

## ● Rest

- 정의 : REST API는 웹 서비스에서 클라이언트와 서버 간의 통신을 가능하게 하는 것으로, REST는 Representational State Transfer의 약자입니다. 이는 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처 스타일을 뜻합니다
- 구성 요소 : 자원(Resource): URI (Uniform Resource Identifier)  
행위(Verb): HTTP Method  
표현(Representations)
- 정리 : 즉, REST는 URI를 통해 자원을 명시하고, HTTP 메소드(GET, POST, PUT, DELETE 등)를 통해 해당 자원에 대한 CRUD(Create, Read, Update, Delete) 연산을 적용하는 방식을 따르고 있습니다.

또한, REST API는 stateless하게 설계되어야 하는데, 이는 각 요청이 서로 독립적이라는 의미입니다. 즉 서버는 클라이언트의 상태 정보를 저장하고 있지 않으며, 각 요청에 필요한 모든 정보는 요청 내에 포함되어 있어야 합니다.

이런 특징 덕분에 REST API는 확장성이 좋고, 여러 플랫폼에서 사용하기 적합하며, 이를 통해 다양한 서비스와 정보를 쉽게 제공할 수 있습니다.

## ● HTTP 통신

HTTP(Hypertext Transfer Protocol)는 웹에서 데이터를 주고받는 프로토콜입니다. 클라이언트와 서버 간에 이루어지는 요청(Request)과 응답(Response)의 형태로 통신이 이루어집니다.

클라이언트가 서버에게 요청(Request)을 보냅니다. 이는 사용자가 웹 브라우저를 통해 웹 페이지에 접속하거나, 웹 어플리케이션에서 API를 호출하는 경우에 해당합니다. 요청 방식(Method)에는 주로 GET, POST, PUT, DELETE 등이 있습니다.

서버는 클라이언트의 요청을 처리하고, 그 결과를 클라이언트에게 응답(Response)합니다. 응답에는 HTTP 상태 코드(Status Code)가 포함되어 있어, 요청의 성공 여부를 알 수 있습니다. 예를 들어, 200은 성공, 404는 Not Found(찾을 수 없음), 500은 Internal Server Error(서버 내부 오류) 등을 나타냅니다.

HTTP는 기본적으로 연결을 유지하지 않는 비연결성(Connectionless) 특성을 가집니다. 즉, 클라이언트와 서버 간의 통신이 끝나면 연결을 끊습니다. 이는 서버의 자원을 효율적으로 사용할 수 있도록 하지만, 매번 연결을 수립하고 끊어야 하므로 오버헤드가 발생할 수 있습니다.

또한, HTTP는 상태 정보를 유지하지 않는 Stateless 특성을 가집니다. 이는 각각의 요청이 독립적으로 관리된다는 것을 의미하며, 이전 요청과 이후 요청이 서로 관련이 없다는 것을 의미합니다. 이러한 특성 때문에 로그인 상태와 같은 정보를 유지하기 위해 쿠키, 세션 등의 기술이 사용됩니다.

이러한 HTTP의 특성과 동작 방식을 이해하는 것은 웹 서비스 개발 및 운영, API 사용 등에 중요한 기초 지식이 됩니다