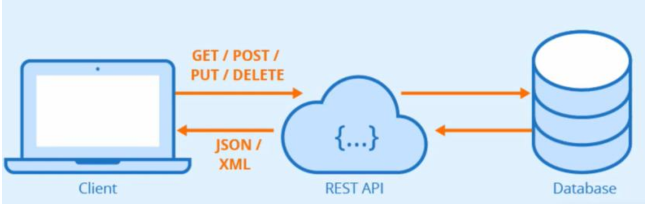
Test Automation using:

- Python
- Postman
- PyCharm
- JSon Beautifier and Editor
- PyTest
- Libraries (ie., requests)

Document highlight:

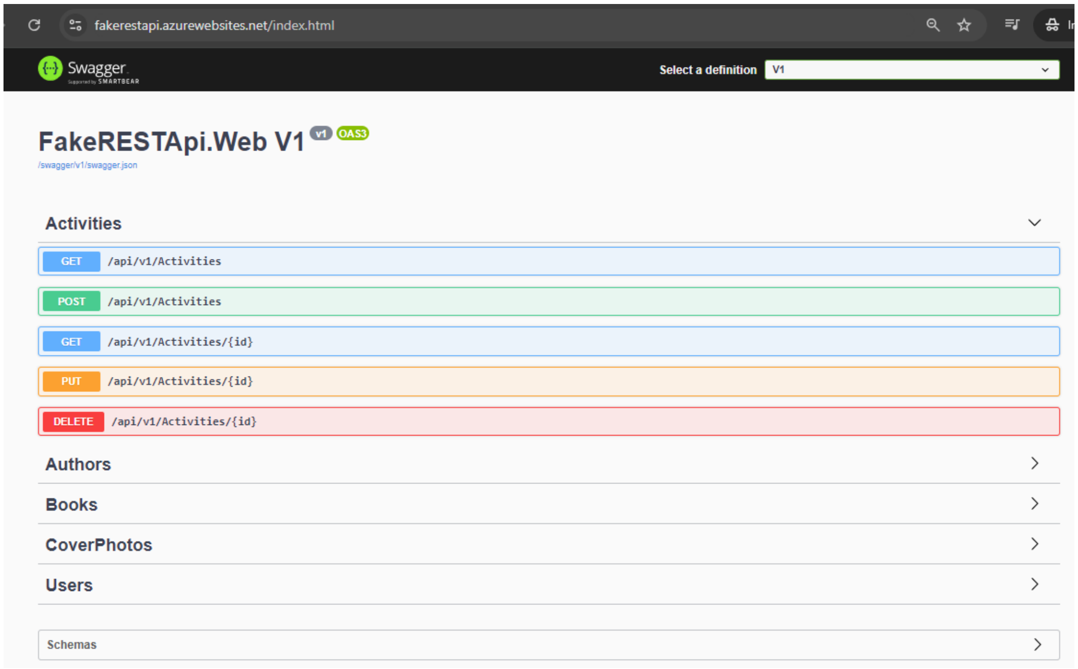
- GET method
- POST method
- PUT method
- OAuth2 Bearer Token request
- Passing parameters in URL
- Passing payload from a JSON file
- Python request API automation with PyTest

1. API Concept (interface between client and server (database), request and response)



2. Sampling APIs (using Swagger)

url:
https://fakerestapi.azurewebsites.net/index.html



fakerestapi.azurewebsites.net/index.html

Swagger

Select a definition V1

FakeRESTApi.Web V1 v1 OAS3

/swagger/v1/swagger.json

Activities

GET /api/v1/Activities

POST /api/v1/Activities

GET /api/v1/Activities/{id}

PUT /api/v1/Activities/{id}

DELETE /api/v1/Activities/{id}

Authors

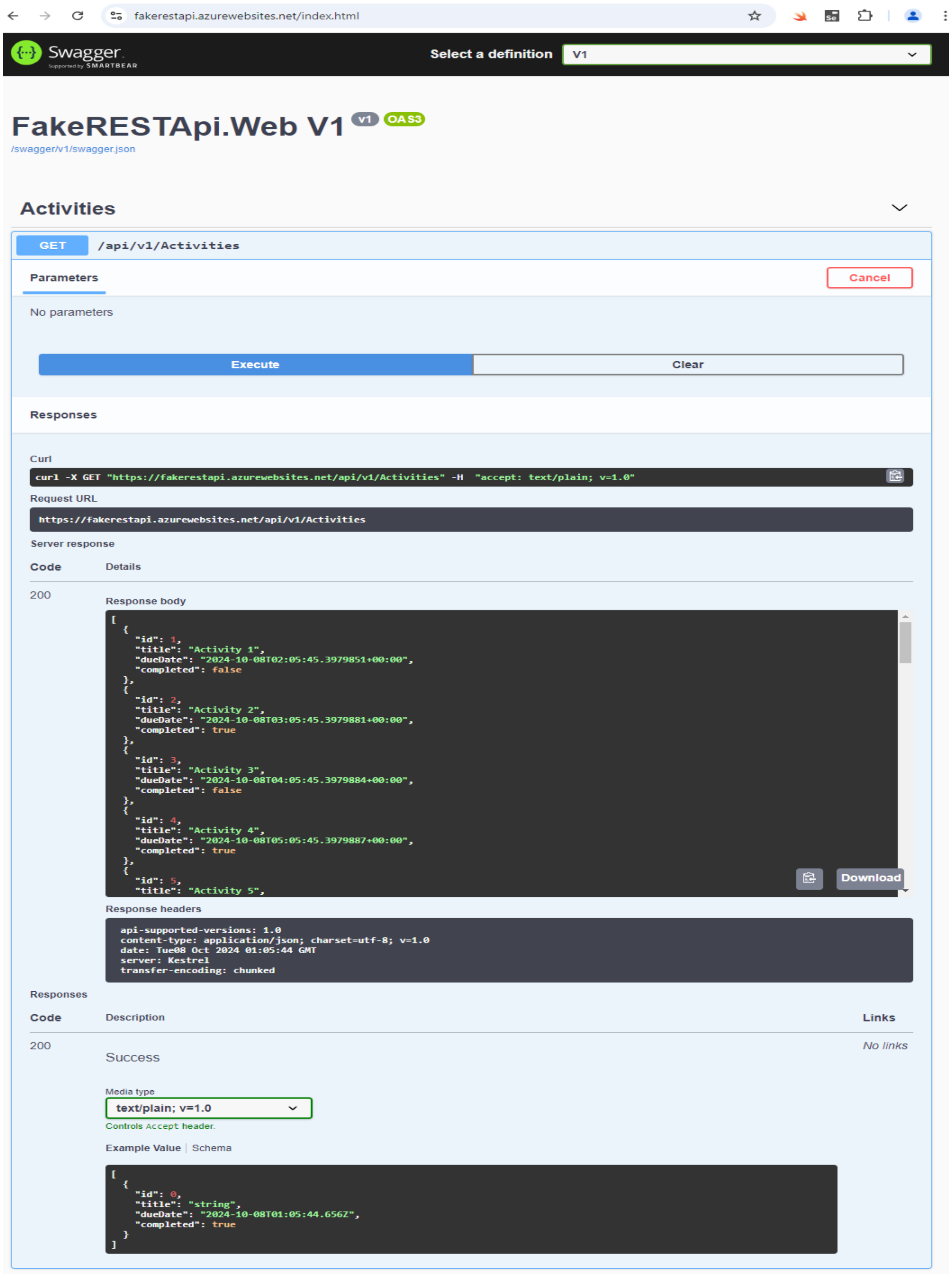
Books

CoverPhotos

Users

Schemas

3. Executing a GET method



FakeRESTApi.Web V1 v1 OAS3

/swagger/v1/swagger.json

Activities

GET /api/v1/Activities

No parameters

Responses

Curl

```
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Activities" -H  "accept: text/plain; v=1.0"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities
```

Server response

Code Details

200

Response body

```
[
  {
    "id": 1,
    "title": "Activity 1",
    "dueDate": "2024-10-08T02:05:45.3979851+00:00",
    "completed": false
  },
  {
    "id": 2,
    "title": "Activity 2",
    "dueDate": "2024-10-08T03:05:45.3979881+00:00",
    "completed": true
  },
  {
    "id": 3,
    "title": "Activity 3",
    "dueDate": "2024-10-08T04:05:45.3979884+00:00",
    "completed": false
  },
  {
    "id": 4,
    "title": "Activity 4",
    "dueDate": "2024-10-08T05:05:45.3979887+00:00",
    "completed": true
  },
  {
    "id": 5,
    "title": "Activity 5",
```

Response headers

```
api-supported-versions: 1.0
content-type: application/json; charset=utf-8; v=1.0
date: Tue08 Oct 2024 01:05:44 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code	Description	Links
200	Success	*No links*

Media type

text/plain; v=1.0

Controls Accept header.

Example Value | Schema

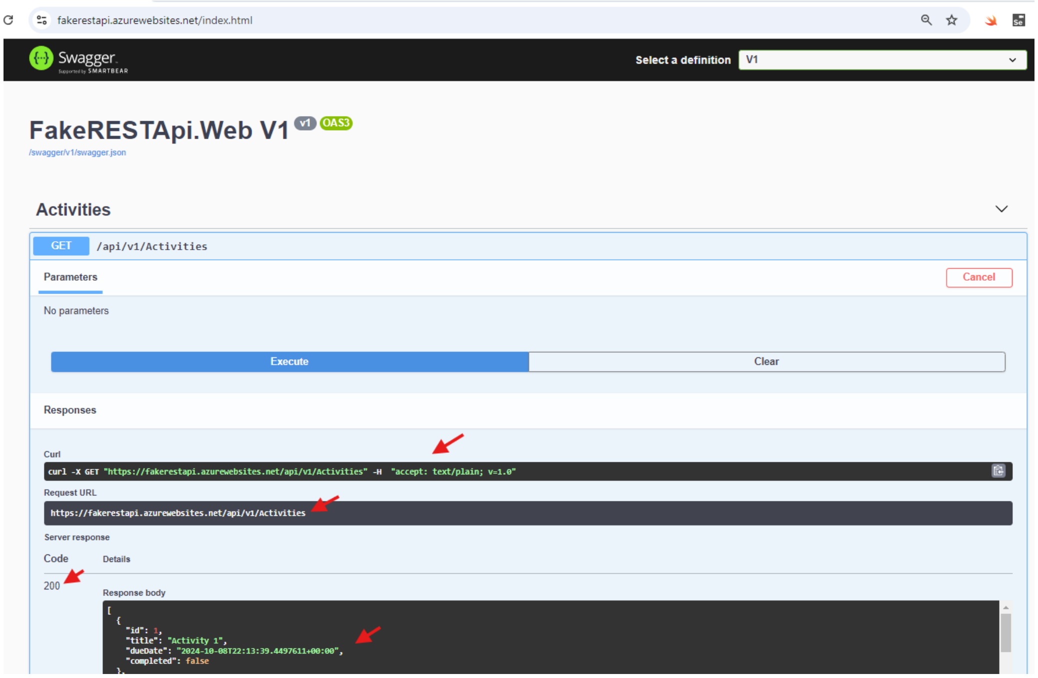
```
[
  {
    "id": 0,
    "title": "string",
    "dueDate": "2024-10-08T01:05:44.656Z",
    "completed": true
  }
]
```

4. Sample GET without parameter (copy the sampling url from Swagger and into Postman)

Validate and compare the following response results:

- **Status**
- **Body**
- **Header**
- **Time**

Swagger



Curl

```
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Activities" -H  "accept: text/plain; v=1.0"
```

Request URL

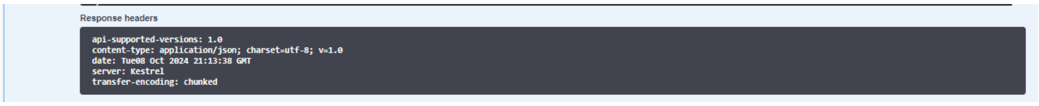
```
https://fakerestapi.azurewebsites.net/api/v1/Activities
```

Server response

Code: 200

Response body

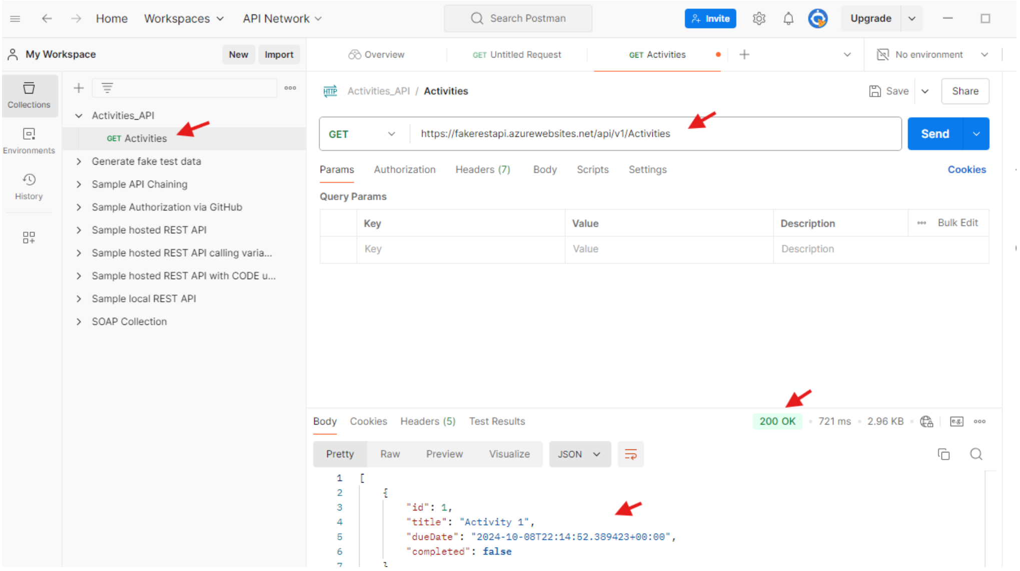
```
[
  {
    "id": 1,
    "title": "Activity 1",
    "dueDate": "2024-10-08T22:13:39.4497611+00:00",
    "completed": false
  },
```



Response headers

```
api-supported-versions: 1.0
content-type: application/json; charset=utf-8; v=1.0
date: Tue08 Oct 2024 21:13:38 GMT
server: Kestrel
transfer-encoding: chunked
```

Postman

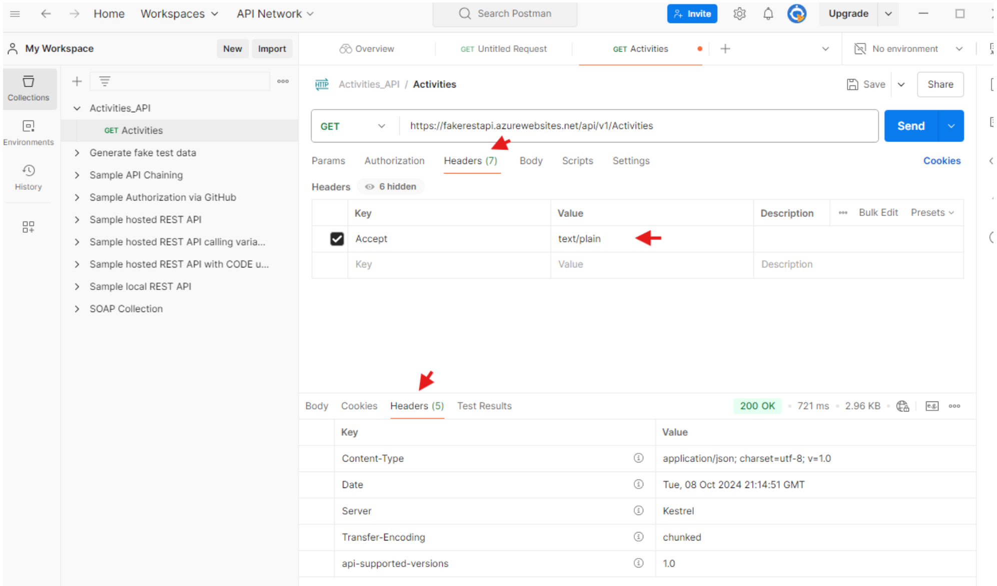


GET https://fakerestapi.azurewebsites.net/api/v1/Activities

Query Params

Key	Value	Description
Key	Value	Description

```
[
    {
        "id": 1,
        "title": "Activity 1",
        "dueDate": "2024-10-08T22:14:52.389423+00:00",
        "completed": false
    }
```



GET https://fakerestapi.azurewebsites.net/api/v1/Activities

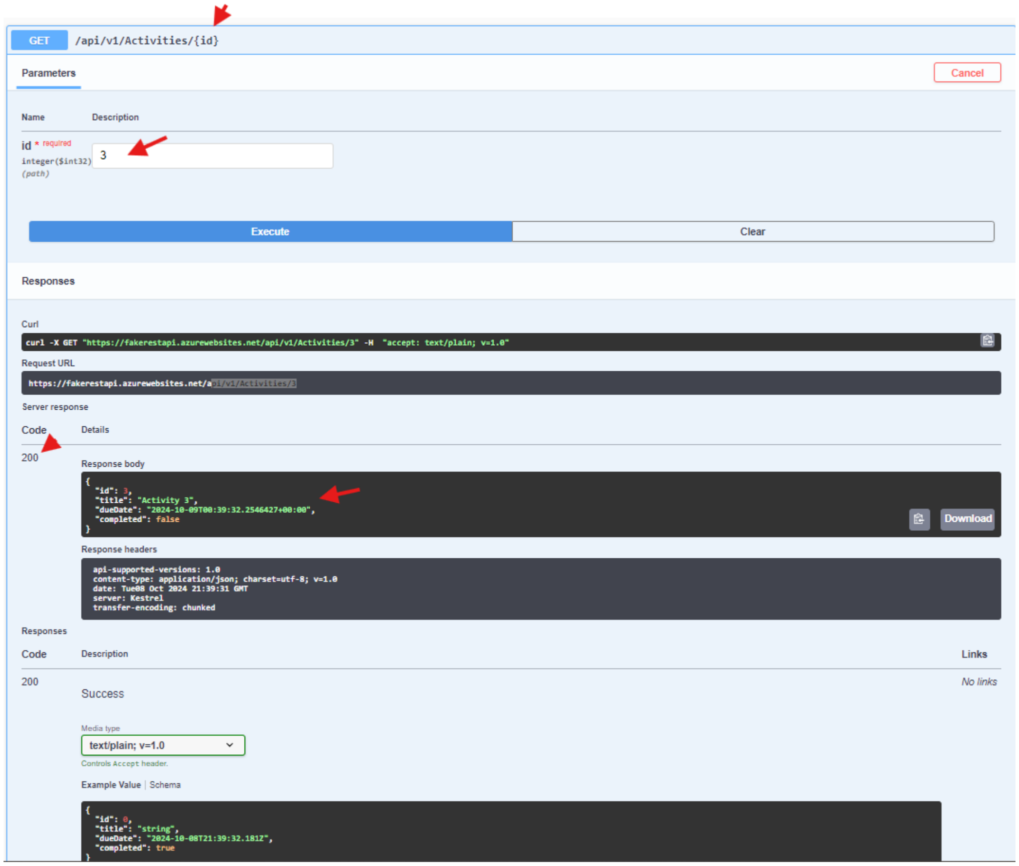
Headers

	Key	Value	Description
☑	Accept	text/plain	
	Key	Value	Description

Key	Value
Content-Type	application/json; charset=utf-8; v=1.0
Date	Tue, 08 Oct 2024 21:14:51 GMT
Server	Kestrel
Transfer-Encoding	chunked
api-supported-versions	1.0

5. Sample GET with parameter (copy the sampling url from Swagger and into Postman)

Swagger



id * required
integer($int32)
(path)
3

Curl

```
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Activities/3" -H  "accept: text/plain; v=1.0"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities/3
```

Server response

Code	Details
200	Response body

```
{
  "id": 3,
  "title": "Activity 3",
  "dueDate": "2024-10-09T00:39:32.2546427+00:00",
  "completed": false
}
```

Response headers

```
api-supported-versions: 1.0
content-type: application/json; charset=utf-8; v=1.0
date: Tue08 Oct 2024 21:39:31 GMT
server: Kestrel
transfer-encoding: chunked
```

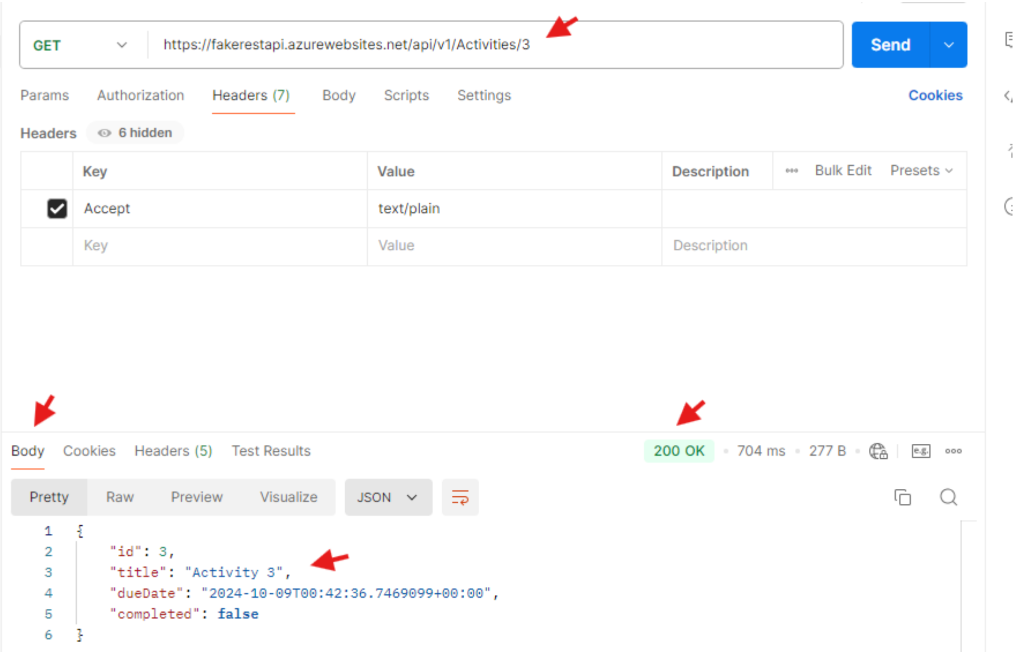
Responses

Code	Description	Links
200	Success	*No links*

Media type
text/plain; v=1.0
Controls Accept header.

```
{
  "id": 0,
  "title": "string",
  "dueDate": "2024-10-08T21:39:32.181Z",
  "completed": true
}
```

Postman

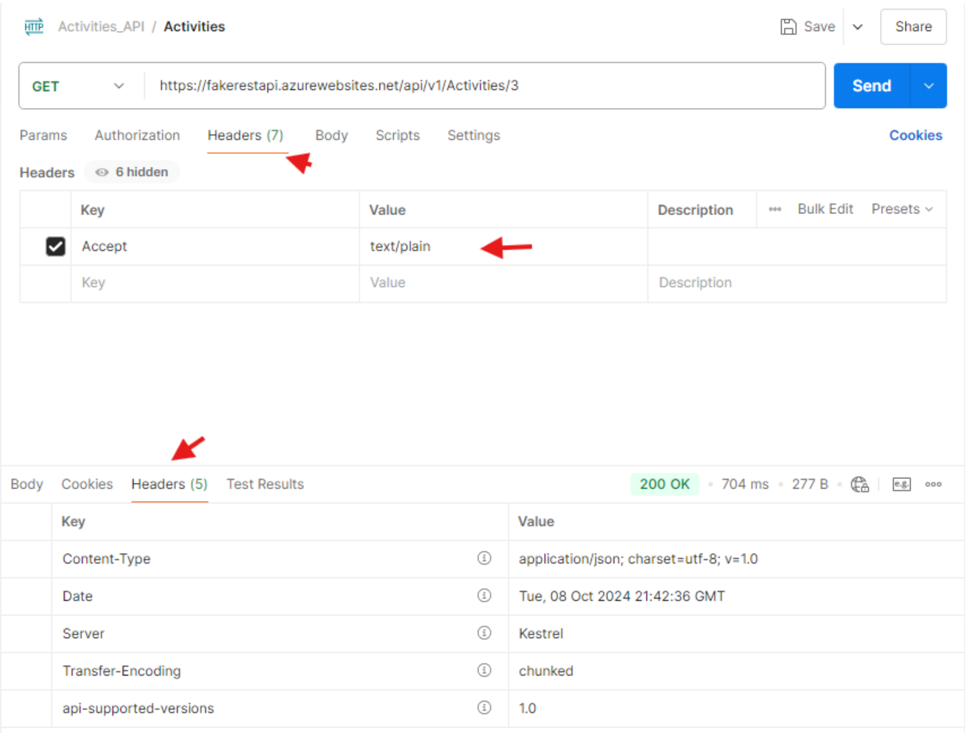


GET https://fakerestapi.azurewebsites.net/api/v1/Activities/3

	Key	Value	Description
☑	Accept	text/plain	
	Key	Value	Description

200 OK · 704 ms · 277 B

```
{
    "id": 3,
    "title": "Activity 3",
    "dueDate": "2024-10-09T00:42:36.7469099+00:00",
    "completed": false
}
```

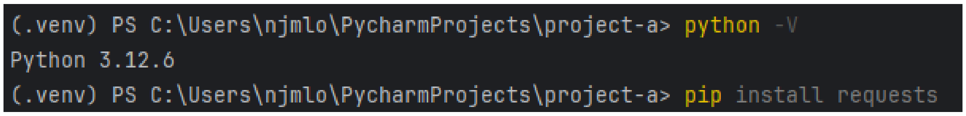


GET https://fakerestapi.azurewebsites.net/api/v1/Activities/3

	Key	Value	Description
☑	Accept	text/plain	
	Key	Value	Description

Key	Value
Content-Type	application/json; charset=utf-8; v=1.0
Date	Tue, 08 Oct 2024 21:42:36 GMT
Server	Kestrel
Transfer-Encoding	chunked
api-supported-versions	1.0

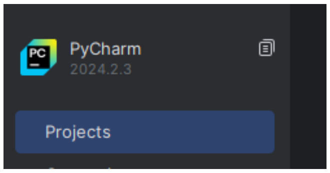
6. Determine Python version, install "requests" library



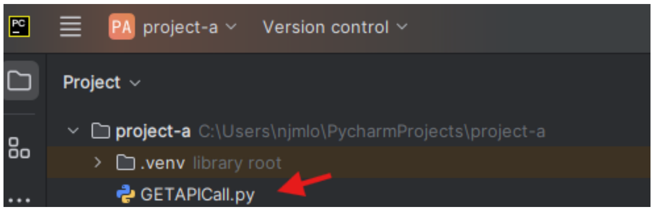
```
(.venv) PS C:\Users\njmlo\PycharmProjects\project-a> python -V
Python 3.12.6
(.venv) PS C:\Users\njmlo\PycharmProjects\project-a> pip install requests
```

7. PyCharm Community Edition (IDE version at the time of documentation)

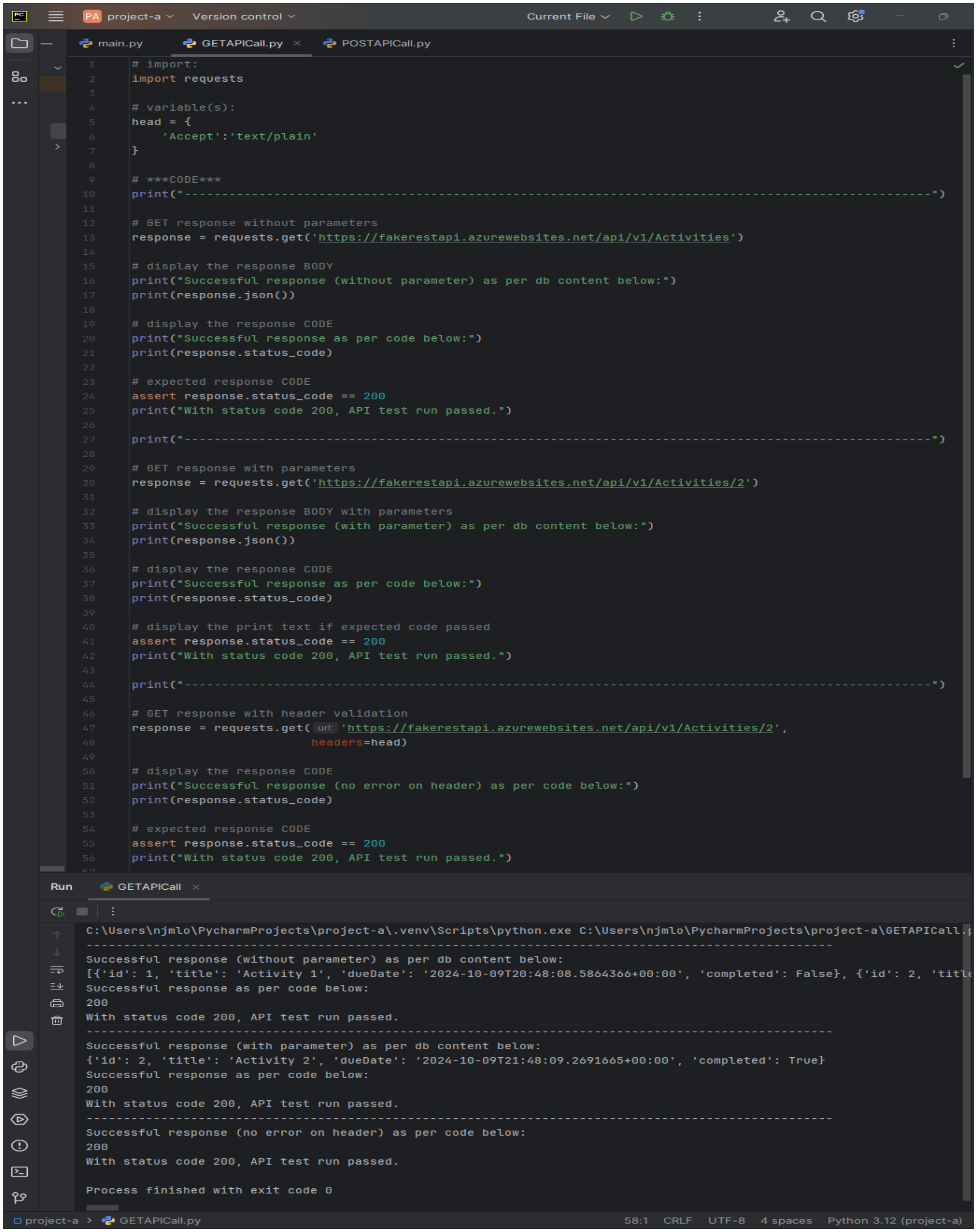
Download:
https://www.jetbrains.com/pycharm/download



8. Create a python file in PyCharm editor



9. Sample GET (in PyCharm using Python code, a sampling of with and without parameters, and header validation)

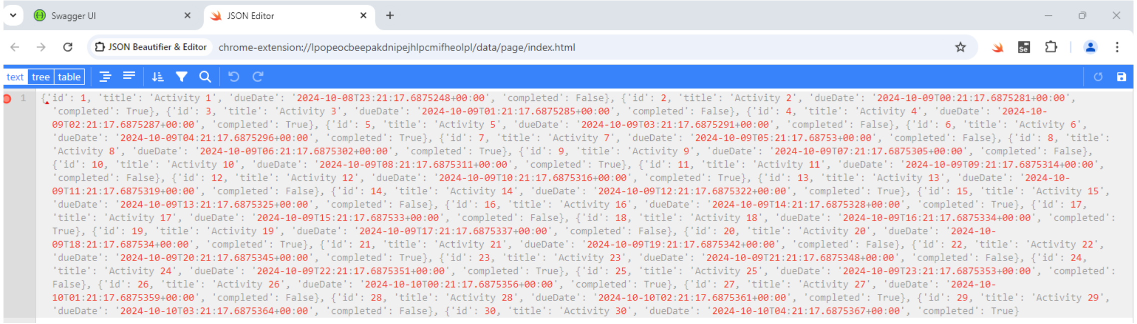


```python
# import:
import requests

# variable(s):
head = {
    'Accept':'text/plain'
}

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# GET response without parameters
response = requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities')

# display the response BODY
print("Successful response (without parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# expected response CODE
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# GET response with parameters
response = requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities/2')

# display the response BODY with parameters
print("Successful response (with parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# GET response with header validation
response = requests.get( url: 'https://fakerestapi.azurewebsites.net/api/v1/Activities/2',
                        headers=head)

# display the response CODE
print("Successful response (no error on header) as per code below:")
print(response.status_code)

# expected response CODE
assert response.status_code == 200
print("With status code 200, API test run passed.")
```

```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\GETAPICall.p
----------------------------------------------------------------------------------------------------
Successful response (without parameter) as per db content below:
[{'id': 1, 'title': 'Activity 1', 'dueDate': '2024-10-09T20:48:08.5864366+00:00', 'completed': False}, {'id': 2, 'title
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------
Successful response (with parameter) as per db content below:
{'id': 2, 'title': 'Activity 2', 'dueDate': '2024-10-09T21:48:09.2691665+00:00', 'completed': True}
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------
Successful response (no error on header) as per code below:
200
With status code 200, API test run passed.

Process finished with exit code 0
```

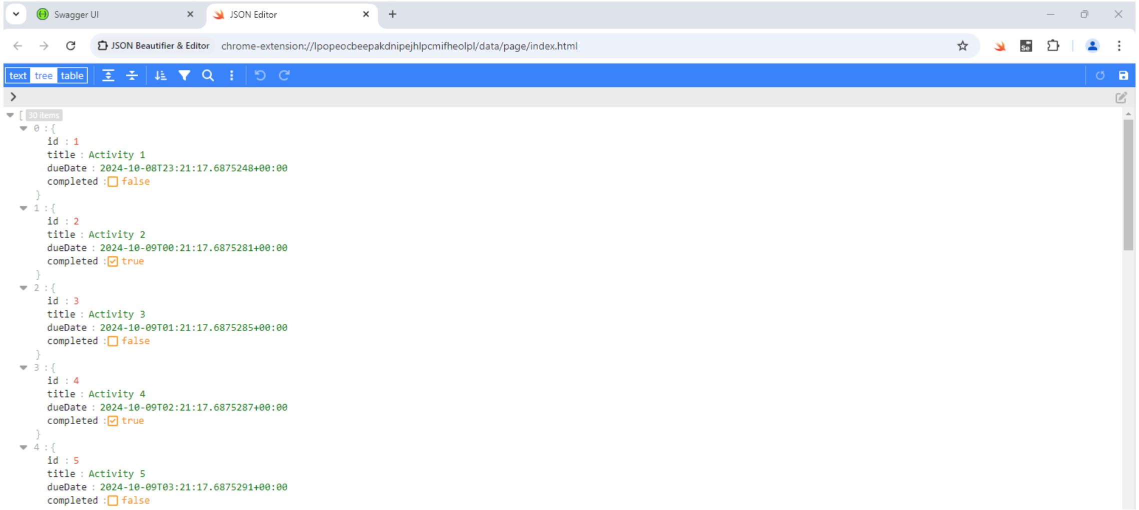
10. Using Json Beautifier (chrome add-on), display response result (copy and paste the results)

Text view


Swagger UI
JSON Editor
JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

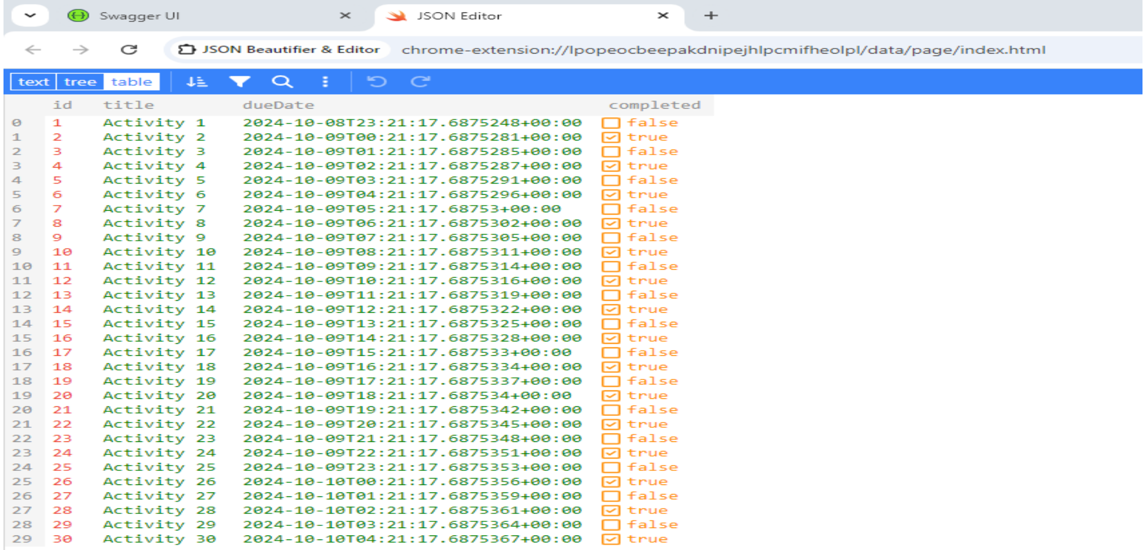
```
{'id': 1, 'title': 'Activity 1', 'dueDate': '2024-10-08T23:21:17.6875248+00:00', 'completed': False}, {'id': 2, 'title': 'Activity 2', 'dueDate': '2024-10-09T00:21:17.6875281+00:00',
'completed': True}, {'id': 3, 'title': 'Activity 3', 'dueDate': '2024-10-09T01:21:17.6875285+00:00', 'completed': False}, {'id': 4, 'title': 'Activity 4', 'dueDate': '2024-10-
09T02:21:17.6875287+00:00', 'completed': True}, {'id': 5, 'title': 'Activity 5', 'dueDate': '2024-10-09T03:21:17.6875291+00:00', 'completed': False}, {'id': 6, 'title': 'Activity 6',
'dueDate': '2024-10-09T04:21:17.6875296+00:00', 'completed': True}, {'id': 7, 'title': 'Activity 7', 'dueDate': '2024-10-09T05:21:17.68753+00:00', 'completed': False}, {'id': 8, 'title':
'Activity 8', 'dueDate': '2024-10-09T06:21:17.6875302+00:00', 'completed': True}, {'id': 9, 'title': 'Activity 9', 'dueDate': '2024-10-09T07:21:17.6875305+00:00', 'completed': False},
{'id': 10, 'title': 'Activity 10', 'dueDate': '2024-10-09T08:21:17.6875311+00:00', 'completed': True}, {'id': 11, 'title': 'Activity 11', 'dueDate': '2024-10-09T09:21:17.6875314+00:00',
'completed': False}, {'id': 12, 'title': 'Activity 12', 'dueDate': '2024-10-09T10:21:17.6875316+00:00', 'completed': True}, {'id': 13, 'title': 'Activity 13', 'dueDate': '2024-10-
09T11:21:17.6875319+00:00', 'completed': False}, {'id': 14, 'title': 'Activity 14', 'dueDate': '2024-10-09T12:21:17.6875322+00:00', 'completed': True}, {'id': 15, 'title': 'Activity 15',
'dueDate': '2024-10-09T13:21:17.6875325+00:00', 'completed': False}, {'id': 16, 'title': 'Activity 16', 'dueDate': '2024-10-09T14:21:17.6875328+00:00', 'completed': True}, {'id': 17,
'title': 'Activity 17', 'dueDate': '2024-10-09T15:21:17.687533+00:00', 'completed': False}, {'id': 18, 'title': 'Activity 18', 'dueDate': '2024-10-09T16:21:17.6875334+00:00', 'completed':
True}, {'id': 19, 'title': 'Activity 19', 'dueDate': '2024-10-09T17:21:17.6875337+00:00', 'completed': False}, {'id': 20, 'title': 'Activity 20', 'dueDate': '2024-10-
09T18:21:17.687534+00:00', 'completed': True}, {'id': 21, 'title': 'Activity 21', 'dueDate': '2024-10-09T19:21:17.6875342+00:00', 'completed': False}, {'id': 22, 'title': 'Activity 22',
'dueDate': '2024-10-09T20:21:17.6875345+00:00', 'completed': True}, {'id': 23, 'title': 'Activity 23', 'dueDate': '2024-10-09T21:21:17.6875348+00:00', 'completed': False}, {'id': 24,
'title': 'Activity 24', 'dueDate': '2024-10-09T22:21:17.6875351+00:00', 'completed': True}, {'id': 25, 'title': 'Activity 25', 'dueDate': '2024-10-09T23:21:17.6875353+00:00', 'completed':
False}, {'id': 26, 'title': 'Activity 26', 'dueDate': '2024-10-10T00:21:17.6875356+00:00', 'completed': True}, {'id': 27, 'title': 'Activity 27', 'dueDate': '2024-10-
10T01:21:17.6875359+00:00', 'completed': False}, {'id': 28, 'title': 'Activity 28', 'dueDate': '2024-10-10T02:21:17.6875361+00:00', 'completed': True}, {'id': 29, 'title': 'Activity 29',
'dueDate': '2024-10-10T03:21:17.6875364+00:00', 'completed': False}, {'id': 30, 'title': 'Activity 30', 'dueDate': '2024-10-10T04:21:17.6875367+00:00', 'completed': True}
```

Tree view


Swagger UI
JSON Editor
JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

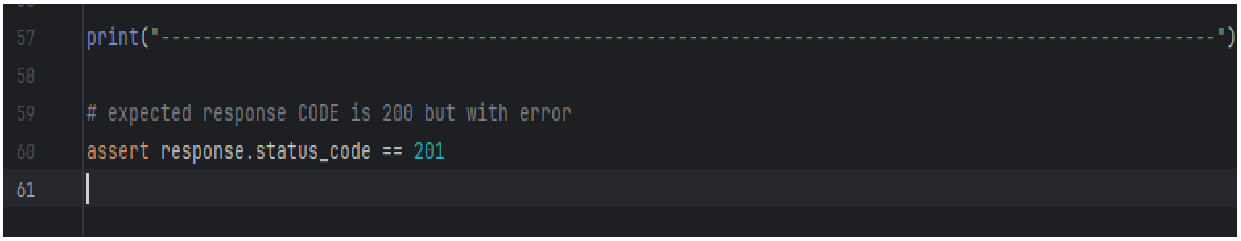
```
▼ [ 30 items
  ▼ 0 : {
      id : 1
      title : Activity 1
      dueDate : 2024-10-08T23:21:17.6875248+00:00
      completed : □ false
    }
  ▼ 1 : {
      id : 2
      title : Activity 2
      dueDate : 2024-10-09T00:21:17.6875281+00:00
      completed : ☑ true
    }
  ▼ 2 : {
      id : 3
      title : Activity 3
      dueDate : 2024-10-09T01:21:17.6875285+00:00
      completed : □ false
    }
  ▼ 3 : {
      id : 4
      title : Activity 4
      dueDate : 2024-10-09T02:21:17.6875287+00:00
      completed : ☑ true
    }
  ▼ 4 : {
      id : 5
      title : Activity 5
      dueDate : 2024-10-09T03:21:17.6875291+00:00
      completed : □ false
```

Table view

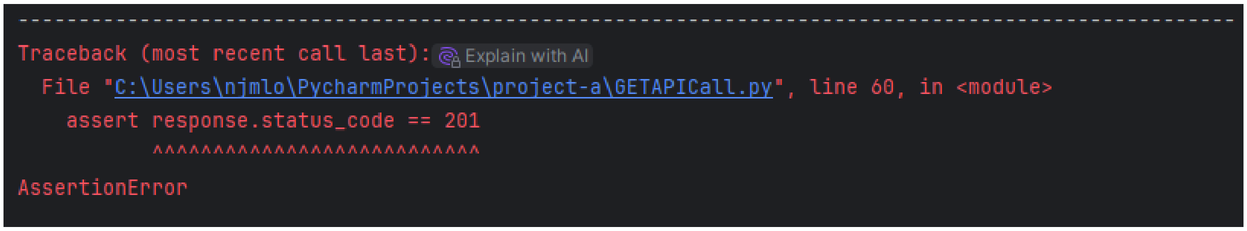

Swagger UI
JSON Editor
JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

	id	title	dueDate	completed
0	1	Activity 1	2024-10-08T23:21:17.6875248+00:00	□ false
1	2	Activity 2	2024-10-09T00:21:17.6875281+00:00	☑ true
2	3	Activity 3	2024-10-09T01:21:17.6875285+00:00	□ false
3	4	Activity 4	2024-10-09T02:21:17.6875287+00:00	☑ true
4	5	Activity 5	2024-10-09T03:21:17.6875291+00:00	□ false
5	6	Activity 6	2024-10-09T04:21:17.6875296+00:00	☑ true
6	7	Activity 7	2024-10-09T05:21:17.68753+00:00	□ false
7	8	Activity 8	2024-10-09T06:21:17.6875302+00:00	☑ true
8	9	Activity 9	2024-10-09T07:21:17.6875305+00:00	□ false
9	10	Activity 10	2024-10-09T08:21:17.6875311+00:00	☑ true
10	11	Activity 11	2024-10-09T09:21:17.6875314+00:00	□ false
11	12	Activity 12	2024-10-09T10:21:17.6875316+00:00	☑ true
12	13	Activity 13	2024-10-09T11:21:17.6875319+00:00	□ false
13	14	Activity 14	2024-10-09T12:21:17.6875322+00:00	☑ true
14	15	Activity 15	2024-10-09T13:21:17.6875325+00:00	□ false
15	16	Activity 16	2024-10-09T14:21:17.6875328+00:00	☑ true
16	17	Activity 17	2024-10-09T15:21:17.687533+00:00	□ false
17	18	Activity 18	2024-10-09T16:21:17.6875334+00:00	☑ true
18	19	Activity 19	2024-10-09T17:21:17.6875337+00:00	□ false
19	20	Activity 20	2024-10-09T18:21:17.687534+00:00	☑ true
20	21	Activity 21	2024-10-09T19:21:17.6875342+00:00	□ false
21	22	Activity 22	2024-10-09T20:21:17.6875345+00:00	☑ true
22	23	Activity 23	2024-10-09T21:21:17.6875348+00:00	□ false
23	24	Activity 24	2024-10-09T22:21:17.6875351+00:00	☑ true
24	25	Activity 25	2024-10-09T23:21:17.6875353+00:00	□ false
25	26	Activity 26	2024-10-10T00:21:17.6875356+00:00	☑ true
26	27	Activity 27	2024-10-10T01:21:17.6875359+00:00	□ false
27	28	Activity 28	2024-10-10T02:21:17.6875361+00:00	☑ true
28	29	Activity 29	2024-10-10T03:21:17.6875364+00:00	□ false
29	30	Activity 30	2024-10-10T04:21:17.6875367+00:00	☑ true

11. Sample GET (Python in PyCharm, a sampling of with status error)



```
57    print("---------------------------------------------------------------------------------------------------------")
58
59    # expected response CODE is 200 but with error
60    assert response.status_code == 201
61
```

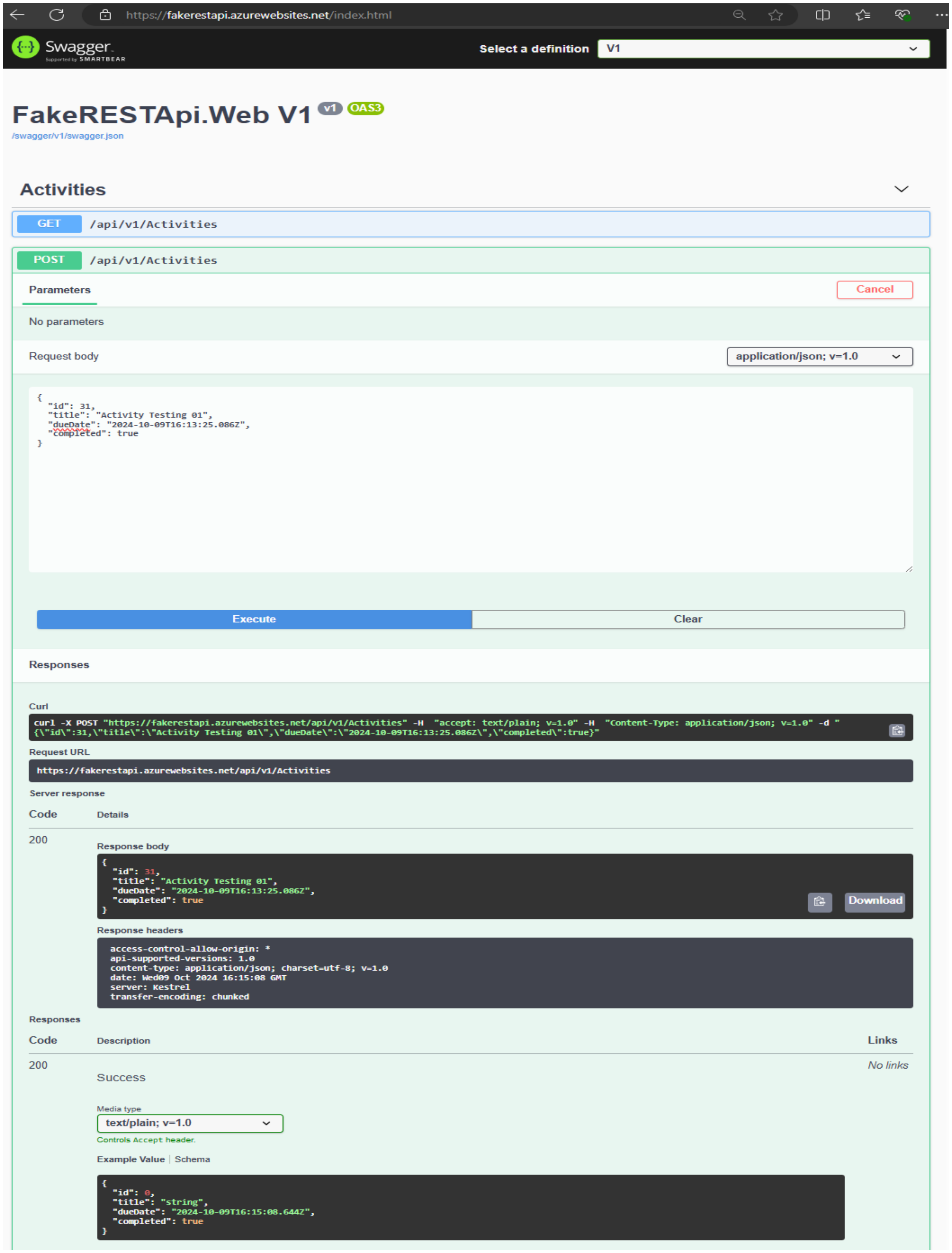


```
--------------------------------------------------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\njmlo\PycharmProjects\project-a\GETAPICall.py", line 60, in <module>
    assert response.status_code == 201
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
```

12. Here is the code in text

```
# import:
import requests

# variable(s):
head = {
    'Accept':'text/plain'
}

# ***CODE***
print("----------------------------------------------------------------------
------------------------------")

# GET response without parameters
response =
requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities')

# display the response BODY
print("Successful response (without parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# expected response CODE
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------
------------------------------")

# GET response with parameters
response =
requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities/2')

# display the response BODY with parameters
print("Successful response (with parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("-----------------------------------------------------------------------
------------------------------")

# GET response with header validation
response =
requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities/2',
                        headers=head)

# display the response CODE
print("Successful response (no error on header) as per code below:")
print(response.status_code)

# expected response CODE
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("-----------------------------------------------------------------------
------------------------------")

# expected response CODE is 200 but with error
assert response.status_code == 201
```

13. Executing a POST method



No parameters

Request body: application/json; v=1.0

```
{
  "id": 31,
  "title": "Activity Testing 01",
  "dueDate": "2024-10-09T16:13:25.086Z",
  "completed": true
}
```

Responses

Curl

```
curl -X POST "https://fakerestapi.azurewebsites.net/api/v1/Activities" -H  "accept: text/plain; v=1.0" -H  "Content-Type: application/json; v=1.0" -d "
{\"id\":31,\"title\":\"Activity Testing 01\",\"dueDate\":\"2024-10-09T16:13:25.086Z\",\"completed\":true}"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities
```

Server response

Code: 200

Details

Response body

```
{
  "id": 31,
  "title": "Activity Testing 01",
  "dueDate": "2024-10-09T16:13:25.086Z",
  "completed": true
}
```

Response headers

```
 access-control-allow-origin: *
 api-supported-versions: 1.0
 content-type: application/json; charset=utf-8; v=1.0
 date: Wed09 Oct 2024 16:15:08 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

Code: 200

Description: Success

Media type: text/plain; v=1.0

Controls Accept header.

Example Value | Schema

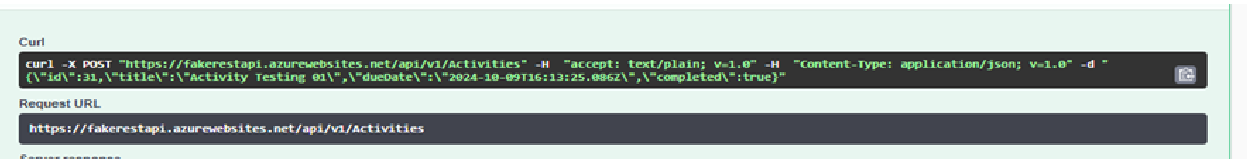
```
{
  "id": 0,
  "title": "string",
  "dueDate": "2024-10-09T16:15:08.644Z",
  "completed": true
}
```

14. Sample POST (copy the sampling url from Swagger and into Postman)

Validate and compare the following response results:

- **Status**
- **Body**
- **Header**
- **Time**

Swagger



Curl

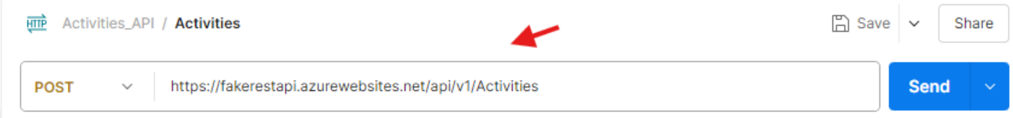
```
curl -X POST "https://fakerestapi.azurewebsites.net/api/v1/Activities" -H  "accept: text/plain; v=1.0" -H  "Content-Type: application/json; v=1.0" -d "
{\"id\":31,\"title\":\"Activity Testing 01\",\"dueDate\":\"2024-10-09T16:13:25.086Z\",\"completed\":true}"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities
```

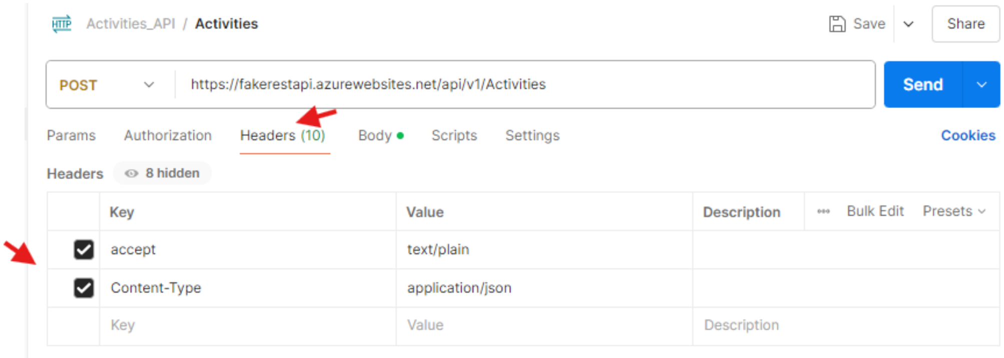
Postman

URL



POST https://fakerestapi.azurewebsites.net/api/v1/Activities

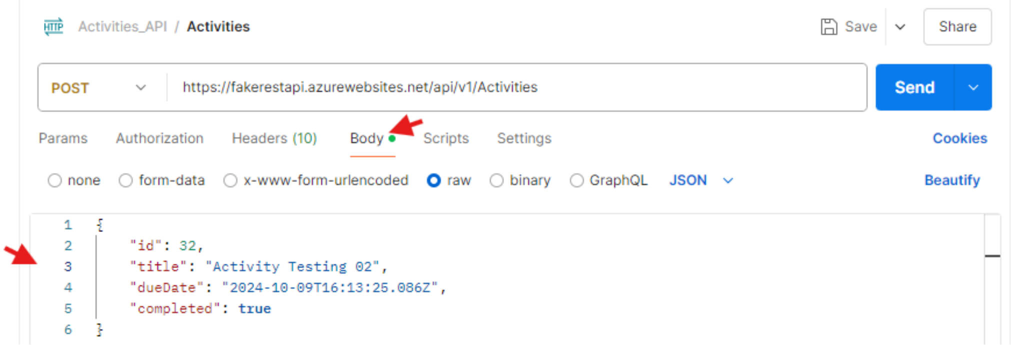
Header



POST https://fakerestapi.azurewebsites.net/api/v1/Activities

	Key	Value	Description
☑	accept	text/plain	
☑	Content-Type	application/json	
	Key	Value	Description

Body



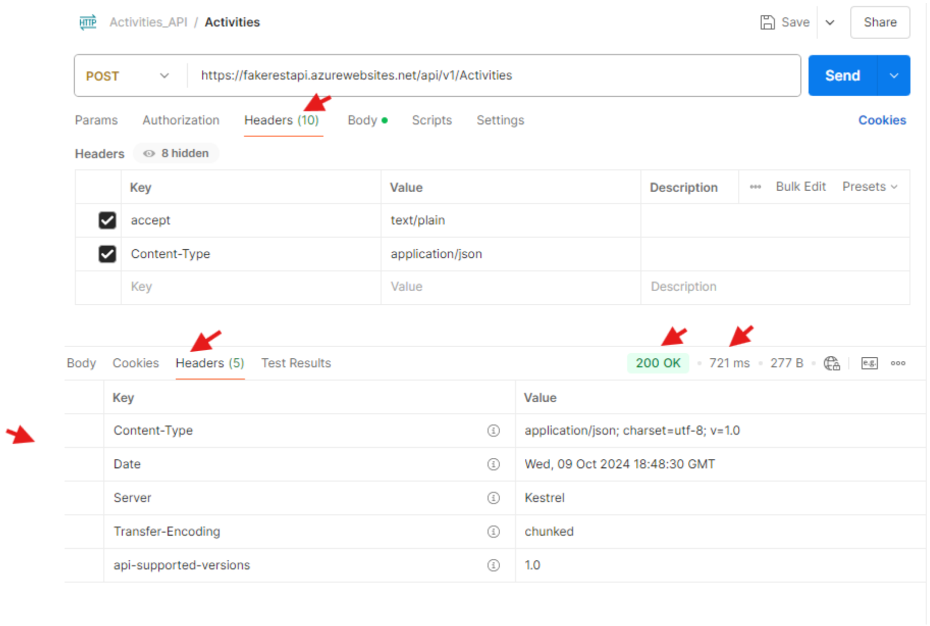
POST https://fakerestapi.azurewebsites.net/api/v1/Activities

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id": 32,
    "title": "Activity Testing 02",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": true
}
```

After Send…

Header



Activities_API / Activities
Save
Share
Send
Params Authorization Headers (10) Body Scripts Settings
Cookies
8 hidden
Bulk Edit Presets
Body Cookies Headers (5) Test Results
200 OK · 721 ms · 277 B

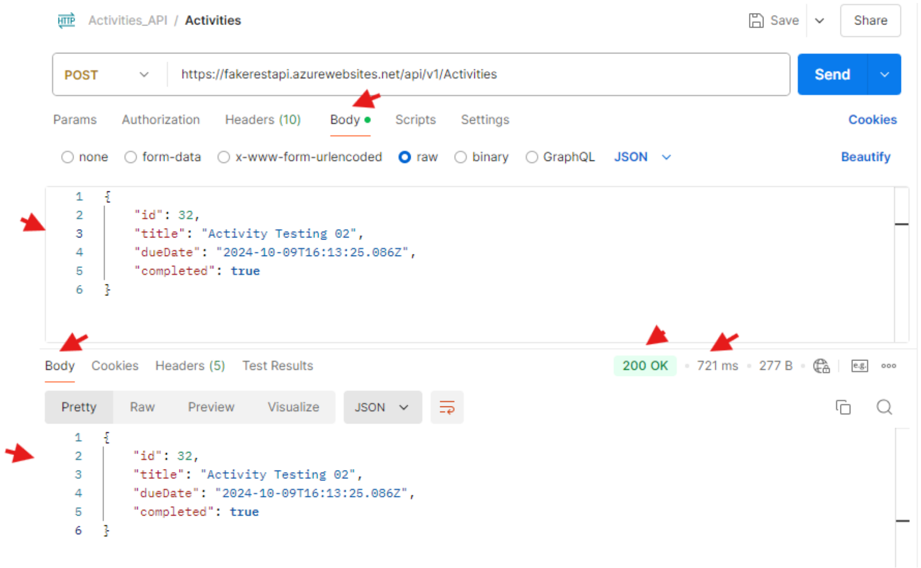
POST https://fakerestapi.azurewebsites.net/api/v1/Activities

Headers

	Key	Value	Description
☑	accept	text/plain	
☑	Content-Type	application/json	
	Key	Value	Description

Key	Value
Content-Type	application/json; charset=utf-8; v=1.0
Date	Wed, 09 Oct 2024 18:48:30 GMT
Server	Kestrel
Transfer-Encoding	chunked
api-supported-versions	1.0

Body



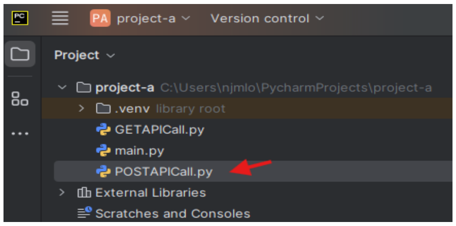
Activities_API / Activities
Save
Share
Send
Params Authorization Headers (10) Body Scripts Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Beautify
Body Cookies Headers (5) Test Results
200 OK · 721 ms · 277 B
Pretty Raw Preview Visualize JSON

POST https://fakerestapi.azurewebsites.net/api/v1/Activities

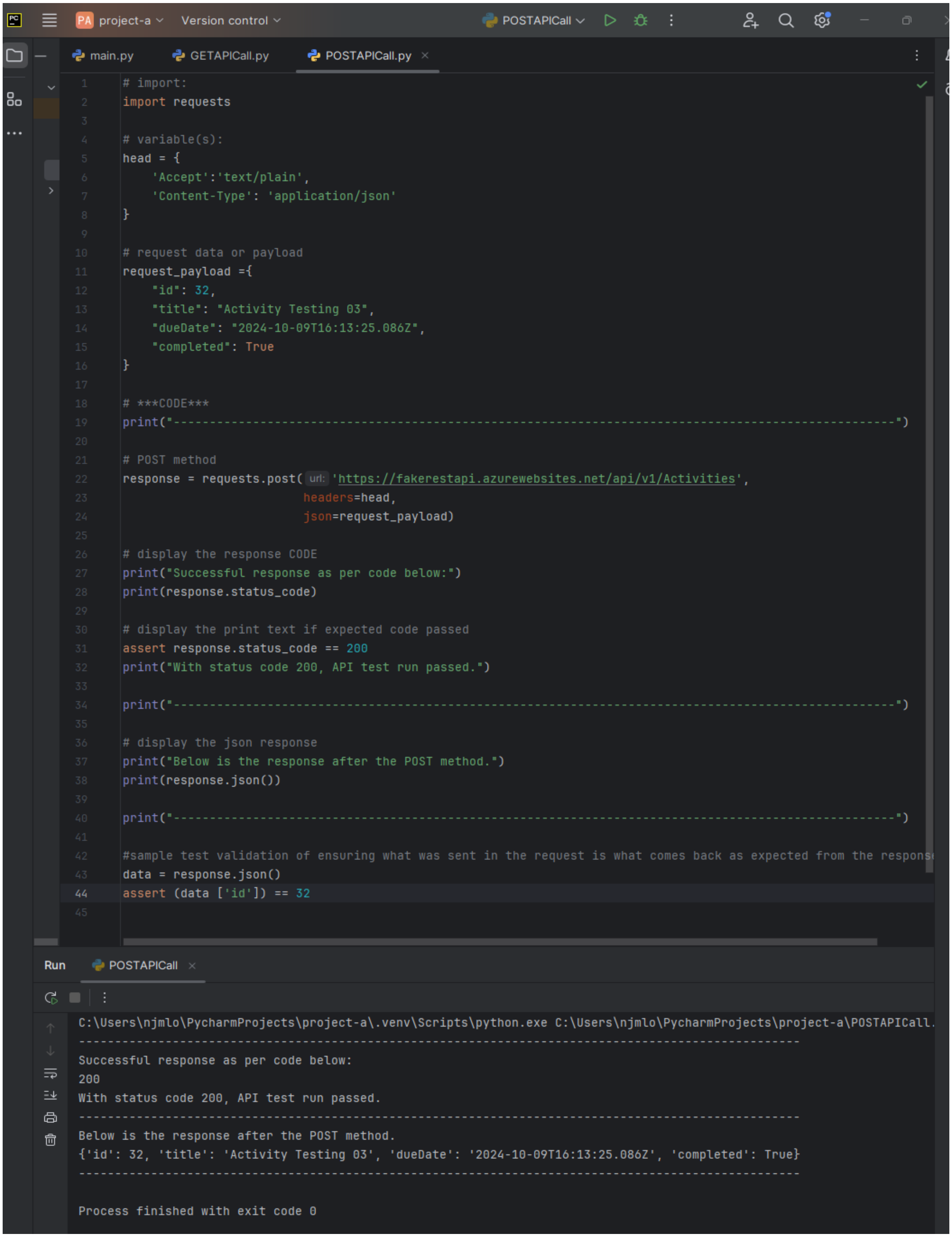
```json
{
    "id": 32,
    "title": "Activity Testing 02",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": true
}
```

```json
{
    "id": 32,
    "title": "Activity Testing 02",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": true
}
```

15. Create a python file in PyCharm editor



16. Sample POST (in PyCharm using Python code)

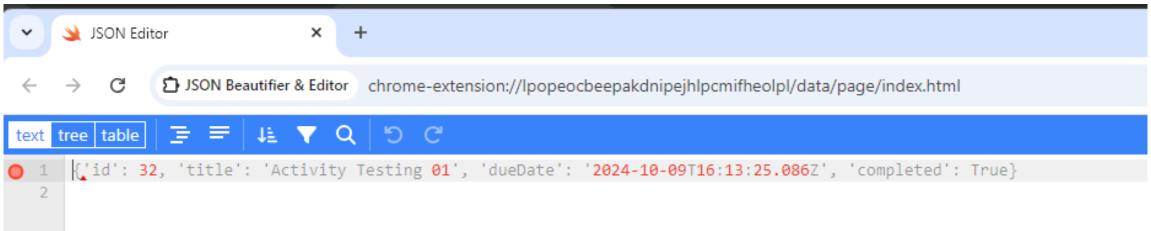


```python
# import:
import requests

# variable(s):
head = {
    'Accept':'text/plain',
    'Content-Type': 'application/json'
}

# request data or payload
request_payload ={
    "id": 32,
    "title": "Activity Testing 03",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": True
}

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# POST method
response = requests.post('https://fakerestapi.azurewebsites.net/api/v1/Activities',
                         headers=head,
                         json=request_payload)

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# display the json response
print("Below is the response after the POST method.")
print(response.json())

print("----------------------------------------------------------------------------------------------------")

#sample test validation of ensuring what was sent in the request is what comes back as expected from the respons
data = response.json()
assert (data ['id']) == 32
```

```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\POSTAPICall.
----------------------------------------------------------------------------------------------------
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------
Below is the response after the POST method.
{'id': 32, 'title': 'Activity Testing 03', 'dueDate': '2024-10-09T16:13:25.086Z', 'completed': True}
----------------------------------------------------------------------------------------------------

Process finished with exit code 0
```

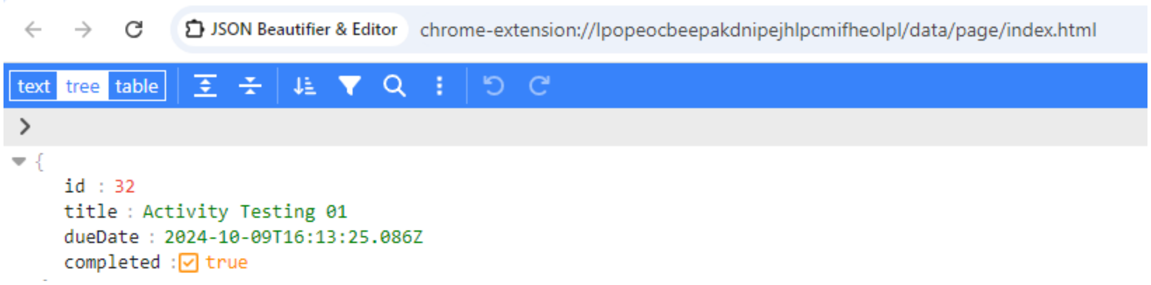
17. Using Json Beautifier to display response result nicely (copy and paste the result)

Text view


JSON Editor
JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

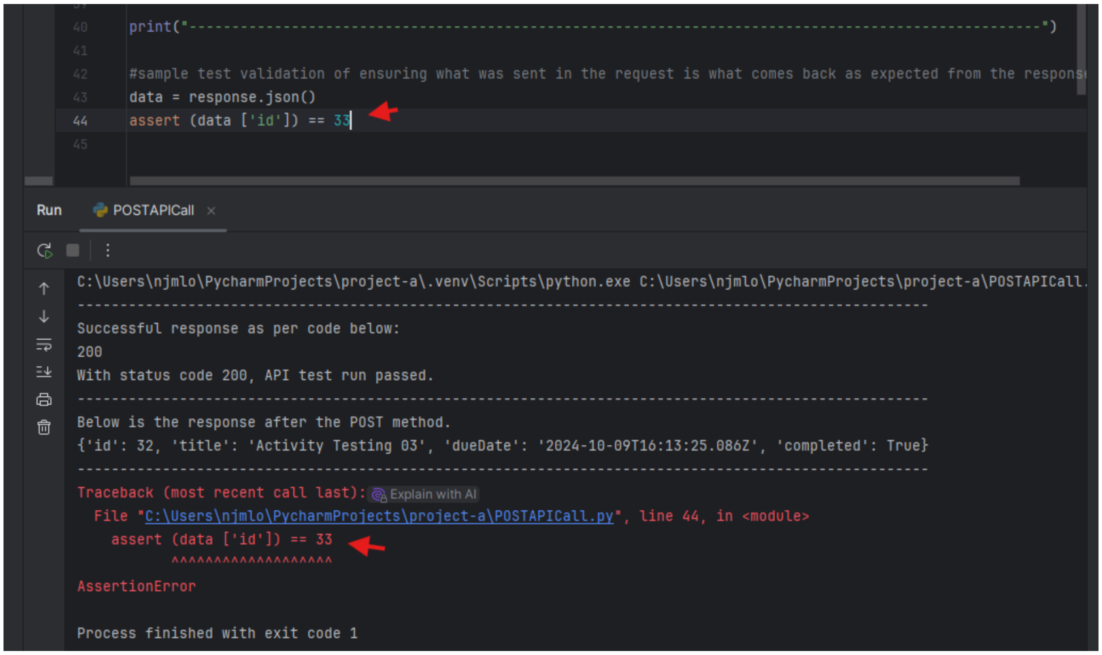
```
{'id': 32, 'title': 'Activity Testing 01', 'dueDate': '2024-10-09T16:13:25.086Z', 'completed': True}
```

Tree view


JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

```
{
  id : 32
  title : Activity Testing 01
  dueDate : 2024-10-09T16:13:25.086Z
  completed : ☑ true
```

18. Sample POST (Python in PyCharm, a sampling of with validation error)



```
print("-----------------------------------------------------------------------------------------")

#sample test validation of ensuring what was sent in the request is what comes back as expected from the respons
data = response.json()
assert (data ['id']) == 33
```

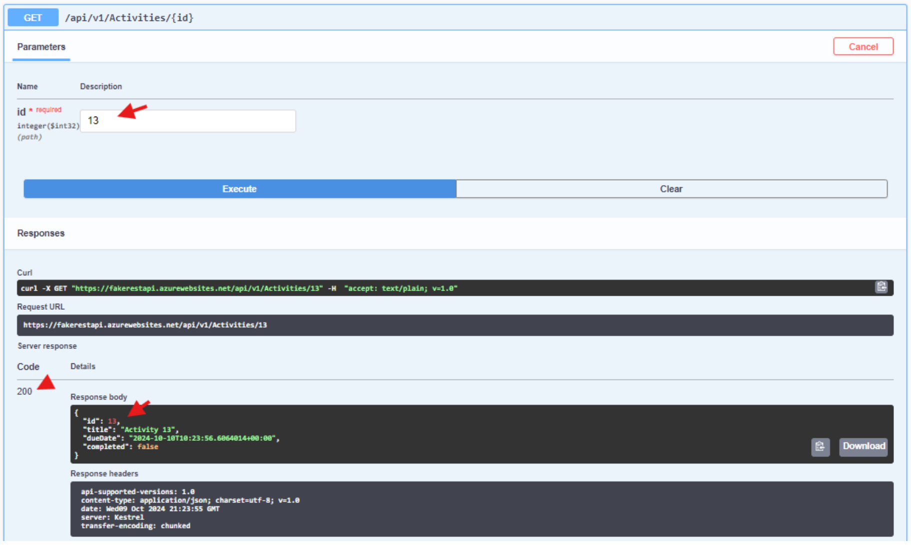
```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\POSTAPICall.
-----------------------------------------------------------------------------------------
Successful response as per code below:
200
With status code 200, API test run passed.
-----------------------------------------------------------------------------------------
Below is the response after the POST method.
{'id': 32, 'title': 'Activity Testing 03', 'dueDate': '2024-10-09T16:13:25.086Z', 'completed': True}
-----------------------------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\njmlo\PycharmProjects\project-a\POSTAPICall.py", line 44, in <module>
    assert (data ['id']) == 33
           ^^^^^^^^^^^^^^^^^^^
AssertionError

Process finished with exit code 1
```

19. Here is the code in text

```
# import:
import requests

# variable(s):
head = {
    'Accept':'text/plain',
    'Content-Type': 'application/json'
}

# request data or payload
request_payload ={
    "id": 32,
    "title": "Activity Testing 03",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": True
}

# ***CODE***
print("----------------------------------------------------------------------
------------------------------")

# POST method
response =
requests.post('https://fakerestapi.azurewebsites.net/api/v1/Activities',
                         headers=head,
                         json=request_payload)

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------
------------------------------")

# display the json response
print("Below is the response after the POST method.")
print(response.json())

print("----------------------------------------------------------------------
------------------------------")

#sample test validation of ensuring what was sent in the request is what
comes back as expected from the response
data = response.json()
assert (data ['id']) == 32
```

20. Executing a PUT method

In SwaggerUI, before executing the method



id * required
integer($int32)
(path)
13

Curl

```
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Activities/13" -H  "accept: text/plain; v=1.0"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities/13
```

Server response

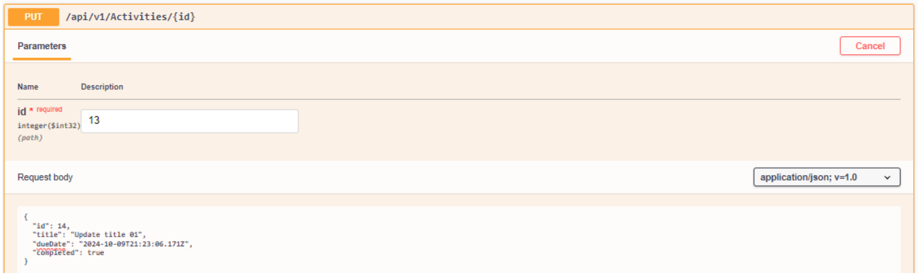
Code: 200

Response body

```
{
  "id": 13,
  "title": "Activity 13",
  "dueDate": "2024-10-10T10:23:56.6064014+00:00",
  "completed": false
}
```

Response headers

```
api-supported-versions: 1.0
content-type: application/json; charset=utf-8; v=1.0
date: Wed09 Oct 2024 21:23:55 GMT
server: Kestrel
transfer-encoding: chunked
```

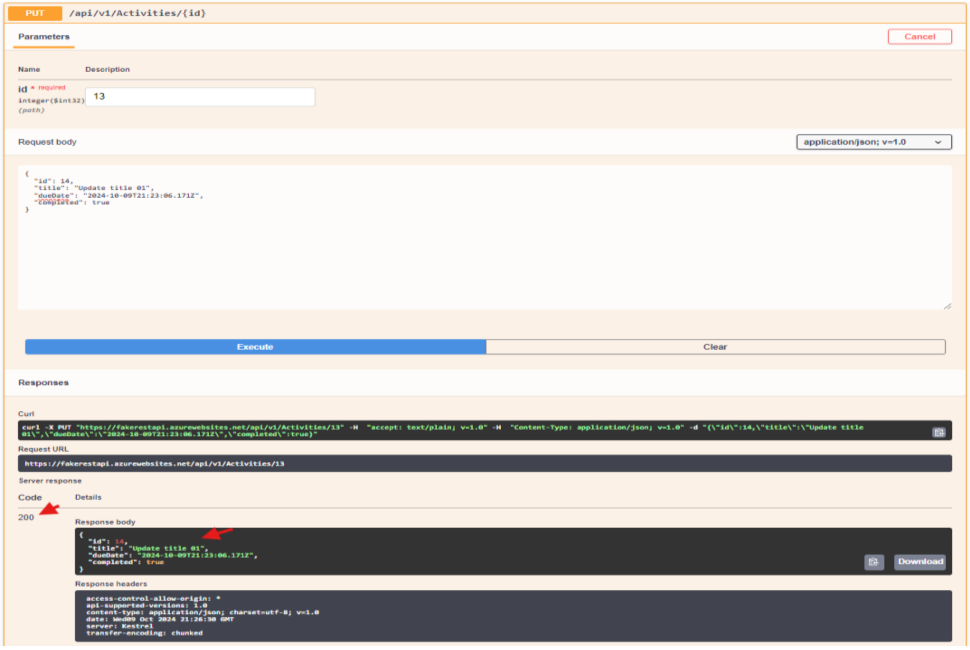


id * required
integer($int32)
(path)
13

Request body: application/json; v=1.0

```
{
  "id": 14,
  "title": "Update title 01",
  "dueDate": "2024-10-09T21:23:06.171Z",
  "completed": true
}
```

In SwaggerUI, after executing the method



id * required
integer($int32)
(path)
13

Request body: application/json; v=1.0

```
{
  "id": 14,
  "title": "Update title 01",
  "dueDate": "2024-10-09T21:23:06.171Z",
  "completed": true
}
```

Curl

```
curl -X PUT "https://fakerestapi.azurewebsites.net/api/v1/Activities/13" -H  "accept: text/plain; v=1.0" -H  "Content-Type: application/json; v=1.0" -d "{\"id\":14,\"title\":\"Update title
01\",\"dueDate\":\"2024-10-09T21:23:06.171Z\",\"completed\":true}"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities/13
```

Server response

Code: 200

Response body

```
{
  "id": 14,
  "title": "Update title 01",
  "dueDate": "2024-10-09T21:23:06.171Z",
  "completed": true
}
```

Response headers

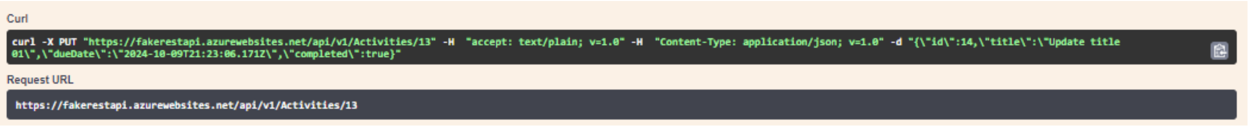
```
access-control-allow-origin: *
api-supported-versions: 1.0
content-type: application/json; charset=utf-8; v=1.0
date: Wed09 Oct 2024 21:26:30 GMT
server: Kestrel
transfer-encoding: chunked
```

21. Sample PUT (copy the sampling url from Swagger and into Postman)

Validate and compare the following response results:

- **Status**
- **Body**
- **Header**
- **Time**

Swagger



Curl

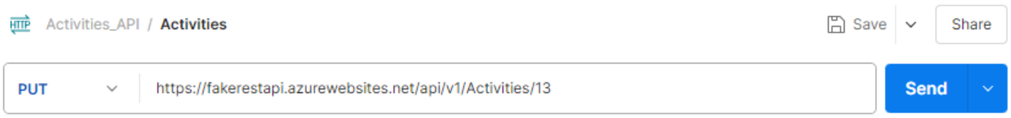
```
curl -X PUT "https://fakerestapi.azurewebsites.net/api/v1/Activities/13" -H  "accept: text/plain; v=1.0" -H  "Content-Type: application/json; v=1.0" -d "{\"id\":14,\"title\":\"Update title 01\",\"dueDate\":\"2024-10-09T21:23:06.171Z\",\"completed\":true}"
```

Request URL

```
https://fakerestapi.azurewebsites.net/api/v1/Activities/13
```

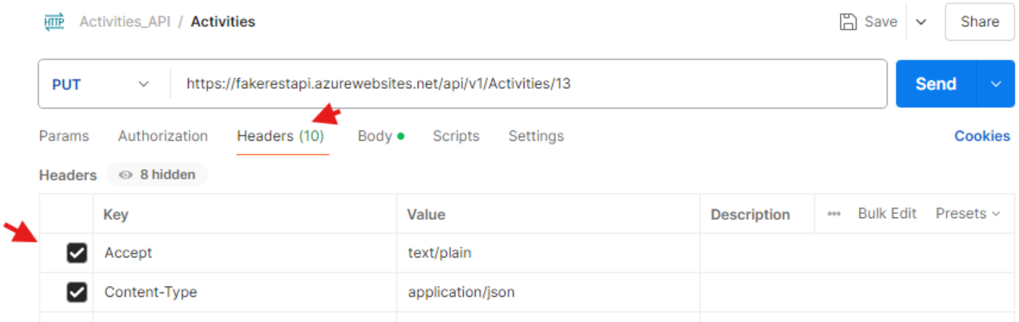
Postman

URL



PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

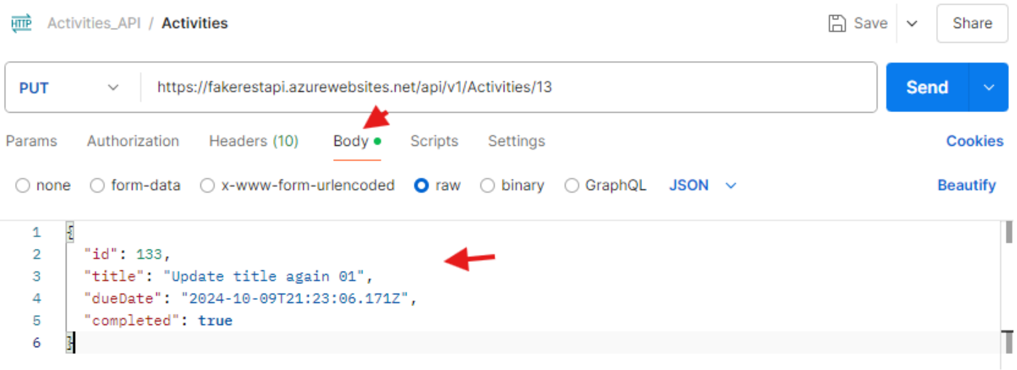
Header



PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

	Key	Value	Description
☑	Accept	text/plain	
☑	Content-Type	application/json	

Body



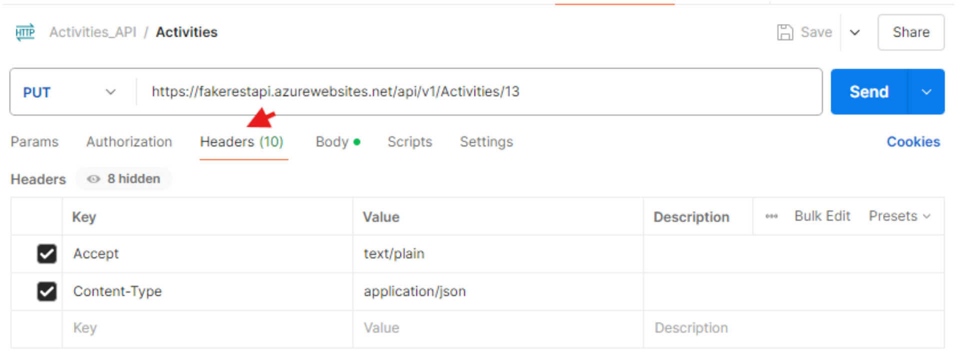
PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
  "id": 133,
  "title": "Update title again 01",
  "dueDate": "2024-10-09T21:23:06.171Z",
  "completed": true
}
```

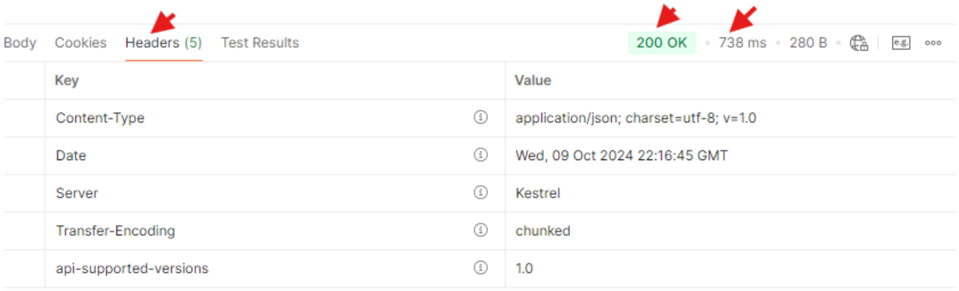
After Send…

Header



PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

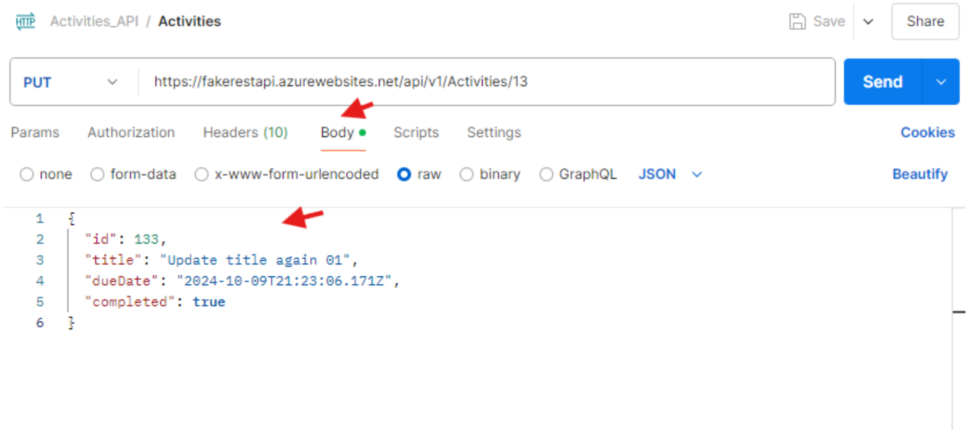
	Key	Value	Description
☑	Accept	text/plain	
☑	Content-Type	application/json	
	Key	Value	Description



200 OK · 738 ms · 280 B

Key	Value
Content-Type	application/json; charset=utf-8; v=1.0
Date	Wed, 09 Oct 2024 22:16:45 GMT
Server	Kestrel
Transfer-Encoding	chunked
api-supported-versions	1.0

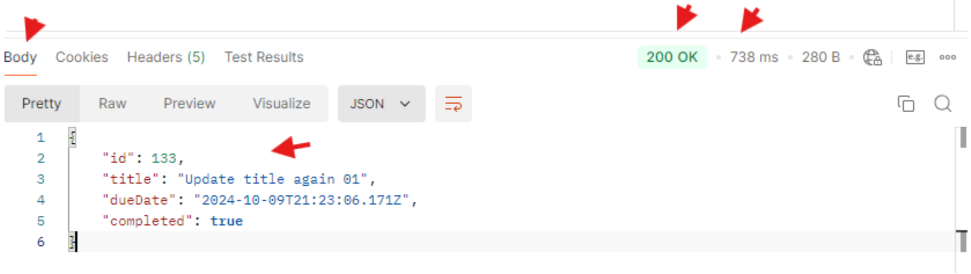
Body



PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

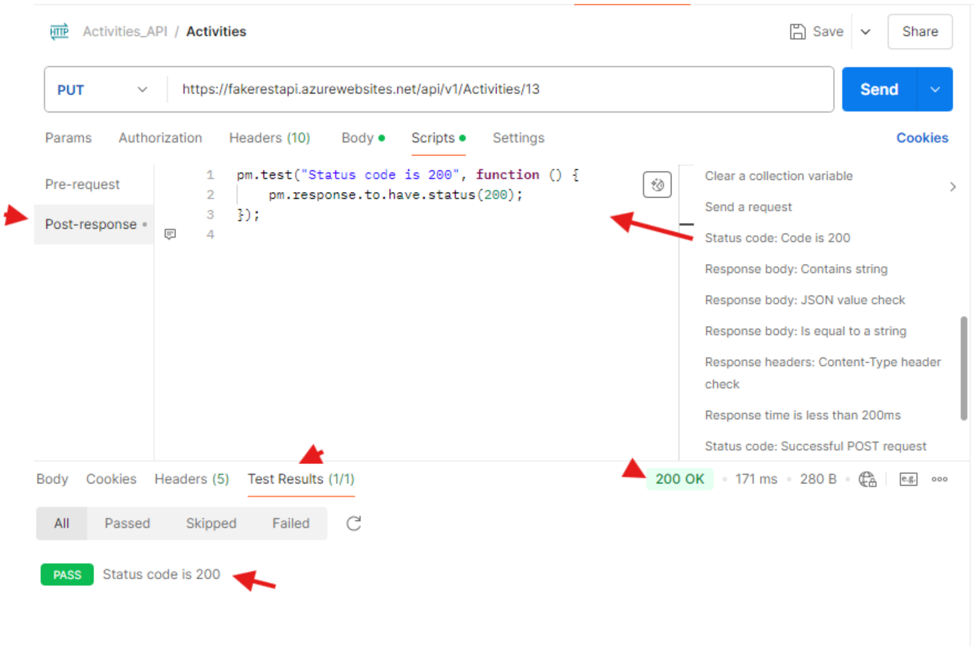
```
{
  "id": 133,
  "title": "Update title again 01",
  "dueDate": "2024-10-09T21:23:06.171Z",
  "completed": true
}
```



200 OK · 738 ms · 280 B

```
{
    "id": 133,
    "title": "Update title again 01",
    "dueDate": "2024-10-09T21:23:06.171Z",
    "completed": true
}
```

22. Sample Test code (using Snippets in Postman) and validating its Test Results (Post-response)



Activities_API / Activities

PUT https://fakerestapi.azurewebsites.net/api/v1/Activities/13

Post-response

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
```

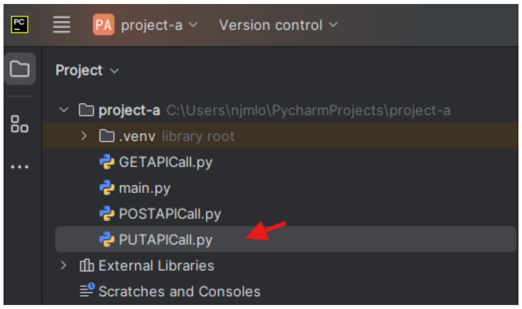
Test Results (1/1)

200 OK • 171 ms • 280 B

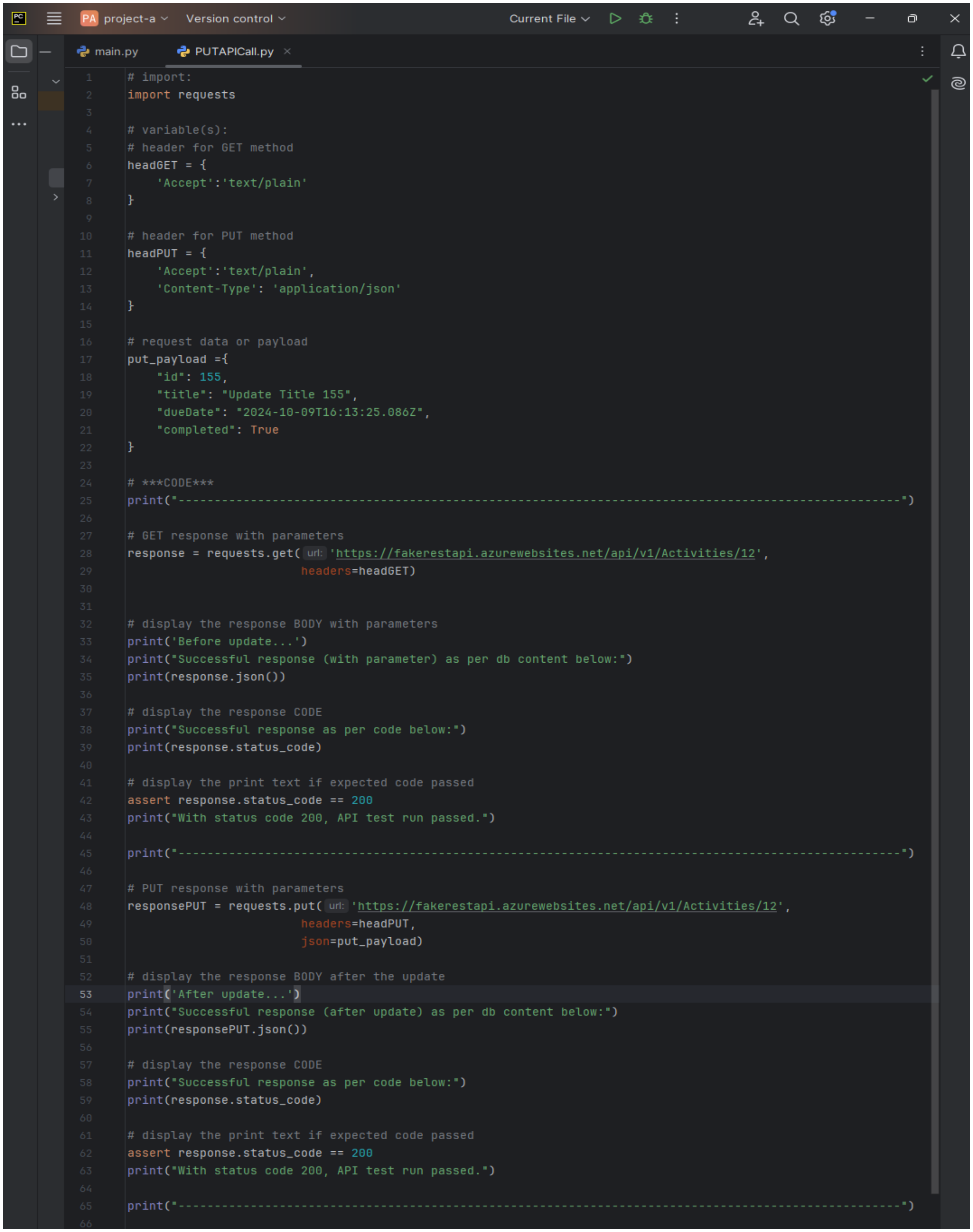
PASS Status code is 200

Save
Share
Send
Params
Authorization
Headers (10)
Body
Scripts
Settings
Cookies
Pre-request
Clear a collection variable
Send a request
Status code: Code is 200
Response body: Contains string
Response body: JSON value check
Response body: Is equal to a string
Response headers: Content-Type header check
Response time is less than 200ms
Status code: Successful POST request
Body
Cookies
Headers (5)
All
Passed
Skipped
Failed

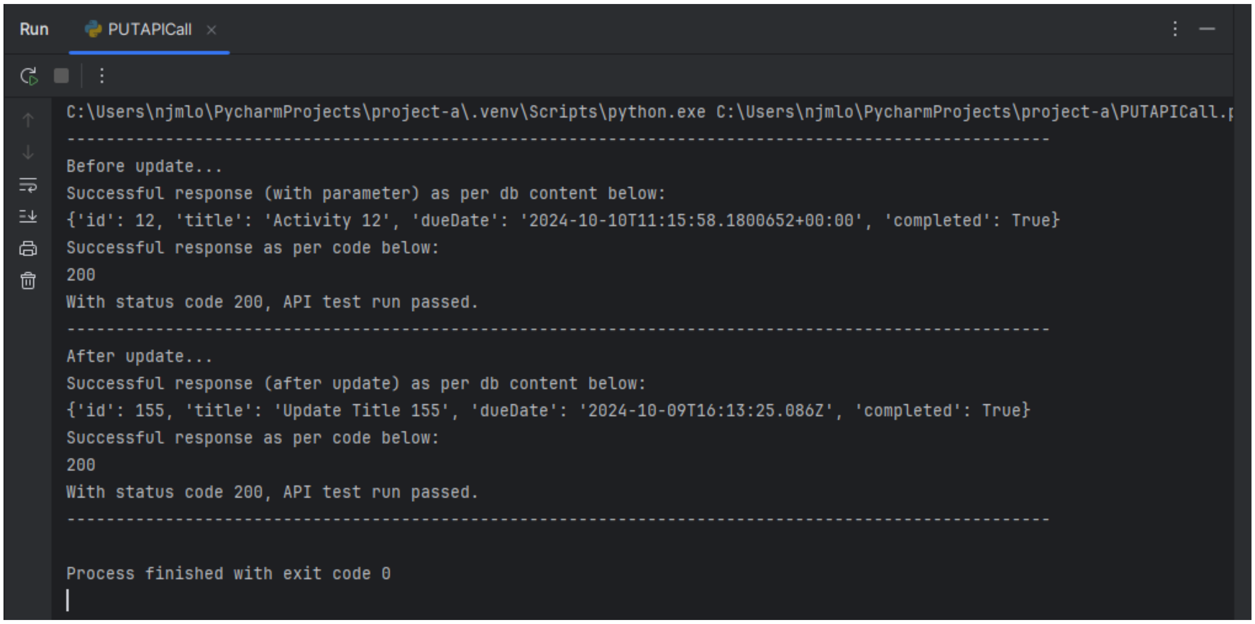
23. Create a python file in PyCharm editor



24. Sample PUT (in PyCharm using Python code)



```python
# import:
import requests

# variable(s):
# header for GET method
headGET = {
    'Accept':'text/plain'
}

# header for PUT method
headPUT = {
    'Accept':'text/plain',
    'Content-Type': 'application/json'
}

# request data or payload
put_payload ={
    "id": 155,
    "title": "Update Title 155",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": True
}

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# GET response with parameters
response = requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities/12',
                        headers=headGET)


# display the response BODY with parameters
print('Before update...')
print("Successful response (with parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# PUT response with parameters
responsePUT = requests.put('https://fakerestapi.azurewebsites.net/api/v1/Activities/12',
                        headers=headPUT,
                        json=put_payload)

# display the response BODY after the update
print('After update...')
print("Successful response (after update) as per db content below:")
print(responsePUT.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")
```



```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\PUTAPICall.p
----------------------------------------------------------------------------------------------------
Before update...
Successful response (with parameter) as per db content below:
{'id': 12, 'title': 'Activity 12', 'dueDate': '2024-10-10T11:15:58.1800652+00:00', 'completed': True}
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------
After update...
Successful response (after update) as per db content below:
{'id': 155, 'title': 'Update Title 155', 'dueDate': '2024-10-09T16:13:25.086Z', 'completed': True}
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------

Process finished with exit code 0
```

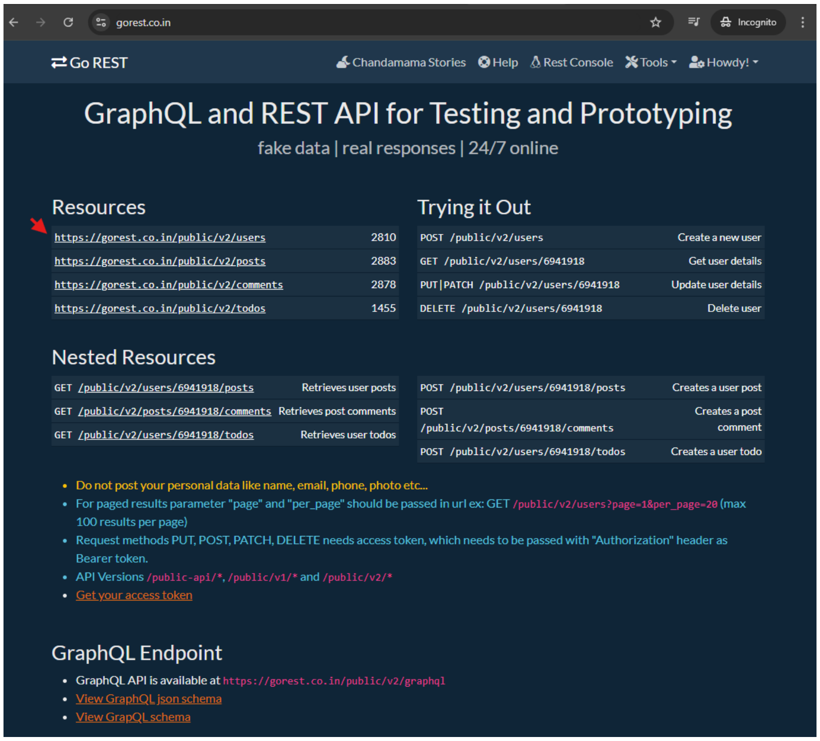
25. Here is the code in text

```
# import:
import requests

# variable(s):
# header for GET method
headGET = {
    'Accept':'text/plain'
}

# header for PUT method
headPUT = {
    'Accept':'text/plain',
    'Content-Type': 'application/json'
}

# request data or payload
put_payload ={
    "id": 155,
    "title": "Update Title 155",
    "dueDate": "2024-10-09T16:13:25.086Z",
    "completed": True
}

# ***CODE***
print("----------------------------------------------------------------------
------------------------------")

# GET response with parameters
response =
requests.get('https://fakerestapi.azurewebsites.net/api/v1/Activities/12',
                        headers=headGET)


# display the response BODY with parameters
print('Before update...')
print("Successful response (with parameter) as per db content below:")
print(response.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------
------------------------------")

# PUT response with parameters
responsePUT =
requests.put('https://fakerestapi.azurewebsites.net/api/v1/Activities/12',
                        headers=headPUT,
                        json=put_payload)

# display the response BODY after the update
print('After update...')
print("Successful response (after update) as per db content below:")
print(responsePUT.json())

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------
------------------------------")
```

26. Setting up and executing an OAuth2 Bearer Token request

url:
https://gorest.co.in/

Go REST sampling APIs



GraphQL and REST API for Testing and Prototyping

fake data | real responses | 24/7 online

Resources

https://gorest.co.in/public/v2/users	2810
https://gorest.co.in/public/v2/posts	2883
https://gorest.co.in/public/v2/comments	2878
https://gorest.co.in/public/v2/todos	1455

Trying it Out

POST /public/v2/users	Create a new user
GET /public/v2/users/6941918	Get user details
PUT|PATCH /public/v2/users/6941918	Update user details
DELETE /public/v2/users/6941918	Delete user

Nested Resources

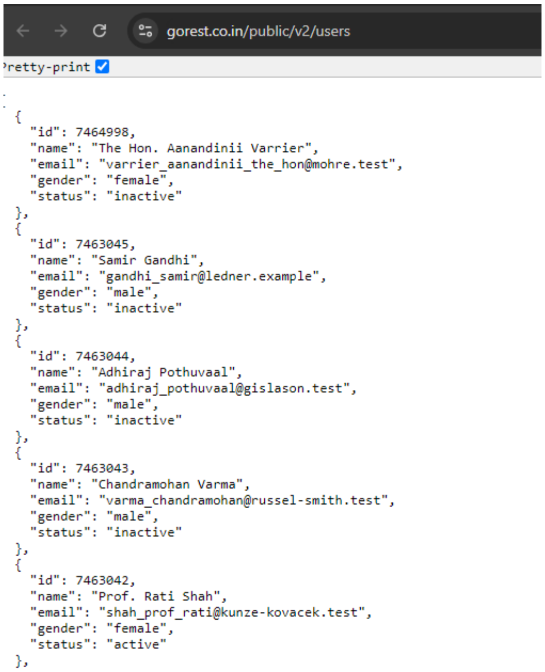
GET /public/v2/users/6941918/posts	Retrieves user posts
GET /public/v2/posts/6941918/comments	Retrieves post comments
GET /public/v2/users/6941918/todos	Retrieves user todos

POST /public/v2/users/6941918/posts	Creates a user post
POST /public/v2/posts/6941918/comments	Creates a post comment
POST /public/v2/users/6941918/todos	Creates a user todo

- Do not post your personal data like name, email, phone, photo etc...
- For paged results parameter "page" and "per_page" should be passed in url ex: GET /public/v2/users?page=1&per_page=20 (max 100 results per page)
- Request methods PUT, POST, PATCH, DELETE needs access token, which needs to be passed with "Authorization" header as Bearer token.
- API Versions /public-api/*, /public/v1/* and /public/v2/*
- Get your access token

GraphQL Endpoint

- GraphQL API is available at https://gorest.co.in/public/v2/graphql
- View GraphQL json schema
- View GrapQL schema



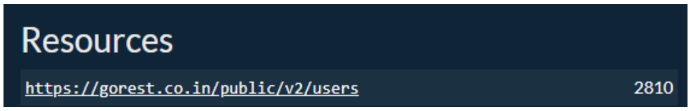
```
  {
    "id": 7464998,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohre.test",
    "gender": "female",
    "status": "inactive"
  },
  {
    "id": 7463045,
    "name": "Samir Gandhi",
    "email": "gandhi_samir@ledner.example",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463044,
    "name": "Adhiraj Pothuvaal",
    "email": "adhiraj_pothuvaal@gislason.test",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463043,
    "name": "Chandramohan Varma",
    "email": "varma_chandramohan@russel-smith.test",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463042,
    "name": "Prof. Rati Shah",
    "email": "shah_prof_rati@kunze-kovacek.test",
    "gender": "female",
    "status": "active"
  },
```

27. Sample POST with OAuth2 bearer key (copy the sampling url from Go REST and into Postman)

Validate and compare the following response results:

- **Status**
- **Body**
- **Header**
- **Time**

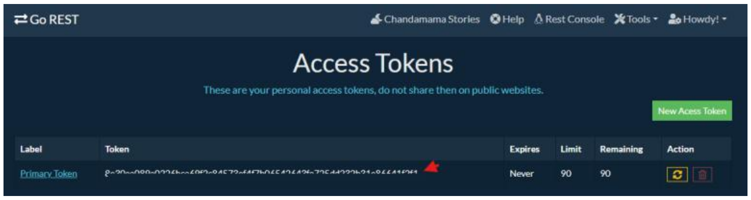
Go REST



Resources

https://gorest.co.in/public/v2/users 2810

Sample Github token



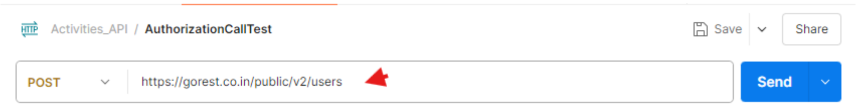
Access Tokens

These are your personal access tokens, do not share then on public websites.

Label	Token	Expires	Limit	Remaining	Action
Primary Token	[illegible]	Never	90	90	

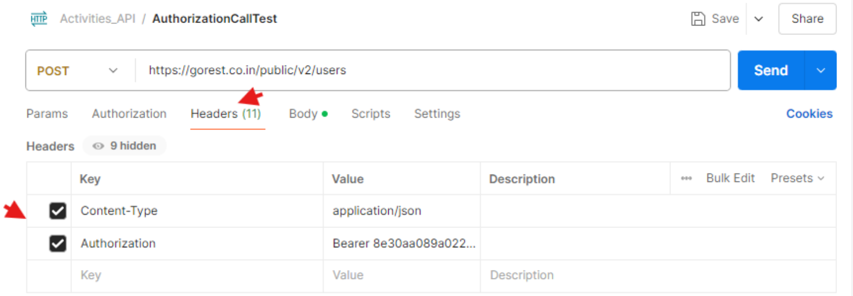
Postman

URL



POST https://gorest.co.in/public/v2/users

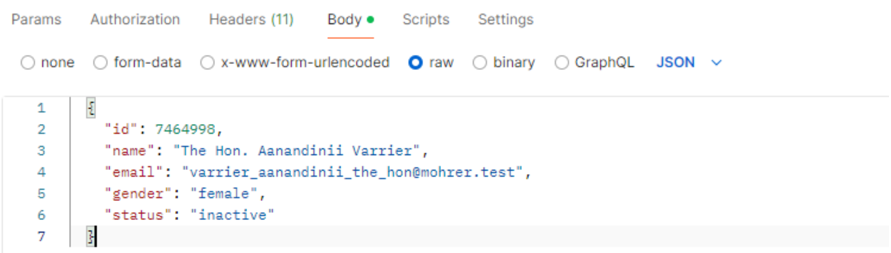
Header



POST https://gorest.co.in/public/v2/users

	Key	Value	Description
☑	Content-Type	application/json	
☑	Authorization	Bearer 8e30aa089a022…	
	Key	Value	Description

Body

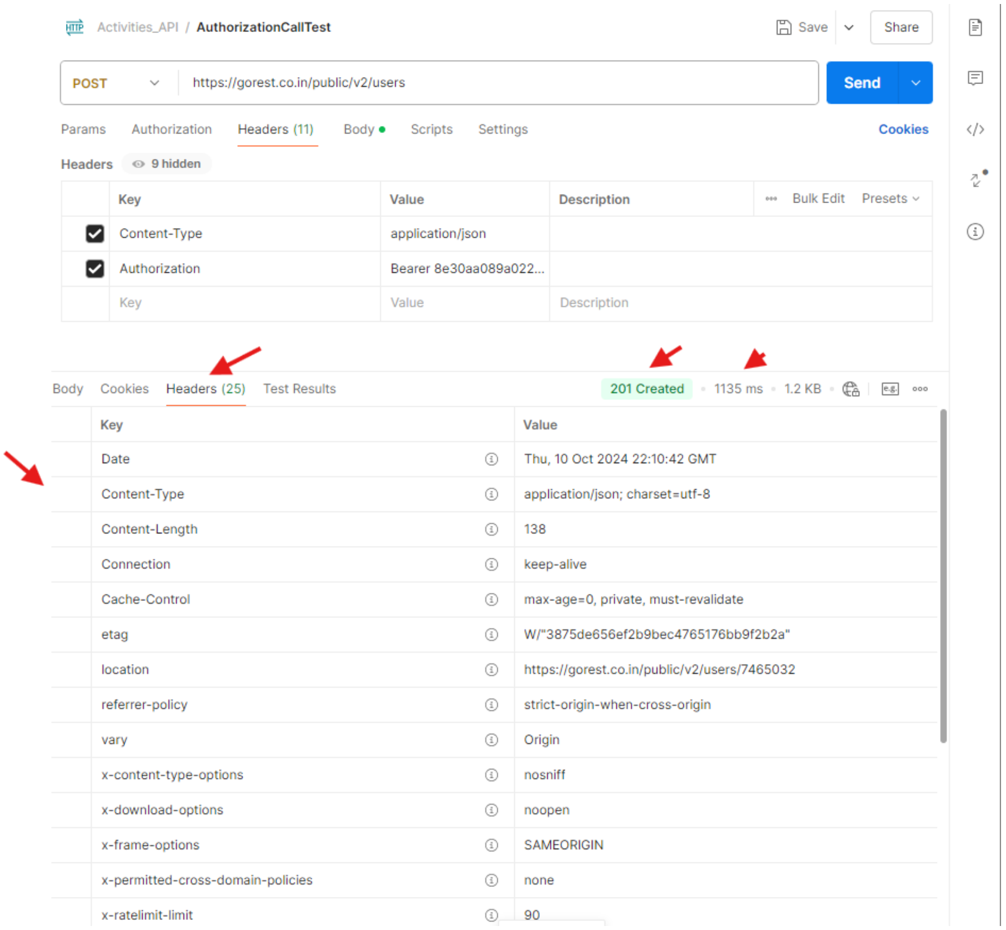


○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
  "id": 7464998,
  "name": "The Hon. Aanandinii Varrier",
  "email": "varrier_aanandinii_the_hon@mohrer.test",
  "gender": "female",
  "status": "inactive"
}
```

After Send…

Header



Activities_API / AuthorizationCallTest

POST https://gorest.co.in/public/v2/users

Params | Authorization | Headers (11) | Body ● | Scripts | Settings | Cookies

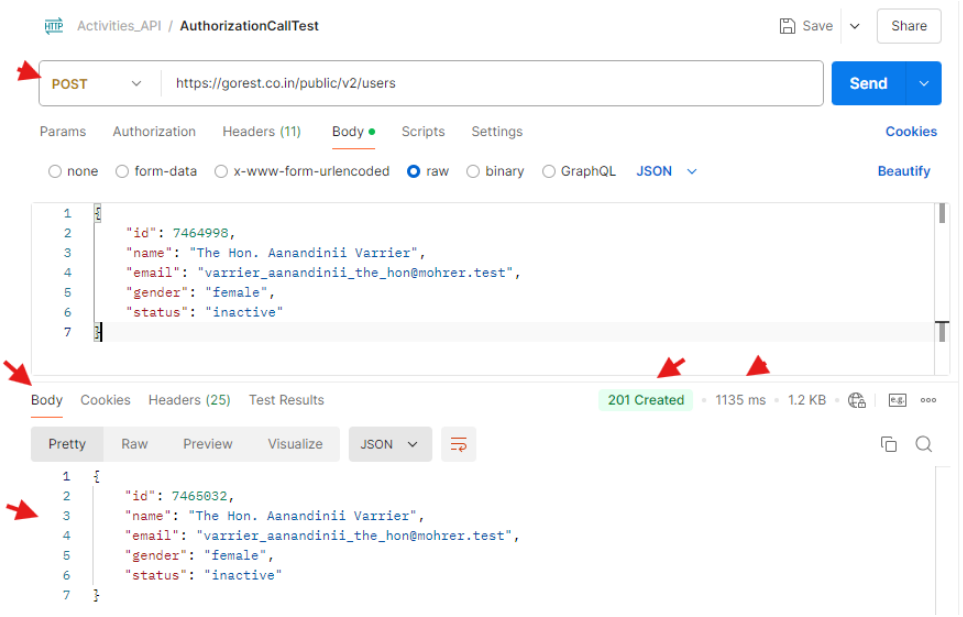
Headers 9 hidden

	Key	Value	Description
☑	Content-Type	application/json	
☑	Authorization	Bearer 8e30aa089a022...	
	Key	Value	Description

Body | Cookies | Headers (25) | Test Results — 201 Created · 1135 ms · 1.2 KB

Key	Value
Date	Thu, 10 Oct 2024 22:10:42 GMT
Content-Type	application/json; charset=utf-8
Content-Length	138
Connection	keep-alive
Cache-Control	max-age=0, private, must-revalidate
etag	W/"3875de656ef2b9bec4765176bb9f2b2a"
location	https://gorest.co.in/public/v2/users/7465032
referrer-policy	strict-origin-when-cross-origin
vary	Origin
x-content-type-options	nosniff
x-download-options	noopen
x-frame-options	SAMEORIGIN
x-permitted-cross-domain-policies	none
x-ratelimit-limit	90

Body



Activities_API / AuthorizationCallTest

POST https://gorest.co.in/public/v2/users

Params | Authorization | Headers (11) | Body ● | Scripts | Settings | Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON — Beautify

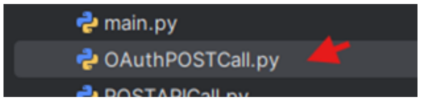
```
{
    "id": 7464998,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohrer.test",
    "gender": "female",
    "status": "inactive"
}
```

Body | Cookies | Headers (25) | Test Results — 201 Created · 1135 ms · 1.2 KB

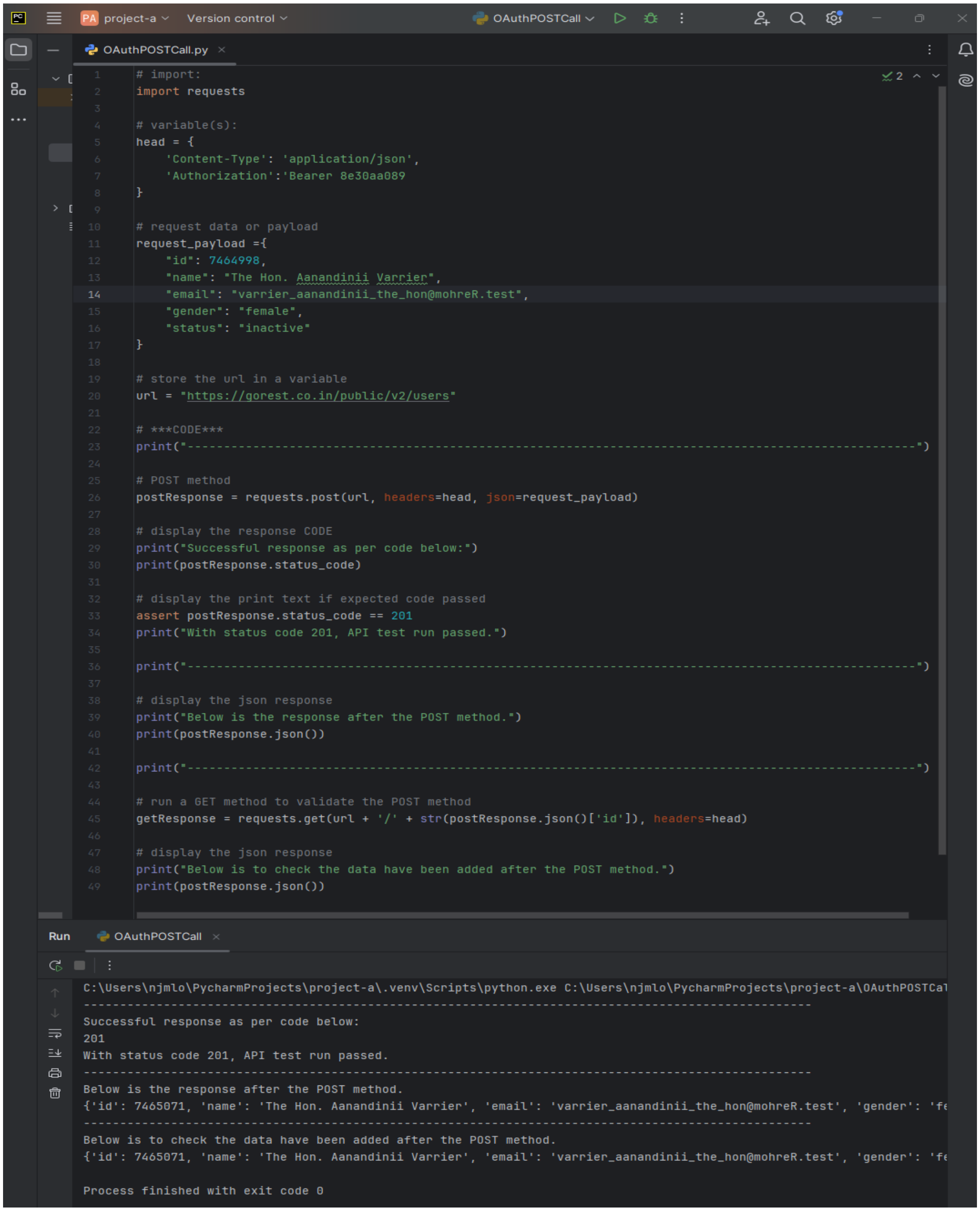
Pretty | Raw | Preview | Visualize | JSON

```
{
    "id": 7465032,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohrer.test",
    "gender": "female",
    "status": "inactive"
}
```

28. Create a python file in PyCharm editor



29. Sample POST (in PyCharm using Python code)



```python
# import:
import requests

# variable(s):
head = {
    'Content-Type': 'application/json',
    'Authorization':'Bearer 8e30aa089
}

# request data or payload
request_payload ={
    "id": 7464998,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohreR.test",
    "gender": "female",
    "status": "inactive"
}

# store the url in a variable
url = "https://gorest.co.in/public/v2/users"

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# POST method
postResponse = requests.post(url, headers=head, json=request_payload)

# display the response CODE
print("Successful response as per code below:")
print(postResponse.status_code)

# display the print text if expected code passed
assert postResponse.status_code == 201
print("With status code 201, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# display the json response
print("Below is the response after the POST method.")
print(postResponse.json())

print("----------------------------------------------------------------------------------------------------")

# run a GET method to validate the POST method
getResponse = requests.get(url + '/' + str(postResponse.json()['id']), headers=head)

# display the json response
print("Below is to check the data have been added after the POST method.")
print(postResponse.json())
```

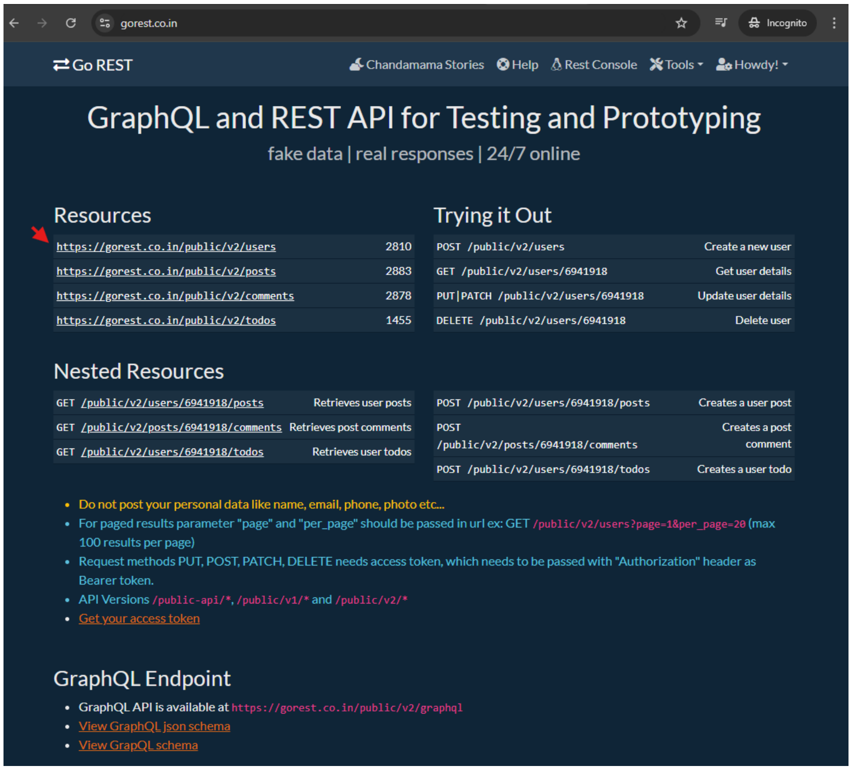
```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\OAuthPOSTCal
----------------------------------------------------------------------------------------------------
Successful response as per code below:
201
With status code 201, API test run passed.
----------------------------------------------------------------------------------------------------
Below is the response after the POST method.
{'id': 7465071, 'name': 'The Hon. Aanandinii Varrier', 'email': 'varrier_aanandinii_the_hon@mohreR.test', 'gender': 'fe
----------------------------------------------------------------------------------------------------
Below is to check the data have been added after the POST method.
{'id': 7465071, 'name': 'The Hon. Aanandinii Varrier', 'email': 'varrier_aanandinii_the_hon@mohreR.test', 'gender': 'fe

Process finished with exit code 0
```

30. Here is the code in text

```
# import:
import requests

# variable(s):
head = {
    'Content-Type': 'application/json',
    'Authorization':'Bearer <place your own token here>' # add your token here
}

# request data or payload
request_payload ={
    "id": 7464998,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohreR.test",
    "gender": "female",
    "status": "inactive"
}

# store the url in a variable
url = "https://gorest.co.in/public/v2/users"

# ***CODE***
print("--------------------------------------------------------------------------------
---------------------")

# POST method
postResponse = requests.post(url, headers=head, json=request_payload)

# display the response CODE
print("Successful response as per code below:")
print(postResponse.status_code)

# display the print text if expected code passed
assert postResponse.status_code == 201
print("With status code 201, API test run passed.")

print("--------------------------------------------------------------------------------
---------------------")

# display the json response
print("Below is the response after the POST method.")
print(postResponse.json())

print("--------------------------------------------------------------------------------
---------------------")

# run a GET method to validate the POST method
getResponse = requests.get(url + '/' + str(postResponse.json()['id']), headers=head)

# display the json response
print("Below is to check the data have been added after the POST method.")
print(postResponse.json())
```

31. Passing Parameters in URL

Go REST sampling APIs



GraphQL and REST API for Testing and Prototyping

fake data | real responses | 24/7 online

Resources

https://gorest.co.in/public/v2/users	2810
https://gorest.co.in/public/v2/posts	2883
https://gorest.co.in/public/v2/comments	2878
https://gorest.co.in/public/v2/todos	1455

Trying it Out

POST /public/v2/users	Create a new user
GET /public/v2/users/6941918	Get user details
PUT|PATCH /public/v2/users/6941918	Update user details
DELETE /public/v2/users/6941918	Delete user

Nested Resources

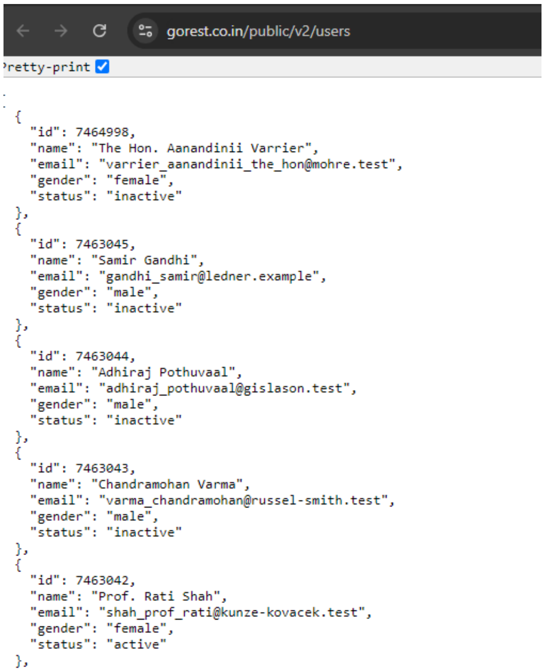
GET /public/v2/users/6941918/posts	Retrieves user posts
GET /public/v2/posts/6941918/comments	Retrieves post comments
GET /public/v2/users/6941918/todos	Retrieves user todos

POST /public/v2/users/6941918/posts	Creates a user post
POST /public/v2/posts/6941918/comments	Creates a post comment
POST /public/v2/users/6941918/todos	Creates a user todo

- Do not post your personal data like name, email, phone, photo etc...
- For paged results parameter "page" and "per_page" should be passed in url ex: GET `/public/v2/users?page=1&per_page=20` (max 100 results per page)
- Request methods PUT, POST, PATCH, DELETE needs access token, which needs to be passed with "Authorization" header as Bearer token.
- API Versions `/public-api/*`, `/public/v1/*` and `/public/v2/*`
- Get your access token

GraphQL Endpoint

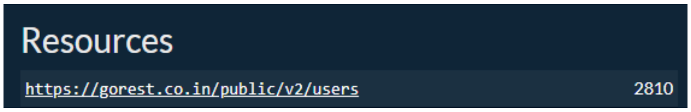
- GraphQL API is available at `https://gorest.co.in/public/v2/graphql`
- View GraphQL json schema
- View GrapQL schema



```
  {
    "id": 7464998,
    "name": "The Hon. Aanandinii Varrier",
    "email": "varrier_aanandinii_the_hon@mohre.test",
    "gender": "female",
    "status": "inactive"
  },
  {
    "id": 7463045,
    "name": "Samir Gandhi",
    "email": "gandhi_samir@ledner.example",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463044,
    "name": "Adhiraj Pothuvaal",
    "email": "adhiraj_pothuvaal@gislason.test",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463043,
    "name": "Chandramohan Varma",
    "email": "varma_chandramohan@russel-smith.test",
    "gender": "male",
    "status": "inactive"
  },
  {
    "id": 7463042,
    "name": "Prof. Rati Shah",
    "email": "shah_prof_rati@kunze-kovacek.test",
    "gender": "female",
    "status": "active"
  },
```

32. Sample GET (copy the sampling url from Go REST and into Postman (add parameters)

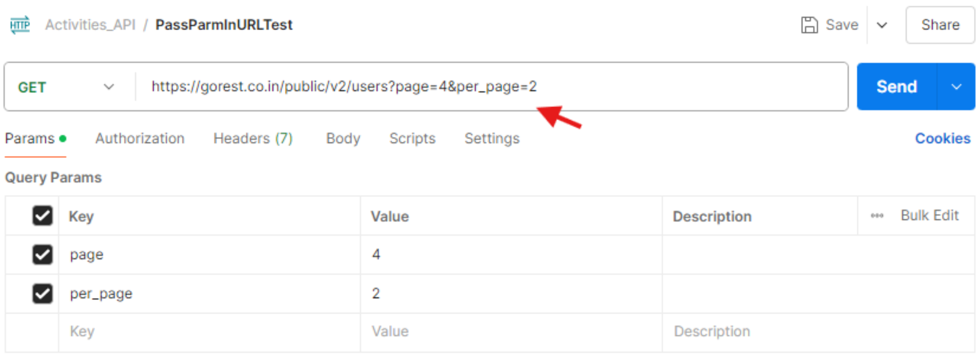
Go REST



Resources

https://gorest.co.in/public/v2/users	2810

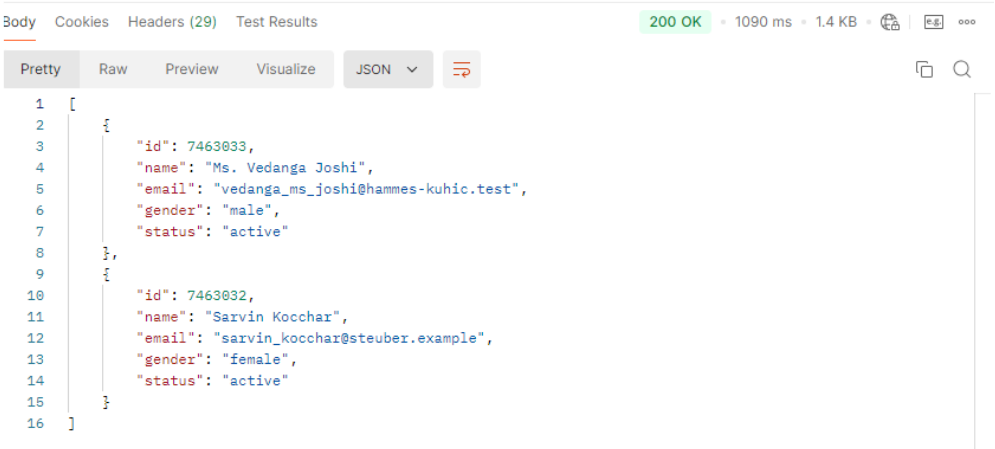
Postman



https://gorest.co.in/public/v2/users?page=4&per_page=2

Query Params

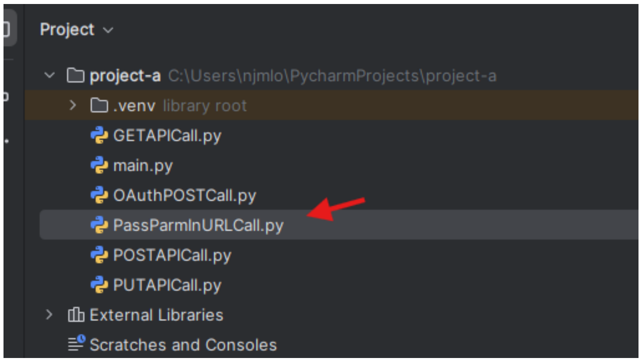
	Key	Value	Description
☑	page	4	
☑	per_page	2	
	Key	Value	Description



200 OK • 1090 ms • 1.4 KB

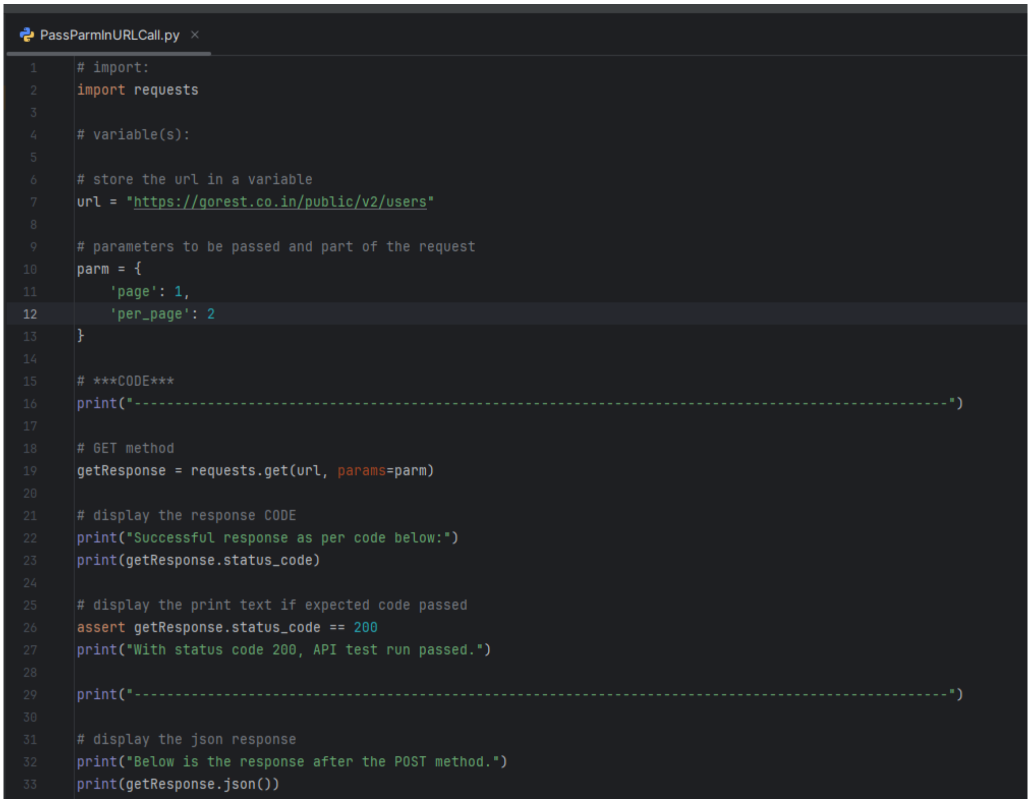
```
[
    {
        "id": 7463033,
        "name": "Ms. Vedanga Joshi",
        "email": "vedanga_ms_joshi@hammes-kuhic.test",
        "gender": "male",
        "status": "active"
    },
    {
        "id": 7463032,
        "name": "Sarvin Kocchar",
        "email": "sarvin_kocchar@steuber.example",
        "gender": "female",
        "status": "active"
    }
]
```

33. Create a python file in PyCharm editor

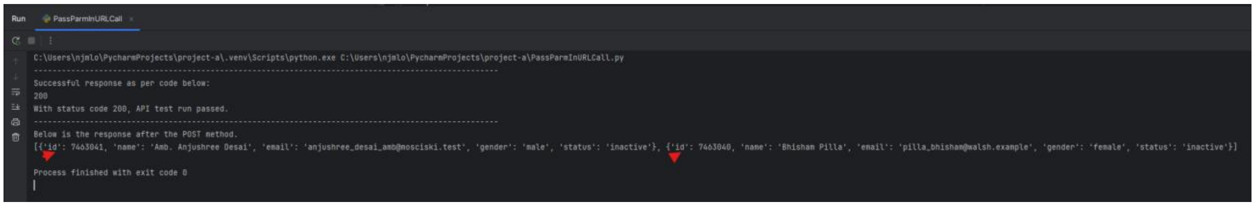


```
Project
project-a C:\Users\njmlo\PycharmProjects\project-a
    .venv library root
    GETAPICall.py
    main.py
    OAuthPOSTCall.py
    PassParmInURLCall.py
    POSTAPICall.py
    PUTAPICall.py
External Libraries
Scratches and Consoles
```

34. Sample GET (in PyCharm using Python code)

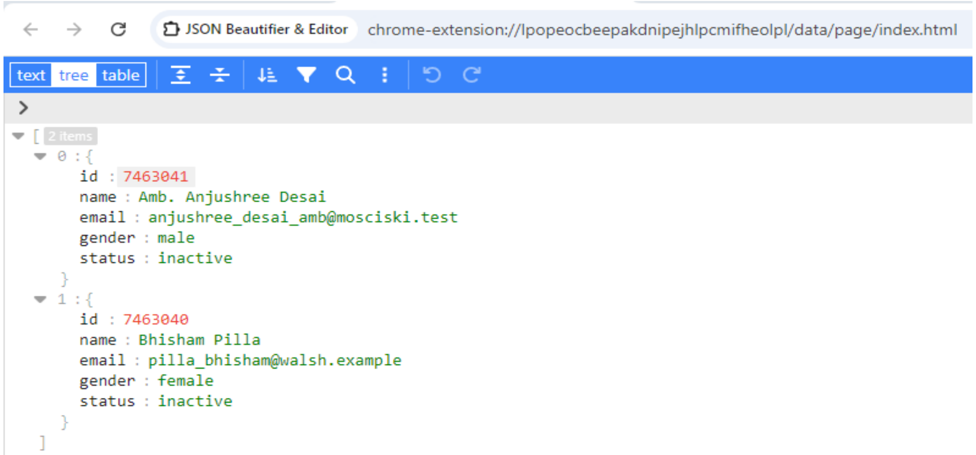


```python
# import:
import requests

# variable(s):

# store the url in a variable
url = "https://gorest.co.in/public/v2/users"

# parameters to be passed and part of the request
parm = {
    'page': 1,
    'per_page': 2
}

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# GET method
getResponse = requests.get(url, params=parm)

# display the response CODE
print("Successful response as per code below:")
print(getResponse.status_code)

# display the print text if expected code passed
assert getResponse.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# display the json response
print("Below is the response after the POST method.")
print(getResponse.json())
```



```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\PassParmInURLCall.py
----------------------------------------------------------------------------------------------------
Successful response as per code below:
200
With status code 200, API test run passed.
----------------------------------------------------------------------------------------------------
Below is the response after the POST method.
[{'id': 7463041, 'name': 'Amb. Anjushree Desai', 'email': 'anjushree_desai_amb@mosciski.test', 'gender': 'male', 'status': 'inactive'}, {'id': 7463040, 'name': 'Bhisham Pilla', 'email': 'pilla_bhisham@walsh.example', 'gender': 'female', 'status': 'inactive'}]

Process finished with exit code 0
```

35. Using Json Beautifier to display response result nicely (copy and paste the result)



JSON Beautifier & Editor
chrome-extension://lpopeocbeepakdnipejhlpcmifheolpl/data/page/index.html
text
tree
table

```
[ 2 items
  0 : {
      id : 7463041
      name : Amb. Anjushree Desai
      email : anjushree_desai_amb@mosciski.test
      gender : male
      status : inactive
    }
  1 : {
      id : 7463040
      name : Bhisham Pilla
      email : pilla_bhisham@walsh.example
      gender : female
      status : inactive
    }
]
```

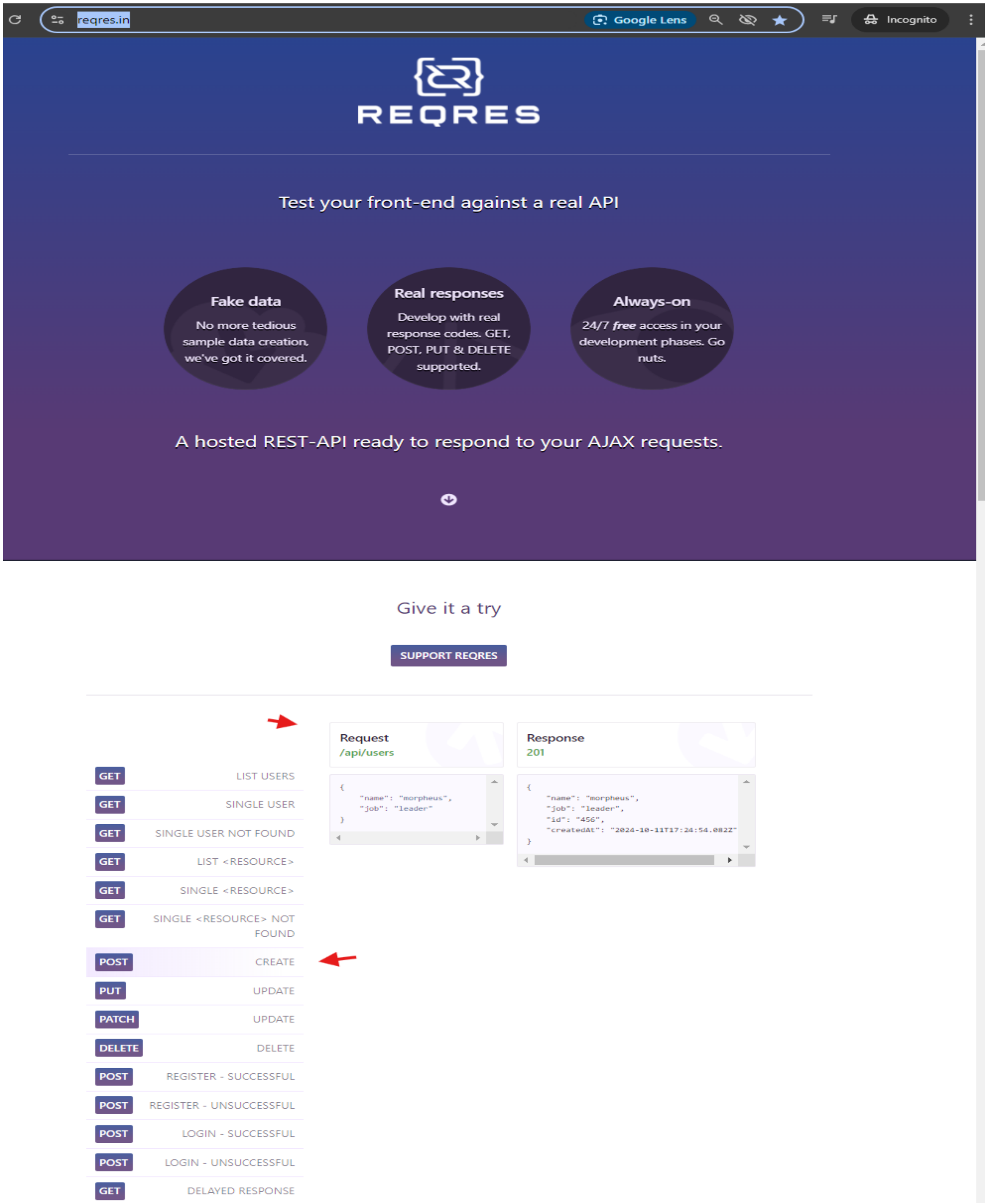
36. Here is the code in text

```python
# import:
import requests

# variable(s):

# store the url in a variable
url = "https://gorest.co.in/public/v2/users"

# parameters to be passed and part of the request
parm = {
    'page': 1,
    'per_page': 2
}

# ***CODE***
print("----------------------------------------------------------------------------------------------------")

# GET method
getResponse = requests.get(url, params=parm)

# display the response CODE
print("Successful response as per code below:")
print(getResponse.status_code)

# display the print text if expected code passed
assert getResponse.status_code == 200
print("With status code 200, API test run passed.")

print("----------------------------------------------------------------------------------------------------")

# display the json response
print("Below is the response after the POST method.")
print(getResponse.json())
```

37. Passing payload from JSON file

url:
https://reqres.in/

REQRES sampling APIs



Test your front-end against a real API

Fake data
No more tedious sample data creation, we've got it covered.

Real responses
Develop with real response codes. GET, POST, PUT & DELETE supported.

Always-on
24/7 *free* access in your development phases. Go nuts.

A hosted REST-API ready to respond to your AJAX requests.

Give it a try

Request
/api/users

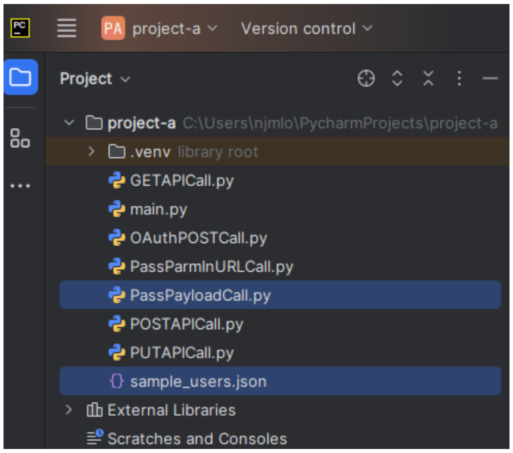
```
{
    "name": "morpheus",
    "job": "leader"
}
```

Response
201

```
{
    "name": "morpheus",
    "job": "leader",
    "id": "456",
    "createdAt": "2024-10-11T17:24:54.082Z"
}
```

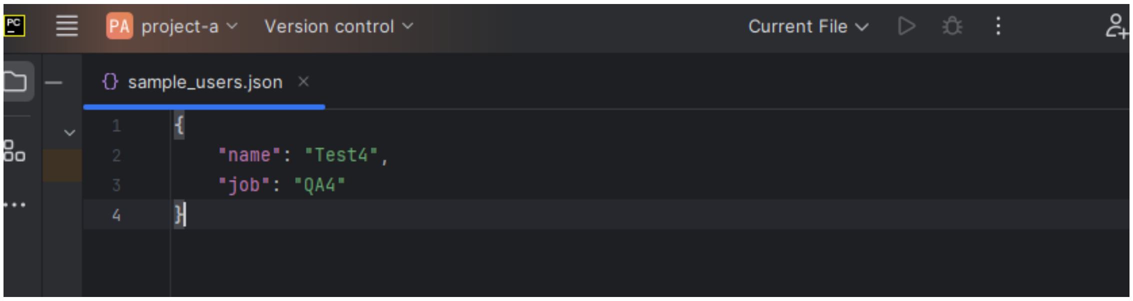
38. Create file(s) in PyCharm editor

Python and JSON file



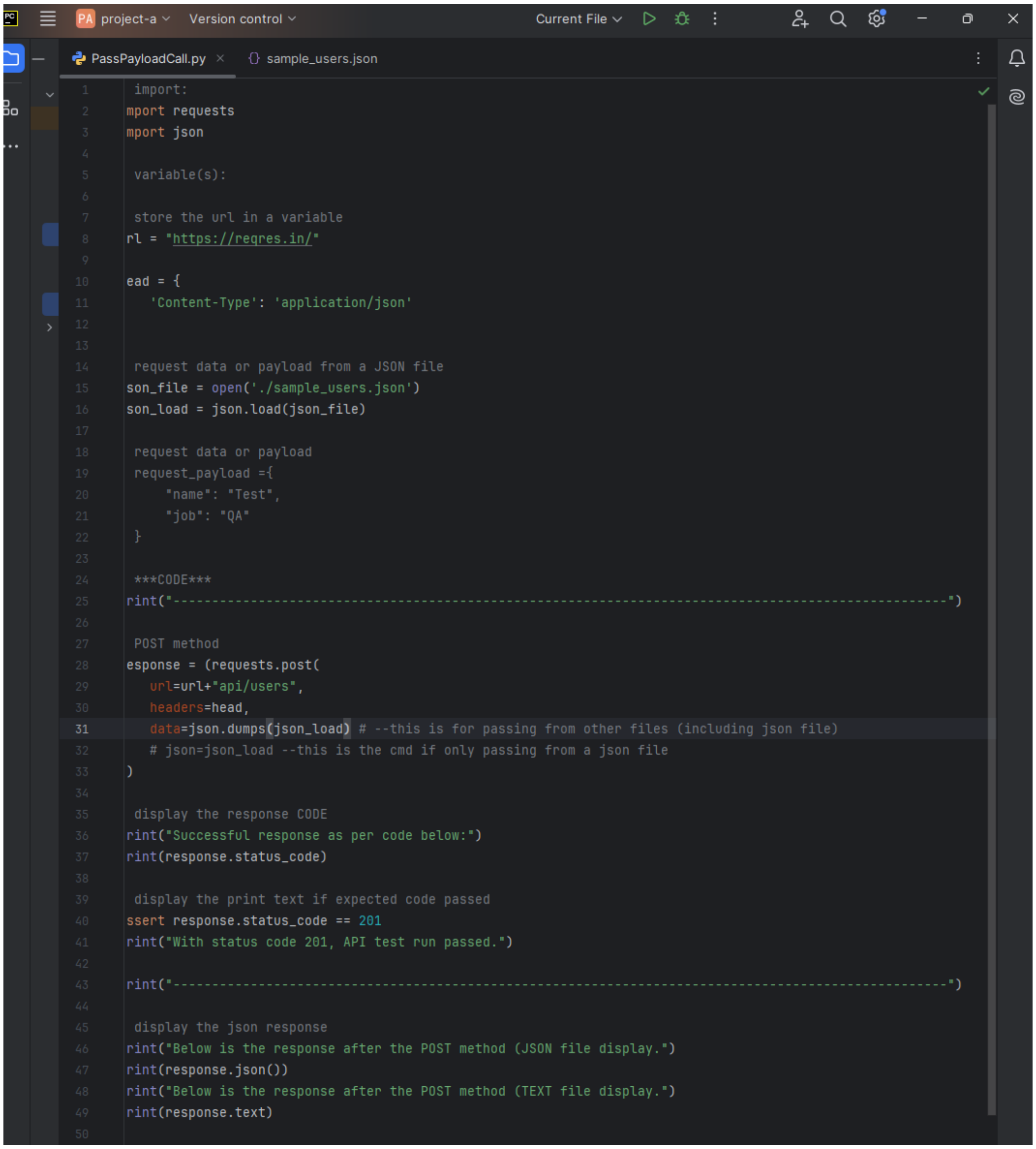
39. Sample POST (in PyCharm using Python code)

JSON file

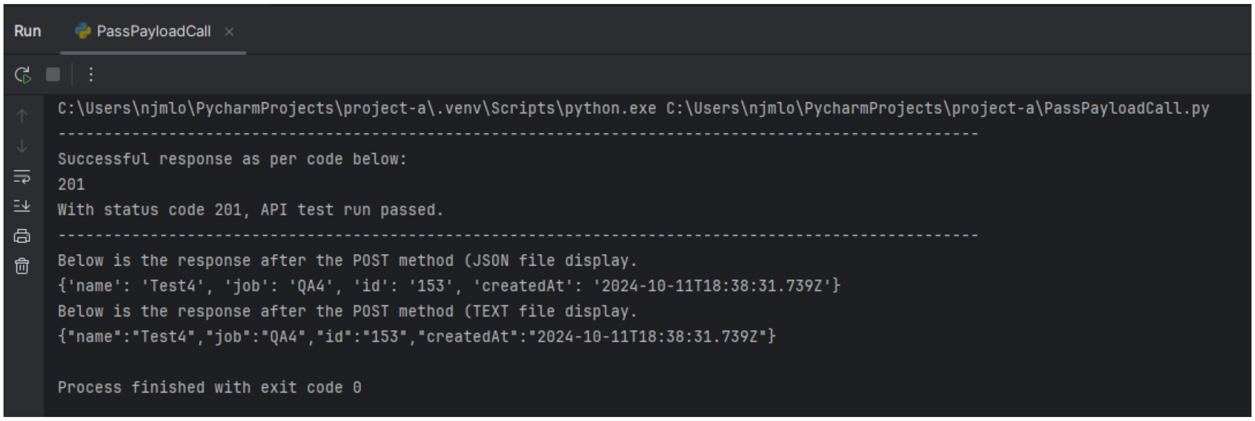


```
{
    "name": "Test4",
    "job": "QA4"
}
```

Python file



```
 import:
mport requests
mport json

 variable(s):

 store the url in a variable
rl = "https://reqres.in/"

ead = {
   'Content-Type': 'application/json'


 request data or payload from a JSON file
son_file = open('./sample_users.json')
son_load = json.load(json_file)

 request data or payload
 request_payload ={
     "name": "Test",
     "job": "QA"
 }

 ***CODE***
rint("----------------------------------------------------------------------------------------------------")

 POST method
esponse = (requests.post(
   url=url+"api/users",
   headers=head,
   data=json.dumps(json_load) # --this is for passing from other files (including json file)
   # json=json_load --this is the cmd if only passing from a json file
)

 display the response CODE
rint("Successful response as per code below:")
rint(response.status_code)

 display the print text if expected code passed
ssert response.status_code == 201
rint("With status code 201, API test run passed.")

rint("----------------------------------------------------------------------------------------------------")

 display the json response
rint("Below is the response after the POST method (JSON file display.")
rint(response.json())
rint("Below is the response after the POST method (TEXT file display.")
rint(response.text)
```



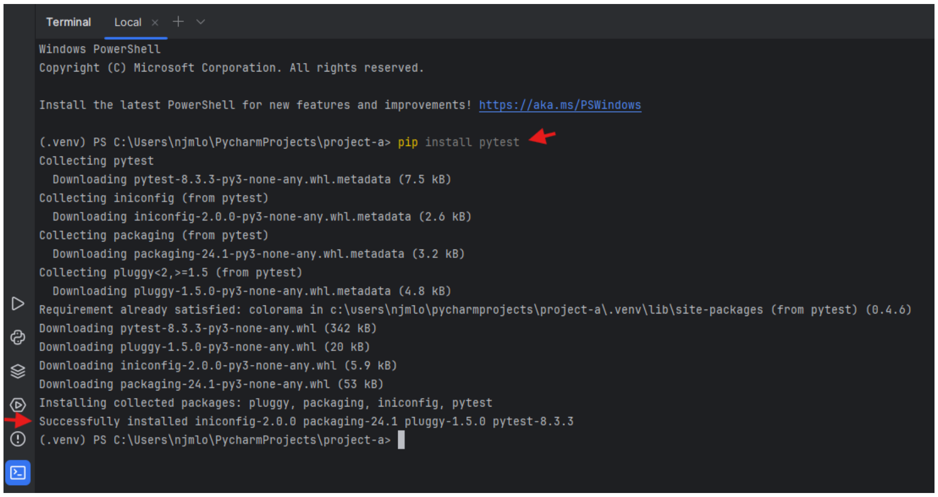
```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\PassPayloadCall.py
----------------------------------------------------------------------------------------------------
Successful response as per code below:
201
With status code 201, API test run passed.
----------------------------------------------------------------------------------------------------
Below is the response after the POST method (JSON file display.
{'name': 'Test4', 'job': 'QA4', 'id': '153', 'createdAt': '2024-10-11T18:38:31.739Z'}
Below is the response after the POST method (TEXT file display.
{"name":"Test4","job":"QA4","id":"153","createdAt":"2024-10-11T18:38:31.739Z"}

Process finished with exit code 0
```

40. Here is the code in text

```
# import:
import requests
import json

# variable(s):

# store the url in a variable
url = "https://reqres.in/"

head = {
    'Content-Type': 'application/json'
}

# request data or payload from a JSON file
json_file = open('./sample_users.json')
json_load = json.load(json_file)

# request data or payload
# request_payload ={
#     "name": "Test",
#     "job": "QA"
# }

# ***CODE***
print("----------------------------------------------------------------------
------------------------------")

# POST method
response = (requests.post(
    url=url+"api/users",
    headers=head,
    data=json.dumps(json_load) # --this is for passing from other files
(including json file)
    # json=json_load --this is the cmd if only passing from a json file
))

# display the response CODE
print("Successful response as per code below:")
print(response.status_code)

# display the print text if expected code passed
assert response.status_code == 201
print("With status code 201, API test run passed.")

print("----------------------------------------------------------------------
------------------------------")

# display the json response
print("Below is the response after the POST method (JSON file display.")
print(response.json())
print("Below is the response after the POST method (TEXT file display.")
print(response.text)
```

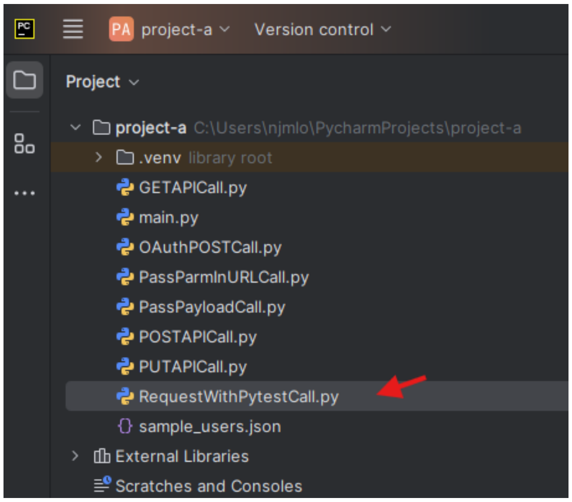
41. Python request API automation with PyTest

PyTest installation (in PyCharm Terminal)



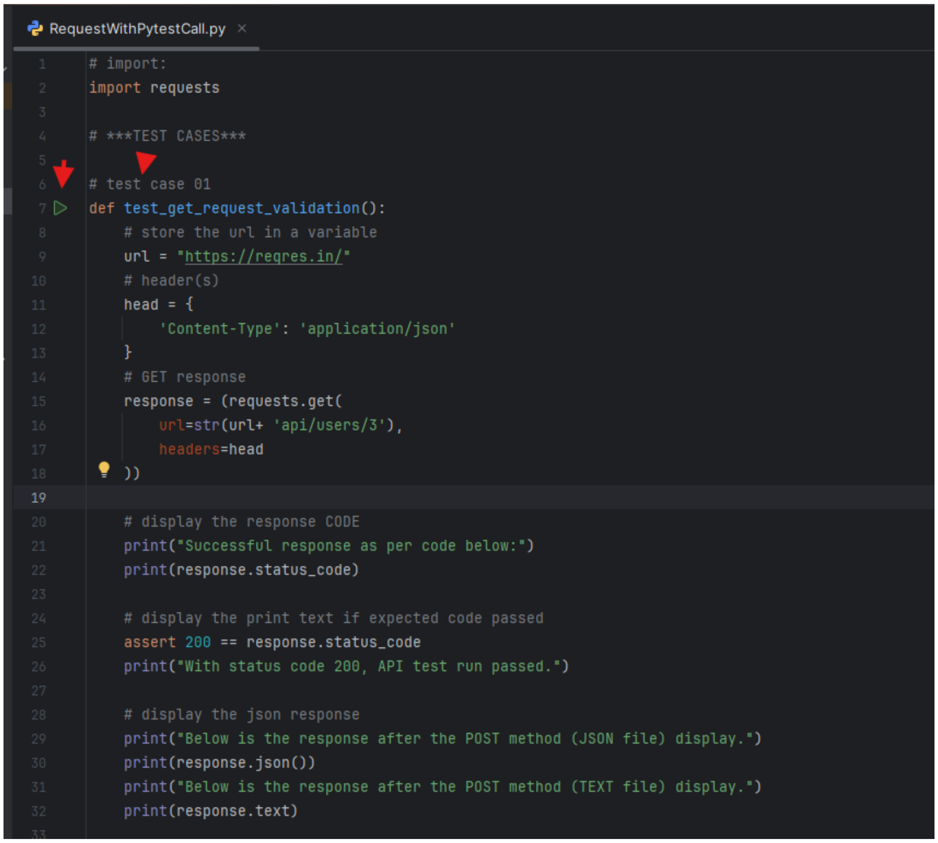
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(.venv) PS C:\Users\njmlo\PycharmProjects\project-a> pip install pytest
Collecting pytest
  Downloading pytest-8.3.3-py3-none-any.whl.metadata (7.5 kB)
Collecting iniconfig (from pytest)
  Downloading iniconfig-2.0.0-py3-none-any.whl.metadata (2.6 kB)
Collecting packaging (from pytest)
  Downloading packaging-24.1-py3-none-any.whl.metadata (3.2 kB)
Collecting pluggy<2,>=1.5 (from pytest)
  Downloading pluggy-1.5.0-py3-none-any.whl.metadata (4.8 kB)
Requirement already satisfied: colorama in c:\users\njmlo\pycharmprojects\project-a\.venv\lib\site-packages (from pytest) (0.4.6)
Downloading pytest-8.3.3-py3-none-any.whl (342 kB)
Downloading pluggy-1.5.0-py3-none-any.whl (20 kB)
Downloading iniconfig-2.0.0-py3-none-any.whl (5.9 kB)
Downloading packaging-24.1-py3-none-any.whl (53 kB)
Installing collected packages: pluggy, packaging, iniconfig, pytest
Successfully installed iniconfig-2.0.0 packaging-24.1 pluggy-1.5.0 pytest-8.3.3
(.venv) PS C:\Users\njmlo\PycharmProjects\project-a>
```

42. Create a python file in PyCharm editor



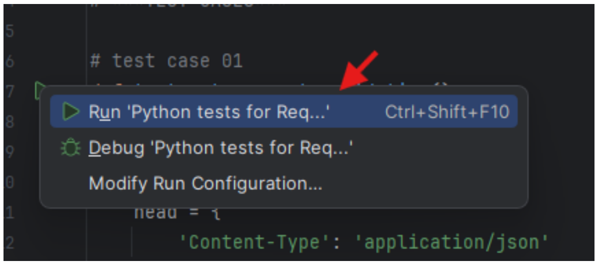
43. Sample GET (in PyCharm using Python code and PyTest to run the “test”)

Python file

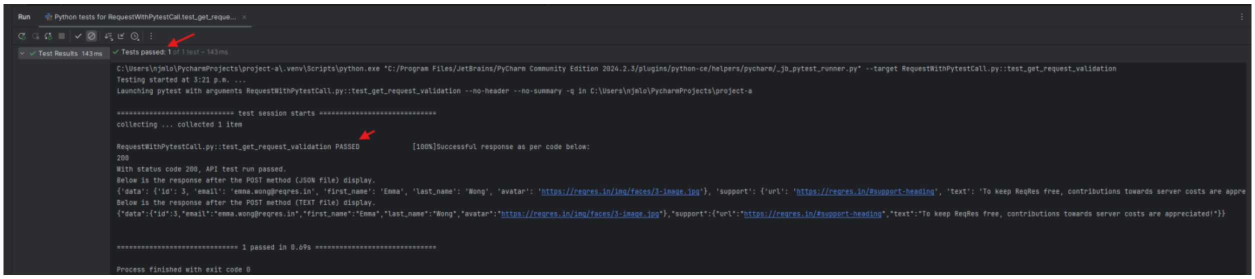


```python
# import:
import requests

# ***TEST CASES***

# test case 01
def test_get_request_validation():
    # store the url in a variable
    url = "https://reqres.in/"
    # header(s)
    head = {
        'Content-Type': 'application/json'
    }
    # GET response
    response = (requests.get(
        url=str(url+ 'api/users/3'),
        headers=head
    ))

    # display the response CODE
    print("Successful response as per code below:")
    print(response.status_code)

    # display the print text if expected code passed
    assert 200 == response.status_code
    print("With status code 200, API test run passed.")

    # display the json response
    print("Below is the response after the POST method (JSON file) display.")
    print(response.json())
    print("Below is the response after the POST method (TEXT file) display.")
    print(response.text)
```

PyTest test run

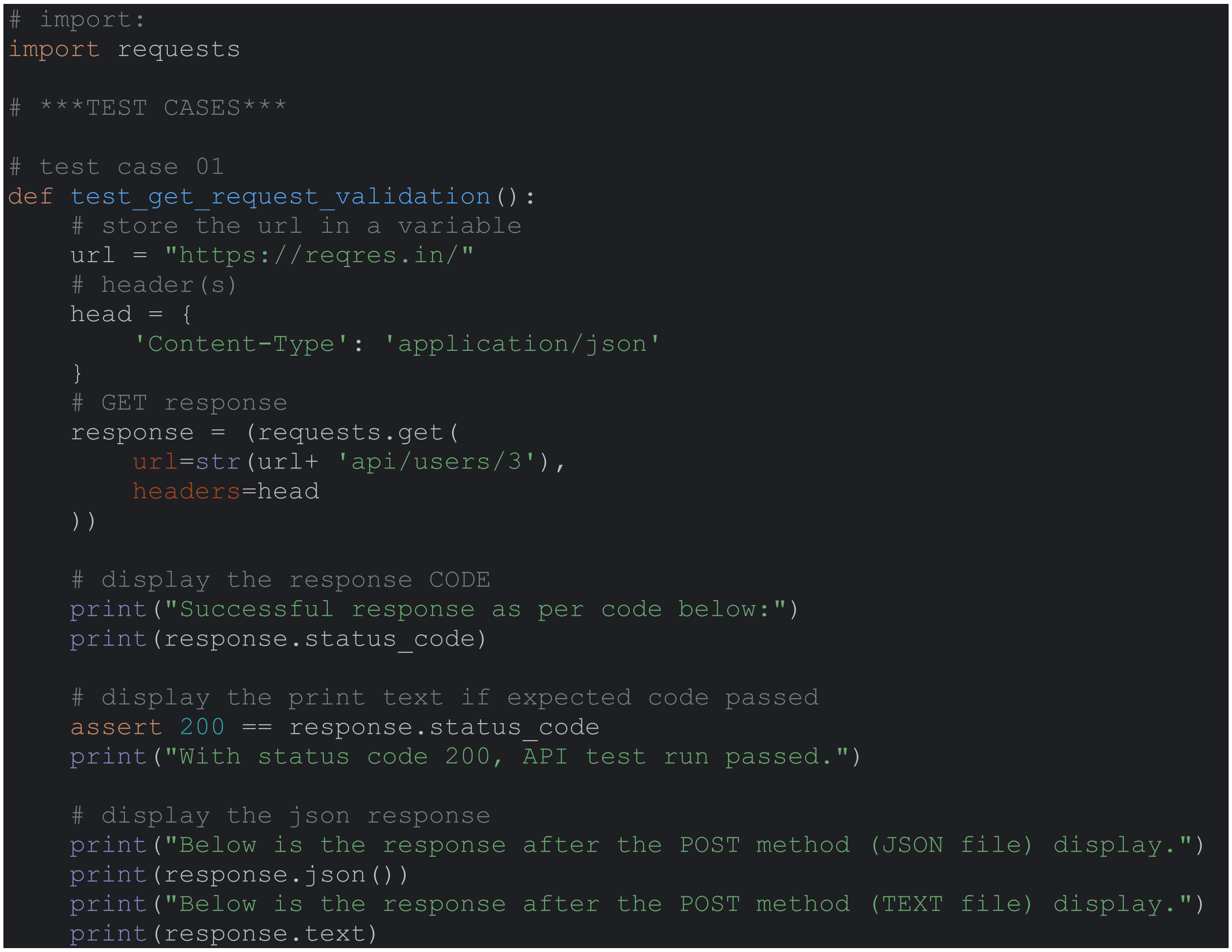

# test case 01
Run 'Python tests for Req...' Ctrl+Shift+F10
Debug 'Python tests for Req...'
Modify Run Configuration...
head = {
'Content-Type': 'application/json'

Response


Run
Python tests for RequestWithPytestCall.test_get_reque...
Test Results 143 ms
Tests passed: 1 of 1 test – 143 ms

```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2024.2.3/plugins/python-ce/helpers/pycharm/_jb_pytest_runner.py" --target RequestWithPytestCall.py::test_get_request_validation
Testing started at 3:21 p.m. ...
Launching pytest with arguments RequestWithPytestCall.py::test_get_request_validation --no-header --no-summary -q in C:\Users\njmlo\PycharmProjects\project-a

============================== test session starts ==============================
collecting ... collected 1 item

RequestWithPytestCall.py::test_get_request_validation PASSED            [100%]Successful response as per code below:
200
With status code 200, API test run passed.
Below is the response after the POST method (JSON file) display.
{'data': {'id': 3, 'email': 'emma.wong@reqres.in', 'first_name': 'Emma', 'last_name': 'Wong', 'avatar': 'https://reqres.in/img/faces/3-image.jpg'}, 'support': {'url': 'https://reqres.in/#support-heading', 'text': 'To keep ReqRes free, contributions towards server costs are appre
Below is the response after the POST method (TEXT file) display.
{"data":{"id":3,"email":"emma.wong@reqres.in","first_name":"Emma","last_name":"Wong","avatar":"https://reqres.in/img/faces/3-image.jpg"},"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}


============================== 1 passed in 0.69s ===============================

Process finished with exit code 0
```

44. Here is the code in text



```
# import:
import requests

# ***TEST CASES***

# test case 01
def test_get_request_validation():
    # store the url in a variable
    url = "https://reqres.in/"
    # header(s)
    head = {
        'Content-Type': 'application/json'
    }
    # GET response
    response = (requests.get(
        url=str(url+ 'api/users/3'),
        headers=head
    ))

    # display the response CODE
    print("Successful response as per code below:")
    print(response.status_code)

    # display the print text if expected code passed
    assert 200 == response.status_code
    print("With status code 200, API test run passed.")

    # display the json response
    print("Below is the response after the POST method (JSON file) display.")
    print(response.json())
    print("Below is the response after the POST method (TEXT file) display.")
    print(response.text)
```

END*