2주차 과제

1. HTTP 통신

Hyper Text Transfer Protocol이란 뜻으로 웹 브라우저와 웹 서버가 HTML로 작성된 웹페이지나 동영상, 음성 파일 등을 주고받기 위한 프로토콜이다.

클라이언트가 서버에 요청 메시지를 보내고 이에 대해 서버가 응답 메시지를 반환한다.

상태가 없는 프로토콜이다. 데이터를 주고받기 위한 요청이 서로 독립적으로 관리가 된다.

전송 계층 프로토콜로 TCP를 사용하고 네트워크 계층 프로토콜로 IP를 사용하는 것이 일반적이다.

80번 포트를 사용한다.

2. 브라우저에 URL을 입력 후 요청하여 서버에서 응답하는 과정까지에 대해 학습한 내용

URL로 요청을 보내면 클라이언트인 웹 브라우저를 통해 서버에 URL의 해당 페이지를 보여 달라는 요청을 하는 것이고 서버는 요청을 받으면 해당 페이지의 HTML 파일을 클라이언트에 응답한다.

HTTP 통신은 연결되어 있지 않다. 따라서 클라이언트가 서버에 요청을 보내면 그제서야 3-way handshake 방식으로 연결이 되고 서버가 클라이언트의 요청에 맞는 응답을 한 뒤 4-way handshake 방식으로 연결을 해제한다.

3. REST API

1) 정의

REST API란 REST 아키텍처의 제약 조건을 준수하는 API다. REST는 Representational State Transfer의 줄임말이다. REST는 프로토콜이나 표준이 아닌 아키텍처다.

REST의 요소로는 크게 리소스, 메소드, 메시지 3가지가 있다.

REST에서는 HTTP 메소드인 GET, POST, PUT, DELETE를 그대로 사용한다.

2)특징

Stateless 성격을 갖는다. 상태 정보를 따로 저장하고 관리하지 않기 때문에 세션 정보나 쿠키 정보를 별도로 저장하지 않고 들어오는 요청만을 단순히 처리하면 된다. 때문에 서비스의 자유도가 높아지고 서버에서 불필요한 정보를 관리하지 않음으로써 구현이 단순해진다.

웹에서 사용하는 기존 인프라를 그대로 사용 가능하다. 따라서 캐싱 구현이 가능하다.

REST API 메시지만 보고도 쉽게 이해 할 수 있는 자체 표현 구조로 되어 있다.

Client-Server 구조라서 Client와 Server에서 개발해야 할 내용이 명확해지고 서로간 의존성이 줄어들게 된다.

3)규칙

URI는 정보의 자원을 꼭 표현해야 한다.

리소스명은 동사보다는 명사를 사용해야 한다.

슬래시는 계층 관계를 나타내는 데 사용하고 마지막에는 슬래시를 쓰지 않는다. 하이픈은 URI 가독성을 높이는 데 사용한다.

밑줄은 사용하지 않는다.

URI 경로에는 소문자가 적합하다. 파일확장자는 URI에 포함시키지 않는다. 대신 Accept header를 사용한다.

연관관계를 표현할 수 있다. Ex) GETE : /users/{userid}/devices ( has 표현 )

리소스를 표현하는 Collection과 Document가 있다. Collection은 문서들의 집합, 객체들의 집합이고 Document가 문서, 객체다. URI에 포함된다.

HTTP 응답 상태 코드를 표현해 준다.