Path Operations

127.0.0.1 : 8000

myproject.com /api/

Dominio PATH

PATHS / ROUTES / ENDPOINTS

/ users
/ tweets

OPERATIONS

- GET
- POST
- PUT
- DELETE
- OPTIONS
- HEAD
- PATCH
- TRACE

HTTP + MÉTODO

HEADERS

Body

PATH OPERATION DECORATOR

```
@app.get("/")
```

PATH OPERATION FUNCTION

```
def home():
    return {"Hello": "World"}
```

API TWITTER

"/" → Home

"/tweets/22" → tweet 22
"/tweets/1"
"/tweets/2"

PATH PARAMETER → OBLIGATORIO

"/tweets/{tweet_id}" → Particular tweet

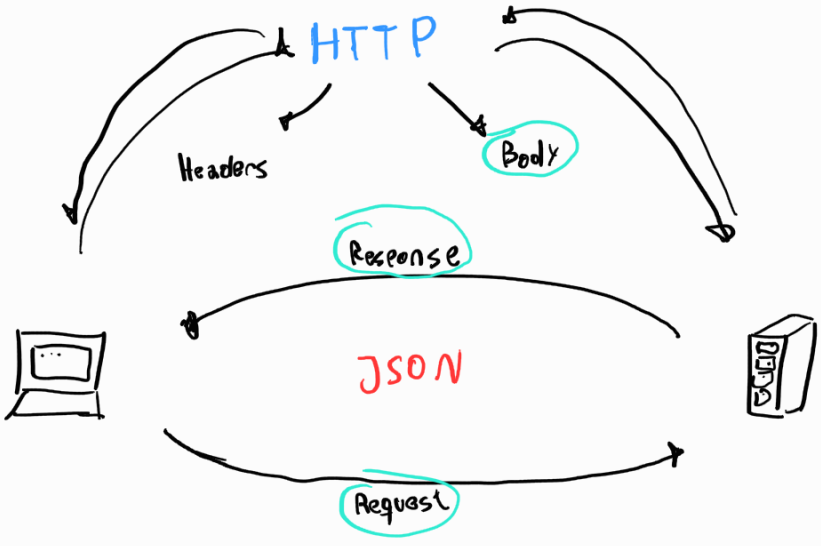
1
2

1000 123787

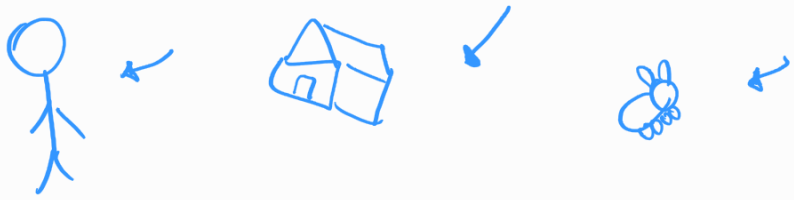
PUT /users/{user_id}/details?age=20&height=184

- Path Parameter
- Query Parameter

Request Body



MODELS ← descriptiva



Pydantic

↓

Base Model

http://myapp.com/api/person/data:1?name=Miguel&age=25

- Name
- Age
- Hair_color

name → aaaaaaa2 .... x1000000

SO > name > 1

Validations:

Query

- max_length
- min_length
- regex

- ge → greater or equal than ≥ >=
- le → less or equal than ≤ <=
- gt → greater than >
- lt → less than <

- Title
- description

Tipos de Datos Especiales

Pydantic → Field

Clásicos

- str
- int
- float
- bool

Exóticos

- Enum
- HttpUrl
  - ✓ http://myproject.com
  - ✓ www.platzi.com
  - ✗ hola
- FilePath
  - ✓ C:/windows/system32/432.dll
- DirectoryPath
  - ✓ /mnt/c/SomeFolder
- EmailStr
  - ✓ hola@hola.com
  - ✗ Facundo.com
- PaymentCardNumber
- IPvAnyAddress
- NegativeFloat
- PositiveFloat
- NegativeInt
- PositiveInt