**\* Rest API**

1. **HTTP 통신에 관하여**

클라이언트가 HTTP request 를 서버에 보내면 서버는 요청을 처리해 HTTP response를 보내는 구조, HTTP는 상태 정보를 저장하지 않고 통신이 끝나면 연결을 끊는 특징을 가진다.

**HTTP Request 구조**

* start line
  + HTTP request 의 첫 라인
    - HTTP Method
      * 해당 request가 의도한 액션을 정의하는 부분
      * GET, POST, PUT, DELETE, OPTIONS 등
    - Request target
      * 해당 request가 전송되는 목표 URI
      * EX) 로그인
    - HTTP Version
      * 사용되는 HTTP 버전
* Headers
  + 해당 request에 대한 추가 정보를 담고 있는 부분
  + Request 메시지 body의 총 길이 등
  + Key:Value 형태
  + 크게 3부분으로 나뉨
  + 자주 사용되는 header 정보에는 Host (request가 전송되는 target의 host uri; ex. Google.com) 가 있다. start line의 target과 headers의 host를 합치면 고유 주소가 나온다.
* Body
  + 해당 request의 실제 내용
  + body가 없는 request도 많음.(ex. GET request)

**HTTP Response 구조**

* Status Line
  + Response 상태를 간략하게 나타내는 부분.
    - HTTP 버전
    - Status code : 응답 상태를 나타내는 코드
    - Status text : 응답 상태를 간략하게 설명해주는 부분
* Headers
  + 응답에 대한 메타 정보가 있는 곳
  + 응답을 주는 주체인 서버에 대한 정보가 들어가 있다.
  + 다만, response에서만 사용되는 header 값들이 있다. (Server 헤더)
* Body
  + Request의 body와 동일
  + body가 없는 response가 있음. (데이터를 전송할 필요가 없는 경우)

**HTTP Methods**

* GET
  + 데이터를 서버로부터 받아올 때 주로 사용하는 Method
  + 데이터 생성/수정/삭제 없이 받아 오기만 할 때 사용
  + 주로 데이터 받아올 때 사용하기 때문에 request에 body를 안 보내는 경우가 많다.
* POST
  + 데이터를 생성/수정/삭제할 때 사용되는 Method
  + 데이터를 생성 및 수정할 때 많이 사용하기 때문에 대부분의 경우 request body가 포함되어서 보내진다.

**2) 브라우저에 URL을 입력 후 요청하여 서버에서 응답하는 과정에 대해 학습한 내용을 작성**

URL 요청을 서버로 보내면 서버는 리소스와 HTTP Method에 따르는 데이터를 Json 이나 xml 형식으로 클라이언트에게 전달한다.