http 통신에 대하여

HTTP(하이퍼텍스트 전송 프로토콜, HyperText Transfer Protocol)는 웹에서 클라이언트와 서버가 데이터를 주고받기 위한 기본적인 프로토콜로 텍스트 기반의 통신 규약이다.

http는 클라이언트-서버 모델을 기반으로 어플리케이션 레벨의 프로토콜로 TCP/IP위에서 작동하며 클라이언트가 HTTP messages 양식에 맞춰서 요청(Request)을 보내면 서버에서는 HTTP messages 양식에 맞춰서 응답(Response)을 보내며 통신하게 된다.

이때 클라이언트는 서버의 IP주소와 포트넘버를 알고 있어야 한다.

HTTP 특징

HTTP의 특징으로는 크게 클라이언트-서버 구조, 무상태성, 비연결성으로 되어 있다.

1) 클라이언트-서버 구조 : 클라이언트가 요청을 보내면 서버가 응답하는 방식으로 동작하며, 이 요청과 응답 사이에는 다양한 작업을 수행하는 여러 개체들이 있는데 예를 들면 게이트웨이, 캐시, 프록시 등이 있다. 이런 클라이언트와 서버사이의 통신 설계방식을 2티어 architecture이라고 하며 여기에 DB가 추가된 방식을 3티어 architecture이라고 한다.

2) 무상태성 : 서버가 클라이언트의 상태를 저장하지 않는 것을 의미하며 이로 인해 클라이언트의 요청은 독립적인 요청들로 처리되며 서버에서 장애가 일어나더라도 다른 서버에서 응답을 전달할 수 있다. 이는 서버의 리소스 소모를 줄이고 통신을 단소화하여 처리속도를 올린다.

3) 비연결성 : HTTP는 클라이언트가 요청을 보내고 서버가 응답을 하게 되면 연결을 끊는다. 따라서 서버는 최소한의 자원만 사용하게 된다. 그러나 새로이 연결하는 과정이 많아지면 시간이 추가로 소요될 수 있다.

HTTP Message

HTTP 메세지는 서버와 클라이언트 간에 데이터가 교환되는 방식이고 메세지 타입은 두가지가 있다. 요청은 클라이언트가 서버에게 전달하여 서버에서 액션이 일어나게 하는 메세지이고, 응답은 요청에 대한 서버의 답변 정도이다. 이 두가지의 메세지들은 서로 구조가 비슷하며 그 구조는 start-line, HTTP header, Empty line, body로 구성되어있다.

1) HTTP 요청(Request) : 클라이언트 -> 서버로 보내는 메세지

# Start Line - Method, Request target, Version으로 구성되어 있고 Method는 GET, POST, PUT 등으로 수행할 작업이나 HEAD, OPTIONS 등으로 방식을 알려준다. request target(path)는 요청이 전송되는 목표 주소이다. version은 각 버전에 따라 메세지 구조나 데이터가 다를 수 있어서 명시되어 있다.

# Header - 헤더는 클라이언와 서버가 요청 또는 응답으로 부가적인 정보를 전송할 수 있도록 해준다. 헤더는 크게 4가지로 나뉘며 General header, request header, response header, Entity header로 나뉜다. General은 요청과 응답 모두 적용되나 바디에서는 최종적으로 전송되는 데이터와는 관련이 없는 헤더이고, request는 패치될 리소스나 클라이언트 자체에 대한 자세한 정보를 포함한다. response는 위치 또는 서버 자체에 대한 정보와 같이 응답에 대한 부가적인 정보를 갖는다. entity는 컨텐츠 길이나 MIME 타입과 같이 엔티티 바디에 대한 자세한 정보를 포함한다.

# Body - 요청의 본문은 메세지 구조의 마지막에 위치한다. 모든 요청에 바디가 필요한 것은 아니며 서버에 리소스를 요청하는 경우에는 본문이 필요하지 않는다. 보통 POST 메소드 타입이 데이터가 포함되어 있다.

2) HTTP 응답(Response) : 서버에서 클라이언트의 요청에 따라 액션을 취한 메세지

# Status line - 응답의 상태를 간략하게 나타내주는 방법으로 3가지 부분으로 구성된다. Version, Status Code, Status text로 되어 있다. 예) "HTTP/1.X 404 Not Found." 처럼 생겼다.

# Header - 요청 부분의 헤더와 동일하나 응답에서만 사용되는 헤더 값들이 있다.

#Body - 응답메세지의 body와 일반적으로 동일하며 데이터를 전송할 필요가 없을 경우에는 빈공란으로 표기된다.

HTTP Method

HTTP 메소드는 클라이언트가 서버에게 요청을 보낼때 사용되는 행동을 정의하는 방법이고 요청 헤더(Request Header)에 포함되며 서버는 이를 통해 클라이언트가 원하는 동작을 이해하고 요청을 처리한다.

#GET : 리소스 조회

- 서버에 전달하고 싶은 데이터는 쿼리를 통해 전달

- 메세지 바디를 사용해서 데이터를 전달할 수 있지만, 보안등의 문제로 인해 권장되지 않음.

#POST : 요청 데이터 처리

- 메세지 바디를 통해 서버로 요청 데이터를 전달하며 바디를 통해 들어온 데이터를 처리하는 모든 기능을 수행한다. 주로 전달된 데이터로 신규 리소스 등록, 프로세스 처리에 사용된다.

#PUT : 리소스를 대체

- 기존에 존재하는 리소스를 대체하고 없으면 새로 생성하게 된다.

#PATCH : 리소스 부분 변경

- 리소스의 일부분을 변경하는 메소드이다.

#DELETE : 리소스 삭제

- 저장된 리소스를 삭제한다.

HTTP Status Code

Status code는 IETF (Internet Engineering Task Force)에서 정의한 인터넷 표준에 따라 개발되며 다섯가지 클래스로 분류된다.

1xx 코드 : Information(요청 정보 처리 중)

- 서버가 요청을 클라이언트에서 성공적으로 수신했으며 서버 끝에서 처리 중이라는 정보를 나타낸다. 서버의 임시 응답이며 일반적으로 상태 줄과 선택적 헤더만 포함하며 빈줄로 끝난다.(현재는 거의 사용하지 않는다)

2xx 코드 : Success(요청을 정상적으로 처리함)

- 서버가 요청을 받고 요청을 성공적으로 처리되었음을 나타낸다.

3xx 코드 : Redirection(요청을 완료하기 위해 추가 동작 필요)

- 브라우저는 자동으로 다른 URL로 리디렉션 되므로 브라우저 창에는 이 코드가 표시되지 않지만 이미지 파일처럼 캐싱된 파일을 새로고침하면 3xx 코드를 확인할 수 있다.

4xx 코드 : Client Error(서버가 요청을 이해 못함.)

- 서버가 해결할 수 없는 클라이언트 측 에러 코드이다. 주로 클라이언트가 서버에 잘못된 요청을 했을 경우 발생한다.

5xx 코드 : Server Error(서버가 요청 처리 실패함)

- 서버가 클라이언트의 요청을 처리하지 못했을때 발생한다. 서버는 보안 상 통신하지 않는 것이 가장 좋으므로 대부분의 에러코드를 5xx번대 에러로 처리한다.