ML



API

- ML
- -
- : FastAPI, SQL, Docker
- Clean Architecture

· () ()

00000

 \mathcal{N}°

ML- , ML-

ML- : ML- :

API (Application Programming Interface).

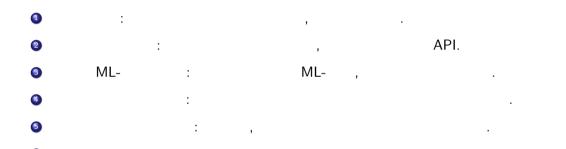
③ :

(, Python).

2 ML- : ML- Scikit-learn.

3 : , , (, PostgreSQL Sqlite).

4 API: API



• ML- , ML-

•

, , API

ML

TBD

HTTP

HTTP:
GET:
POST:
PUT:
DELETE:
HTTP (status codes):
200 OK:
400 Bad Request:
404 Not Found:

API

- RESTful API:
 - REST.
 - URL.
 - •
- GraphQL API:
 - API.
 - ,
 - .

- Swagger: API. API **JSON** YAML.
- API Blueprint:
 - API Markdown.
 - API.
- RAML:
 - YAML-

 - API

API.

Swagger,

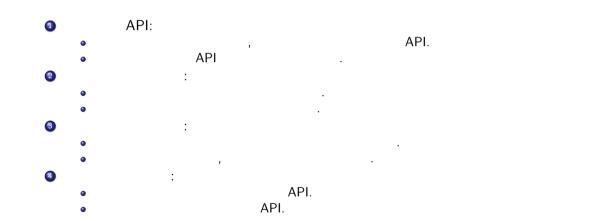
HTTP, API.

 \mathcal{N}°

API

HTTP
API
API,

API



GET /items

GET /items/{id}

POST /items

PUT /items/{id}

"

RESTful API

GET /getAllItems

GET /getItemById/{id}

- "Byld"

POST /addItemToInventory

PUT /updateItem/{id}

- "update"

RESTful

No

Stateful API

- 0
- COC
- cookie

Cookie

"userId": "12345",

```
cookie:

GET /items HTTP/1.1

Host: example.com

Cookie: sessionId=abcd1234
```

 \aleph

```
JWT:
GET /items HTTP/1.1
```

Host: example.com Authorization: Bearer eyJhbGciOiJIUzl1NilsInR5cCl6lkpXVCJ9.eyJzdW

JWT

. evJhbGciOiJIUzl1NilsInR5cCl6lkpXVCJ9 SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV adQssw5c **JWT** (payload). "sub" "1234567890"

JWT?								
• JWT (JSON Web Token) -								
JWT	:	,						
•								
•	(payload)		,					
•								

	JWT	Python		
0		PyJWT: pip	install	PyJWT
2			:	
import	jwt			
secret_	key =	"my_secret	_key"	

```
JWT
                 Python (
 1
def create_token(payload):
    token = jwt.encode(payload, secret_key, algorithm="HS256"
    return token
 0
user id = 123
username = "john_doe"
payload = {"user_id": user_id, "username": username}
token = create_token(payload)
print(token)
```

```
•
```

"user id": 123.

"username": "iohn doe"

```
JWT
                 Python (
 0
def decode_token(token):
    decoded = jwt.decode(token, secret_key, algorithms=["HS256"])
    return decoded
decoded_token = decode_token(token)
print(decoded_token)
```

 \mathcal{N}_{0}



- API (Application Programming Interface)
- API

API

- POST /users -
- GET /users/{userId} -
- PUT /users/{userId} -
- DELETE /users/{userId} -
- GET /items/{itemId} -
- POST /items -
- PUT /items/{itemId} -
- DELETE /items/{itemId} -

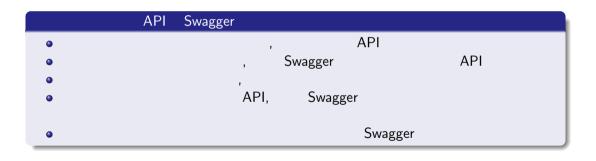
info:

API Swagger

swagger: "2.0"

```
API
  title:
  version: 1.0.0
paths:
  /users:
    post:
      summary:
      parameters:
        - in: body
          name: user
          required: true
          schema:
            $ref: "#/definitions/User"
```

	API Swagger				
•	: API			,	
	, API.				
•	: Swagger		,	,	
		API.			
•	: Swagger				
	API, ,				
•	: Swagger				,
•	: Swagger				
	,				
	<u> </u>				



API
Swagger API, , ,
API Swagger ,
API Swagger , ,

API

Slide 1

- Swagger API
- APSwagger

Slide 2 URL API

- URL API: /api/v1
- API: POST, GET, PUT, DELETE

Slide 3

API

POST /api/v1/users/register -name, email, password

POST /api/v1/users/login -

• : id

 \aleph_0

API.