

2024-04-03 6. RESTful API Design

! [현행 문서: 6. RESTful API Design](#)

- 1 Prerequisites
- 2 교재
- 3 Definition of Done
 - 3.1 REST란 무엇인가?
 - 3.2 리소스 중심 API 디자인
 - 3.3 HTTP 측면에서의 API 디자인
 - 3.4 HTTP 의미 체계 준수

1 Prerequisites

2 교재

[REST API Tutorial](#)

[REST API Quick Tips](#)

[Using HTTP Methods for RESTful Services](#)

[Idempotence](#)

[RESTful 웹 API 디자인](#)

[RESTful API를 설계하기 위한 디자인 팁](#)

[그럼 REST API로 괜찮은가 - Youtube](#)

3 Definition of Done

3.1 REST란 무엇인가?

- REST 가 무엇인지 알고 있다.
- RESTful API 가 무엇인지 알고 있다.
- 다음 RESTful API 의 특성에 대해 간단히 알고 있다.
 - Client-Server
 - Stateless
 - Cachable
 - Uniform Interface
 - 리소스 중심 URI 디자인을 통한 리소스 식별(Identification of Resource)
 - 표현을 활용한 리소스 조작(Manipulation Of Resource through Representations)
 - 자기 서술적인 메시지(Self-descriptive Message)
 - HATEOAS(Hypermedia As The Engine Of Application State)
 - Layered System
- REST와 HTTP가 어떤 관계를 가지는지 알고 있다.
- REST에서 Resource가 어떤 의미인지 알고 있다.

3.2 리소스 중심 API 디자인

- 리소스 중심으로 설계한다는 말의 의미와 목적을 알고 있다.
- 리소스를 표현하기 위해 URI가 어떻게 형성되는지 알고 있다.
- 리소스 메소드가 무엇이고 어떤 역할을 하는지 알고 있다.

3.3 HTTP 측면에서의 API 디자인

- HTTP의 메소드가 RESTful API 디자인에서 각각 어떤 행동을 정의하는지 알고 있다.
 - GET
 - POST
 - PUT
 - PATCH
 - DELETE
- POST, PUT 및 PATCH의 차이점을 알고 있다.
- URI와 HTTP Method만으로 표현되지 않는 경우에는 어떻게 표현할 수 있는지 알고 있다.
 - Filtering

- Pagination

3.4 HTTP 의미 체계 준수

- 리소스의 유형을 정의하기 위해 HTTP의 MIME type이 어떻게 지정되는지 알고 있다.
 - Content-Type
 - Accept
- GET 메소드 활용 시 요청 성공 여부, 리소스 존재 여부, 권한 등 상황에 따라 어떤 응답 혹은 응답코드가 전달되는지 이해하고 있다.
- POST 메소드 활용 시 리소스 생성 여부, 요청 데이터 적합성 체크 결과, 권한 등 상황에 따라 어떤 응답 혹은 응답코드가 전달되는지 이해하고 있다.
- PUT 메소드 활용 시 리소스 생성 여부, 데이터 충돌 여부, 권한 등 상황에 따라 어떤 응답 혹은 응답코드가 전달되는지 이해하고 있다.
- PATCH 메소드 활용 시 리소스가 패치되는 사양에 대해 알고 있고, 상황에 따라 어떤 응답 혹은 응답코드가 전달되는지 이해하고 있다.
- DELETE 메소드 활용 시 성공 여부, 데이터 존재 여부, 권한 등 상황에 따라 어떤 응답 혹은 응답코드가 전달되는지 이해하고 있다.