# HTTP란

서로 다른 시스템들 사이에서 통신을 주고 받게 해주는 application 계층의 protocol

웹에서만 사용하는 프로토콜로 TCP/IP 기반으로 서버와 클라이언트 간의 요청과 응답을 전송함

## HTTP 특징

1. Transfer layer

TCP/IP 기반으로 서버와 클라이언트 간의 요청과 응답을 전송한다.

TCP, UDP란?

1. 비 연결성 (connectionless)

클라이언트와 서버가 한번 연결을 맺은 후에 클라이언트의 요청에 대해 서버가 응답을 마치면 연결을 끊어버림

1. 무상태성 (stateless)

Connectionless로 인해 서버가 두 요청 간의 어떠한 데이터도 유지하지 않음

즉, 서버는 클라이언트와 연결에 대한 정보를 저장하지 않아 서버는 클라이언트를 식별하지 못함

# HTTPS가 무엇인고 이를 HTTP와 비교하라

HTTPS (하이퍼텍스트 전송 프로토콜 보안): 웹 브라우저와 웹 사이트 간에 데이터를 전송하는 데 사용되는 기본 프로토콜인 HTTP의 보안 버전

* HTTPS는 데이터 전송의 보안을 강화하기 위해 암호화됨 이는 사용자가 은행 계좌, 이메일 서비스, 의료 보험 공급자에 로그인하는 등 중요한 데이터를 전송할 때 특히 중요함
* 일반 HTTP를 통해 정보를 전송할 때, 정보는 무료 소프트웨어를 사용하여 쉽게 '스니핑'할 수 있는 데이터 패킷으로 나뉨 따라서 공용 Wi-Fi와 같이 안전하지 않은 매체를 통한 통신은 도청에 매우 취약함 실제로 HTTP를 통해 발생하는 모든 통신은 일반 텍스트로 이루어지므로 올바른 도구만 있으면 누구나 쉽게 접근할 수 있으며 경로상 공격에 취약함

- HTTPS를 사용하면 트래픽이 암호화되므로 패킷을 스니핑하거나 가로챈다고 해도 무의미한 문자로만 인식됨

ex) 암호화 전:

완전히 읽을 수 있는 텍스트 문자열입니다

암호화 후:

ITM0IRyiEhVpa6VnKyExMiEgNveroyWBPlgGyfkflYjDaaFf/Kn3bo3OfghBPDWo6AfSHlNtL8N7ITEwIXc1gU5X73xMsJormzzXlwOyrCs+9XCPk63Y+z0=

# 아래 API에 대한 RESTful한 URI를 설계하라.

이벤트 목록 조회 : GET /api/events

이벤트 조회: GET /api/events/{event\_id}

이벤트 등록: POST /api/events

이벤트 수정: PUT /api/events/{event\_id}

//PUT과 PATCH

PUT: 클라이언트가 리소스의 전체 표현을 업데이트하려고 할 때 사용 즉, 클라이언트가 전체 리소스의 상태를 제공하며, 서버는 해당 리소스를 주어진 정보로 완전히 교체함 따라서 PUT을 사용할 때는 클라이언트가 모든 필수 필드를 포함하고, 비변경 필드에 대해서도 업데이트를 수행해야 함

PATCH: 클라이언트가 리소스의 일부만 업데이트하려고 할 때 사용됨. 즉, 클라이언트는 업데이트할 필드와 해당 필드의 새 값을 제공함. 서버는 해당 필드만 업데이트하고 나머지는 그대로 둠

이벤트 삭제: DELETE /api/events/{event\_id}

이벤트 상태 변경: PATCH /api/events/{event\_id}/status

특정 이벤트의 주문 목록 조회: GET /api/events/{event\_id}/orders

멤버 목록 조회: GET /api/members

특정 멤버 권한 변경: PATCH /api/members/{member\_id}/permissions

특정 멤버 정보 조회: GET /api/members/{member\_id}

특정 멤버 정보 변경: PUT /api/members/{member\_id}

멤버 등록: POST /api/members