Kotlin Spring Boot MVC

by steve


Web 이란 무엇인가?

Web 이란?

(World Wide Web, WWW, W3)은 인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간을 말한다

Web의 용도는 다양하게 나눌 수 있습니다.

그중에서 우리가 제일 많이 접하는 부분을 정리 해보겠습니다.

1. Web Site
   google, naver, daum, yahoo etc...

2. User Interface
   Chrome, Safari, Explorer, Smart Watch, IP TV etc...

3. API (Application Programming Interface) * Web Service
   Kakao Open API, Google Open API, Naver Open API etc...

Web의 기반

HTTP
Hypertext Transfer Protocol

어플리케이션 컨트롤

GET
POST
PUT
DELETE
OPTIONS
HEAD
TRACE
CONNECT

의 Method가 존재

URI
Uniform Resource Identifier

리소스 식별자

특정 사이트
특정 쇼핑 목록
동영상 목록

모든 정보에 접근 할 수 있는 정보

HTML
Hyper Text Markup Language

하이퍼미디어 포맷

XML을 바탕으로한 범용 문서 포맷

이를 이용하여 Chrome, Safari, Explorer에서 사용자가 알아보기 쉬운 형태로 표현


REST

REST

REST (Representational State Transfer, 자원의 상태 전달)
네트워크 아키텍처 원리

1. Client , Server : 클라이언트와 서버가 서로 독립적으로 분리되어져 있어야 한다.

2. Stateless : 요청에 대해서 클라이언트의 상태가 서버에 저장을 하지 않는다.

3. 캐시 : 클라이언트는 서버의 응답을 캐시 할 수 있어야 한다.
   클라이언트가 캐시를 통해서 응답을 재사용할 수 있어야 하며, 이를 통해서 서버의 부하를 낮춘다.

4. 계층화 (Layered System) : 서버와 클라이언트 사이에, 방화벽, 게이트웨이, Proxy 등 다계층 형태를 구성할 수 있어야 하며, 확장 할 수 있어야 한다.

5. 인터페이스 일관성 : 아키텍처를 단순화시키고 작은 단위로 분리하여서,
   클라이언트, 서버가 독립적으로 개선될 수 있어야 한다.

6. Code On Demand (optional) 자바 애플릿, 자바스크립트 플래시 등 특정기능을
   서버가 클라이언트에 코드를 전달하여 실행 할 수 있어야 한다.

REST

인터페이스의 일관성 : 인터페이스 일관성이 잘 지켜졌는지에 따라 REST를 잘 사용했는지 판단을 할 수 있다.

1. 자원 식별
2. 메시지를 통한 리소스 조작
3. 자기 서술적 메시지
4. 애플리케이션 상태에 대한 엔진으로서 하이퍼미디어

REST

[자원식별]

웹 기반의 REST에서는 리소스 접근을 URI를 사용합니다.

https://foo.co.kr/user/100

Resource : user
식별자 : 100

REST

[메시지를 통한 리소스 조작]

Web에서는 다양한 방식으로 데이터를 전송 할 수 있습니다.

그중에는 HTML, XML, JSON, TEXT 등등 다양한 방법이 있습니다.

이 중에서 리소스의 타입을 알려주기 위해서 header 부분에 content-type 를 통해서 어떠한 타입인지를 지정할 수 있습니다.

REST

[자기서술적 메시지]

요청 하는 데이터가 어떻게처리 되어져야 하는지 충분한 데이터를 포함 할 수 있어야 합니다.

HTTP 기반의 REST에서는 HTTP Method 와 Header의 정보로 이를 표현할 수 있습니다.

REST

[애플리케이션 상태에 대한 엔진으로서 하이퍼미디어]

REST API를 개발할때에도 단순히 Client 요청에 대한 데이터만 내리는것이 아닌 관련된 리소스에 대한 Link 정보까지 같이 포함되어야 한다.

이러한 조건들을 잘 갖춘 경우 REST Ful 하다고 말하고 이를 REST API 라고 부릅니다.

URI 설계 (REST API 설계)

URI

1. URI (Uniform Resource Identifier)
   인터넷에서 특정 자원을 나타내는 주소값. 해당 값은 유일합니다.
   ex : https://www.foo.co.kr/resource/sample/1
   response : sample1.pdf, sample2.pdf, sample.doc

2. URL (Uniform Resource Locator)
   인터넷 상에서의 자원, 특정 파일이 어디에 위치하는지 식별 하는 주소
   ex : https://www.foo.co.kr/sample1.pdf

URL는 URI의 하위 개념입니다.

URI 설계

URI 설계원칙 (RFC-3986)

- 슬래시 구분자 (/)는 계층 관계를 나타내는 데 사용한다.
  https://foo.co.kr/vehicles/suv/q6
- URI 마지막 문자로 (/) 는 포함하지 않는다.
  https://foo.co.kr/vehicles/suv/q6/
- 하이픈(-)은 URI가독성을 높이는데 사용한다
  https://foo.co.kr/vehicles/suv/q-series/6
- 밑줄(_)은 사용하지 않는다.
  https://foo.co.kr/vehicles/suv/q_series/6

URI 설계

- URI 경로에는 소문자가 적합하다.
  https://foo.co.kr/vehicles/suv/q6 (O)
  https://Foo.co.kr/Vehicles/SUV/Q6 (X)

- 파일 확장자는 URI에 포함하지 않는다.
  https://foo.co.kr/vehicles/suv/q6.jsp

- 프로그래밍 언어에 의존적인 확장자를 사용하지 않는다.
  https://foo.co.kr/vehicles/suv/q6.do

- 구현에 의존적인 경로를 사용하지 않는다.
  https://foo.co.kr/servlet/vehicles/suv/q6

URI 설계

- 세션 ID를 포함하지 않는다.
  https://foo.co.kr/vehicles/suv/q6?session-id=abcdef
- 프로그래밍 언어의 Method명을 이용하지 않는다.
  https://foo.co.kr/vehicles/suv/q6?action=intro
- 명사에 단수형 보다는 복수형을 사용해야 한다. 컬렉션에 대한 표현은 복수로 사용
  https://foo.co.kr/vehicles/suv/q6
- 컨트롤러 이름으로는 동사나 동사구를 사용한다.
  https://foo.co.kr/vehicles/suv/q6/re-order

URI 설계

- 경로 부분 중 변하는 부분은 유일한 값으로 대체 한다.
  https://foo.co.kr/vehicles/suv/q7/{car-id}/users/{user-id}/release
  https://foo.co.kr/vehicles/suv/q7/117/users/steve/release

- CRUD 기능을 나타내는것은 URI에 사용하지 않는다.
  GET : https://foo.co.kr/vehicles/q7/delete/{car-id} (X)
  DELETE : https://foo.co.kr/vehicles/q7/{car-id} (O)

- URI Query Parameter 디자인
  - URI 쿼리 부분으로 컬렉션 결과에 대해서 필터링 할 수 있다.
    https://foo.co.kr/vehicles/suv?model=q7

  - URI 쿼리는 컬렉션의 결과를 페이지로 구분하여 나타내는데 사용한다.
    https://foo.co.kr/vehicles/suv?page=0&size=10&sort=asc

URI 설계

- API에 있어서 서브 도메인은 일관성 있게 사용해야 한다.
  https://foo.co.kr
  https://api.foo.co.kr

- 클라이언트 개발자 포탈 서브 도메인은 일관성 있게 만든다.
  https://dev-api.foo.co.kr/vehicles/suv/q6
  https://developer-api.foo.co.kr/vehicles/suv/q6


HTTP

HTTP

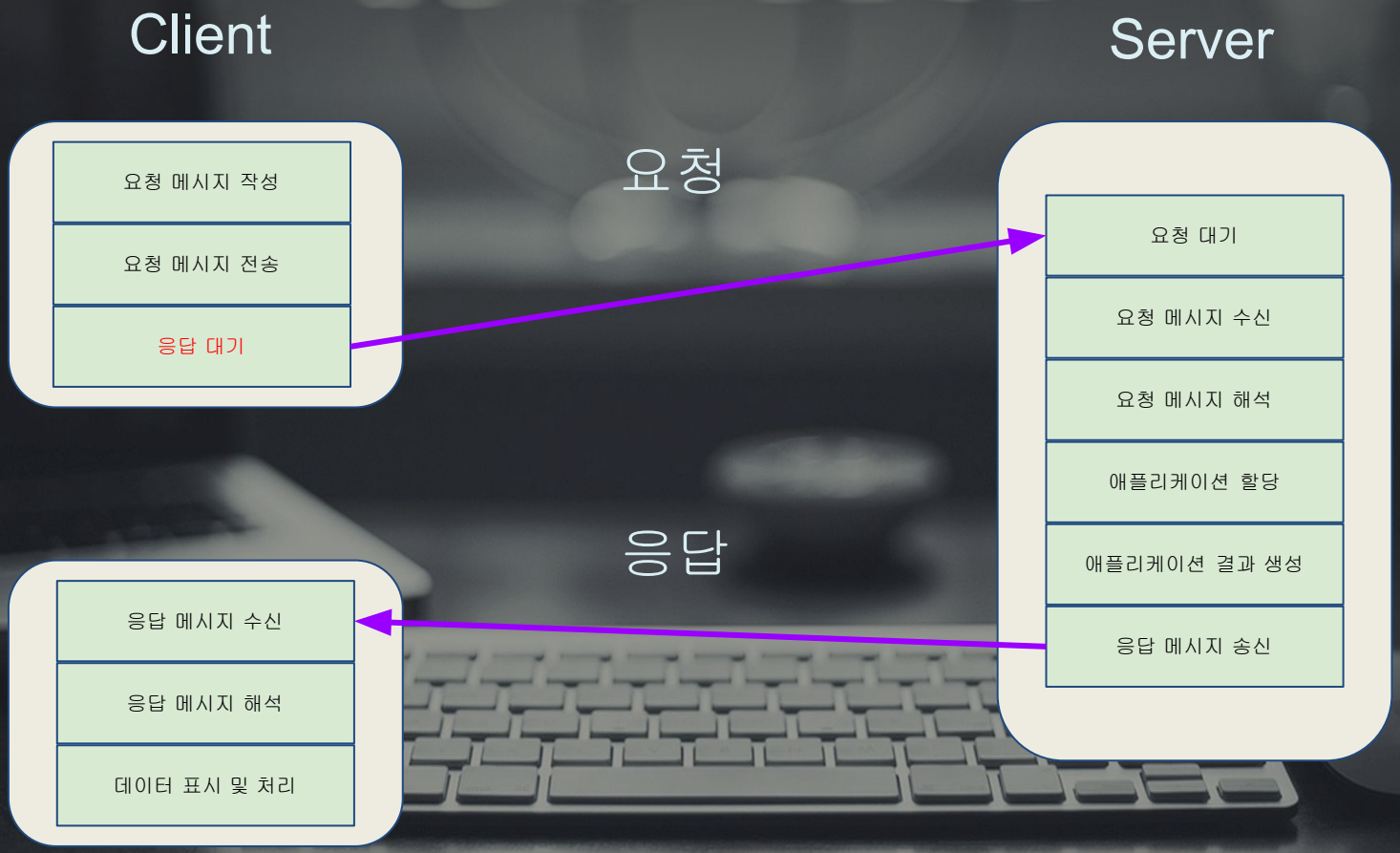
1. HTTP (Hyper Text Transfer Protocol) 로 RFC 2616에서 규정된 Web에서 데이터를 주고 받는 프로토콜.

2. 이름에는 하이퍼텍스트 전송용 프로토콜로 정의되어 있지만 실제로는 HTML, XML, JSON, Image, Voice, Video, Javascript, PDF 등 다양한 컴퓨터에서 다룰 수 있는 것은 모두 전송 할 수 있습니다.

3. HTTP는 TCP를 기반으로한 REST의 특징을 모두 구현하고있는 Web기반의 프로토콜

HTTP

HTTP는 메시지를 주고 (Request) 받는 (Response) 의 형태의 통신 방식 입니다.



HTTP

HTTP Method

HTTP의 요청을 특정하는 Method는 8가지가 있습니다. REST를 구현하기 위한 인터페이스이니 알아둬야 합니다.

	의미	CRUD	멱등성	안정성	Path Variable	Query Parameter	DataBody
GET	리소스 취득	R	O	O	O	O	X
POST	리소스 생성, 추가	C	X	X	O	△	O
PUT	리소스 갱신, 생성	C / U	O	X	O	△	O
DELETE	리소스 삭제	D	O	X	O	O	X
HEAD	헤더 데이터 취득	-	O	O	-	-	-
OPTIONS	지원하는 메소드 취득	-	O	-	-	-	-
TRACE	요청메시지 반환	-	O	-	-	-	-
CONNECT	프록시 동작의 터널 접속으로 변경	-	X	-	-	-	-

HTTP

HTTP Status Code
응답의 상태를 나타내는 코드

	의미	내용
1XX	처리중	처리가 계속 되고 있는 상태. 클라이언트는 요청을 계속 하거나 서버의 지시에 따라서 재요청
2XX	성공	요청의 성공
3XX	리다이렉트	다른 리소스로 리다이렉트. 해당 코드를 받았을때는 Response의 새로운 주소로 다시 요청
4XX	클라이언트 에러	클라이언트의 요청에 에러가 있는 상태. 재전송 하여도 에러가 해결되지 않는다.
5XX	서버에러	서버 처리중 에러가 발생한 상태. 재전송시 에러가 해결 되었을 수도 있다.

HTTP

자주 사용되는 Code

200	성공
201	성공. 리소스를 생성 성공
301	리다이렉트, 리소스가 다른 장소로 변경됨을 알림
303	리다이렉트, Client에서 자동으로 새로운 리소스로 요청 처리
400	요청 오류, 파라미터 에러
401	권한 없음 (인증실패)
404	리소스 없음 (페이지를 찾을 수 없음)
500	서버 내부 에러 (서버 동작 처리 에러)
503	서비스 정지 (점검 등등)


마치며...
by steve