



REST란?

Representational State Transfer의 약자.

- 자원 (Resource): URI.
- 행위 (Verb): HTTP Method
- 표현 (Representations)

CRUD란? - 대부분의 컴퓨터 소프트웨어가 가지는

- Create 기본적인 데이터 추가 기능.
- Read - 사용자와 인터페이스가 갖추어야 할 기능.
- Update
- Delete



HTTP Method.

보통 CRUD에서
조회: GET
등록: POST
수정: PUT
삭제: DELETE

멤통성 (Idempotence) - 여러번 실행해도 결과가 같음.
흔히로 인하여 데이터 변형 X.

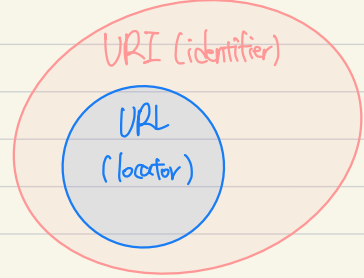
전체 종류

GET - 서버로부터 데이터를 취득
POST - 서버에 데이터를 추가, 작성
PUT - 서버의 데이터를 갱신, 작성
DELETE - 서버의 데이터를 삭제
HEAD - 서버 리소스의 헤더 (메타데이터의 취득)
OPTIONS - 리소스가 지원하는 메소드의 취득
PATCH - 리소스의 일부분을 수정
CONNECT - 프록시 동작의 터널 설정을 변경

URI - 특정 리소스를 식별하는 통합과원 식별자

식별 웹 환경에서 사용하는 논리적 또는 물리적 리소스를 식별하는
고유한 물리적 시스템

URL - 웹 주소, 컴퓨터 네트워크 상에서 리소스가 어디 있는지
위치 알려주기 위한 규약. URI의 서브셋.



• GET

주로 데이터를 읽거나 (READ) 검색(Retrieve)할 때에 사용되는 메소드.

GET 요청 - 성공 - XML이나 JSON과 함께 200 (OK) HTTP 응답 코드를 반환한다.

 - 에러 - 주로 404 (Not found) 에러나 400 (Bad request) 에러 발생.

특징 - HTTP 명세에 의하면 GET 요청은 오직 데이터를 읽을 때만 사용되고 수정할 때는 사용X

 - GET 요청은 idempotent 하다. (여러번 해도 결과가 동일하다.)

 - 데이터 변경하는 연산에 사용하면 안된다.

ex) GET /user/1

데이터 조회하는 것이기 때문에 요청시 Body 값과 Content-Type 가 비워져있다.

조회할 데이터에 대한 정보는 URL을 통해서 파라미터를 받고 있는 모양을 볼 수 있다.

데이터 조회에 성공한다면 Body 값에 데이터 값을 저장하여 성공응답을 보낸다.

• POST

POST 메소드는 주로 새로운 메소드를 생성(Create)할 때 사용된다.

POST는 리소스 (부호리소스나 하위리소스)들을 생성하는데 사용된다.

성공적으로 creation을 완료하면 201 (Created) HTTP 응답을 반환한다.

특징 - POST 요청은 idempotent 하지 않다.

같은 POST 요청을 반복해서 했을 때 항상 같은 결과가 나오지 않을 수 있다.

두개의 같은 POST 요청을 보내면 같은 정보를 담은 두개의 다른 resource를 반환할 가능성이 높다.

ex) POST /user

body : { "name": "example" }

Content-Type : "application/json"

데이터를 생성하는 것이기 때문에 요청시 Body 값과 Content-Type 값을 작성해야한다.

URL을 통해서 데이터를 받지 않고 Body 값을 통해서 받는다.

데이터 조회에 성공한다면 Body 값에 저장한 데이터 값을 저장하여 성공응답을 보낸다.

• PUT

리소스를 생성/업데이트하기 위해 서버로 데이터를 보내는 데 사용.

PUT 요청은 idempotent 합니다.

동일한 PUT 요청을 여러 번 전송하면 항상 동일한 결과가 생성됩니다.

ex) PUT /user/1

body : { "date": "update example" }

Content-Type : "application/json"

데이터를 수정하는 것이 때문에 요청시 Body 값과 Content-Type 값을 작성해야 합니다.

URL을 통해서 어떤 데이터를 수정할지 파라미터를 받습니다.

그리고 수정할 데이터 값은 Body 값을 통해서 받습니다.

데이터 조회에 성공한다면 Body 값에 저장한 데이터 값을 그대로 성공 응답을 보냅니다.

• DELETE

Delete 메소드는 지정된 리소스를 삭제합니다.

ex) DELETE /user/1

데이터를 삭제하는 것이 때문에 요청시 Body 값과 Content-Type 값이 바뀌어 있습니다.

URL을 통해서 어떠한 데이터를 삭제할지 파라미터를 받습니다.

데이터 삭제에 성공하면 Body 값 없이 성공 응답만 보내게 됩니다.