Restful

sechung

WHY

- 당시 폭팔적인 웹 성장에 비해 표준이라는 개념이 미비
- 표준을 만들기 위한 다양한 움직임 등장
- 구조적 모델을 통해 웹 프로토콜 표준에 가이드를 제공하기 위해 등장

REST

- Roy T.Fielding - Architectural Styles and the Design of Network-based Software Architectures
- 웹의 우수한 설계, 장점을 최대한 활용할 수 있는 아키텍처
  - 분산 하이퍼미디어 시스템의 구조화 스타일
  - 여러 네트워크 기반 구조화 스타일을 접목
  - 동일한 인터페이스를 위해 추가적인 제약

REST Principle

- Client-Server
- Stateless
- Cache
- Uniform Interface
- Layered System
- Code-On-Demand

REpresentational State Transfer

- REpresentational
  - 자원의 상태가 어떤 형태(Media type)로든 클라이언트에게 전달
- 상태 전송
  - 서버가 요청한 자원의 "표현" 으로 클라이언트의 상태를 변경

REpresentational

- Identifier
  - URI
- Resource
  - 고유한 ID로 식별되는 자원
- Representation
  - Media type

State Transfer

- 웹페이지의 상태가 클라이언트에 의해 요청
- 서버로부터 자원의 "표현"을 응답
- 전송된 표현은 클라이언트의 상태를 변경

REST

- 기존의 URL을 자원 그 자체로 보는 시야에서 벗어나
  - URI는 자원을 식별하는 "의미"
    - 자원은 실제 정보와 그것을 담는 "표현" 으로 구분
      - 다양한 미디어 타입을 지원함을 설명

Restful API

- Restful은 Rest 원칙을 잘 따르는 것
- Restful API는 Rest 원칙을 잘 따르는 API

Restful API

GET /user/1/comment/24 HTTP/1.1
Host: example.com
Authorization: Bearer jwt.jwt.jwt
cache-Control: no-cache
Accept: application/json

Http/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Vary: Accept
Content-length: 46

```
{
    "id": 24,
    "writer_id": 1,
    "content": "Hello World"
}
```

- Client-Server
- Stateless
- Cache
- Uniform Interface
- Layered System

Finish