REST API

REST API란 RESTful한 API를 말하며 일련의 특징과 규칙 등을 지키는 API를 일컫는다.

2000년에 Roy Thomas Fielding이 쓴 논문에서 처음 등장한 개념임.

1. Uniform-Interface

API에서 자원들은 각각의 독립적인 인터페이스를 가지며 각각의 자원들이 url 자원식별, 표현을 통한 자원조작, self-descriptive message, HATEOAS 구조를 가지는 것을 말한다.

독립적인 인터페이스라는 것은 서로 종속적이지 않은 인터페이스를 말함.

예를 들어 웹페이지를 변경했다고 웹 브라우저를 업데이트하는 일은 없어야 하고 HTTP 명세나 HTML 명세가 변경되어도 웹페이지는 잘 작동해야하는 것이다.

- url 자원 식별 - identification of resources, 자원은 url로 식별되어야한다.
- 표현을 통한 자원조작 - manipulation of resources through representations은 url과 GET, DELETE 등 HTTP 표준 메서드 등을 통해 자원을 조회, 삭제 등 작업을 설명할 수 있는 정보가 담겨야하는 것을 말함.
- self-descriptive messages - HTTP Header에 타입을 명시하고 각 메시지들은 MIME types에 맞춰 표현되어야 한다. 예를 들어 .json를 반환한다면 application/json으로 명시해주어야 한다. MIME types는 문서, 파일 등의 특성과 형식을 나타내는 표준이다.
- HATEOAS 구조 - 하이퍼링크에 따라 다른 페이지를 보여줘야 하며 데이터마다 어떤 URL에서 원했는지 명시해주어야하는 것을 말함. 보통 href, links, link, url 속성 중 하나에 해당 데이터의 URL을 담아서 표기해야 함.

2. Stateless

이 규칙은 HTTP 자체가 Stateless이기 때문에 HTTP를 이용하는 것 만으로도 만족됨. 그리고 이는 REST API를 제공해주는 서버는 세션을 해당 서버쪽에 유지하지 않는다는 의미이다.

3. Cacheable

HTTP는 원래 캐싱이 됨. 아무런 로직을 구현하지 않더라도 자동적으로 캐싱이 됨.

새로고침을 하면 304가 뜨면서 원래 있던 js와 css이미지 등을 불러오는 것을 볼 수 있음

이는 HTTP 메서드 중 GET에 한정되며 한정된 시간을 정할 수 있음. 캐싱된 데이터가 유효한지를 판단하기 위해 Last-Modified와 Etag라는 헤더값을 사용함. Etag는 전달되는 값에 태그를 붙여서 캐싱되는 자원인지를 확인해주는 것임.

4. Client-Server 구조

클라이언트와 서버가 서로 독립적인 구조를 가져야 함. 물론 이는 HTTP를 통해 가능한 구조이다. 서버에서 HTTP 표준만 지킨다면 웹에서는 그에 따른 화면이 잘 나타나게 된다. 서버는 그저 API를 제공하고 그 API에 맞는 비즈니스 로직을 처리하면 된다. 마찬가지로 클라이언트에서는 HTTP로 받는 로직만 잘 처리하면 되는 것이다.

5. Layered System

계층구조로 나눠져 있는 아키텍쳐를 뜻함. WEB기반 서비스를 하면 보통 이러한 시스템을 구축함

REST API의 URI규칙

1. 동작은 HTTP 메서드로만 해야하고 url에 해당 내용이 들어가면 안된다. 수정=put, 삭제=delete... 예를 들어 books/delete/1 이렇게 표기하면 안됨
2. .jpg,.png등 확장자는 표시하지 말아야한다.

3. 동사가 아닌 명사로만 표기해야 한다. 유저가 책을 소유한다라는 것을 표현한다면 유저/유저아이디/inclusion/책/책아이디 로 표현하고 유저가 소유한 아파트를 조회한다고 users/{userId}/aparts 이런식
4. 계층적인 내용을 담고 있어야 한다. /집/아파트/전세 이런식으로 내려가야함
5. 대문자가 아닌 소문자로만 쓰며 너무 길 경우에 바를 써야할 경우 언더바가 아닌 그냥 바 - 를 쓴다
6. HTTP응답 상태코드를 적재적소에 활용한다. 성공시에 200, 리다이렉트는 301 등..