REST APIs with Python & FastAPI
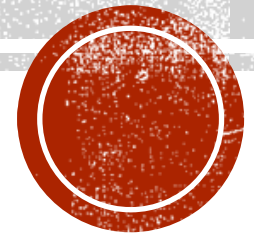
APIS, WHY ?
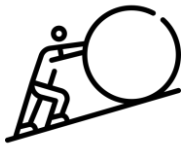

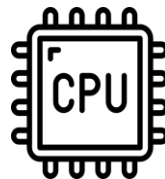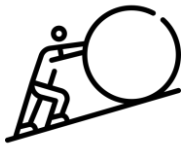API, You are a good man. Thank you

ISSUE

ISSUE
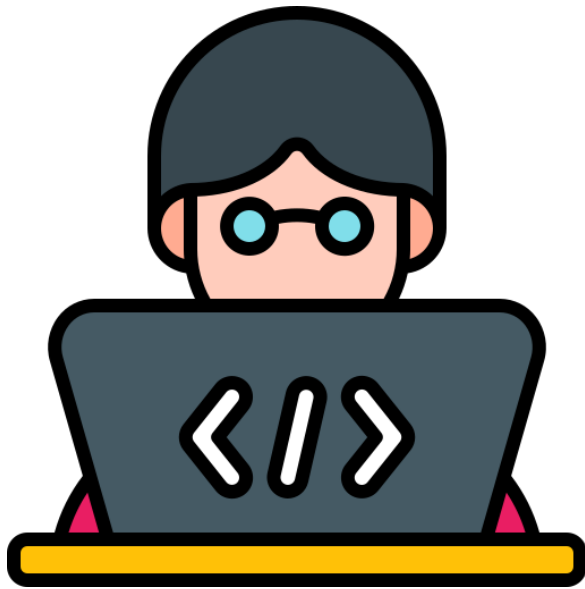
ISSUE

ISSUE

ISSUE

ISSUE

ISSUE
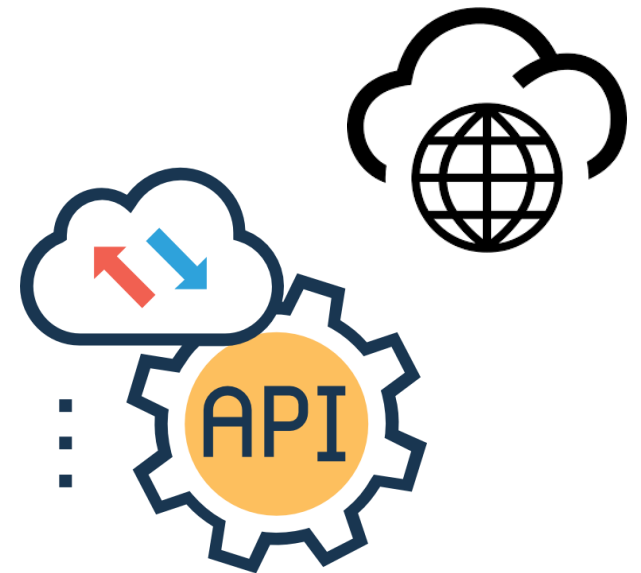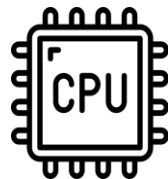
WHAT IS AN API?

- API is the acronym for Application Programming Interface. It is a software interface that allows two applications to interact with each other without any user intervention.

- APIs provides product or service to communicate with other products and services without having to know how they're implemented.

Client / Application

Requête

Réponse

API

Requête

Réponse

Base de données

Fonctionnalités et Données
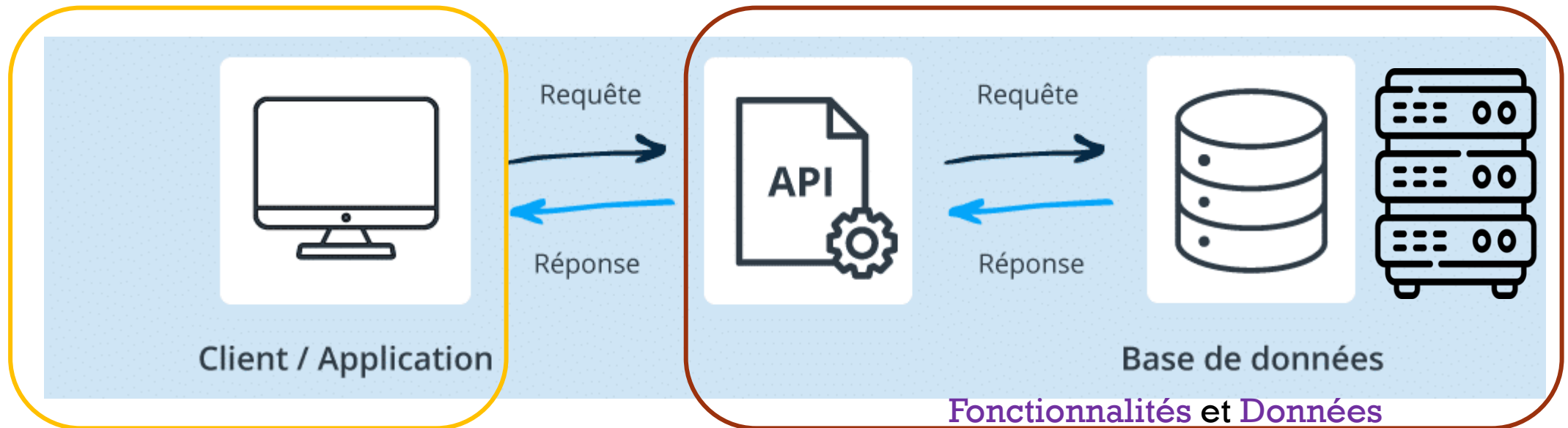
WHAT IS AN API?

REST API MODEL

GET | POST | PUT | DELETE

JSON | XML | HTML

CLIENT

API

HTTP REQUEST

HTTP RESPONSE
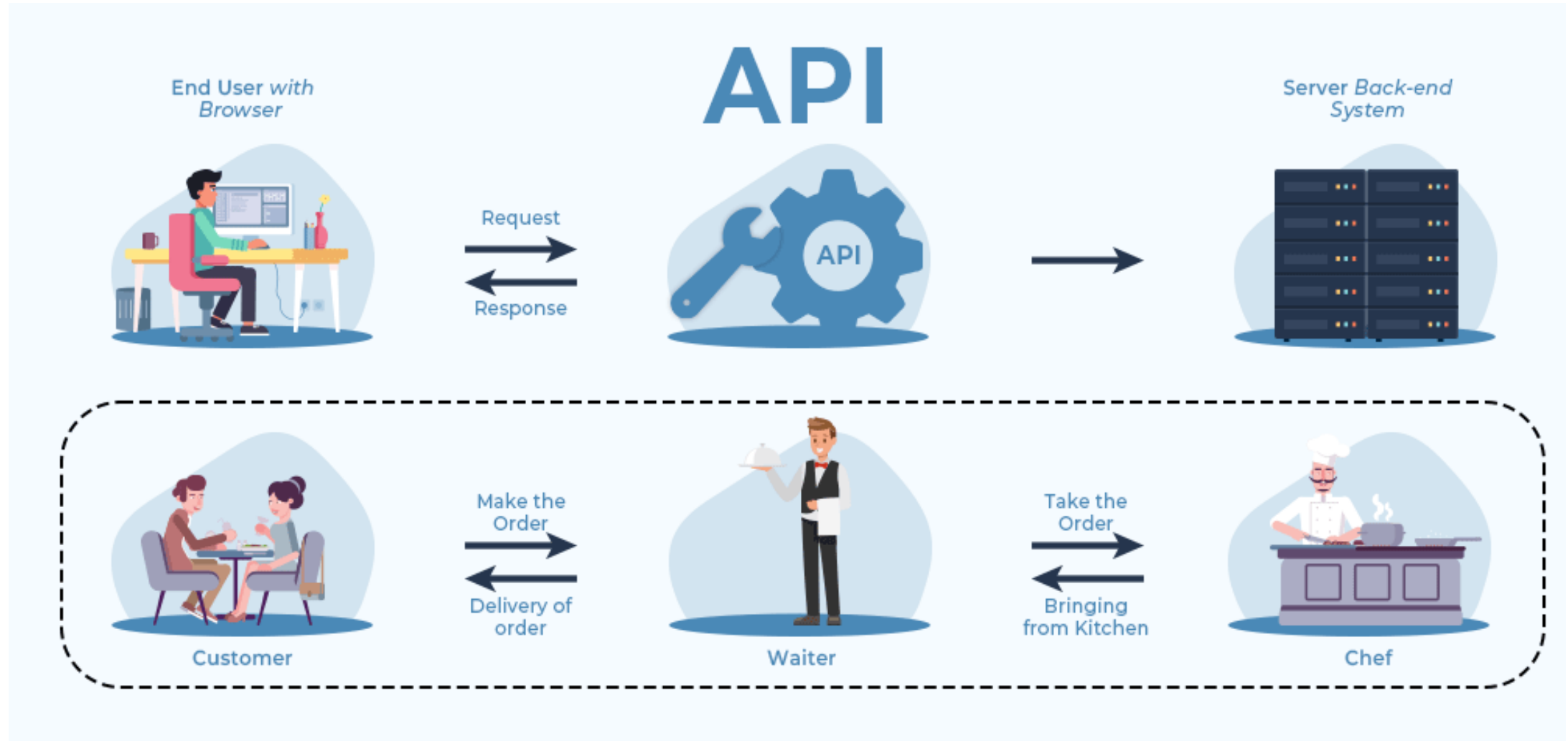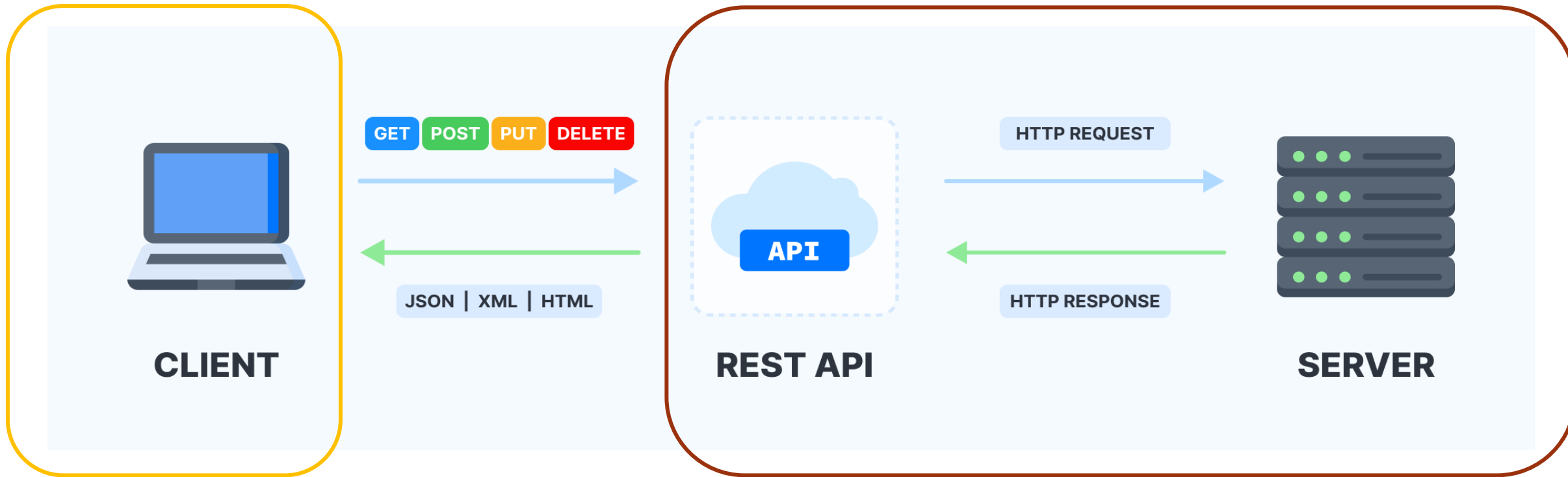
REST API

SERVER

client-server architecture

Data and **Functionality** are considered **resources** and are accessed using **Uniform Resource Identifiers** (**URIs endpoints**) and are developed as **functions**.

REST API MODEL

GET **POST** **PUT** **DELETE**

JSON | XML | HTML

CLIENT

API

REST API

HTTP REQUEST

HTTP RESPONSE

SERVER

C R U D

REST API MODEL

GET **POST** **PUT** **DELETE**

JSON | XML | HTML

CLIENT

API

HTTP REQUEST

HTTP RESPONSE

REST API

SERVER

JSON **XML**
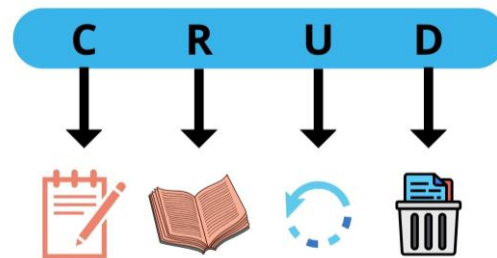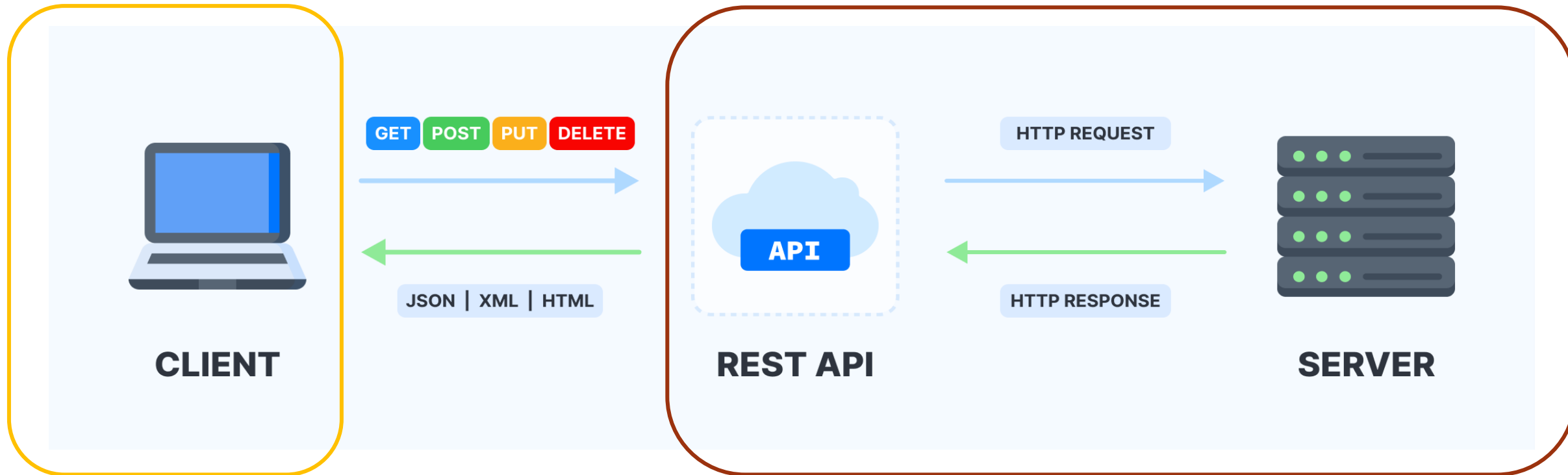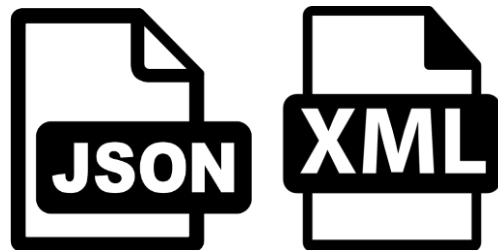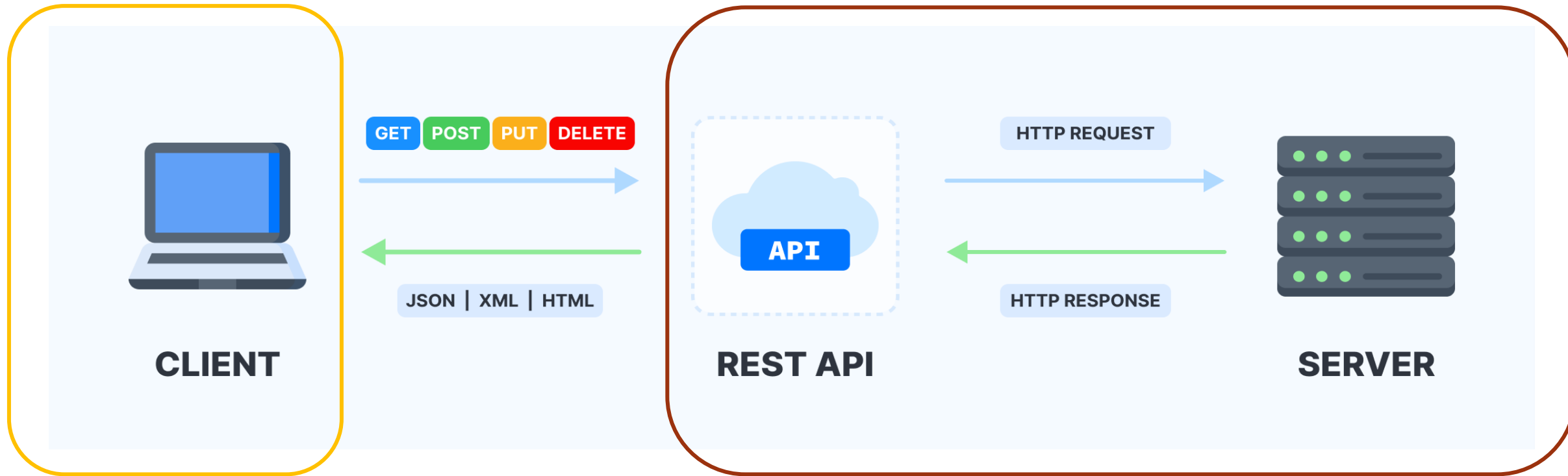
REST API MODEL

REST API MODEL
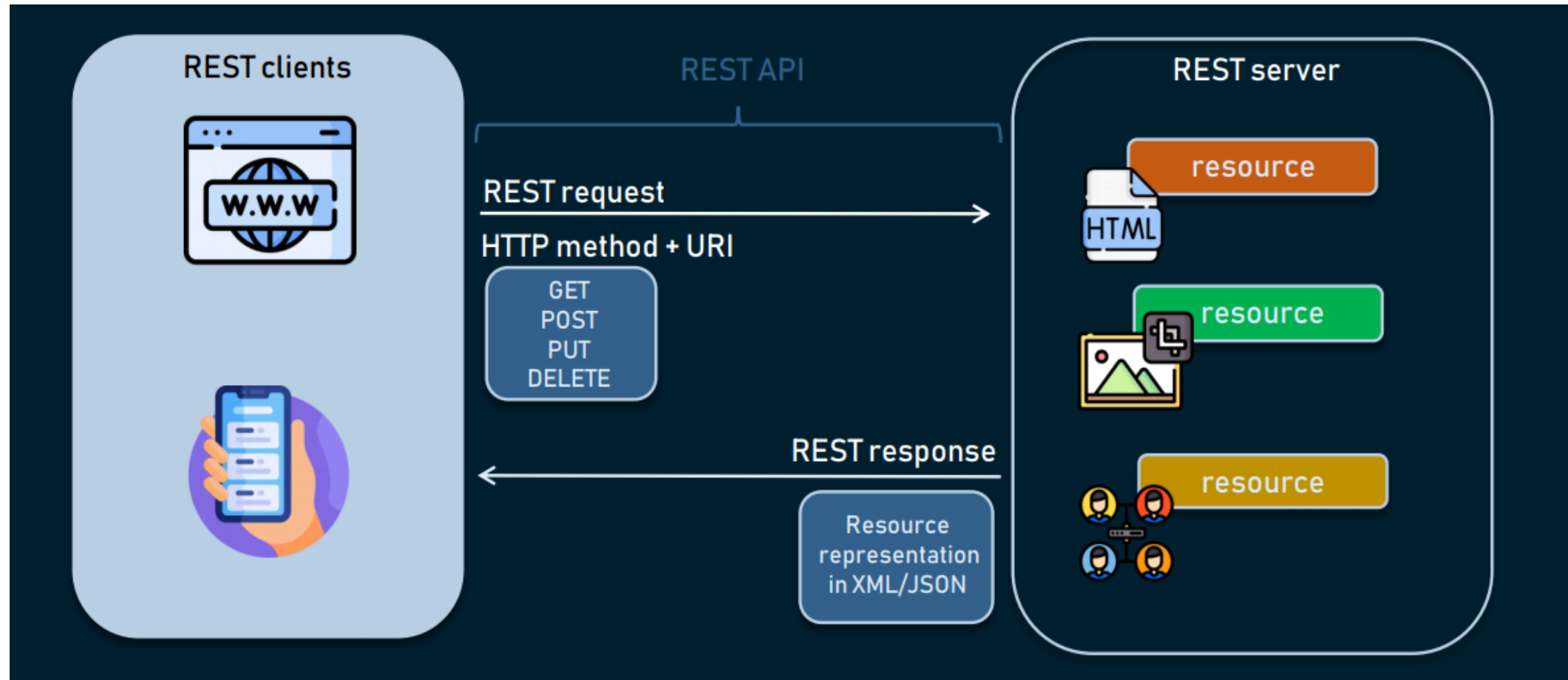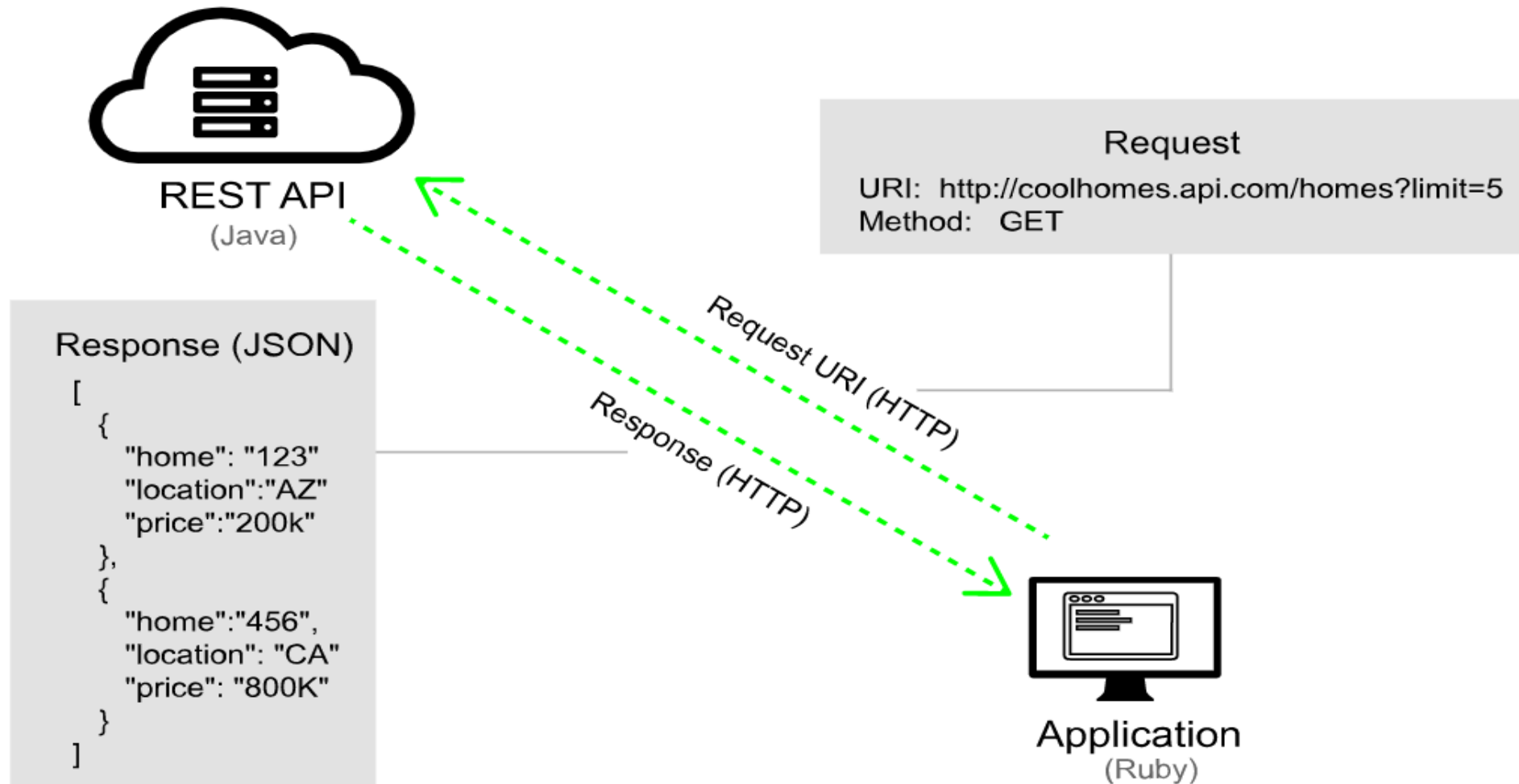
REST API
(Java)
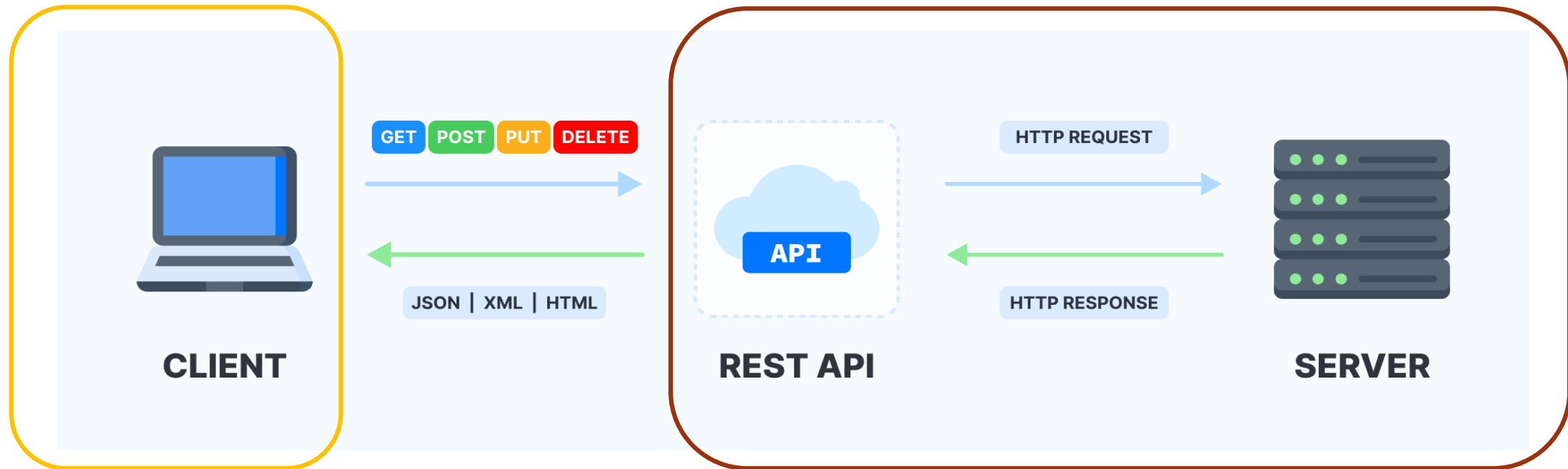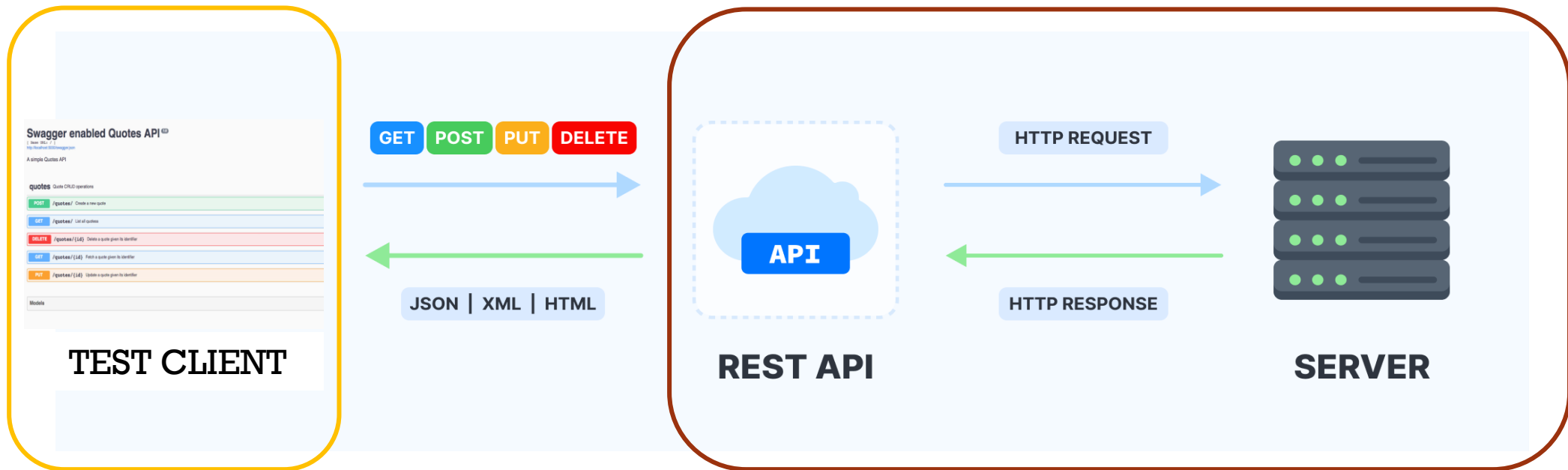
Request
URI: http://coolhomes.api.com/homes?limit=5
Method: GET

Request URI (HTTP)

Response (HTTP)

Response (JSON)
```
[
  {
    "home": "123"
    "location":"AZ"
    "price":"200k"
  },
  {
    "home":"456",
    "location": "CA"
    "price": "800K"
  }
]
```

Application
(Ruby)

REST API WITH FASTAPI

REST API WITH FASTAPI

GET | POST | PUT | DELETE

JSON | XML | HTML

TEST CLIENT

API

REST API

HTTP REQUEST

HTTP RESPONSE

SERVER

REST API WITH FASTAPI

http://localhost:8000/...

URL endpoint resource	Method
/	GET
/shows/{id}	GET
/create/{show}	POST
/update/{id}	PUT

REST API HTTP REQUEST HTTP RESPONSE **SERVER**

python
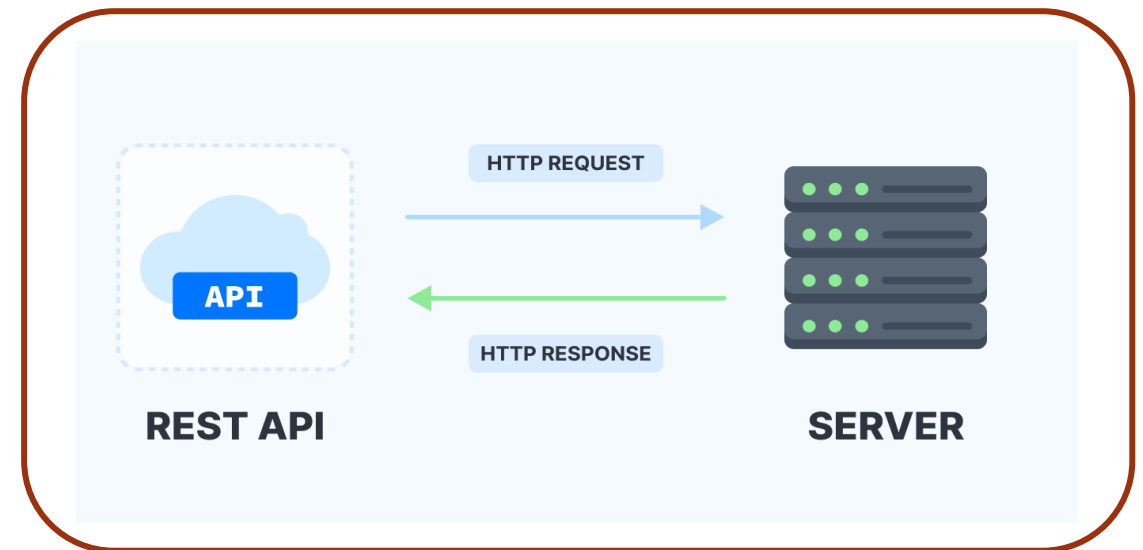
⚡ FastAPI

REST API WITH FASTAPI

- Installation : FastAPI, Uvicorn   https://fastapi.tiangolo.com/

- Create new Folder : myrestapi

- Add new file in myrestapi folder : main.py
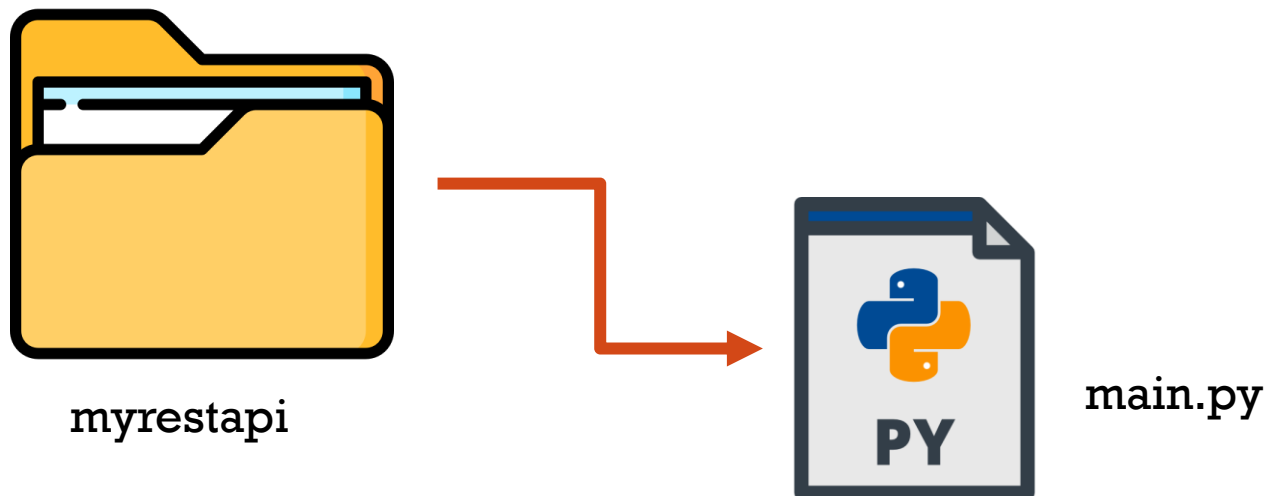
myrestapi   main.py

REST API WITH FASTAPI

main.py

```python
from fastapi import FastAPI

app = FastAPI()
```

REST API WITH FASTAPI

main.py

```python
from fastapi import FastAPI

app = FastAPI()


@app.get("/")
async def home():
    return {"message": "Hello World"}
```

URL endpoint resource	Method
/	GET

http://localhost:8000/

REST API WITH FASTAPI

main.py

Run the live server:

```bash
$ uvicorn main:app --reload

INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C
to quit)
INFO:     Started reloader process [28720]
INFO:     Started server process [28722]
INFO:     Waiting for application startup.
INFO:     Application startup complete.

                                                    restart ↻
```

..\myrestapi > uvicorn main:app --reload

REST API WITH FASTAPI

main.py

Check it

Open your browser at http://127.0.0.1:8000 [↩].

You will see the JSON response as:

```
{"message": "Hello World"}
```

← → C ⌂ ⓘ localhost:8000

{"message":"Hello World"}

http://localhost:8000/

REST API WITH FASTAPI

TEST CLIENT

GET POST PUT DELETE

JSON | XML | HTML

REST API

API

HTTP REQUEST

HTTP RESPONSE

SERVER

FastAPI

python™
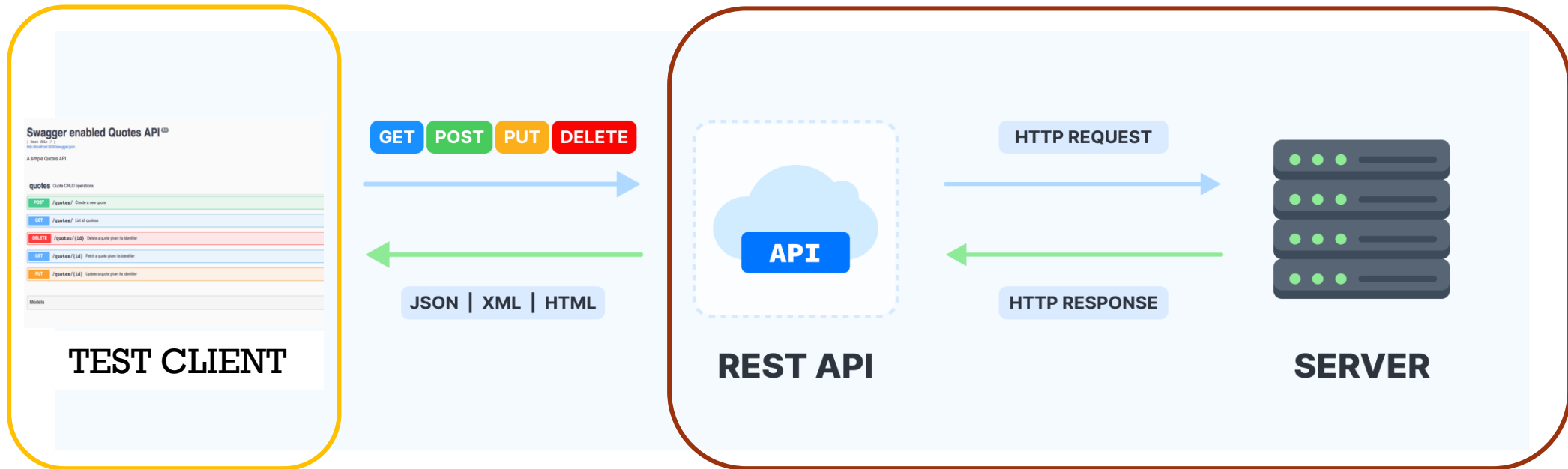
FastAPI

REST API WITH FASTAPI

main.py

Interactive API docs

Now go to http://127.0.0.1:8000/docs [↵].

You will see the automatic interactive API documentation (provided by Swagger UI [↵]):

FastAPI 0.1.0 OAS3

/openapi.json

http://localhost:8000/docs

default

GET / Home

Parameters   Try it out

No parameters

REST API WITH FASTAPI

http://localhost:8000/docs

main.py

Request URL

http://localhost:8000/

Server response

Code	Details
200	**Response body**

```
{
  "message": "Hello World"
}
```

Download

Response headers

```
content-length: 25
content-type: application/json
date: Thu,04 May 2023 11:07:53 GMT
server: uvicorn
```

REST API MODEL

REST API WITH FASTAPI

main.py

```python
shows_db = [
    {'name': 'The Wire',
        'producer': 'HBO',
        'description': 'This is the tv show description',
        'show_id': '1'},

    {'name': 'Breaking Bad',
        'producer': 'HBO',
        'description': 'This is the tv show description',
        'show_id': '2'},

    {'name': 'Mindhunter',
        'producer': 'Netflix',
        'description': 'This is the tv show description',
        'show_id': '3'},

    {'name': 'Vikings',
        'producer': 'History',
        'description': 'This is the tv show description',
        'show_id': '4'},
]
```

REST API WITH FASTAPI

main.py

```python
38
39  @app.get("/")
40  async def home():
41      return {"all_shows": shows_db}
42
```

URL endpoint resource	Method
/	GET

http://localhost:8000/docs

REST API WITH FASTAPI

GET POST PUT DELETE

JSON | XML | HTML

CLIENT

API

HTTP REQUEST

HTTP RESPONSE

REST API

SERVER

python™

python™

FastAPI

REST API WITH FASTAPI

- Create new Folder : mytestapp

- Add new file in mytestapp folder : app.py
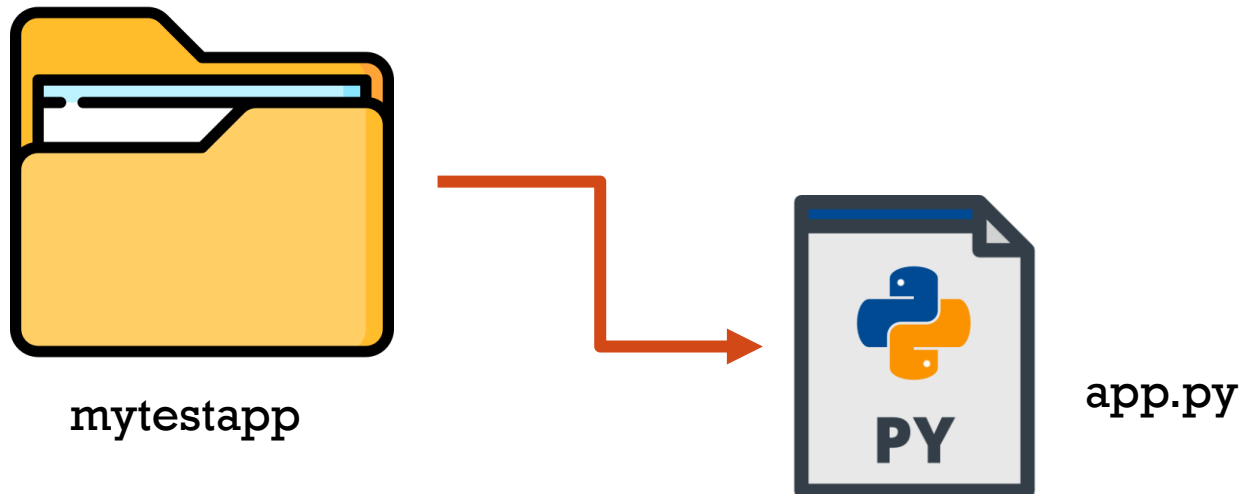
mytestapp

app.py

REST API WITH FASTAPI

app.py

```python
1  import requests
2
3
4  resp = requests.get('http://127.0.0.1:8000/').json()
5  print(resp)
```

print(resp['all_shows'] [1])

print(resp ['all_shows'] [1]['name'])

…mytestapp > python app.py

```
C:\Users\LeE\Documents\Python ML DL\Final Projects\Language_Detector_FastAPI\mytestapp>
python app.py
[{'name': 'The Wire', 'producer': 'HBO', 'description': 'This is the tv show descriptio
n', 'show_id': '1'}, {'name': 'Breaking Bad', 'producer': 'HBO', 'description': 'This i
s the tv show description', 'show_id': '2'}, {'name': 'Mindhunter', 'producer': 'Netfli
x', 'description': 'This is the tv show description', 'show_id': '3'}, {'name': 'Viking
s', 'producer': 'History', 'description': 'This is the tv show description', 'show_id':
 '4'}]
```
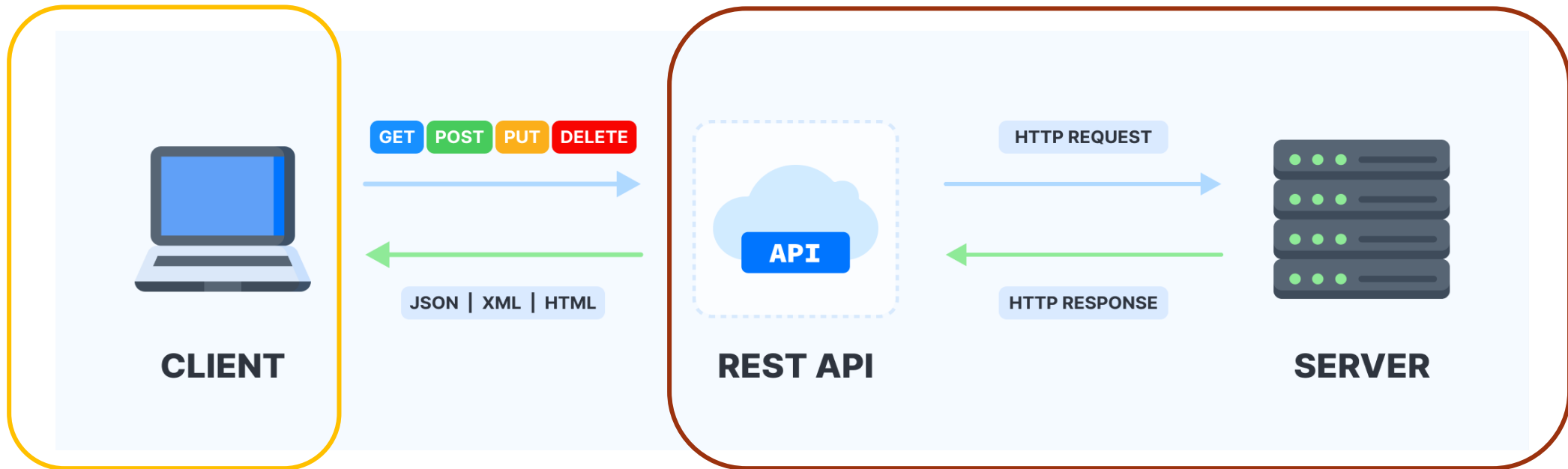
REST API WITH FASTAPI

http://localhost:8000/...

URL endpoint resource	Method
/	GET
/shows/{id}	GET
/create/{show}	POST
/update/{id}	PUT

REST API   HTTP REQUEST   HTTP RESPONSE   SERVER

REST API with FastAPI

```python
43
44  @app.get("/shows/{show_id}")
45  async def get_show(show_id: str):
46      for show in shows_db:
47          if show['show_id'] == show_id:
48              return show
49      return {"Error": "No show in DB with such id"}
50
```

URL endpoint resource	Method
/shows/{show_id}	GET

REST API WITH FASTAPI

main.py

Request URL

```
http://localhost:8000/shows/1
```

Server response

Code	Details
200	**Response body**
```
{
  "name": "The Wire",
  "producer": "HBO",
  "description": "This is the tv show description",
  "show_id": "1"
}
```
Response headers
```
content-length: 98
content-type: application/json
date: Thu,04 May 2023 12:53:11 GMT
server: uvicorn
``` |

REST API WITH FASTAPI

main.py

```python
6  class Show(BaseModel):
7      name: str
8      producer: str
9      description: str
10     show_id: str
11
12
13 app = FastAPI()
14
```

URL endpoint resource	Method
/create/{show}	POST

REST API WITH FASTAPI

main.py

```python
50
51  @app.post("/create")
52  async def create_show(show: Show):
53      shows_db.append(show)
54      return show
```

URL endpoint resource	Method
/create/{show}	POST

REST API WITH FASTAPI

main.py

Request URL

```
http://localhost:8000/create
```

POST **/create** Create Show ^

| Parameters | Cancel | Reset |

No parameters

Request body ^{required}   application/json ▾

```
{
    "name": "Game of Thrones",
    "producer": "HBO",
    "description": "This is the tv show description",
    "show_id": "5"
}
```

```
{
    "name": "Game of Thrones",
    "producer": "HBO",
    "description": "This is the tv show description",
    "show_id": "5"
}
```

REST API WITH FASTAPI

PY

main.py

Request URL

http://localhost:8000/

Server response

Code   Details

200
            Response body

                "show_id": "1"
              },
              {
                "name": "Breaking Bad",
                "producer": "HBO",
                "description": "This is the tv show description",
                "show_id": "2"
              },
              {
                "name": "Mindhunter",
                "producer": "Netflix",
                "description": "This is the tv show description",
                "show_id": "3"
              },
              {
                "name": "Vikings",
                "producer": "History",
                "description": "This is the tv show description",
                "show_id": "4"
              },
              {
                "name": "Game of Thrones",
                "producer": "HBO",
                "description": "This is the tv show description",
                "show_id": "5"
              }
            ]

REST API WITH FASTAPI

main.py

```python
58  @app.put("/update/{show_id}", response_model=Show)
59  async def update_show(show_id: str, show: Show):
60      update_show_encoded = jsonable_encoder(show)
61      for i in range(len(shows_db)):
62          if shows_db[i]['show_id'] == show_id:
63              shows_db[i] = update_show_encoded
64
65      return update_show_encoded
```

from fastapi.encoders import jsonable_encoder

URL endpoint resource	Method
/update/{id}	PUT

REST API WITH FASTAPI

http://localhost:8000/update/2

main.py

Request URL

```
http://localhost:8000/update/2
```

Server response

Code	Details
200	**Response body**

```
{
  "name": "erytez",
  "producer": "tet",
  "description": "rtet",
  "show_id": "zata"
}
```

Response headers

```
content-length: 72
content-type: application/json
date: Thu,04 May 2023 13:04:08 GMT
server: uvicorn
```

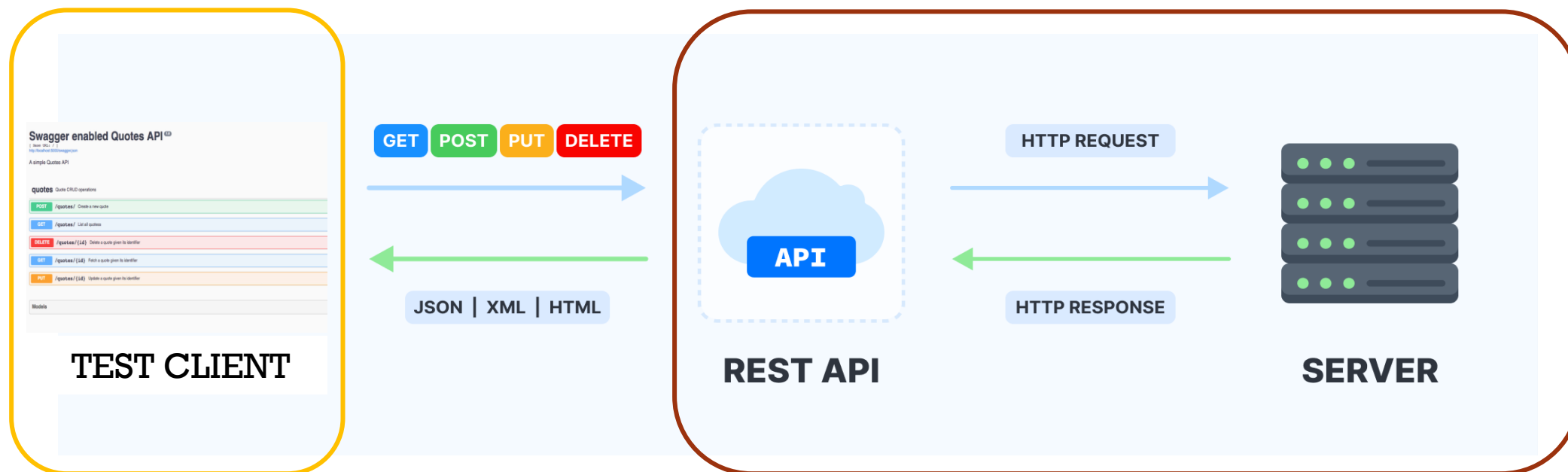| Name	Description
show_id * required string *(path)*	2

Request body required

```
{
  "name": "erytez",
  "producer": "tet",
  "description": "rtet",
  "show_id": "zata"
}
```

REST API WITH FASTAPI

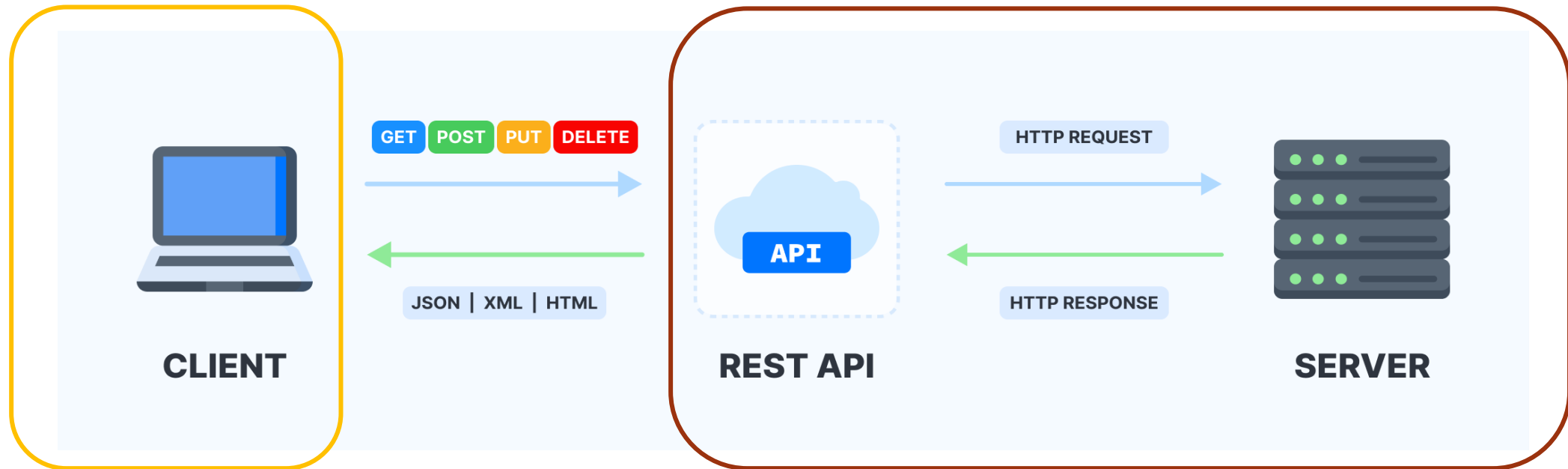# REST API WITH FASTAPI

REST API WITH FASTAPI

- Run CMD : …mytestapp/flaskapp > python app.py

- Then go to :   http://localhost:5000

```
main.py          ×    app.py — mytestapp    ×    app.py — mytestapp\flaskapp  ●    home.html          ×

 1   from flask import Flask, render_template
 2   import requests
 3
 4
 5   app = Flask(__name__)
 6
 7   app.config['SECRET_KEY'] = '40361900361905b5b9f99e0361905b9f'
 8
 9
10   @app.route("/", methods=['GET'])
11   def home():
12       shows = requests.get('http://127.0.0.1:8000/').json()
13       return render_template('home.html', shows=shows['all_shows'])
14
15
16   if __name__ == '__main__':
17       app.run(debug=True)
18
```

REST API WITH FASTAPI