

REST API

박가람

이 강의에서는

- REST API 소개
- REST API 구성 요소
- REST API 규칙

REST API란?

Representational **S**tate **T**ransfer

웹의 구조적 스타일(Web's architectural Style)

웹의 구조적 스타일(Web's architectural Style)

1. 클라이언트/서버(Client / Server)
2. 정형화된 인터페이스 (Uniformed Interface)
3. 계층 시스템 (Layered System)
4. 캐시 (Cache)
5. 상태 없음 (Stateless)
6. 주문형 코드 (code-on-command)

클라이언트 / 서버 (Client / Server)

서로는 어떻게 구현되어있는지 알 수 없고 '일관적 규칙'으로 통신
클라이언트와 서버는 독립적

정형화된 인터페이스 (Uniformed Interface)

1. 리소스 식별 - URI(Unified Resource Identity)
2. 표현을 통한 리소스 처리 - HTML/JSON/XML
3. 자기서술적 메시지 - 요청은 메시지 / 헤더의 메타데이터

계층 시스템 (Layered System)

- 네트워크에 중간 메체 강화
 - 미들 웨어
 - 보안(HTTPS)
- 부하 분산
 - 로드 밸런싱

캐시 (Cache)

콘텐츠 전송망 **CDN** 단, 또는 클라이언트 내에도 가능

상태 없음 (Stateless)

- 서버가 클라이언트의 상태를 관리할 필요가 없음
- 필요하다면 해당 상황에 대한 정보를 직접 관리

주문형 코드 (code - on - command)

선택 사항 - 자바 스크립트 코드 등과 같은 것

URI 식별자 설계

URI 기본 원칙

1. 마지막 문자로 '/' 를 포함하지 않는다.
2. 가독성을 높이는 데에는 '-'를 사용한다.
3. 소문자만 사용
4. URI 에는 확장자를 넣지 않는다.

URI vs URL

URI : Uniform Resourc Identifier

URL : Uniform Resourc Locator

리소스 형식 (Resource Type)

- 도큐먼트 (Document)
- 컬렉션 (Collection)
- 스토어 (Store)
- 컨트롤러 (Controller)

도큐먼트 (Document)

가장 기본이 되는 리소스 형식

데이터 베이스의 레코드와 같은 것

ex) `http://api.your-service-books.com/books/1`

컬렉션 (Collection)

도큐먼트의 디렉터리 리소스

도큐먼트의 리스트

ex) <http://api.your-service-books.com/books>

스토어 (Store)

클라이언트가 특별히 관리하는 형태를 갖는 것

favorites, mark, done

ex) <http://api.your-service-books.com/users/1/favorites>

컨트롤러 (Controller)

CRUD 이외의 것

ex) `http://api.your-service-books.com/books/1/buy`

명명 규칙 (Conventional Naming Rules)

명명 규칙 (Conventional Naming Rules)

- **CRUD 는 URI 에 표시 X**
 - **도큐먼트 : 단수**
 - 컬렉션 : 복수
 - 스토어 : 복수
 - **컨트롤러 : 동사/동사구**
 -

Query * 선택 사항 *

- 페이지네이션
 - 타입
 - 소팅

HTTP 프로토콜 이용

HTTP 프로토콜의 이용

- 요청 메서드
- 응답 상태 코드
- 메시지 헤더

요청 메서드

- **GET** : 리소스 상태
- **HEAD** : 리소스의 메타 데이터
- **PUT** : '스토어 리소스' 및 '리스소' 갱신
- **POST** : 컬렉션에 리소스 추가 / 스토어에 리소스 추가 / 제한 없는 사용
 - 가지고 오는 것(get) x
 - 저장하는 것(put:store) x
 - 지우는 것 (delete) x
- **DELETE** : 리소스 제거

응답 상태 코드

- 2xx : 정상 처리 (성공)
- 3xx : 실패하지 않은 요청 중 비정상 처리 (변경 및 이동)
- 4xx : 비정상 처리 (실패)
- 5xx : 서버 에러

200/201/202/204

- 200 : 일반적인 성공 코드(에러메시지는 200 으로 보내서는 절대 안된다)
- 201 : 리소스 생성 성공
- 202 : 비동기 처리 성공
- 204 : 의도적으로 응답 바디가 없는 것

301

301 : 재설계된 리소스 헤더에 **Location** 헤더에 **URI** 기술

4xx

400 : 일반적인 요청 실패

401 : Unauthorized 인증 안됨 / 누구인지 모름

403 : 권한 없음 / 거부된 , 인증 되었지만 권한이 없음 / 누군지는 아는데 보여줄수는 없다

404 : 리소스 없음

405 : 메서드 지원 안함

406 : 받아들일 수 없음 (Not Acceptable)

409 : 불가능한 요청 (Conflict)

412 : 조건부 연산

415 : Content-type 처리 못 하는 타입이다.

표현 (Representation) - JSON 이 대세

JSON

요구 사항

도서 관리 시스템

- 책의 생성 / 조회 / 변경 / 삭제 -> resource 및 collection
- 찜하기(마크) -> store
- 대출 / 반납 -> controller

요구 사항

1. 도서 CRUD 연산
2. 도서에 mark 표시
3. 대출/반납