

REST API란?

API(애플리케이션 프로그래밍 인터페이스)는 애플리케이션이나 디바이스가 서로 간에 연결하여 통신할 수 있는 방법을 정의하는 규칙 세트입니다. **REST API**는 **REST(REpresentational State Transfer)** 아키텍처 스타일의 디자인 원칙을 준수하는 API입니다. 이러한 이유로 **REST API**를 **RESTful API**라고도 합니다.

CRUD Operation이란?

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 **Create(생성)**, **Read(읽기)**, **Update(갱신)**, **Delete(삭제)**를 묶어서 일컫는 말로 REST에서의 **CRUD Operation** 동작 예시는 다음과 같다.

REST 구성 요소

1. 자원(Resource) : HTTP URI
2. 자원에 대한 행위(Verb) : HTTP Method
3. 자원에 대한 행위의 내용 (Representations) : HTTP Message Pay Load

REST API의 특징

1. 자원 기반 (Resource-Based): 각각의 자원은 고유한 식별자인 **URI(Uniform Resource Identifier)**를 가지며, 클라이언트는 **URI**를 통해 자원에 접근합니다.
2. 상태 없음 (Stateless): 서버는 클라이언트의 상태를 저장하지 않으며, 클라이언트의 각 요청은 모든 필요한 정보를 포함해야 합니다. 이를 통해 서버의 부담을 줄이고 확장성을 높일 수 있습니다.
3. 통일된 인터페이스 (Uniform Interface): **REST API**는 통일된 인터페이스를 제공하여 일관성을 유지합니다. **HTTP** 메서드(**GET**, **POST**, **PUT**, **DELETE** 등)를 사용하여 자원에 대한 작업을 수행하고, **HTTP** 상태 코드를 통해 요청의 결과를 알려줍니다.
4. 자체 표현 (Self-Descriptive): **REST API**는 자체 표현적인 메시지를 사용하여 자원에 대한 정보를 전달합니다. 예를 들어, **JSON**이나 **XML** 형식으로 데이터를 반환하며, 이러한 데이터 포맷은 클라이언트가 이해하고 처리할 수 있도록 합니다.
5. 계층 구조 (Layered System): **REST API**는 서버와 클라이언트 사이에 여러 개의 중간 서버를 두고 요청을 전달할 수 있습니다. 이를 통해 시스템의 확장성과 보안성을 향상시킬 수 있습니다.

REST API는 다양한 프로그래밍 언어와 프레임워크에서 구현될 수 있으며, 웹 애플리케이션, 모바일 앱, 마이크로서비스 등 다양한 분야에서 사용됩니다. 이를 통해 서로 다른 시스템 간의 상호 작용을 효율적으로 구현할 수 있습니다.