

HTTP 메서드

Network

CONTENTS

01
HTTP API

02
GET, POST

03
PUT, PATCH
DELETE

04
HTTP 메서드의 속성

01

HTTP API

01. CONTENTS

요구사항

요구사항 : 회원 정보 관리 API를 만들어주세요

- 회원 목록 조회
- 회원 조회
- 회원 등록
- 회원 수정
- 회원 삭제

01. CONTENTS

API URI 설계

- 회원 목록 조회 /read-member-list
- 회원 조회 /read-member-by-id
- 회원 등록 /create-member
- 회원 수정 /update-member
- 회원 삭제 /delete-member

01. CONTENTS

URI 설계

URI 에서 가장 중요한 것은 리소스
리소스의 의미를 담아야 함

즉, 회원을 등록하고 수정하고 조회하는 것을 배제해야함

01. CONTENTS

API URI 설계

- 회원 목록 조회 /members
- 회원 조회 /members/{id}
- 회원 등록 /members/{id}
- 회원 수정 /members/{id}
- 회원 삭제 /members/{id}



어떻게 구분하는가?

01. CONTENTS

HTTP 메서드

GET - 리소스 조회

POST - 요청 데이터 처리

PUT - 리소스를 대체한다, 해당 리소스가 없다면 생성함

PATCH - 리소스를 부분 변경한다

DELETE - 리소스를 삭제함

02

HTTP 메서드 (GET,POST)

02. CONTENTS

HTTP METHOD

GET

- PATH에 있는 자원을 달라고 요청하는 메서드
- 서버에 전달하고 싶은 데이터를 쿼리를 통해 전달
- 바디를 이용하여 데이터 전송이 가능한 하지만 지원하지 않는 곳이 많음

02. CONTENTS

HTTP METHOD

POST

- 메시지 바디를 통해 서버로 요청 데이터를 전달함
- 메시지 바디를 통해 들어온 데이터를 처리하는 모든 기능을 수행한다
- 주로 전달된 데이터로 신규 리소스를 등록하거나 프로세스를 처리하는데 사용한다.

02. CONTENTS

HTTP METHOD

POST 데이터 처리 예시

- HTML FORM 에 입력한 정보를 처리 (회원가입,주문)
- 게시판 글쓰기 , 댓글 쓰기
- 서버가 식별하지 않은 새로운 리소스 생성 (신규주문 생성)
- 기존 자원에 데이터를 추가 (문서 끝에 내용을 추가)

02. CONTENTS

HTTP METHOD

GET vs POST ?

POST를 이용하여 조회 기능을 구현할수도 있다. 그러나 그러지 않는 이유는 GET 메서드를 이용한 경우 캐싱을 약속을 하기때문에 GET을 사용하는 것이 유리하다. 조회와 같이 명확하게 GET 메서드를 사용해야하는 경우 POST를 사용할 이유가 없다.

03

HTTP 메서드 (PUT,PATCH,DELETE)

03. CONTENTS

HTTP METHOD

PUT

- 리소스를 대체하는 메서드
- 리소스가 있으면 대체, 없으면 생성
- 클라이언트가 리소스의 위치를 정확하게 알고 식별한다는 점이 POST와의 차이점이다
- 리소스를 부분수정 할 수 없다. 기존 리소스가 존재하면 해당 리소스를 삭제하고 대체하기 때문이다.

03. CONTENTS

HTTP METHOD

PATCH

- 리소스를 부분 변경하는 메서드
- PUT과 달리 변경할 데이터가 있다면 PATCH 메서드를 사용하면 된다.
- PATCH를 지원하지 않을 경우 POST를 사용하면 됨

03. CONTENTS

HTTP METHOD

DELETE

- 리소스를 제거할 때 사용한다
- 어떤 리소스를 제거할지 정확하게 명시함

04

HTTP 메서드의 속성

04. CONTENTS

HTTP METHOD의 속성

HTTP 메소드 ◆	RFC ◆	요청에 Body가 있음 ◆	응답에 Body가 있음 ◆	안전 ◆	멱등(Idempotent) ◆	캐시 가능 ◆
GET	RFC 7231	아니오	예	예	예	예
HEAD	RFC 7231	아니오	아니오	예	예	예
POST	RFC 7231	예	예	아니오	아니오	예
PUT	RFC 7231	예	예	아니오	예	아니오
DELETE	RFC 7231	아니오	예	아니오	예	아니오
CONNECT	RFC 7231	예	예	아니오	아니오	아니오
OPTIONS	RFC 7231	선택 사항	예	예	예	아니오
TRACE	RFC 7231	아니오	예	예	예	아니오
PATCH	RFC 5789	예	예	아니오	아니오	예

출처: <https://ko.wikipedia.org/wiki/HTTP>

04. CONTENTS

HTTP METHOD의 속성

안전 (GET, HEAD)

- 호출해도 리소스를 변경하지 않는다.
- 호출해서 리소스의 변경이 일어나면 안전하지 않다
- 계속 호출해서 로그가 쌓여 장애가 발생한다고 해도 해당 부분은 고려하지 않는다. 안전은 해당 리소스만 고려한다.

04. CONTENTS

HTTP METHOD의 속성

멱등 (GET,PUT,DELETE)

- 한번 호출하던, 두번 호출하던 100번 호출하던 결과가 동일하다
- PUT의 경우 기존 데이터를 날리고 덮어쓴다는 결과가 동일하기 때문에 멱등이다.

04. CONTENTS

HTTP METHOD의 속성

먹등

- 먹등 또한 외부요인으로 중간에 리소스가 변경되는 것 까지 고려하지 않음
- 왜 이런 개념이 필요할까??
 - 서버가 TIME OUT 등으로 정상 응답을 못주었을 때, 클라이언트가 같은 요청을 다시 해도 되는가의 판단 근거로 사용함.

04. CONTENTS

HTTP METHOD의 속성

캐시가능 (GET,HEAD,POST,PATCH)

- 응답 결과 리소스를 캐시해서 사용해도 되는가?
- 실제로 GET HEAD 정도만 캐시로 사용한다
- POST,PATCH의 경우 본문 내용까지 캐시키로 고려해야하기 때문에 구현이 쉽지 않다.

QUIZ

- GET vs POST 둘의 차이점을 서술해주세요
- HTTP 메서드의 특징에 대해 설명하고, 이러한 특징이 왜 필요한지도 서술해주세요.



Thank you

HTTP 메서드

