

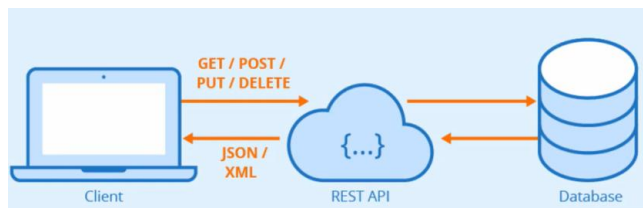
### Test Automation using:

- 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

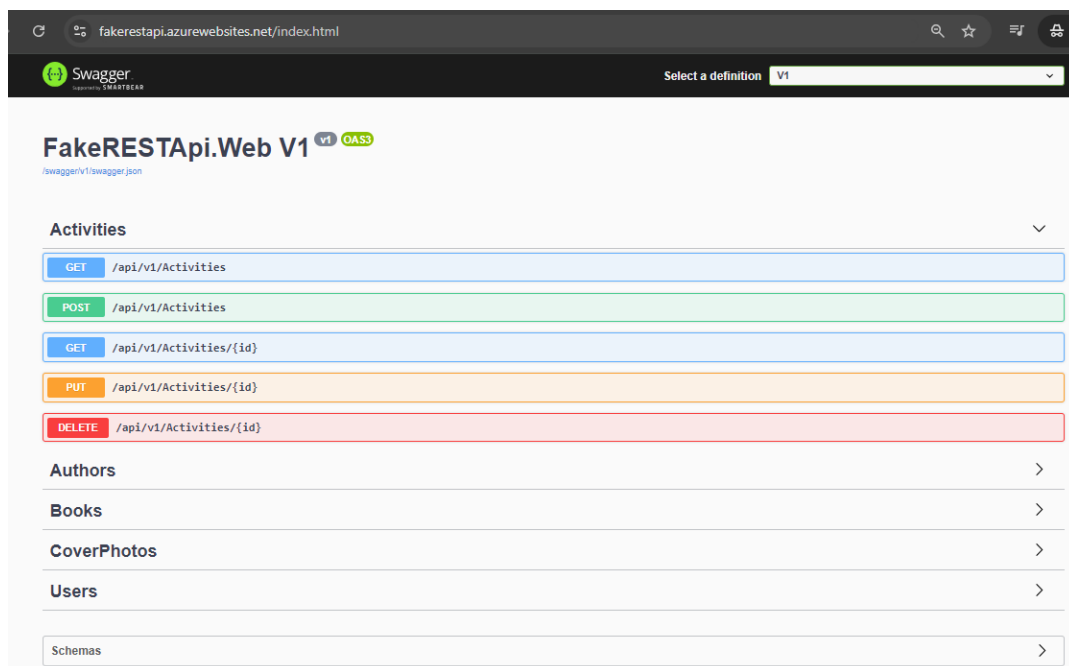
## 1. API Concept (interface between client and server (database), request and response)



## 2. Sampling APIs (using Swagger)

url:

<https://fakerestapi.azurewebsites.net/index.html>



### 3. Executing a GET method

Swagger  
Supported by SMARTBEAR

Select a definition V1

## FakeRESTApi.Web V1 v1 OAS3

/swagger/v1/swagger.json

### Activities

GET /api/v1/Activities

Parameters

No parameters

Execute Clear

Responses

Curl

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

Request URL

```
https://fakerestopi.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",
    "dueDate": "2024-10-08T06:05:45.3979890+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 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

The screenshot displays the Swagger UI for 'FakeRESTApi.Web V1'. The 'Activities' endpoint is selected, showing a GET request to '/api/v1/Activities'. The 'Parameters' section is empty. The 'Responses' section shows a 200 status code with a JSON response body. Red arrows highlight the 'Execute' button, the 'Request URL', the '200' status code, and the 'Response body'.

Swagger

Select a definition V1

### FakeRESTApi.Web V1 v1 OAS3

/swagger/v1/swagger.json

#### Activities

GET /api/v1/Activities

Parameters

No parameters

Execute Clear

Responses

Curl

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

Request URL

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

Server response

Code Details

200

Response body

```
{
  "id": 1,
  "title": "Activity 1",
  "startDate": "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: Tue 08 Oct 2024 21:13:38 GMT
server: Kestrel
transfer-encoding: chunked
```

## Postman

The screenshot shows the Postman interface with a GET request to `https://fakerestapi.azurewebsites.net/api/v1/Activities`. The response is `200 OK` with a status of `721 ms` and a size of `2.96 KB`. The response body is displayed in JSON format:

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

The screenshot shows the Postman interface with the same GET request, but the `Headers` tab is selected. The response is `200 OK` with a status of `721 ms` and a size of `2.96 KB`. The response body is displayed in JSON format:

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

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

### Swagger

The Swagger UI displays a GET endpoint at `/api/v1/Activities/{id}`. The parameter `id` is required and is of type `integer(int32)`. A red arrow points to the input field where the value `3` is entered. Below the parameters, there are buttons for `Execute` and `Clear`. The `Responses` section shows a `200` status code with a response body containing JSON data. A red arrow points to the `id` field in the JSON response. The response headers include `api-supported-versions: 1.0`, `content-type: application/json; charset=utf-8; v=1.0`, `date: Tue, 01: 2024-10-09T00:39:32.2546427+00:00`, `server: Kestrel`, and `transfer-encoding: chunked`. At the bottom, there is a table of responses with a `200` status code and a description of `Success`. The media type is set to `text/plain; v=1.0`. An example value is shown in the JSON format.

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

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

### Postman

The Postman interface shows a GET request to `https://fakereapi.azurewebsites.net/api/v1/Activities/3`. A red arrow points to the URL. The `Headers` tab is selected, showing a table with one header: `Accept` with the value `text/plain`. The `Body` tab is selected, showing the response in JSON format. A red arrow points to the `id` field in the JSON response. The response status is `200 OK` with a response time of `704 ms` and a response size of `277 B`. The response body is displayed in the `Body` tab, showing the JSON data.

```
GET https://fakereapi.azurewebsites.net/api/v1/Activities/3
```

Key	Value	Description
Accept	text/plain	

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

Activities\_API / Activities Save Share

GET ▼  Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers 6 hidden

	Key	Value	Description	***	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/>	Accept	text/plain				
	Key	Value	Description			

Body Cookies Headers (5) Test Results 200 OK 704 ms 277 B 🌐 📄 ⋮

	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

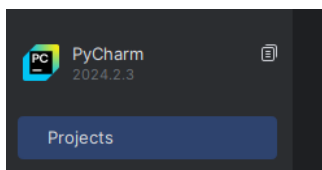
## 6. Determine Python version, install “requests” library

```
(.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
```

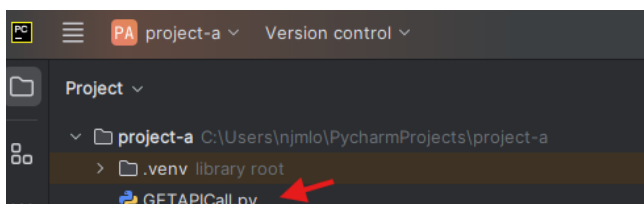
## 7. PyCharm Community Edition (IDE version at the time of documentation)

Download:

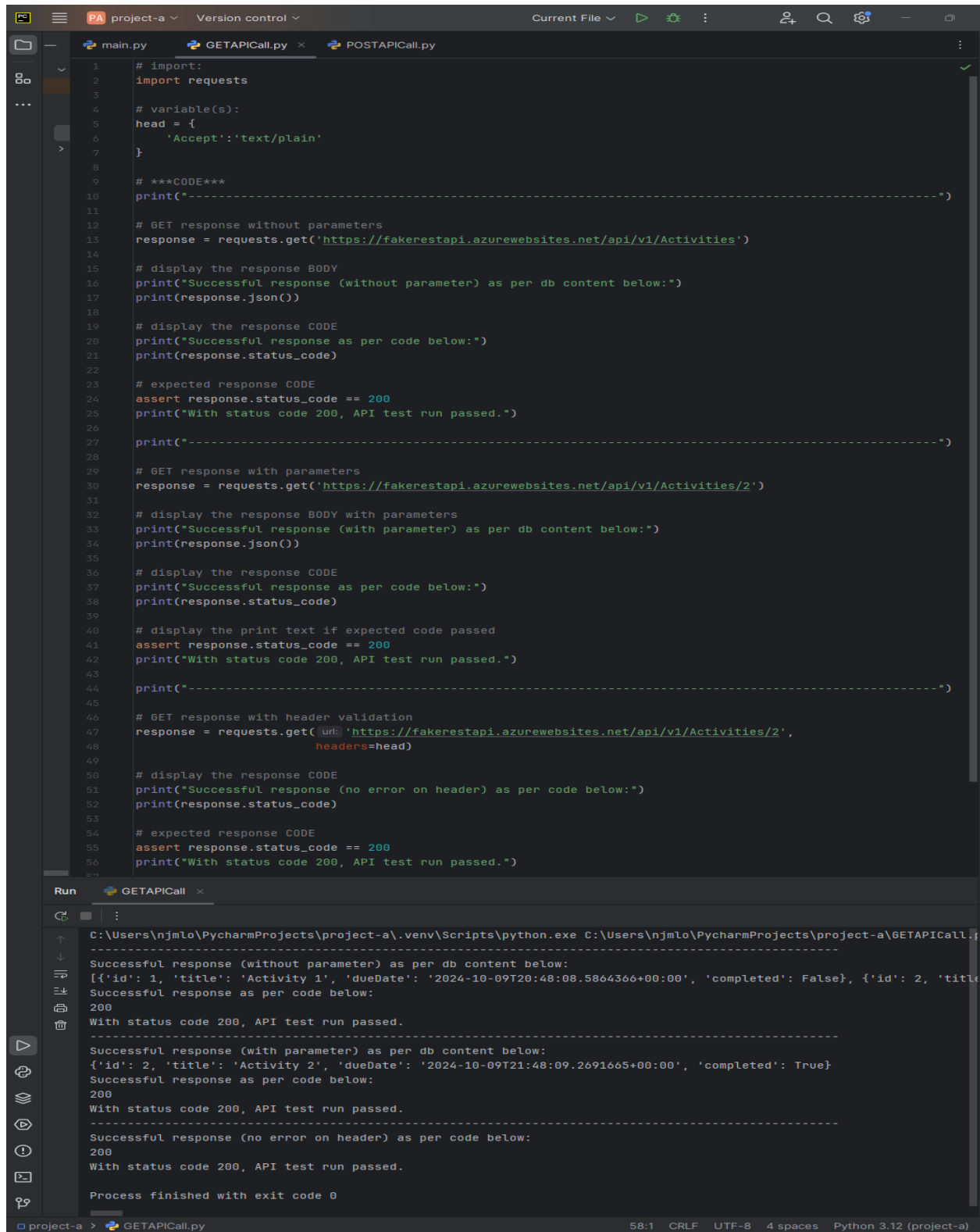
<https://www.jetbrains.com/pycharm/download>



## 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)



```
1 # import:
2 import requests
3
4 # variable(s):
5 head = {
6     'Accept': 'text/plain'
7 }
8
9 # ***CODE***
10 print("-----")
11
12 # GET response without parameters
13 response = requests.get('https://fakereestapi.azurewebsites.net/api/v1/Activities')
14
15 # display the response BODY
16 print("Successful response (without parameter) as per db content below:")
17 print(response.json())
18
19 # display the response CODE
20 print("Successful response as per code below:")
21 print(response.status_code)
22
23 # expected response CODE
24 assert response.status_code == 200
25 print("With status code 200, API test run passed.")
26
27 print("-----")
28
29 # GET response with parameters
30 response = requests.get('https://fakereestapi.azurewebsites.net/api/v1/Activities/2')
31
32 # display the response BODY with parameters
33 print("Successful response (with parameter) as per db content below:")
34 print(response.json())
35
36 # display the response CODE
37 print("Successful response as per code below:")
38 print(response.status_code)
39
40 # display the print text if expected code passed
41 assert response.status_code == 200
42 print("With status code 200, API test run passed.")
43
44 print("-----")
45
46 # GET response with header validation
47 response = requests.get(url='https://fakereestapi.azurewebsites.net/api/v1/Activities/2',
48                        headers=head)
49
50 # display the response CODE
51 print("Successful response (no error on header) as per code below:")
52 print(response.status_code)
53
54 # expected response CODE
55 assert response.status_code == 200
56 print("With status code 200, API test run passed.")
57
```

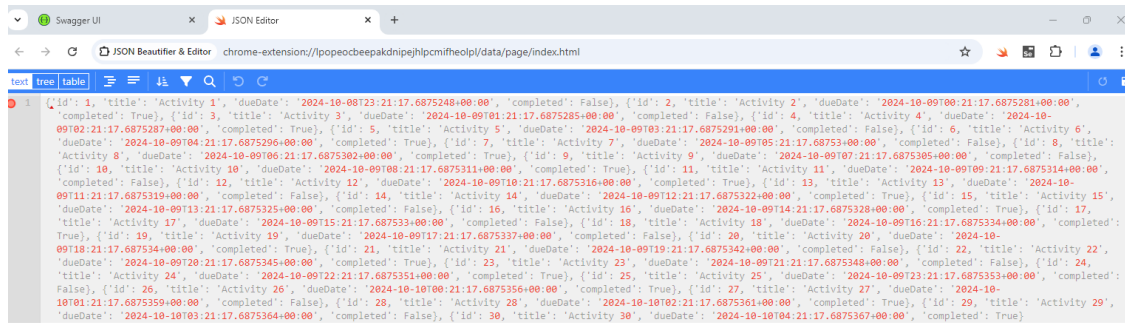
Run GETAPICall x

```
C:\Users\njm\lo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njm\lo\PycharmProjects\project-a\GETAPICall.py
-----
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': '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 (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
```

project-a > GETAPICall.py 58:1 CRLF UTF-8 4 spaces Python 3.12 (project-a)

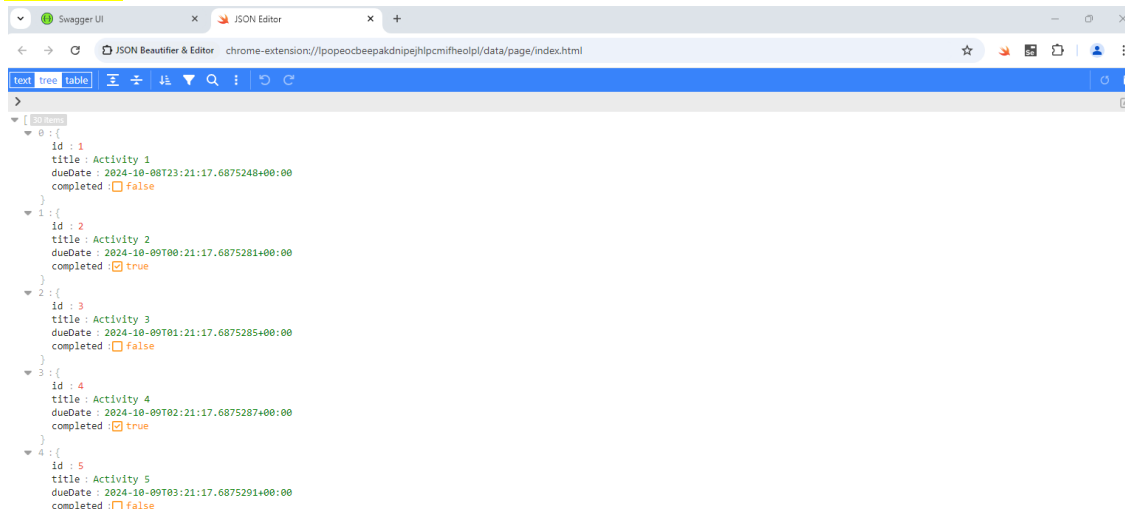
## 10. Using Json Beautifier (chrome add-on), display response result (copy and paste the results)

### Text view



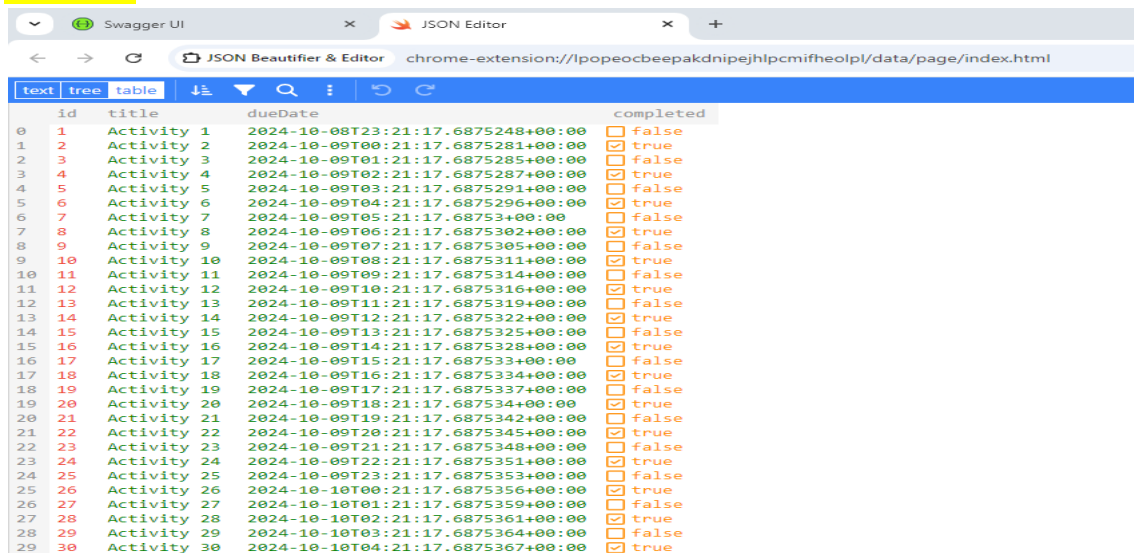
```
[{"id": 1, "title": "Activity 1", "dueDate": "2024-10-08T23:21:17.6875248+00:00", "completed": false}, {"id": 2, "title": "Activity 2", "dueDate": "2024-10-09T00:21:17.6875281+00:00", "completed": true}, {"id": 3, "title": "Activity 3", "dueDate": "2024-10-09T01:21:17.6875285+00:00", "completed": 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



```
{
  "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



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

Swagger  
Powered by SMARTBEAR

Select a definition V1

### FakeRESTApi.Web V1 V1 OAS3

/swagger/v1/swagger.json

#### Activities

GET /api/v1/Activities

POST /api/v1/Activities

Parameters Cancel

No parameters

Request body application/json; v=1.0

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

Execute Clear

#### Responses

Curl

```
curl -X POST "https://fakerestopi.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://fakerestopi.azurewebsites.net/api/v1/Activities

Server response

Code Details

200

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: Wed 09 Oct 2024 16:15:08 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": "2024-10-09T16:15:08.644Z",
  "completed": true
}
```

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

Validate and compare the following response results:

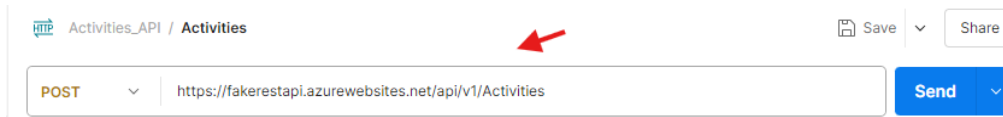
- Status
- Body
- Header
- Time

### Swagger

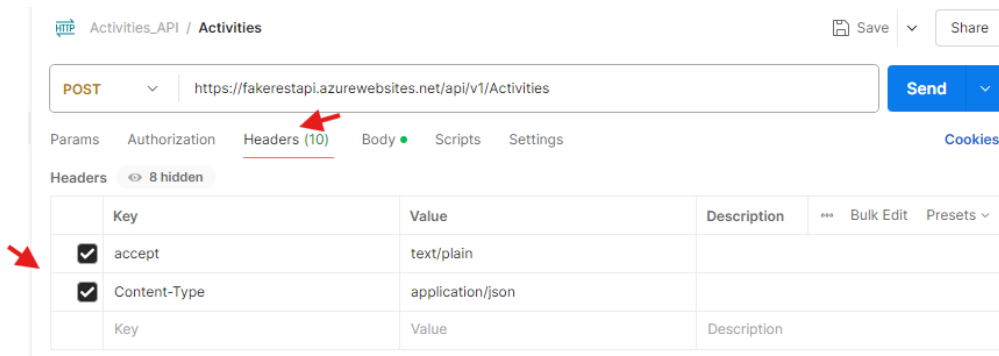


### Postman

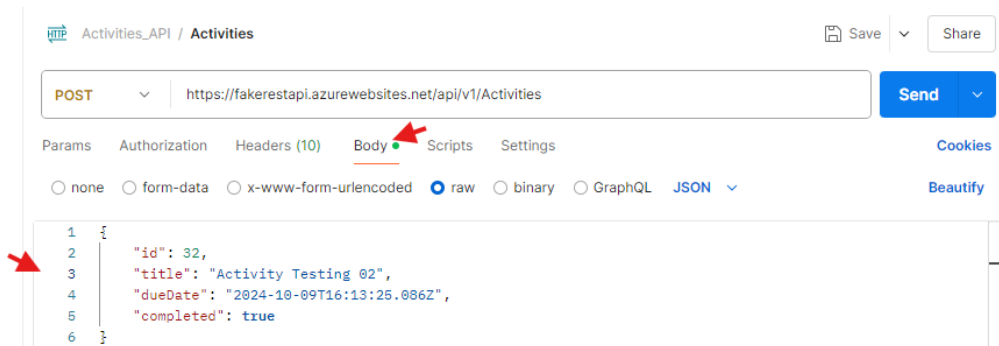
#### URL



#### Header

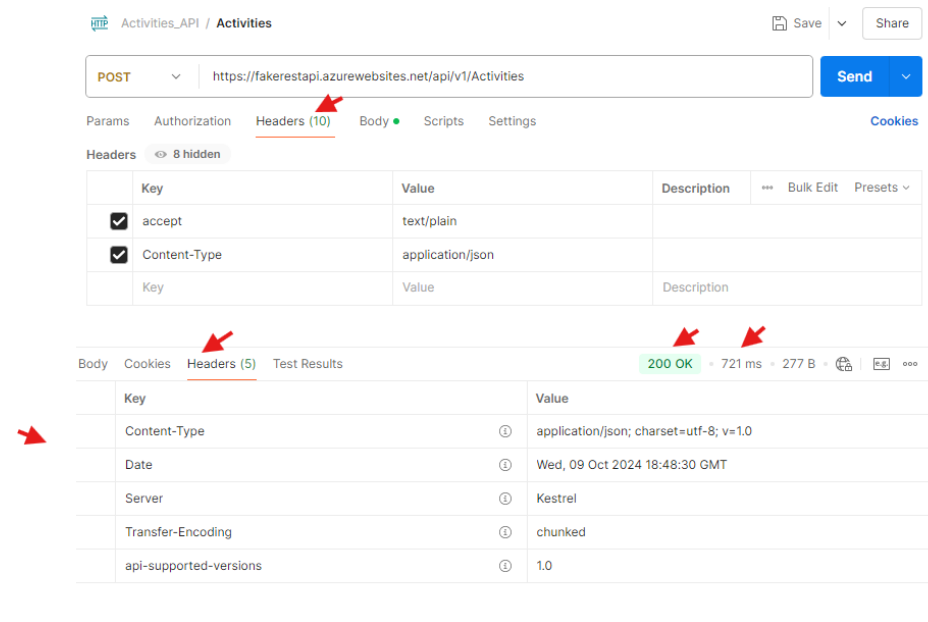


#### Body



## After Send...

### Header



Activities\_API / Activities

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

Send

Params Authorization Headers (10) Body Scripts Settings Cookies

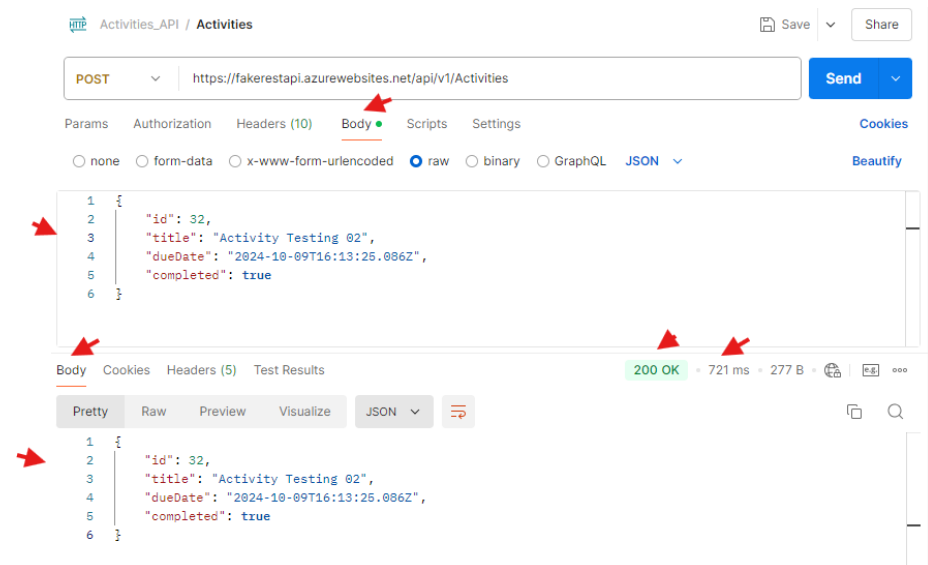
Headers 8 hidden

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

Body Cookies Headers (5) Test Results 200 OK • 721 ms • 277 B

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

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

Send

Params Authorization Headers (10) Body Scripts Settings Cookies

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

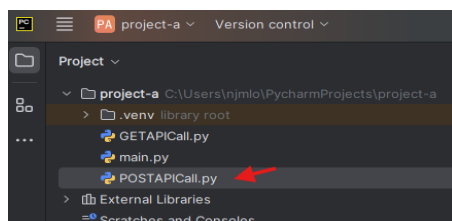
```
1 {
2   "id": 32,
3   "title": "Activity Testing 02",
4   "dueDate": "2024-10-09T16:13:25.086Z",
5   "completed": true
6 }
```

Body Cookies Headers (5) Test Results 200 OK • 721 ms • 277 B

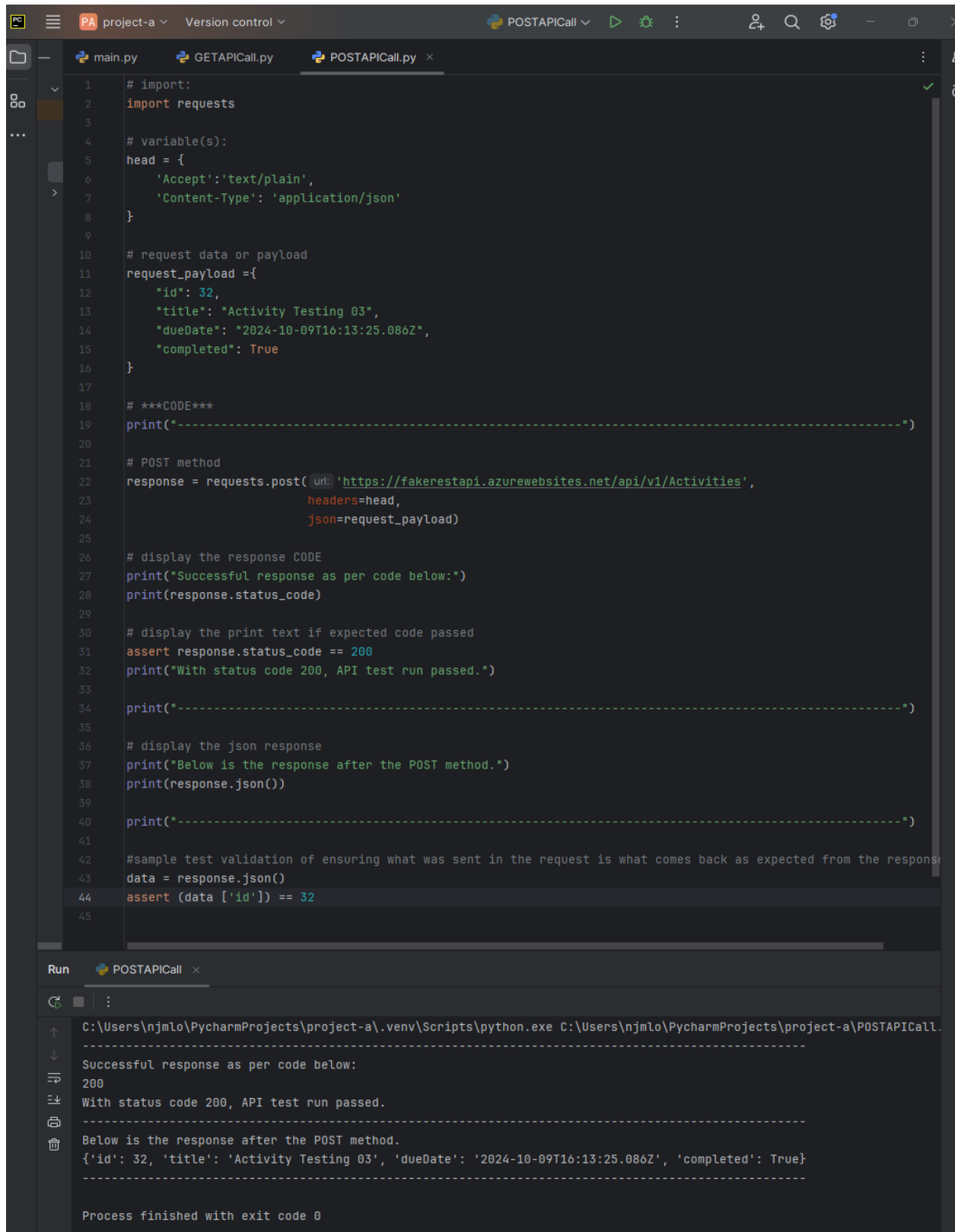
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 32,
3   "title": "Activity Testing 02",
4   "dueDate": "2024-10-09T16:13:25.086Z",
5   "completed": true
6 }
```

## 15. Create a python file in PyCharm editor



## 16. Sample POST (in PyCharm using Python code)



```
1  # import:
2  import requests
3
4  # variable(s):
5  head = {
6      'Accept': 'text/plain',
7      'Content-Type': 'application/json'
8  }
9
10 # request data or payload
11 request_payload = {
12     "id": 32,
13     "title": "Activity Testing 03",
14     "dueDate": "2024-10-09T16:13:25.086Z",
15     "completed": True
16 }
17
18 # ***CODE***
19 print("-----")
20
21 # POST method
22 response = requests.post(url='https://fakereastapi.azurewebsites.net/api/v1/Activities',
23                          headers=head,
24                          json=request_payload)
25
26 # display the response CODE
27 print("Successful response as per code below:")
28 print(response.status_code)
29
30 # display the print text if expected code passed
31 assert response.status_code == 200
32 print("With status code 200, API test run passed.")
33
34 print("-----")
35
36 # display the json response
37 print("Below is the response after the POST method.")
38 print(response.json())
39
40 print("-----")
41
42 #sample test validation of ensuring what was sent in the request is what comes back as expected from the response
43 data = response.json()
44 assert (data['id']) == 32
45
```

Run POSTAPICall ×

C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\POSTAPICall.py

-----

Successful response as per code below:

200

With status code 200, API test run passed.

-----

Below is the response after the POST method.

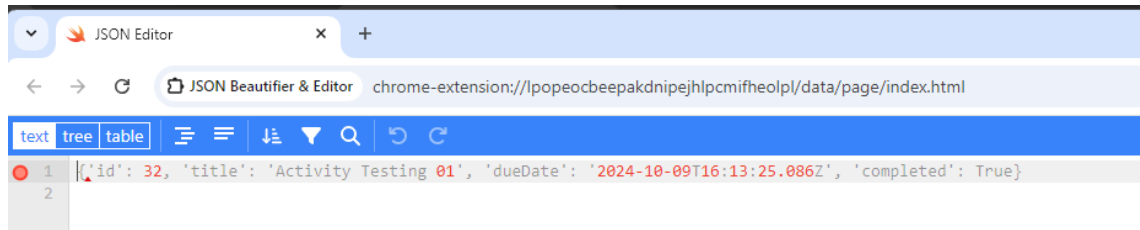
{'id': 32, 'title': 'Activity Testing 03', 'dueDate': '2024-10-09T16:13:25.086Z', 'completed': True}

-----

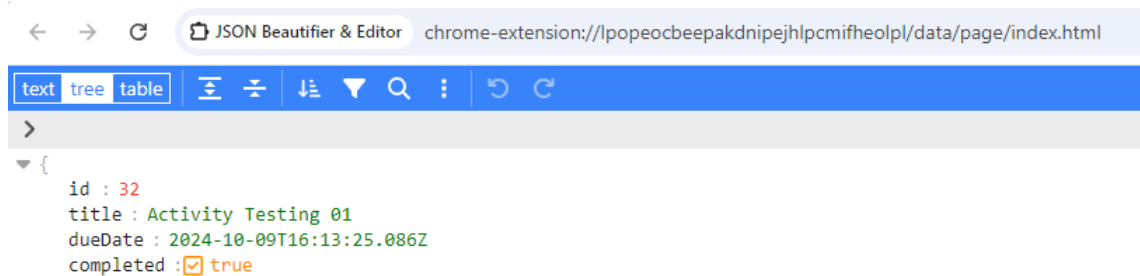
Process finished with exit code 0

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

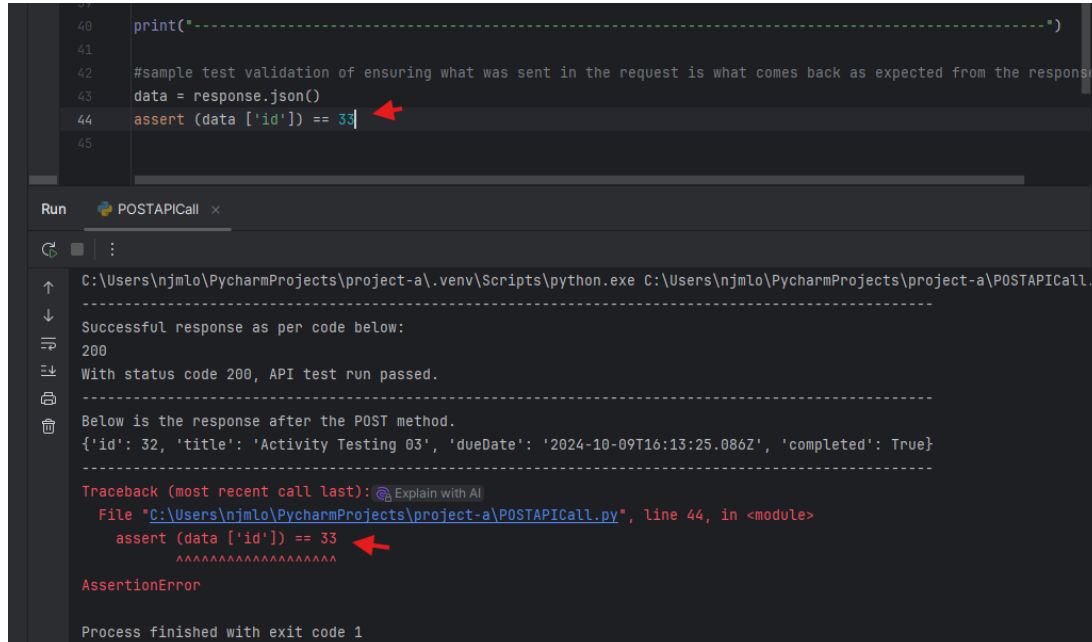
### Text view



### Tree view



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



## 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

The screenshot shows the SwaggerUI interface for the GET method at the endpoint `/api/v1/Activities/{id}`. The **Parameters** tab is active, showing a required path parameter `id` of type `integer(int32)` with the value `13`. The **Execute** button is highlighted. Below, the **Responses** tab shows a 200 status code with a response body containing JSON data: `{ "id": 13, "title": "Activity 13", "dueDate": "2024-10-20T10:23:56.604814+00:00", "completed": false }`. The response headers include `api-supported-versions: 1.0`, `content-type: application/json; charset=utf-8; v=1.0`, `date: Mon Oct 2024 21:23:55 GMT`, `server: Kestrel`, and `transfer-encoding: chunked`.

The screenshot shows the SwaggerUI interface for the PUT method at the endpoint `/api/v1/Activities/{id}`. The **Parameters** tab is active, showing a required path parameter `id` of type `integer(int32)` with the value `13`. The **Request body** tab is also active, showing a JSON payload: `{ "id": 14, "title": "Update title 01", "dueDate": "2024-10-09T21:23:06.1712", "completed": true }`. The **Execute** button is highlighted.

In SwaggerUI, after executing the method

The screenshot shows the SwaggerUI interface for the PUT method at the endpoint `/api/v1/Activities/{id}` after execution. The **Parameters** tab is active, showing a required path parameter `id` of type `integer(int32)` with the value `13`. The **Request body** tab is also active, showing the same JSON payload as before. The **Execute** button is highlighted. Below, the **Responses** tab shows a 200 status code with a response body containing JSON data: `{ "id": 14, "title": "Update title 01", "dueDate": "2024-10-09T21:23:06.1712", "completed": true }`. The response headers include `Access-Control-Allow-Origin: *`, `api-supported-versions: 1.0`, `content-type: application/json; charset=utf-8; v=1.0`, `date: Mon Oct 2024 21:26:38 GMT`, `server: Kestrel`, and `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**

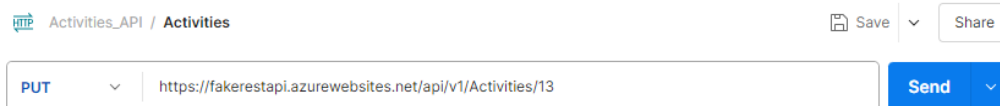
### Swagger

```
Curl
curl -X PUT "https://fakereapi.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}'

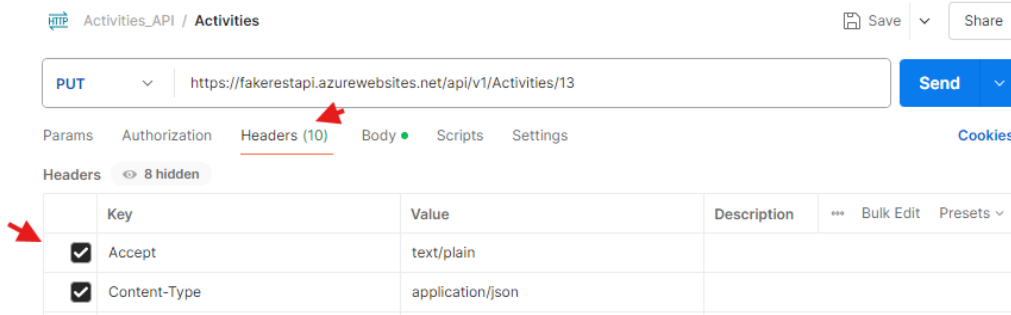
Request URL
https://fakereapi.azurewebsites.net/api/v1/Activities/13
```

### Postman

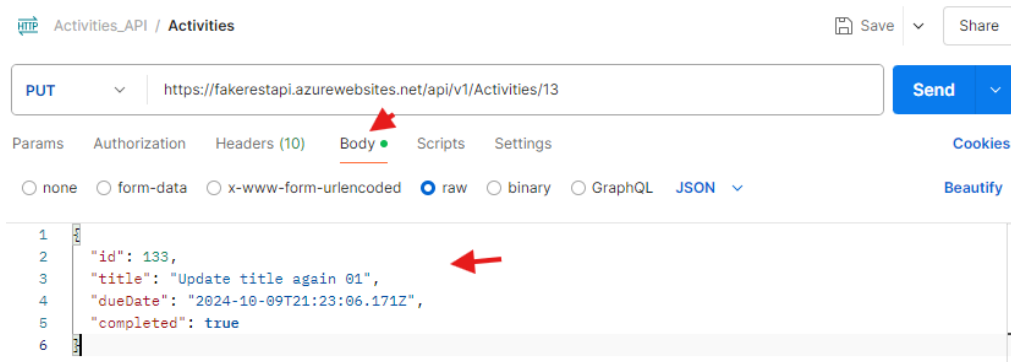
#### URL



#### Header



#### Body



After Send...

Header

HTTP

Activities\_API / Activities

Save

Share

PUT

https://fakerestdapi.azurewebsites.net/api/v1/Activities/13

Send

Params

Authorization

Headers (10)

Body

Scripts

Settings

Cookies

Headers

8 hidden

	Key	Value	Description		Bulk Edit	Presets
<input checked="" type="checkbox"/>	Accept	text/plain				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Body

Cookies

Headers (5)

Test Results

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

HTTP

Activities\_API / Activities

Save

Share

PUT

https://fakerestdapi.azurewebsites.net/api/v1/Activities/13

Send

Params

Authorization

Headers (10)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautiful

1

{

2

"id": 133,

3

"title": "Update title again 01",

4

"dueDate": "2024-10-09T21:23:06.171Z",

5

"completed": true

6

}

Body

Cookies

Headers (5)

Test Results

200 OK

738 ms

280 B

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"id": 133,

3

"title": "Update title again 01",

4

"dueDate": "2024-10-09T21:23:06.171Z",

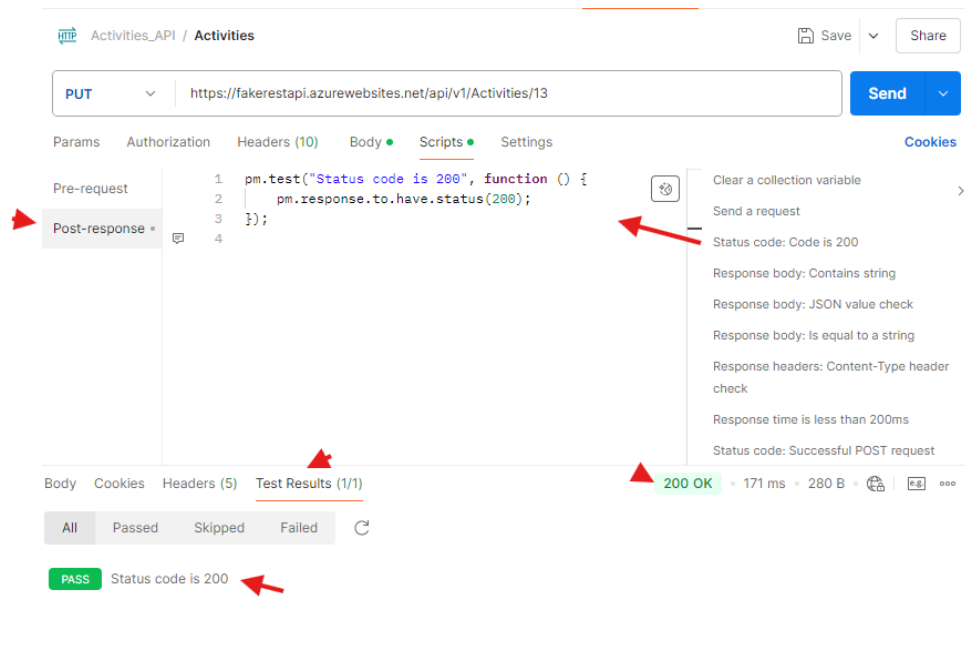
5

"completed": true

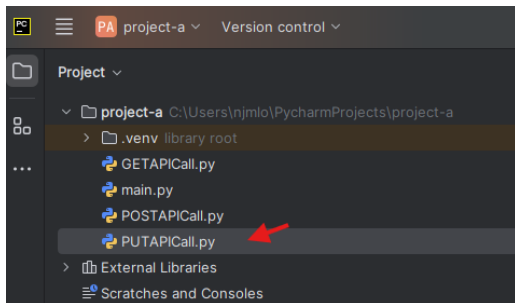
6

}

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



## 23. Create a python file in PyCharm editor



## 24. Sample PUT (in PyCharm using Python code)

```
1  # import:
2  import requests
3
4  # variable(s):
5  # header for GET method
6  headGET = {
7      'Accept': 'text/plain'
8  }
9
10 # header for PUT method
11 headPUT = {
12     'Accept': 'text/plain',
13     'Content-Type': 'application/json'
14 }
15
16 # request data or payload
17 put_payload = {
18     "id": 155,
19     "title": "Update Title 155",
20     "dueDate": "2024-10-09T16:13:25.086Z",
21     "completed": True
22 }
23
24 # ***CODE***
25 print("-----")
26
27 # GET response with parameters
28 response = requests.get(url='https://fakereapi.azurewebsites.net/api/v1/Activities/12',
29                          headers=headGET)
30
31 # display the response BODY with parameters
32 print('Before update...')
33 print("Successful response (with parameter) as per db content below:")
34 print(response.json())
35
36 # display the response CODE
37 print("Successful response as per code below:")
38 print(response.status_code)
39
40 # display the print text if expected code passed
41 assert response.status_code == 200
42 print("With status code 200, API test run passed.")
43
44 print("-----")
45
46 # PUT response with parameters
47 responsePUT = requests.put(url='https://fakereapi.azurewebsites.net/api/v1/Activities/12',
48                            headers=headPUT,
49                            json=put_payload)
50
51 # display the response BODY after the update
52 print('After update...')
53 print("Successful response (after update) as per db content below:")
54 print(responsePUT.json())
55
56 # display the response CODE
57 print("Successful response as per code below:")
58 print(responsePUT.status_code)
59
60 # display the print text if expected code passed
61 assert responsePUT.status_code == 200
62 print("With status code 200, API test run passed.")
63
64 print("-----")
65
66
```

```
Run PUTAPICall x
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\PUTAPICall.py
-----
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(responsePUT.status_code)

# display the print text if expected code passed
assert responsePUT.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

The screenshot shows the Go REST API interface. At the top, there's a navigation bar with 'Go REST', 'Chandamama Stories', 'Help', 'Rest Console', 'Tools', and 'Howdy!'. Below this, the main heading is 'GraphQL and REST API for Testing and Prototyping' with a subtitle 'fake data | real responses | 24/7 online'. The interface is divided into two main sections: 'Resources' and 'Trying it Out'. The 'Resources' section lists four endpoints with their respective HTTP methods and descriptions. The 'Trying it Out' section shows the same endpoints with their corresponding REST client actions. Below these, there's a 'Nested Resources' section with more endpoints. At the bottom, there's a 'GraphQL Endpoint' section with links to the GraphQL API and schema.

Resources	
<a href="https://gorest.co.in/public/v2/users">https://gorest.co.in/public/v2/users</a>	2810
<a href="https://gorest.co.in/public/v2/posts">https://gorest.co.in/public/v2/posts</a>	2883
<a href="https://gorest.co.in/public/v2/comments">https://gorest.co.in/public/v2/comments</a>	2878
<a href="https://gorest.co.in/public/v2/todos">https://gorest.co.in/public/v2/todos</a>	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

- 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 GraphQL schema](#)

The screenshot shows the REST client interface with the URL 'gorest.co.in/public/v2/users'. The response is a JSON array of user objects. The first user is 'The Hon. Aanandini Varrier' with ID 7464998, email 'varrier\_aanandinii\_the\_hon@mohre.test', gender 'female', and status 'inactive'. The second user is 'Samir Gandhi' with ID 7463045, email 'gandhi\_samir@ledner.example', gender 'male', and status 'inactive'. The third user is 'Adhiraj Pothuvaal' with ID 7463044, email 'adhiraj\_pothuvaal@gislason.test', gender 'male', and status 'inactive'. The fourth user is 'Chandramohan Varma' with ID 7463043, email 'varma\_chandramohan@russel-smith.test', gender 'male', and status 'inactive'. The fifth user is 'Prof. Rati Shah' with ID 7463042, email 'shah\_prof\_rati@kunze-kovacek.test', gender 'female', and status 'active'.

```
{
  "id": 7464998,
  "name": "The Hon. Aanandini 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"
},
}
```

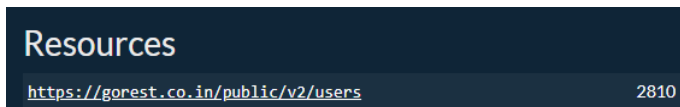


## 27. Sample POST with OAuth2 bearer key (copy the sampling url from Go REST and into Postman)

Validate and compare the following response results:

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

### Go REST

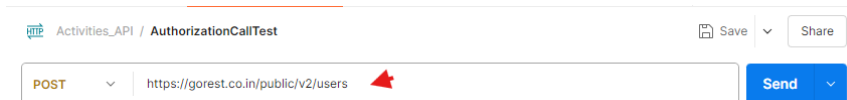


### Sample Github token

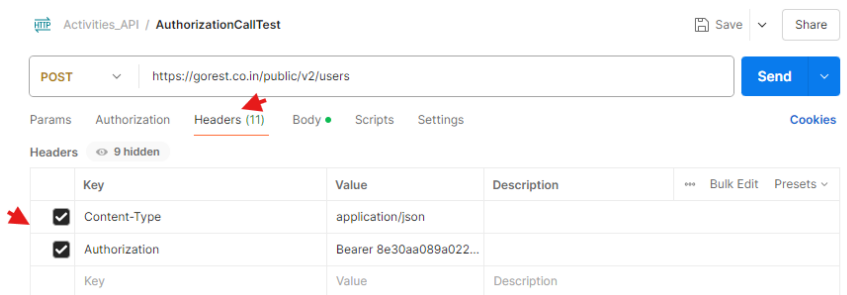


### Postman

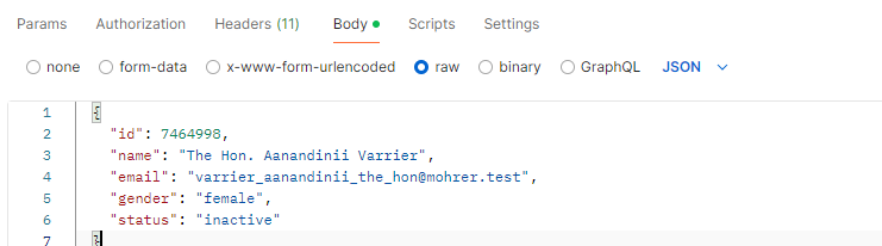
#### URL



#### Header



#### Body



## After Send...

### Header

The screenshot shows the 'Header' tab of a REST client interface. At the top, the request method is 'POST' and the URL is 'https://gorest.co.in/public/v2/users'. Below this, the 'Headers' tab is selected, displaying a table of request headers. The table has columns for 'Key', 'Value', and 'Description'. Two headers are listed: 'Content-Type' with value 'application/json' and 'Authorization' with value 'Bearer 8e30aa089a022...'. Below the request headers, the 'Response Headers' tab is selected, showing a table with 'Key' and 'Value' columns. The response status is '201 Created' with a response time of '1135 ms' and a size of '1.2 KB'. The response headers include 'Date', 'Content-Type', 'Content-Length', 'Connection', 'Cache-Control', 'etag', 'location', 'referrer-policy', 'vary', 'x-content-type-options', 'x-download-options', 'x-frame-options', 'x-permitted-cross-domain-policies', and 'x-ratelimit-limit'.

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

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

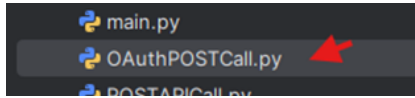
The screenshot shows the 'Body' tab of the REST client interface. The request method is 'POST' and the URL is 'https://gorest.co.in/public/v2/users'. The 'Body' tab is selected, showing the request body in JSON format. The response status is '201 Created' with a response time of '1135 ms' and a size of '1.2 KB'. The response body is also in JSON format, showing the created user details.

```
1 {
2   "id": 7464998,
3   "name": "The Hon. Aanandinii Varrier",
4   "email": "varrier_aanandinii_the_hon@mohzer.test",
5   "gender": "female",
6   "status": "inactive"
7 }
```

```
1 {
2   "id": 7465032,
3   "name": "The Hon. Aanandinii Varrier",
4   "email": "varrier_aanandinii_the_hon@mohzer.test",
5   "gender": "female",
6   "status": "inactive"
7 }
```

## 28. Create a python file in PyCharm editor



## 29. Sample POST (in PyCharm using Python code)

```
1  # import:
2  import requests
3
4  # variable(s):
5  head = {
6      'Content-Type': 'application/json',
7      'Authorization': 'Bearer 8e30aa089'
8  }
9
10 # request data or payload
11 request_payload = {
12     'id': 7464998,
13     'name': 'The Hon. Aanandinii Varrier',
14     'email': 'varrier_aanandinii_the_hon@mohreR.test',
15     'gender': 'female',
16     'status': 'inactive'
17 }
18
19 # store the url in a variable
20 url = "https://qonest.co.in/public/v2/users"
21
22 # ***CODE***
23 print("-----")
24
25 # POST method
26 postResponse = requests.post(url, headers=head, json=request_payload)
27
28 # display the response CODE
29 print("Successful response as per code below:")
30 print(postResponse.status_code)
31
32 # display the print text if expected code passed
33 assert postResponse.status_code == 201
34 print("With status code 201, API test run passed.")
35
36 print("-----")
37
38 # display the json response
39 print("Below is the response after the POST method.")
40 print(postResponse.json())
41
42 print("-----")
43
44 # run a GET method to validate the POST method
45 getResponse = requests.get(url + '/' + str(postResponse.json()['id']), headers=head)
46
47 # display the json response
48 print("Below is to check the data have been added after the POST method.")
49 print(getResponse.json())
```

Run OAuthPOSTCall

```
C:\Users\njmlo\PycharmProjects\project-a\.venv\Scripts\python.exe C:\Users\njmlo\PycharmProjects\project-a\OAuthPOSTCall.py
-----
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

The screenshot shows the Go REST website interface. At the top, there's a navigation bar with links to Chandamama Stories, Help, Rest Console, Tools, and Howdy!. The main heading is "GraphQL and REST API for Testing and Prototyping" with a subtitle "fake data | real responses | 24/7 online". Below this, there are two main sections: "Resources" and "Trying it Out".

**Resources**

URL	Status
<a href="https://gorest.co.in/public/v2/users">https://gorest.co.in/public/v2/users</a>	2810
<a href="https://gorest.co.in/public/v2/posts">https://gorest.co.in/public/v2/posts</a>	2883
<a href="https://gorest.co.in/public/v2/comments">https://gorest.co.in/public/v2/comments</a>	2878
<a href="https://gorest.co.in/public/v2/todos">https://gorest.co.in/public/v2/todos</a>	1455

**Trying it Out**

Method	URL	Action
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**

Method	URL	Action
GET	/public/v2/users/6941918/posts	Retrieves user posts
POST	/public/v2/users/6941918/posts	Creates a user post
GET	/public/v2/posts/6941918/comments	Retrieves post comments
POST	/public/v2/posts/6941918/comments	Creates a post comment
GET	/public/v2/users/6941918/todos	Retrieves user todos
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**

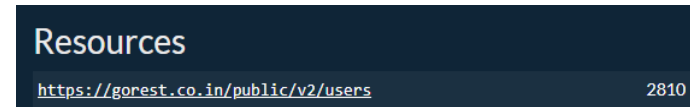
- GraphQL API is available at <https://gorest.co.in/public/v2/graphql>
- [View GraphQL json schema](#)
- [View GraphQL schema](#)

The screenshot shows a REST client interface with the URL `gorest.co.in/public/v2/users` and the "pretty-print" checkbox checked. The response is a JSON array of user objects, each with fields: id, name, email, gender, and status.

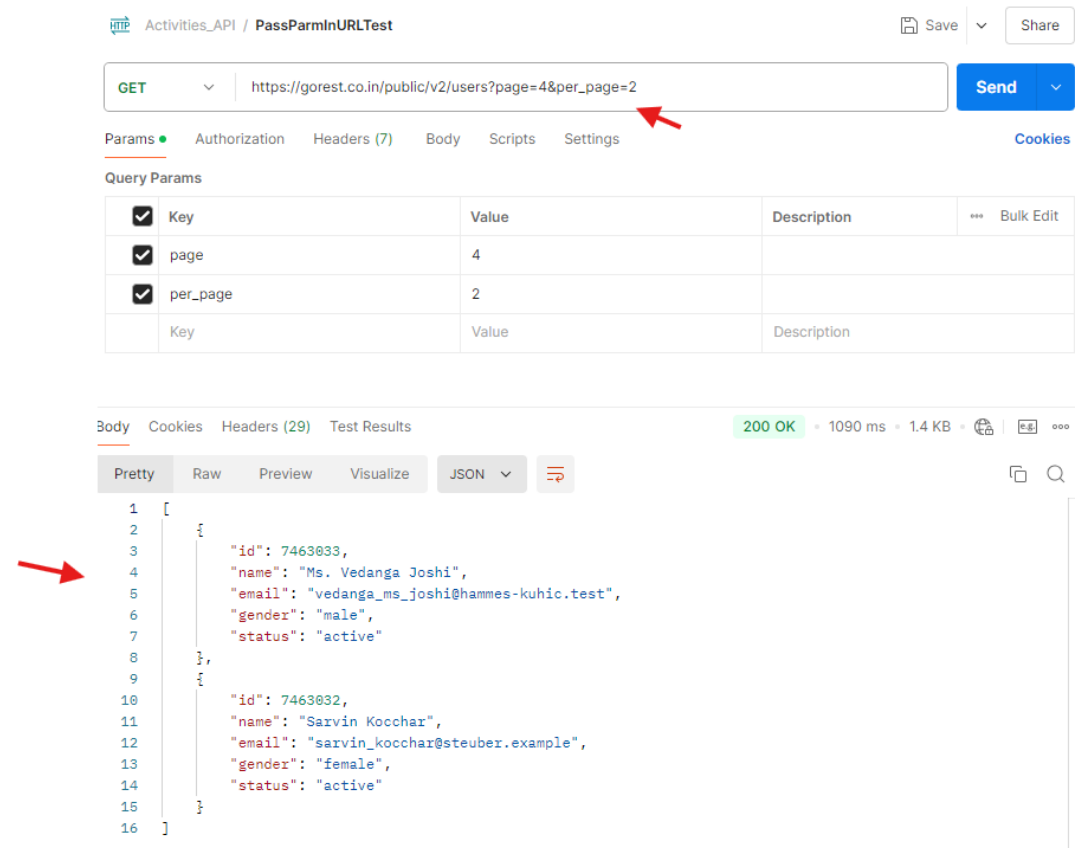
```
{
  "id": 7464998,
  "name": "The Hon. Aanandini Varrier",
  "email": "varrier_aanandinii_the_hon@mohre.test",
  "gender": "female",
  "status": "inactive"
},
{
  "id": 7463045,
  "name": "Samir Gandhi",
  "email": "gandhi_samir@ledner.example",
  "gender": "male",
  "status": "inactive"
},
{
  "id": 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))

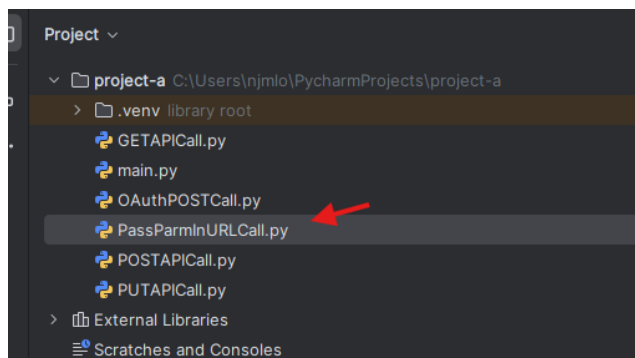
### Go REST



### Postman



## 33. Create a python file in PyCharm editor



## 34. Sample GET (in PyCharm using Python code)

```
PassParamURLCall.py x
1 # import:
2 import requests
3
4 # variable(s):
5
6 # store the url in a variable
7 url = "https://gorest.co.in/public/v2/users"
8
9 # parameters to be passed and part of the request
10 parm = {
11     'page': 1,
12     'per_page': 2
13 }
14
15 # ***CODE***
16 print("-----")
17
18 # GET method
19 getResponse = requests.get(url, params=parm)
20
21 # display the response CODE
22 print("Successful response as per code below:")
23 print(getResponse.status_code)
24
25 # display the print text if expected code passed
26 assert getResponse.status_code == 200
27 print("With status code 200, API test run passed.")
28
29 print("-----")
30
31 # display the json response
32 print("Below is the response after the POST method.")
33 print(getResponse.json())
```

```
Run PassParamURLCall
C:\Users\nejo\PycharmProjects\project-a\venv\Scripts\python.exe C:\Users\nejo\PycharmProjects\project-a\PassParamURLCall.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

```
# 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())
```

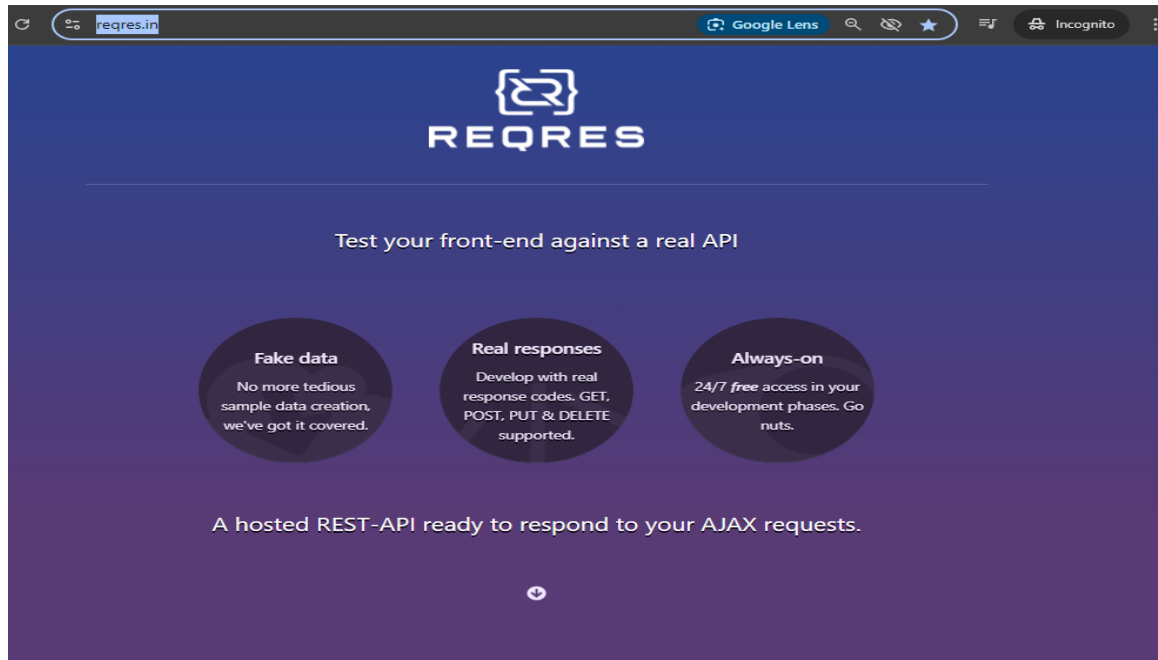


## 37. Passing payload from JSON file

url:

<https://reqres.in/>

REQRES sampling APIs



Give it a try

SUPPORT REQRES

The screenshot displays the Reqres API endpoint list. On the left, there's a table of endpoints with columns for HTTP method and description. A red arrow points to the "POST" method in the "CREATE" row. To the right of the table, there are two panels: "Request" and "Response". The "Request" panel shows a JSON payload for creating a user. The "Response" panel shows a 201 status code and a JSON response for the created user. A red arrow points to the "POST" method in the "CREATE" row of the table.

Method	Description
GET	LIST USERS
GET	SINGLE USER
GET	SINGLE USER NOT FOUND
GET	LIST <RESOURCE>
GET	SINGLE <RESOURCE>
GET	SINGLE <RESOURCE> NOT FOUND
POST	CREATE
PUT	UPDATE
PATCH	UPDATE
DELETE	DELETE
POST	REGISTER - SUCCESSFUL
POST	REGISTER - UNSUCCESSFUL
POST	LOGIN - SUCCESSFUL
POST	LOGIN - UNSUCCESSFUL
GET	DELAYED RESPONSE

**Request**  
/api/users  

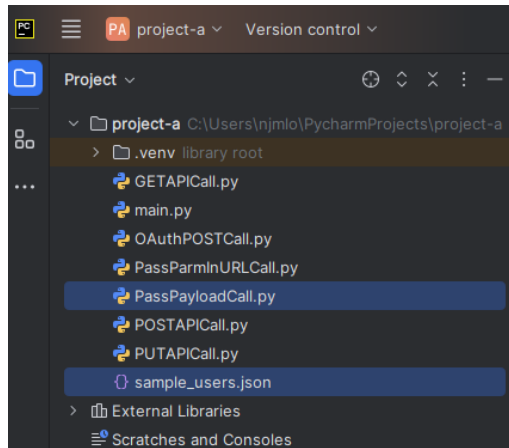
```
{  "name": "morpheus",  "job": "leader"}
```

**Response**  
201  

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

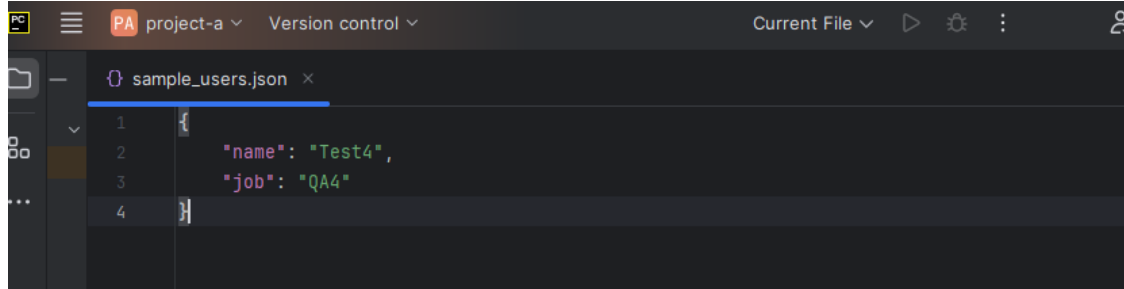
### 38. Create file(s) in PyCharm editor

Python and JSON file



### 39. Sample POST (in PyCharm using Python code)

JSON file



## Python file

```
1 import:
2 import requests
3 import json
4
5 variable(s):
6
7 store the url in a variable
8 url = "https://reqres.in/"
9
10 head = {
11     'Content-Type': 'application/json'
12 }
13
14 request data or payload from a JSON file
15 json_file = open('./sample_users.json')
16 json_load = json.load(json_file)
17
18 request data or payload
19 request_payload = {
20     "name": "Test",
21     "job": "QA"
22 }
23
24 ***CODE***
25 print("-----")
26
27 POST method
28 response = (requests.post(
29     url=url+"api/users",
30     headers=head,
31     data=json.dumps(json_load) # --this is for passing from other files (including json file)
32     # json=json_load --this is the cmd if only passing from a json file
33 ))
34
35 display the response CODE
36 print("Successful response as per code below:")
37 print(response.status_code)
38
39 display the print text if expected code passed
40 assert response.status_code == 201
41 print("With status code 201, API test run passed.")
42
43 print("-----")
44
45 display the json response
46 print("Below is the response after the POST method (JSON file display.)")
47 print(response.json())
48 print("Below is the response after the POST method (TEXT file display.)")
49 print(response.text)
50
```

```
Run PassPayloadCall x
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://regres.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)

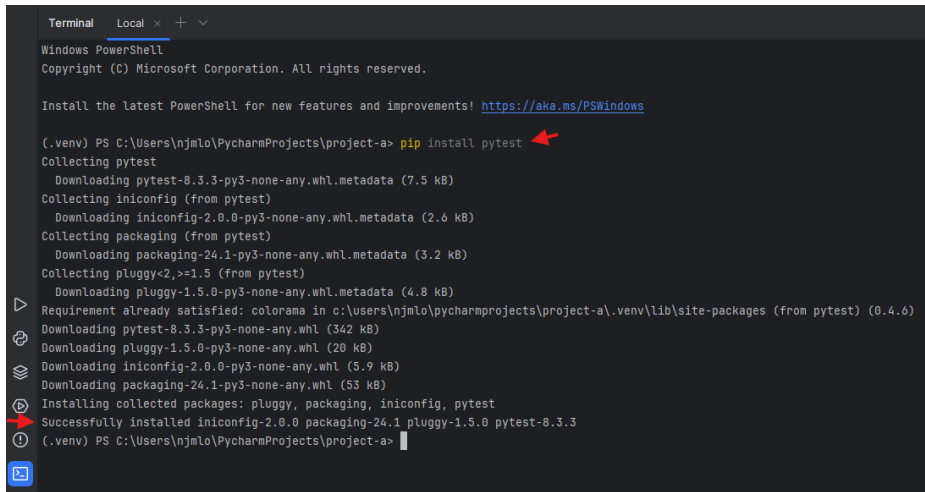
# display the print text if expected code passed
assert response.status_code == 201
print("With status code 201, API test run passed.")

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)

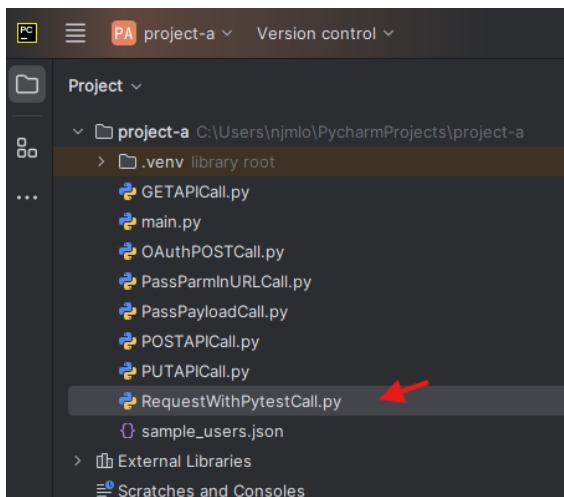


```
Terminal Local x + -
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

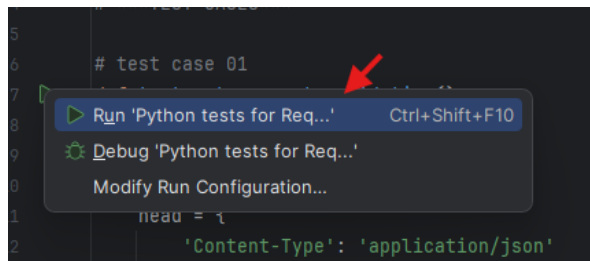


### 43. Sample GET (in PyCharm using Python code and PyTest to run the “test”)

#### Python file

```
RequestWithPytestCall.py x
1  # import:
2  import requests
3
4  # ***TEST CASES***
5
6  # test case 01
7  def test_get_request_validation():
8      # store the url in a variable
9      url = "https://reqres.in/"
10     # header(s)
11     head = {
12         'Content-Type': 'application/json'
13     }
14     # GET response
15     response = (requests.get(
16         url=str(url+ 'api/users/3'),
17         headers=head
18     ))
19
20     # display the response CODE
21     print("Successful response as per code below:")
22     print(response.status_code)
23
24     # display the print text if expected code passed
25     assert 200 == response.status_code
26     print("With status code 200, API test run passed.")
27
28     # display the json response
29     print("Below is the response after the POST method (JSON file) display.")
30     print(response.json())
31     print("Below is the response after the POST method (TEXT file) display.")
32     print(response.text)
```

#### PyTest test run



#### Response

```
Run Python tests for RequestWithPytestCall.py
Test Results 143ms
C:\Users\jela\PycharmProjects\project-a\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2024.2.3\plugins\python-ce\helpers\pycharm\jh_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\jela\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/1-image.jpg'}, 'support': {'url': 'https://reqres.in/support-heading', 'text': 'To keep ReqRes free, contributions towards server costs are appreciated!'}}
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/1-image.jpg'}, 'support': {'url': 'https://reqres.in/support-heading', 'text': 'To keep ReqRes free, contributions towards server costs are appreciated!'}}
===== 1 passed in 0.00s =====
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\*\*\*\*\*