**HTTP 통신**

**HTTP(Hyper Text Transfer Protocol)**: 서버-클라이언트 간 데이터를 주고 받기 위한 프로토콜(**80번 포트 사용**)

프로토콜: 컴퓨터들 간 정의한 규칙, 특정 기기 간 데이터를 주고받기 위한 규약

**HTTPS(Hyper Text Transfer Protocol Secure)**: HTTP에 **보안(암호화)이 추가**된 프로토콜

**HTTP 메서드 종류**

* **GET: 리소스 조회**
* **POST: 요청 데이터 처리, 주로 등록에 사용**
* **PUT: 리소스를 대체, 해당 리소스가 없으면 생성**
* **PATCH: 리소스 부분 변경**
* **DELETE: 리소스 삭제**

**HTTP 통신**

**HTTP 요청**

* ***클라이언트가 서버에게 어떠한 동작을 요청하기 위해 보내는 정보***
* 사용자가 원하는 작업에 따라 동작방식이 달라짐

ex) 구글 접속 요청HTTP 요청 메시지

GET (동작방식)

<https://www.google.com> (URL)

HTTP/1.1 (버전정보)

ex) 구글에서 “사과” 검색 시 HTTP 요청 메시지

GET

<https://www.google.com/search?q=>애플

HTTP/1.1

ex) 구글에서 회원가입 시 HTTP 요청 메시지

[요청 메시지 헤더]

POST

[www.google.com/members](http://www.google.com/members)

HTTP/1.1

[요청 메시지 본문]

name = 홍길동 &

userid = 홍홍 &

password = 111

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**HTTP 응답**

* ***요청에 맞는 정보를 찾아 다시 클라이언트에게 전송***
* 웹 브라우저는 응답 받은 페이지를 사용자에게 띄워 줌

**응답 메시지**

HTTP/1.1 200 OK (응답메시지 헤더)

(응답 메시지 본문)

<html>

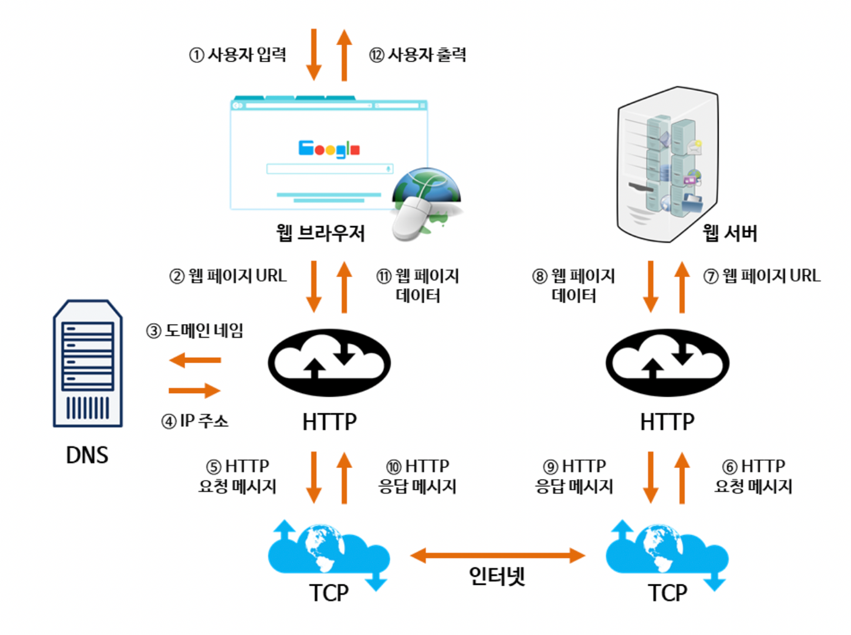
(구글 메인 페이지의 소스코드)

</html>

**주요 상태코드**

|  |  |  |
| --- | --- | --- |
| **상태코드** | **메시지** | **의미** |
| 200 | OK | 요청이 성공적으로 수행됨 |
| 404 | Not Found | 요청한 페이지 없음 |
| 400 | Bad Request | 사용자의 잘못된 요청 |
| 500 | Internal Server Error | 서버 내부 에러 |

**웹 동작 원리**



사용자가 동작을 입력하면 해당 도메인 네임에 해당하는 IP주소를 반환하여 HTTP 요청 메시지 전달 → 웹 서버는 입력에 해당하는 정보를 응답 메시지로 전달하여 응답 제공

**Rest API**

**REST(Representational State Transfer): 웹의 장점을 최대한 활용할 수 있는 아키텍처**

* **REST API 구성**
  + 자원(Resource) → URI
  + 행위(Verb) → HTTP Method
  + 표현(Representations)
* REST API 유의점
  + URI는 정보의 자원을 표현해야 함
  + 자원에 대한 행위는 GET, POST, PUT, DELETE 등 HTTP Method로 표현

**REST API 규칙**

* URI는 정보의 자원 표현해야 함 → 동사보다는 **명사 사용**

ex) GET /members

* **자원에 대한 행위는 HTTP Method로 표현**

ex) GET /members/delete/1 → DELETE /members/1

* 슬래시 구분자는 계층 관계 나타내는 데 사용

**단, URI 마지막 문자로 슬래시 포함 X**

* 하이픈(-)은 URI 가독성을 높이는데 사용

단, 밑줄(\_)은 사용하지 X

* URI 경로는 소문자가 적합

ex) http://api.example.com/firstProject X

ex) http://api.example.com/first-project O

* 파일 확장자는 URI에 포함하지 않음 → header에 넣기
* Collection과 Document 이용 → 단수, 복수 표현하기
  + Collection: 문서들의 집합, 객체들의 집합
  + Document: 문서

ex) http://restapi.example.com/sports/soccer