

# 2주차 과제

## REST API

자원을 URI 로 구분하고 HTTP 메서드(GET, POST, PUT, DELETE) 를 통해 자원에 대해 CRUD 를 할 수 있게 도와주는 인터페이스

- 장점
  - HTTP 를 그대로 사용하므로 쉽게 사용할 수 있다
  - 클라이언트와 서버 역할을 명확하게 구분할 수 있다
- 단점
  - 표준이 존재하지 않아 각각 정의가 필요하다
  - 사용할 수 있는 메서드가 제한적이다
  - 화면에 필요없는 데이터도 받아와야 한다

## HTTP

인터넷 환경에서 데이터를 주고받기 위해 도와주는 프로토콜

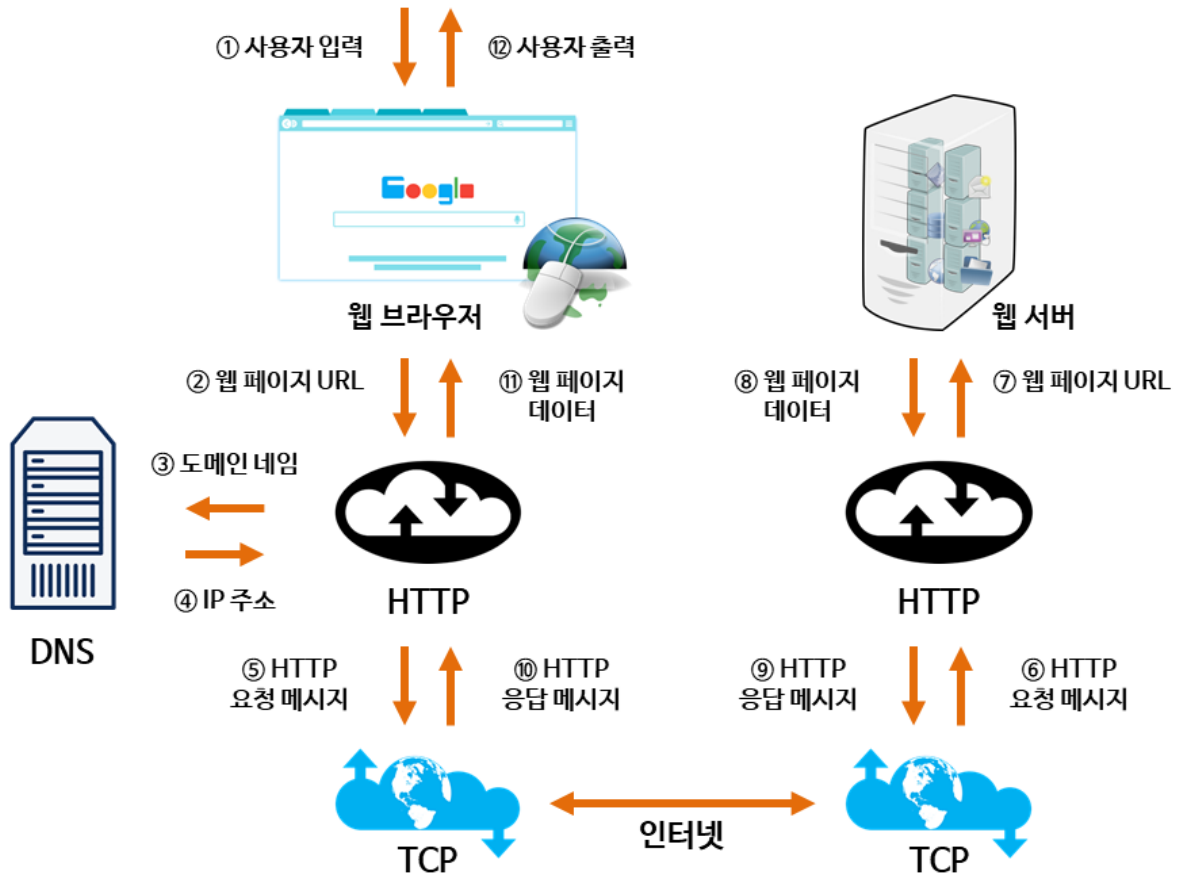
### 특징

- **비연결성** : 클라이언트 요청에 서버가 응답을 한 뒤 연결을 종료하여 서버의 리소스를 효율적으로 사용할 수 있다
- **비상태성** : 클라이언트 상태를 서버가 저장하지 않는다
- TCP/IP 를 이용하는 **응용계층 프로토콜** 이다

### 통신과정

- HTTP 의 통신과정은 클라이언트의 요청과 서버의 응답으로 이루어져 있다.
- **클라이언트 요청 메시지**
  - **Method** : 리소스를 가져오기 위한 방법을 명시 (GET, POST, PUT, DELETE .. )
  - **Path** : 가져오려는 리소스의 경로를 명시 (URL, URI)
  - Version : 프로토콜의 버전을 명시
  - Header : 서버에게 추가 정보를 전달하기 위해 명시
- **서버 요청 메시지**
  - Version : 프로토콜 버전을 명시
  - **Status Code** : 요청 결과를 명시

## HTTP 의 동작과정



1. 사용자가 웹 페이지 URL 을 입력을 한다.
  - 만약 호스트 파일에서 도메인의 ip 주소가 존재한다면 DNS 에 질의하지 않고 바로 원격지의 서버에게 요청할 수 있다
2. DNS 서버에게 도메인의 IP 를 질의 후 IP 주소를 응답받음
3. 응답받은 ip 주소와 도메인 이름을 확인하여 HTTP 요청 메세지 생성 후 전송
4. 서버에 HTTP 요청이 도착
5. 서버는 클라이언트의 IP 를 알고있기 때문에 바로 HTTP 응답 메세지를 생성 후 전송