**Restful API 와 HTTP 통신**

* HTTP? HyperText Transfer Protocol. 웹 환경에서 정보를 주고받기 위한 프로토콜
  + 클라이언트는 HTTP의 상태 코드를 확인하여 오청의 성공, 실패를 확인
  + HTTP를 사용하는 클라이언트와 서버 간의 약속, 프로토콜
* REST? (Representational State Transfer). 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처
* RESTful API? 두 컴퓨터 시스템이 인터넷을 통해 정보를 안전하게 교환하기 위해 사용하는 인터페이스
  + 확장성, 유연성, 독립성
* API? (Application Programming Interface) 다른 소프트웨어 시스템과 통신하기 위해 따라야 하는 규칙을 정의
* 클라이언트? 웹에서 정보에 액세스하려는 사용자
* 리소스? 다양한 애플리케이션이 클라이언트에게 제공하는 정보
* HTTP는 웹 환경에서 정보를 송수신할 때 사용하는 약속, REST는 소프트웨어 아키텍처
  + REST에서 클라이언트-서버 통신을 할 때 HTTP를 사용
  + 웹 환경 통신의 대부분이 HTTP를 사용, 따라서 REST에서도 HTTP가 필수가 아니지만 그것이 일상적임.
* HTTP Status Code(상태코드)
  + HTTP 요청이 성공했는지 실패했는지를 서버에서 알려주는 코드
  + 예제

Table

Description automatically generated

* + - 200번대 상태 코드는 서버가 클라이언트 요청을 성공적으로 처리했다는 의미
* 200 OK

Graphical user interface, text, application

Description automatically generated

* + 상태코드는 200으로 성공인데 내용엔 실패에 관한 내용을 리턴하는 경우가 있음.
  + 모든 응답을 200으로 처리하고 body 내용으로 성공, 실패를 판단하는 구조에서 사용함(웹이서는 문제가 되지 않으나, API에서는 이상한 구조임)
* 201 Created
  + POST, PUT 요청에 대한 응답에 주로 사용
  + 성공과 동시에 새로운 리소스가 생성되었다는 의미를 포함(정확한 의미 전달)
* HTTP 헤더의 Content-Location를 이용하여 만들어진 리소스 생성된 위치를 알려주면 좋음.

Graphical user interface, text, application

Description automatically generated

* 202 Accepted
  + Client의 요청은 정상적이나, 서버가 요청을 완료하지 못한 상태
* 204 No Content
  + 클라이언트의 요청은 정상적이나, 컨텐츠를 제공하지 않는 상태
    - 흔하지 않음
* 400 Bad Request
  + 클라이언트의 요청이 유효하지 않아 더 이상 작업을 진행하지 않는 경우
  + 서버에서 잘못되었다고 판단하는 응답.
  + 서버는 그 요청이 왜 잘못되었는지 알 수 있지만 클라이언트는 알기 어려운 내용
* 401 Unauthorized
  + 클라이언트가 권한이 없기 때문에 작업을 진행할 수 없는 경우
  + 인증에 대한 부분
* 403 Forbidden
  + 클라이언트가 권한이 없기 때문에 작업을 진행할 수 없는 경우
* 404 Not Found
  + 클라이언트가 요청한 자원이 존재하지 않는 경우
  + 자원에 대한 존재 여부를 파악
  + 경로에 대한 요청이라 할지라도 자원이 존재하는 지 파악해야 함.
* 405 Method Not Allowed
* 409 Conflict
  + 클라이언트의 요청이 서버의 상태와 충돌이 발생한 경우
* 429 Too Many Request
  + 클라이언트가 일정 시간 동안 너무 많은 요청을 보낸 경우