📚 Code Examples: https://github.com/toliktemp/Postman\_Exmpl

Swagger: <https://rs-language-api.herokuapp.com/doc/#/Users/post_users>

Swagger Petstore: https://petstore.swagger.io/v2/swagger.json

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

📚 JSON Evaluator:

<https://jsonbeautifier.org/>

<https://jsonpath.com/>

https://jsoneditoronline.org/

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

200 (OK)

201 (Created - Создано)

202 (Accepted - Принято)

204 (No Content - Нет контента)

301 (Moved Permanently - Перемещено навсегда)

302 (Found - Найдено)

303 (See Other - Смотреть другое)

304 (Not Modified - Не изменен)

307 (Temporary Redirect - Временный редирект)

400 (Bad Request - Плохой запрос)

401 (Unauthorized - Неавторизован)

403 (Forbidden - Запрещено)

404 (Not Found - Не найдено)

405 (Method Not Allowed - Метод не разрешен)

406 (Not Acceptable - Неприемлемый)

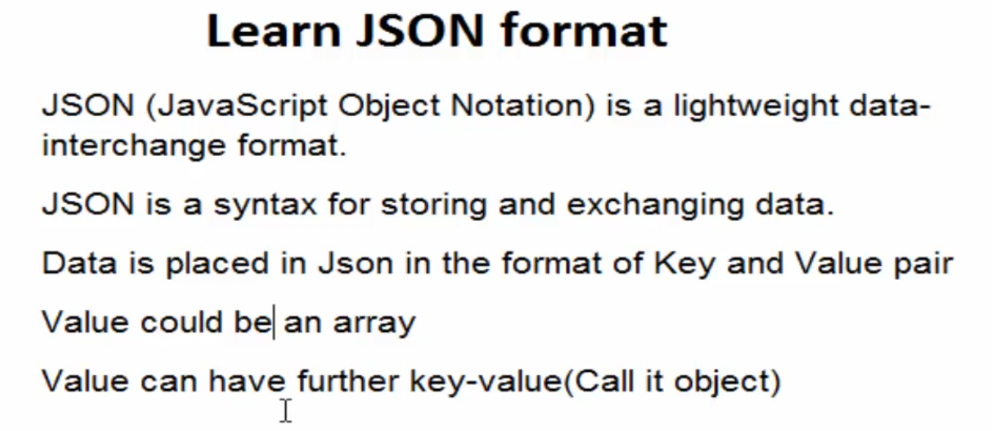
412 (Precondition Failed - Предусловие провалено)

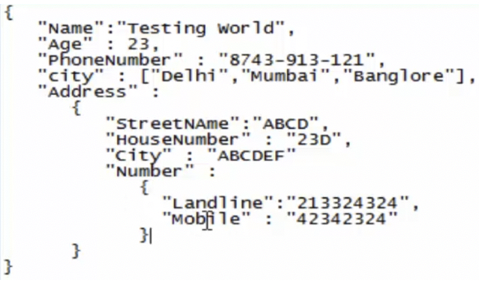
415 (Unsupported Media Type - Неподдерживаемый медиа тип)

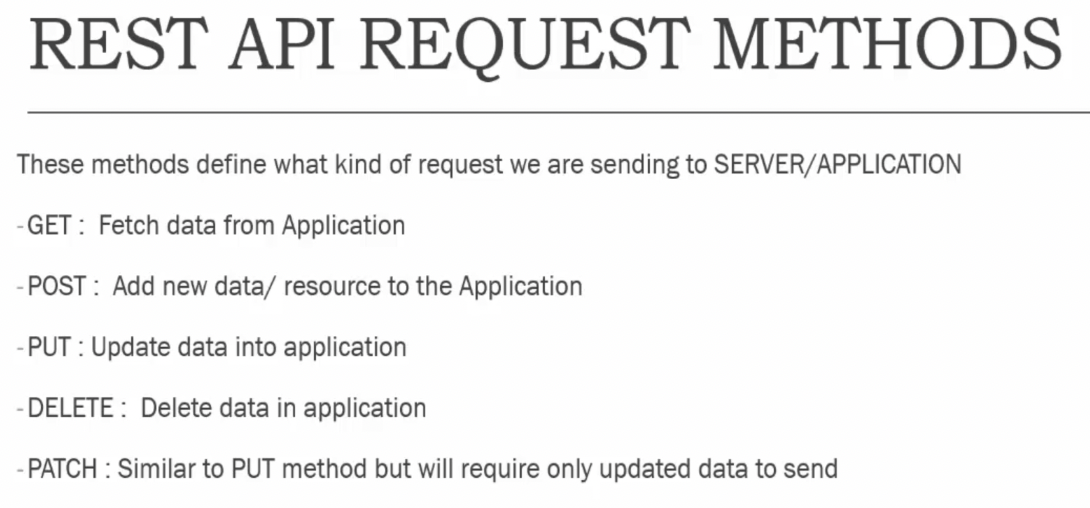
500 (Internal Server Error - Внутренняя ошибка сервера)

501 (Not Implemented - Не реализован)

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣



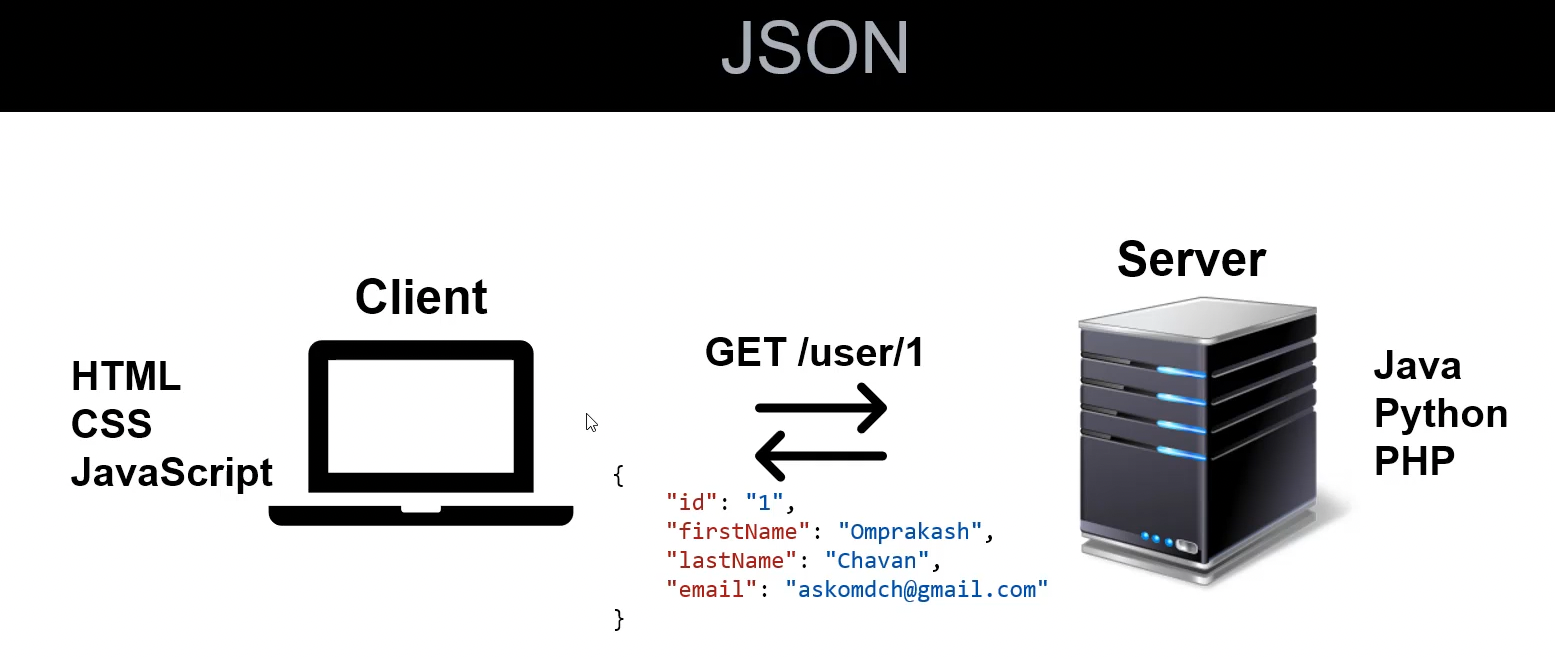


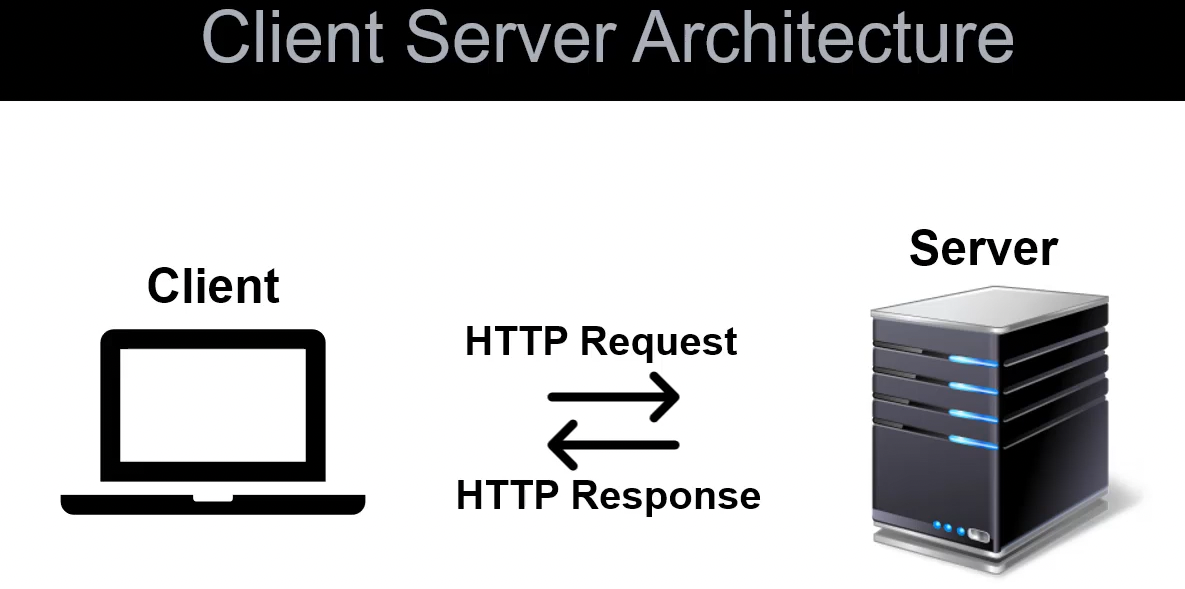
****

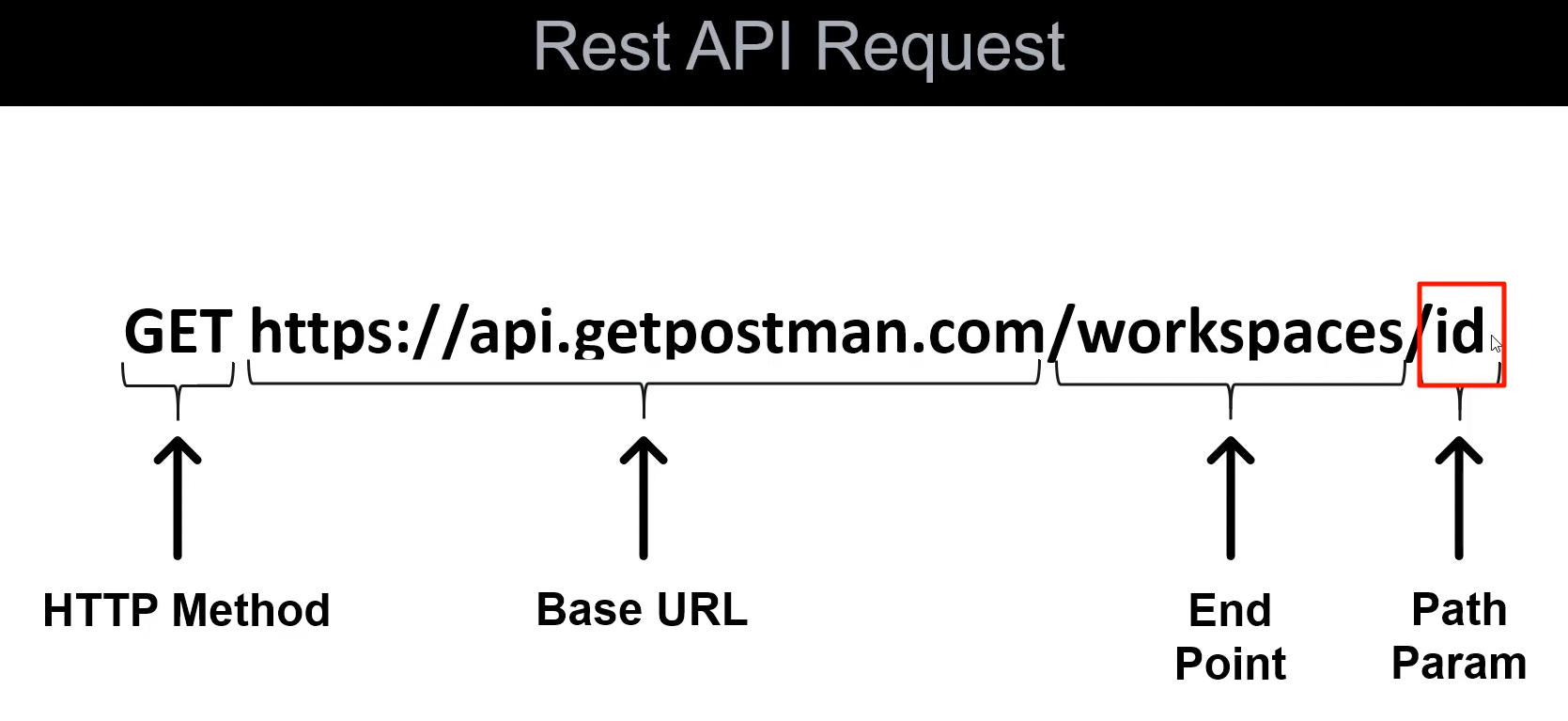
CRUD => Create / Insert : Post (insert) Read => Get (select)

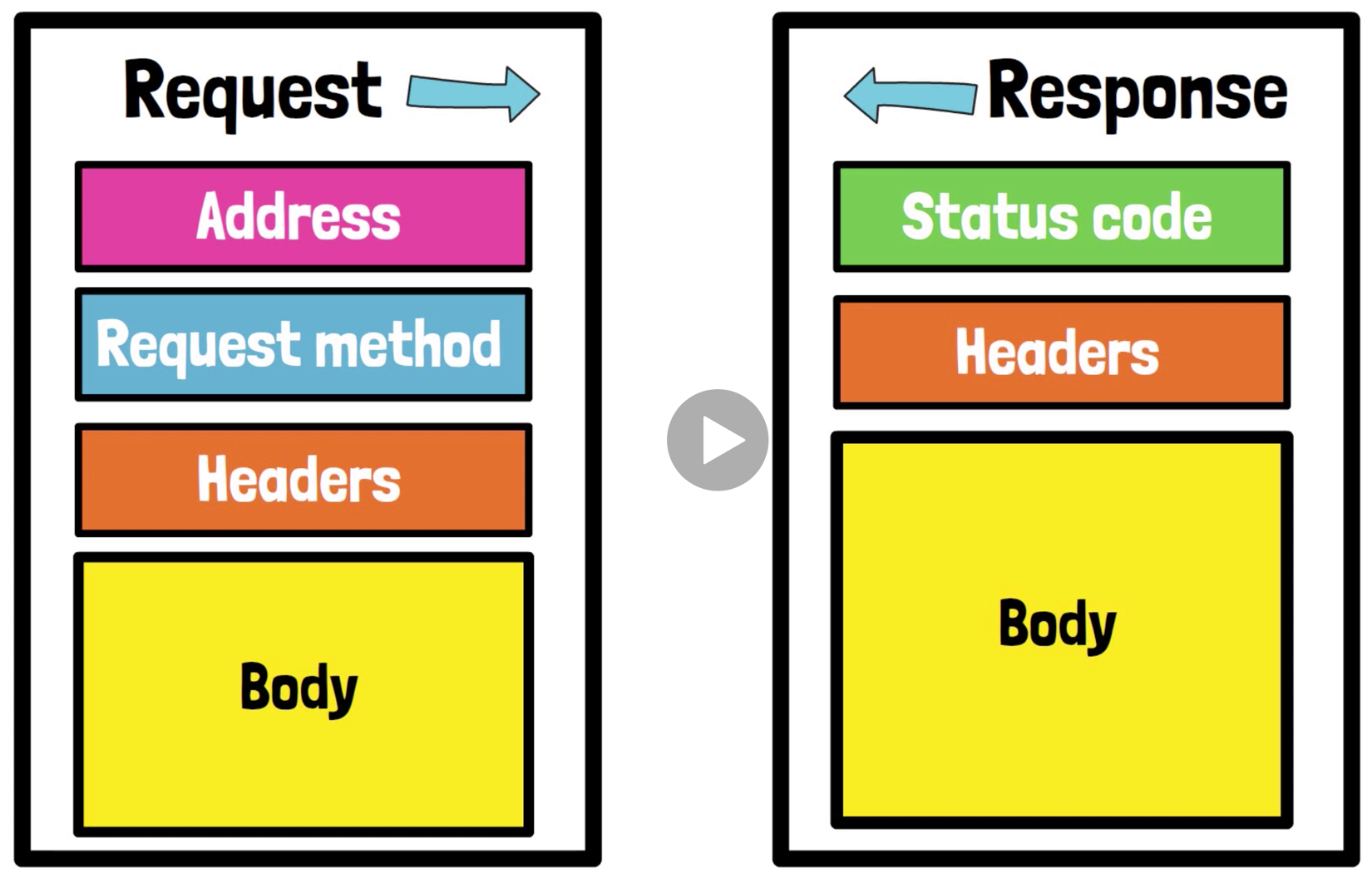
Update: Put (update) Delete (delete)

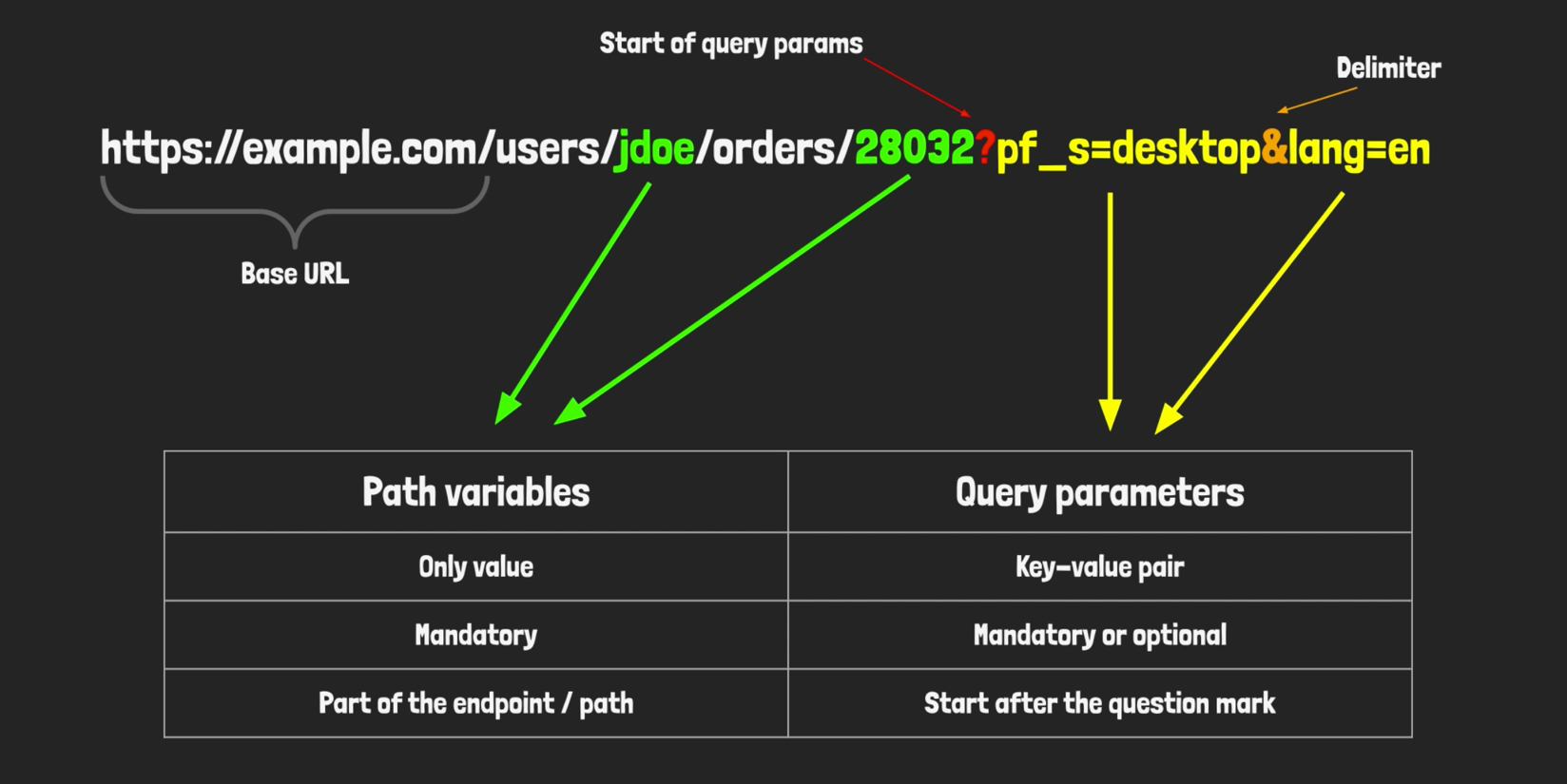
‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣











‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

API Testing

* Get Request
* Pick Response
* Display Response
* Validate Status Code
* Validate Response Content
* Fetch Cookies
* Fetch Elapsed Time

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

import requests  
  
# API URL  
url = "https://reqres.in/api/users?page=2"  
  
# Send Get Request  
response = requests.get(url)  
print(f"Response: => {response}\n")  
  
# Display Response Content  
print(f"Response Content: => {response.content}\n")  
print(f"Header Response: => {response.headers}")

# Validate Status Code  
print(f"Status Code: => {response.status\_code}")

assert response.status\_code == 200

# Fetch Response Header  
print(f"Header Response: => {response.headers}")  
print(f"Date: => {response.headers.get('Date')}")  
print(f"Server: => {response.headers.get('Server')}")

# Fetch Cookies  
print(f"Fetching Cookies: => {response.cookies}")

# Fetch Encoding  
print(f"Encoding: => {response.encoding}")

# Fetch Elapsed time  
print(f"Elapsed time: => {response.elapsed}")

Send Headers with Request

URL: "http://httpbin.org/get"

Sending Customized Headers =>

import requests

# Customized header  
header\_data = {'T1': 'First\_Header', 'T2': 'Second\_Header'}

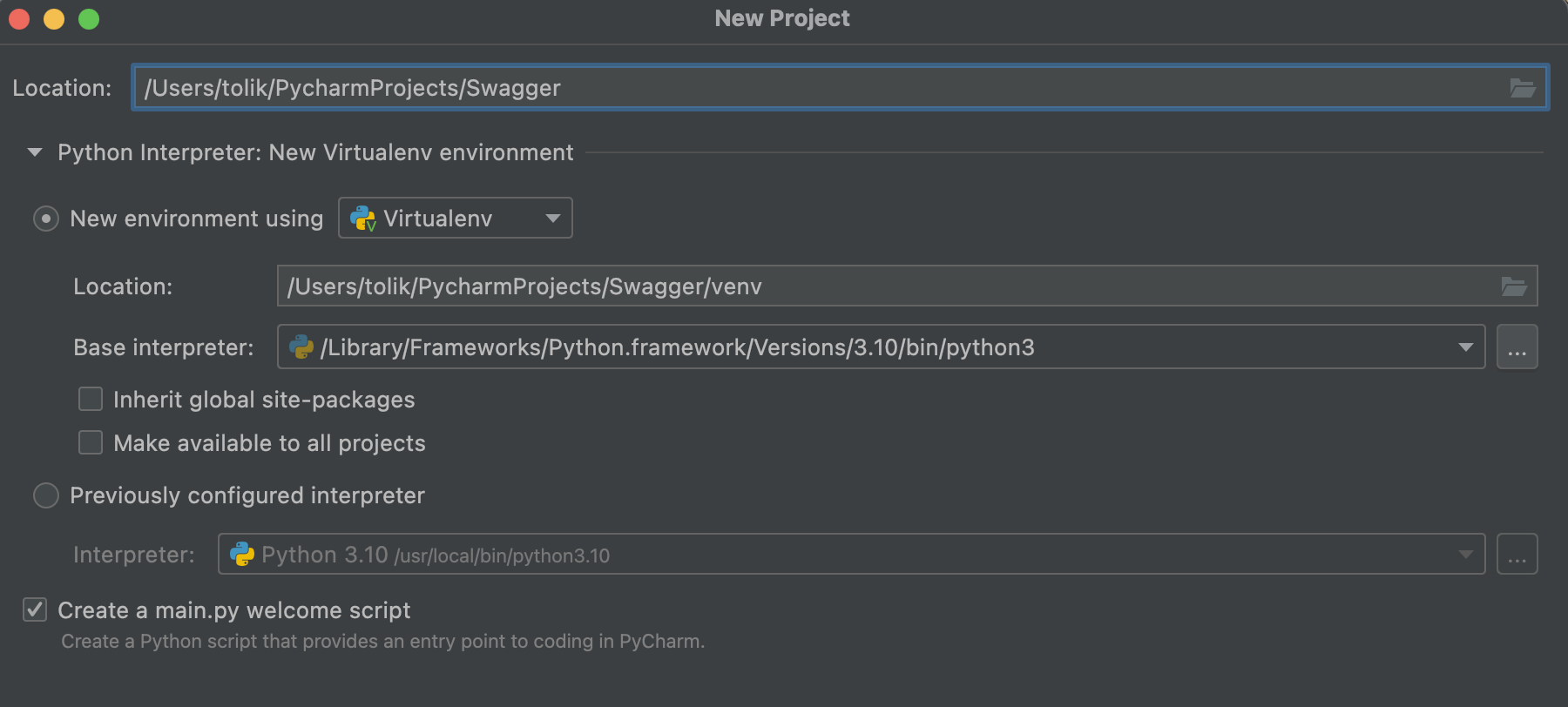
response = requests.get('http://httpbin.org/get', headers=header\_data)

print(response.text)

''' Output:  
{  
 "args": {},   
 "headers": {  
 "Accept": "\*/\*",   
 "Accept-Encoding": "gzip, deflate",   
 "Host": "httpbin.org",   
 "T1": "First\_Header",   
 "T2": "Second\_Header",   
 "User-Agent": "python-requests/2.28.1",   
 "X-Amzn-Trace-Id": "Root=1-6328f8fe-20701f847153dd4f6eade3d2"  
 },   
 "origin": "67.180.123.128",   
 "url": "http://httpbin.org/get"  
}  
'''

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

📚 Create project Swagger



Install the following packages: requests, configparser, pytest

* Create the folder <TestCases> in the root of the project

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

Method GET:

Gets a chunk of words

* **Example Value**
* Schema

[

{

"id": "string",

"group": 0,

"page": 0,

"word": "string",

"image": "string",

"audio": "string",

"audioMeaning": "string",

"audioExample": "string",

"textMeaning": "string",

"textExample": "string",

"transcription": "string",

"wordTranslate": "string",

"textMeaningTranslate": "string",

"textExampleTranslate": "string"

}

]

import json  
import requests  
  
  
def test\_get\_words():  
 url = "https://rs-language-api.herokuapp.com/words"  
 response = requests.get(url)  
 data = response.json()  
 print(f"\n\nStatus Response: {response.status\_code}")  
  
 print('\n\*\*\*\*\* Example of getting response\n')  
 print(response.content)  
 print('\ntext\n')  
 print(response.text)  
  
 print(f"Type: {type(data)}")  
 print(f"Length: {len(data)}")  
 print(f"1st record: {data[0]}")  
 print(f"2nd record: {data[1]}")  
 print(f"3rd record: {data[2]}")  
 print(f"20th record(last): {data[19]}")  
  
 print(f"Value of the first element from the list")  
 print(data[0]['word'])  
 print(data[0]['textExample'])  
  
 print(f"\n \*\*\*\*\* Get values of particular key in list of dictionaries \*\*\*\*\*")  
 key\_word = [key['word'] for key in data]  
 print(key\_word)  
  
 assert response.status\_code == 200  
 assert data[0]['word'] == 'enjoy'  
 assert (data[0]['textExample']) == "The woman <b>enjoys</b> riding her bicycle."

Gets a word with assets by id

/words/{id}

def test\_get\_words():  
  
 global word\_id  
 url = "https://rs-language-api.herokuapp.com/words"  
 response = requests.get(url)  
 data = response.json()  
 print(f"\n\nStatus Response: {response.status\_code}")  
  
   
 assert response.status\_code == 200  
 assert data[0]['word'] == 'enjoy'  
 assert (data[0]['textExample']) == "The woman <b>enjoys</b> riding her bicycle."  
 word\_id = data[0]['id']  
 # print(word\_id) # 5e9f5ee35eb9e72bc21af4aa  
  
  
def test\_get\_word\_by\_id():  
 url = "https://rs-language-api.herokuapp.com/words/"+str(word\_id)  
 response = requests.get(url)  
 data = response.json()  
 print(f"\nResponse Status Code: {response.status\_code}")  
 assert response.status\_code == 200  
 print(f"Data: {type(data)}") # class 'dict'  
 print(data)  
 assert data['id'] == '5e9f5ee35eb9e72bc21af4aa'  
 assert data['word'] == "enjoy"  
 assert data['image'] == "files/01\_0011.jpg"  
 assert data['textMeaning'] == "To <i>enjoy</i> is to like something."  
 assert data['textExample'] == "The woman <b>enjoys</b> riding her bicycle."  
 assert data['textExampleTranslate'] == "Женщина любит кататься на велосипеде"

‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣ ‣

import requests  
  
response = requests.get('http://216.10.245.166/Library/GetBook.php',  
 params={'AuthorName': 'Rahul Shetty'},)  
  
print(f"Response : {response.text}")

import json  
  
import requests  
  
response = requests.get('http://216.10.245.166/Library/GetBook.php',  
 params={'AuthorName': 'Rahul Shetty3'},)  
  
print(f"Response : {response.text}")

*# Response Output: [{"book\_name":"Jmeter","isbn":"Amit","aisle":"234"},{"book\_name":"Jmeter","isbn":"Sumit","aisle":"234"},{"book\_name":"Automation Everywhere - Siddhant","isbn":"test","aisle":"32659"},{"book\_name":"Automation Everywhere - Siddhant","isbn":"sicddh","aisle":"12345"},{"book\_name":"Automation Everywhere - Siddhant","isbn":"priyca","aisle":"65986"}]*print(type(response.text)) *# <class 'str'>*

*# First method to extract data*dict\_response = json.loads(response.text)  
*# print(type(dict\_response))  
# print(dict\_response[0]['isbn'])  
# print(dict\_response[1]['book\_name'])*

*# Second method to extract data*json\_response = response.json()  
print(type(json\_response)) *# <class 'list'>*print(json\_response[0]['isbn'])  
print(json\_response[0]['book\_name'])

def test\_signin\_with\_authentication():  
 auth\_token = 'swagger'  
 head = {'Authorization': 'Bearer ' + auth\_token}  
 response = requests.post("https://rs-language-api.herokuapp.com/signin",  
 auth=HTTPBasicAuth('doe@example.com', 'passw0rd'),  
 headers=head, )  
 print(response.headers.get('Server'))  
 print(response.headers.get('X-Frame-Options'))  
 print(response.headers.get('Strict-Transport-Security'))  
 print(response.headers.get('Content-Type'))  
 print(response.status\_code)  
 print(response.text)