

ML

07 2023

No

1

2 Brief: ML-

3

4 API

5 stateful API

6 cookie JWT RESTful API

7 API

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

- (\quad)
- $\quad / \quad (\quad)$
- (\quad)
- (\quad)



○○○●○

Brief:
○○○○○

ML-

○○

○○○○○○○
API

○○○

stateful API

○○○○○○○○○

cookie

JWT

RESTful API

○





- ML- , ML-
- .

1 ML- :
ML-
API (Application Programming Interface).

2 :
.
.
3 :
.
4 :
.
.

- ① : , (, Python).
- ② ML- : ML- Scikit-learn.
- ③ : , (, PostgreSQL , Sqlite).
- ④ API: API , .
- ⑤ : .

①

:

,

.

②

:

,

API.

③

ML-

:

ML-

,

.

④

:

.

⑤

:

,

.

⑥

:

-

,

.

- ML- , ML-
- . ,
- . ,
- , ,
- , , API .

○○○○○

Brief:
○○○○○

ML-

●●

○○○○○○○
API

○○○

stateful API

○○○○○○○○○

cookie

JWT

RESTful API

○

ML

TBD

HTTP

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

API

① RESTful API:



REST.



URL.



.

② GraphQL API:



API.



,

.



.

API

1 Swagger:

- , API JSON API.
- API YAML.
- .

2 API Blueprint:

- API Markdown.
- .
- API.

3 RAML:

- YAML- API.
- .
- .

API

, Swagger, HTTP,
API.

API

1

○ :
○
○ , HTTP .

2

○ :
○ , API .
○ API,

3

○ :
○
○ .
○ .

API

1

API:

•

API

,

API.

•

.

2

:

•

.

•

.

3

:

•

.

•

,

.

4

:

•

API.

•

API.

RESTful API

GET /items

- GET /items
- - :

GET /items/{id}

- GET /items/{id}
- - : ,

POST /items

- POST /items
- - :

PUT /items/{id}

- PUT /items/{id}
- - : ,

DELETE /items/{id}

RESTful API

GET /getAllItems

- .
- - : “ ” , , .

GET /getItemById/{id}

- .
- - : “ById” , , .

POST /addItemToInventory

- .
- - : , .

PUT /updateItem/{id}

- .
- - : “update” RESTful .

Stateful API

-
-

cookie

.

.

JWT (JSON Web Token)

- JWT
- Authorization

1

2

3

4

5

JWT

JWT

JWT

Authorization

JWT

•

•

•

cookie.

•

cookie

Cookie

cookie:

GET /items HTTP/1.1

Host: example.com

Cookie: sessionId=abcd1234

Cookie

sessionId

Cookie

{

"userId": "12345",

JWT (JSON Web Token)

JWT:

GET /items HTTP/1.1

Host: example.com

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW

JWT

, eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

SflKxwRJSMekKF2QT4fwpmEJf36POk6yJV_adQssw5c

JWT

(payload).

{

"sub": "1234567890".

1

JWT?

- JWT (JSON Web Token) - ,
.
- JWT : , .
- .
- (payload) , .
- .

2

JWT

Python

1

PyJWT: pip install PyJWT

2

:

import jwt

secret_key = "my_secret_key"

3

JWT Python ()

```
1 :  
  
def create_token(payload):  
    token = jwt.encode(payload, secret_key, algorithm="HS256")  
    return token
```

```
1 :  
  
user_id = 123  
username = "john_doe"  
  
payload = {"user_id": user_id, "username": username}  
token = create_token(payload)  
print(token)
```

4

JWT

Python ()

```

1      :

def decode_token(token):
    decoded = jwt.decode(token, secret_key, algorithms=["HS256"])
    return decoded

```

```

1      :

decoded_token = decode_token(token)
print(decoded_token)

```

```

•      :

{
    "user_id": 123,
    "username": "john_doe"
}

```

5

!

,

.

.

,

.

1

- API (Application Programming Interface)
- API

2

API

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

3

API Swagger

```

swagger: "2.0"
info:
  title:    API
  version: 1.0.0
paths:
  /users:
    post:
      summary:
      parameters:
        - in: body
          name: user
          required: true
          schema:
            $ref: "#/definitions/User"
    
```

4

API Swagger

- : API ,
- , API.
- : Swagger , ,
- . API.
- : Swagger API, , .
- : Swagger ,
- : Swagger .

API	Swagger
•	, API
•	, Swagger API
•	,
•	API, Swagger
•	Swagger

6

- API
- Swagger API, ,
- API Swagger ,
- API Swagger , ,

API

Slide 1

Swagger API

- Swagger API.

Slide 2

URL API

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

Slide 3

API

- 1 POST /api/v1/users/register -
 - : name, email, password
 - : id
- 2 POST /api/v1/users/login -