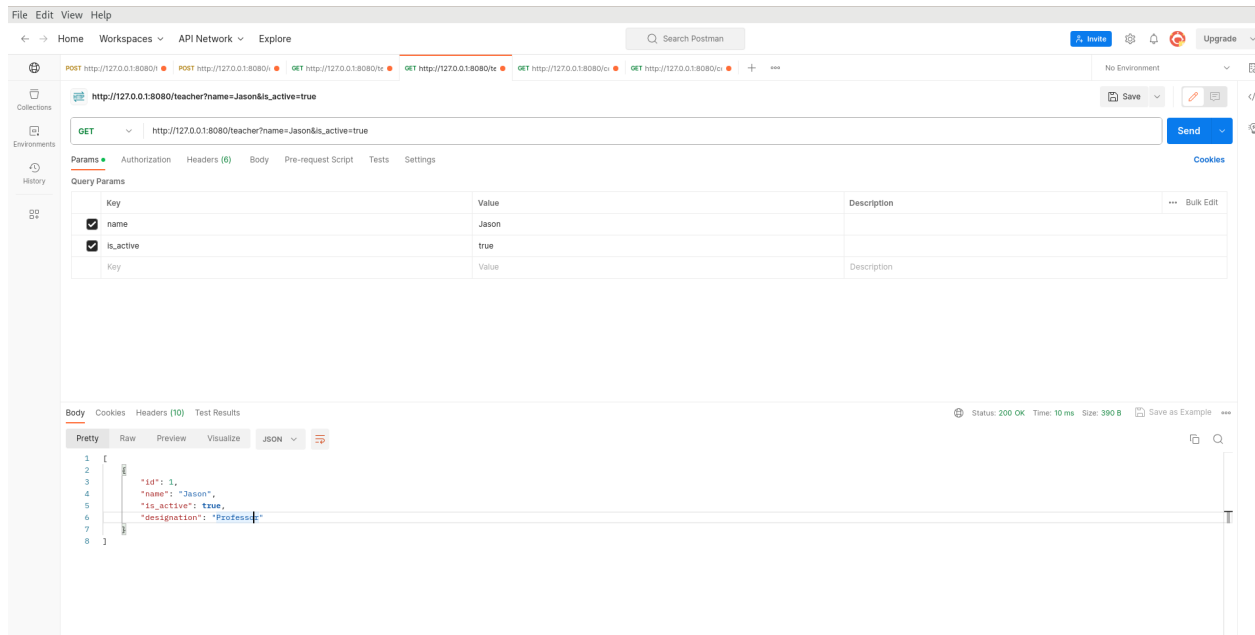


2. GET - /teacher?<field=value> - Get a list of teachers based on filters.

a. Example Request: /teacher?name=Jason&is_active=true

Example Response:

```
[
  {
    "teacher_id":1,
    "name": "Jason",
    "is_active":true,
    "designation": "Professor"
  },
  {
    "teacher_id":5,
    "name": "Jason",
    "is_active":true,
    "designation": "Associate Professor"
  }
]
```



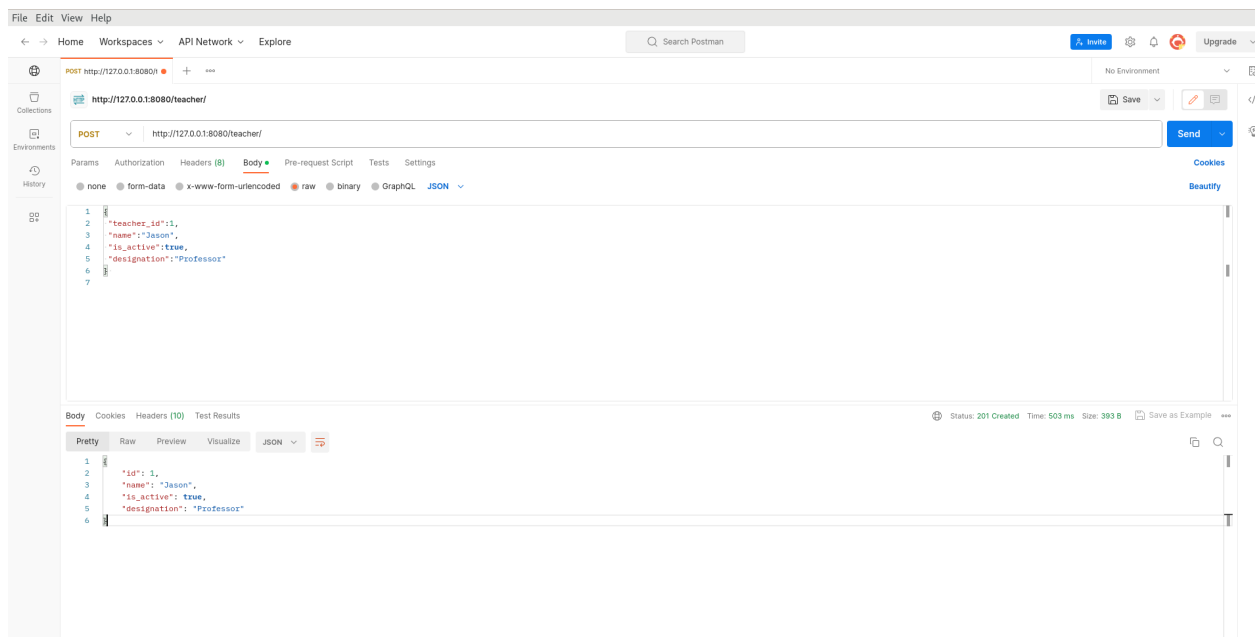
3. Post - /teacher - Create a new teacher record using the JSON payload

a. Example request: - /teacher

```
{  
  "teacher_id":1,  
  "name":"Jason",  
  "is_active:true,  
  "designation":"Professor"  
}
```

Example response:

```
{  
  "teacher_id":1,  
  "name":"Jason",  
  "is_active:true,  
  "designation":"Professor"  
}
```



4. GET - /course?<field=value> - Get a list of courses based on filters.

a.

Example Request: - /course/

Example Response:

```
{
  "course_id":1,
  "Course_mentor":{
    "teacher_id":6,
    "name":"John",
    "Is_active":true,
    "designation":"Assistant Professor"
  },
  "name":"Introduction to Devops",
  "start_date":"2024-01-01",
  "end_date":"2024-02-02"
  "description":"Basic concepts of DevOps",
  "is_active":true
}
```

FileEditViewHelp

HomeWorkspacesAPI NetworkExplore

Search Postman

WriteNo EnvironmentSaveSendCookiesBeauty

POSThttp://127.0.0.1:8080/course/

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1{"course_id": 1,
2  "course_mentor": 6,
3  "name": "Introduction to DevOps",
4  "start_date": "2024-01-01",
5  "end_date": "2024-02-02",
6  "description": "Basic concepts of DevOps",
7  "is_active": true
8
9
10
```

BodyCookiesHeaders (10)Test ResultsStatus: 201 CreatedTime: 519 msSize: 570 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1{"id": 1,
2  "course_mentor": {
3    "id": 6,
4    "name": "Olivia",
5    "is_active": false,
6    "designation": "Visiting Professor"
7  },
8  "name": "Introduction to DevOps",
9  "start_date": "2024-01-01",
10 "end_date": "2024-02-02",
11 "description": "Basic concepts of DevOps",
12 "is_active": true
13
14
```

127.0.0.1:8080/course/

Django REST framework

Api Root / Course List

Course List

FiltersOPTIONSGET

GET /course/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
[
  {
    "id": 1,
    "course_mentor": {
      "id": 6,
      "name": "Olivia",
      "is_active": false,
      "designation": "Visiting Professor"
    },
    "name": "Introduction to DevOps",
    "start_date": "2024-01-01",
    "end_date": "2024-02-02",
    "description": "Basic concepts of DevOps",
    "is_active": true
  }
]
```

Raw dataHTML form

Name

Start date

End date

Description

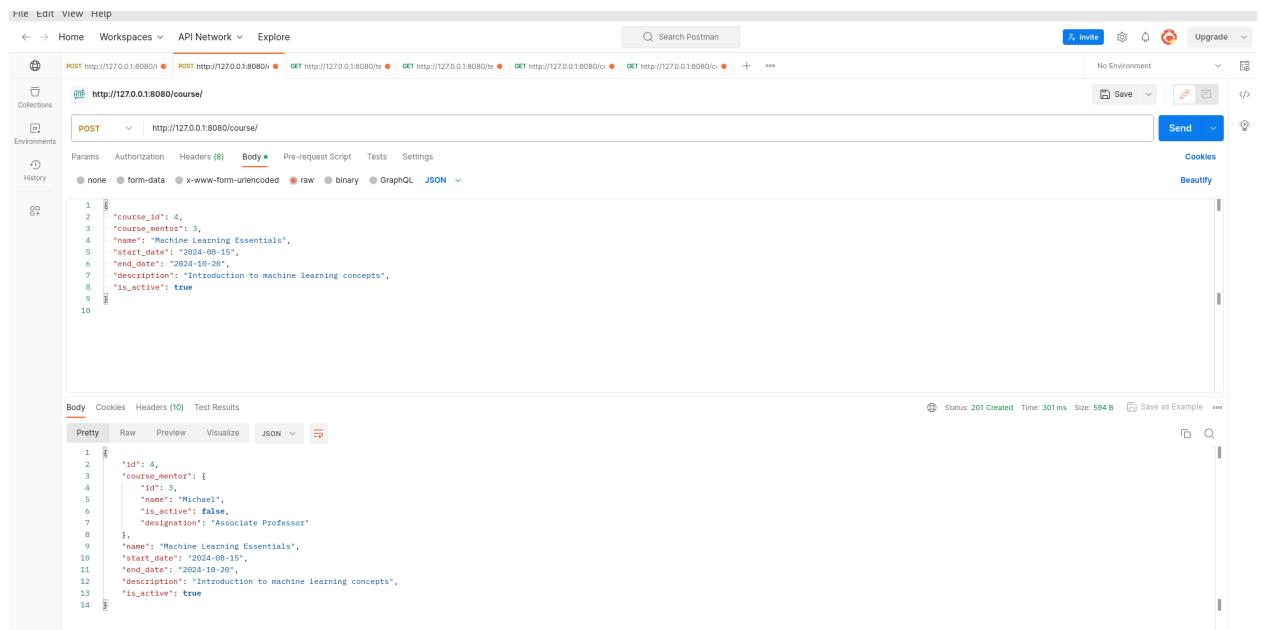
5. POST - /course - create a new course using the JSON payload

a. Example Request: - /course/

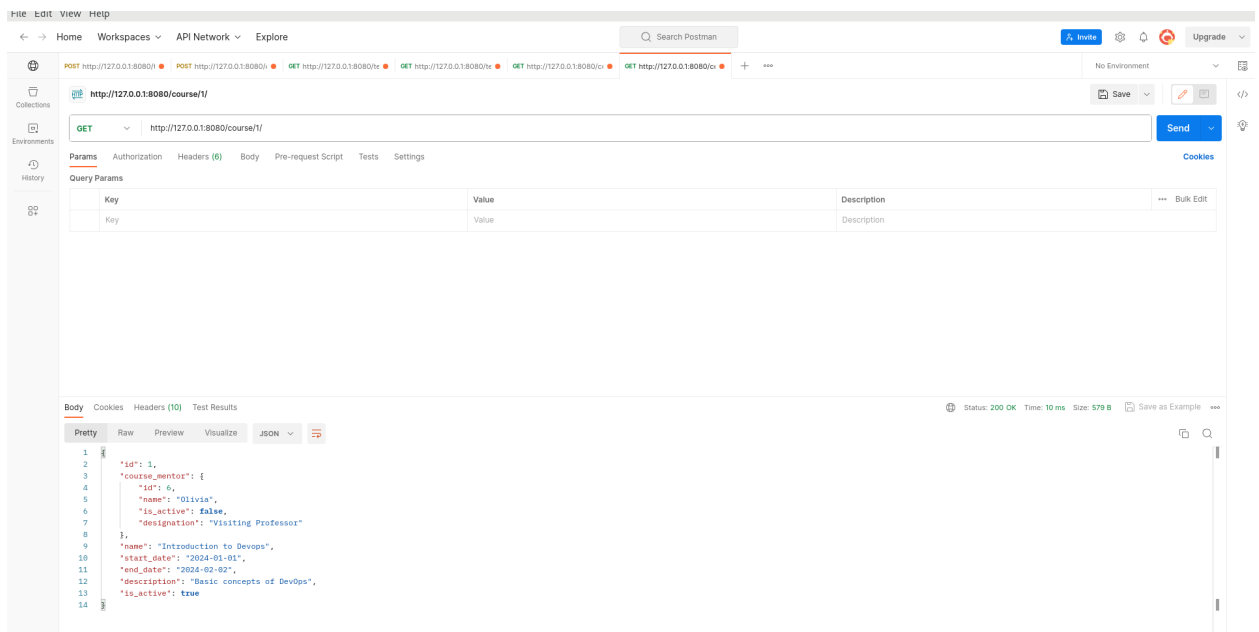
```
{
  "course_id":1,
  "course_mentor":6,
  "name":"Introduction to Devops",
  "start_date":"2024-01-01",
  "end_date":"2024-02-02"
  "description":"Basic concepts of DevOps",
  "is_active":true
}
```

Example Response:

```
{
  "course_id":1,
  "Course_mentor":{
    "Teacher_id":6,
    "name":"John",
    "Is_active":true,
    "designation":"Assistant Professor"
  },
  "name":"Introduction to Devops",
  "start_date":"2024-01-01",
  "end_date":"2024-02-02"
  "description":"Basic concepts of DevOps",
  "is_active":true
}
```



6. GET - /course/{id}
- a. Example Request: - /courses/1
- Example Response: {
- ```
"course_id":1,
"Course_mentor":{
"Teacher_id":6,
"name":"John",
"Is_active":true,
"designation":"Assistant Professor"
},
"name":"Introduction to Devops",
"start_date":"2024-01-01",
"end_date":"2024-02-02"
"description":"Basic concepts of DevOps",
"is_active":true
}
```



## Sample Test cases

Test case 1 :

Command:

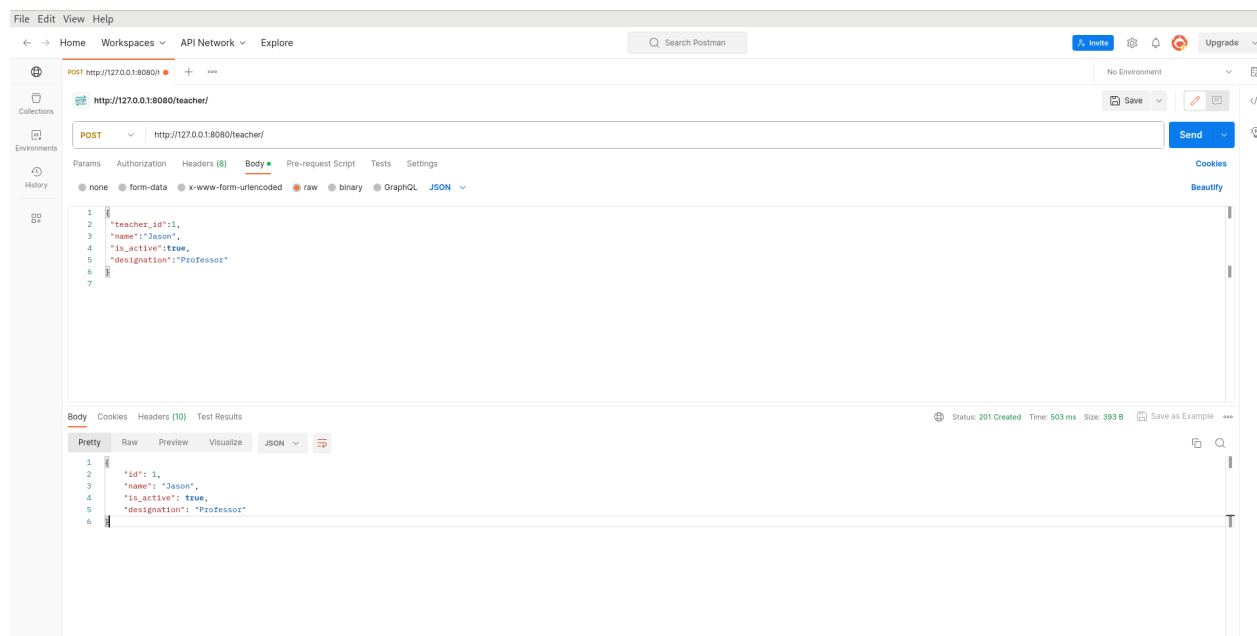
```
curl --location --request POST 'http://localhost:8080/teacher' \
--header 'accept: application/json' \
--header 'Content-Type: application/json' \
--data '{
 "teacher_id":1,
 "name":"Jason",
 "is_active":true,
 "designation":"Professor"
'
```

Output:

Response Code: 200

JSON:

```
{
 "teacher_id":1,
 "name":"Jason",
 "is_active":true,
 "designation":"Professor"
}
```



Test case 2 :

Command:

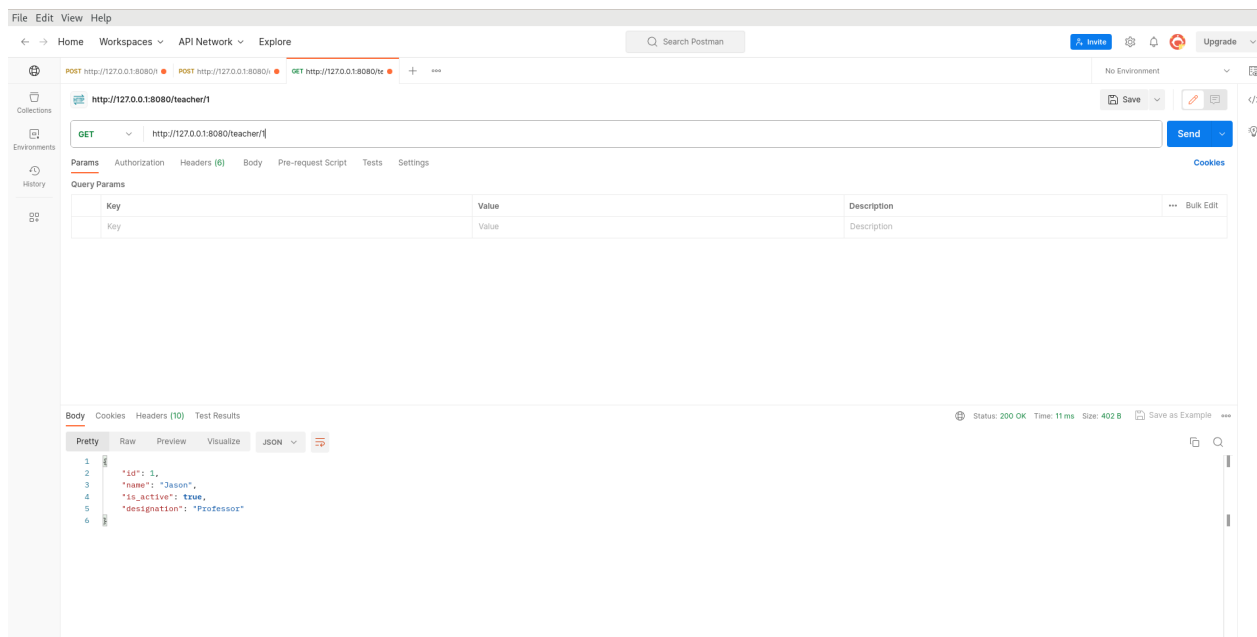
```
curl --location --request GET 'http://localhost:8080/teacher/1' \
--header 'accept: application/json' \
--header 'Content-Type: application/json'
```

Output:

Response Code: 200

JSON:

```
{
 "teacher_id":1,
 "name":"Jason",
 "is_active":true,
 "designation":"Professor"
}
```



Test case 3:

Command:

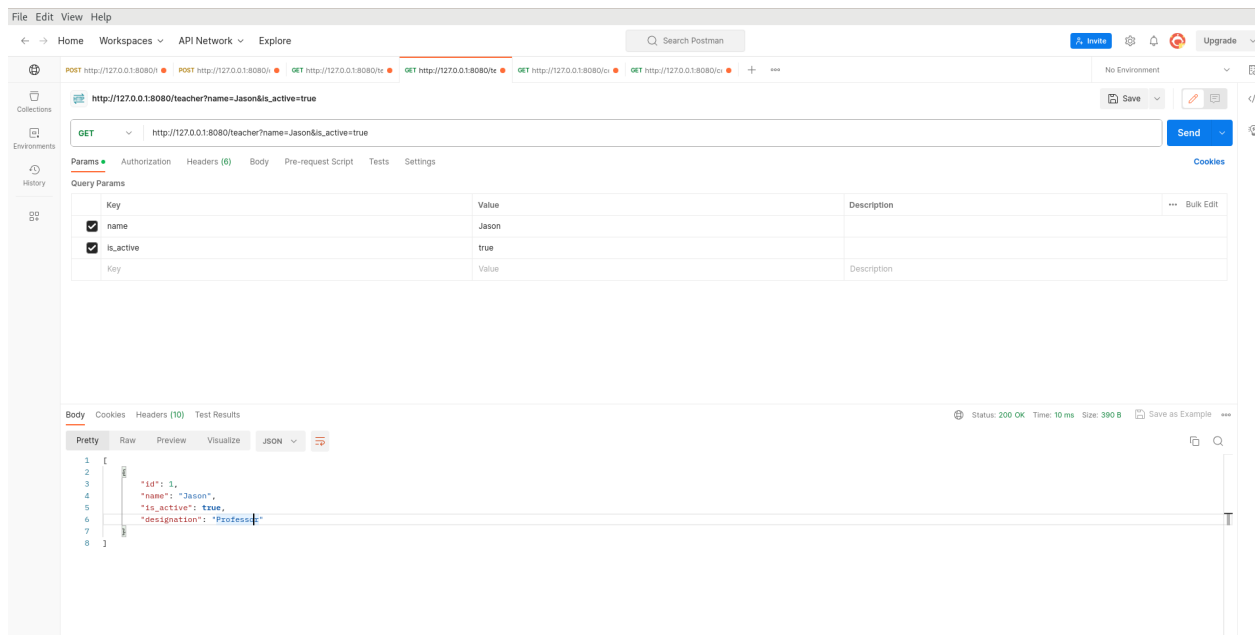
```
curl --location 'http://localhost:8080/teacher?name=Jason&is_active=true' \
--header 'accept: application/json' \
--header 'Content-Type: application/json'
```

Output:

Response Code: 200

JSON:

```
[
 {
 "teacher_id":1,
 "name":"Jason",
 "is_active":true,
 "designation":"Professor"
 }
]
```



## References & Documentation

- [Docker](#)
- [Django](#)
- [Python](#)
- [Postman API](#)
- [SQLite3](#)
- [Google](#)
- [YouTube: Home](#)