

# HTTP 정리

## Resource

- 회원가입 API일 경우 리소스는 회원
- 리소스와 행위를 구분하여 API를 설계 → 이러한 행위들은 HTTP Method들로 구분이 가능

## HTTP Method

### GET

- 리소스 조회
- 서버에 전달하고 싶은 데이터는 쿼리 파라미터를 통해 전달
- 메시지 바디에 데이터를 실어 보낼 수는 있지만 지원하는 서버가 적어 권장하지 않음

### POST

- 서버로 보낸 요청 데이터를 처리하는 메서드
- 메시지 바디를 통해 서버로 보낼 데이터를 전달
- 주로 새로운 리소스 생성 및 데이터 처리에 사용

## HTTP Method 속성

1. 안전 : GET, HEAD → 쉽게 말해서 해당 메서드를 호출해도 리소스가 변하지 않는다.
2. 멍등 : 한번 호출하든 두번 호출하든, 백번을 호출하든 응답값이 동일하다. 즉, 요청에 대한 응답이 없을 경우 이러한 요청을 몇번을 다시해도 응답값이 동일하여 서비스에 에러가 나지 않는 메서드 요청이다.
  - a. GET
  - b. PUT
  - c. DELETE

3. 캐시가능 : 응답 결과 리소스를 캐싱하여 사용가능한가

- a. GET
- b. HEAD

## HTML Form 전송

- POST : 바디에 data 싣어서 보냄
- GET : 쿼리 파라미터에 데이터 싣어서 보냄

## HTTP 헤더

- HTTP 전송에 필요한 모든 부가정보를 가지고 있다.
- 필요시 사용자 임의의 헤더를 추가할 수 있다.
- 기본적인 구조 그림

GET /search?q=hello&hl=ko HTTP/1.1	HTTP/1.1 200 OK
Host: www.google.com	Content-Type: text/html; charset=UTF-8
	Content-Length: 3423
	<html>
	<body>...</body>
	</html>

- RFC 7230 스펙에서 Entity → Representation이라는 표현으로 바뀜
- 그래서 헤더가 표현 헤더로 변경
- 바디는 표현 데이터로 변경됨
- 표현 헤더는 표현 데이터를 해석할 수 있는 정보를 제공함, 즉 메타 데이터라고 생각해도 됨
- 표현 헤더는 요청, 응답 시 둘다 사용됨

## Content-Negotiation

- Client가 선호하는 표현을 요청하는 헤더
- 그래서 HTTP 요청 시에만 사용가능하다.
- 협상은 헤더에 담긴 Quality Value의 값들을 기준으로 설정되고 가중치가 높으면 1순위이고 해당 하는 표현을 서버가 지원하지 않을 경우 그 다음 순위를 응답해준다.
- 다음과 같이 4가지 종류의 헤더가 존재함

- **Accept:** 클라이언트가 선호하는 미디어 타입 전달
- **Accept-Charset:** 클라이언트가 선호하는 문자 인코딩
- **Accept-Encoding:** 클라이언트가 선호하는 압축 인코딩
- **Accept-Language:** 클라이언트가 선호하는 자연 언어