REST(Representational State Transfer)

1. REST란 자원을 이름으로 구분하여 해당 자원의 상태(정보)를 주고 받는 모든 것을 의미.
2. 자원(resource)의 표현(representation)에 의한 상태 전달을 말하며 REST는 ROA(Resource Oriented Architecture)를 따르는 웹 서비스 아키텍쳐이다.
3. HTTP URI를 통해 자원을 명시하고, HTTP Method(POST, GET, PUT, DELETE)를 통해 해당 자원에 대한 CRUD Operation을 적용함.
4. 특징
   1. Server-Client(서버-클라이언트 구조)
   2. Stateless(무상태성)
   3. Cacheable(캐시 가능)
   4. Layered System(계층화)
   5. Addressability(URI 접근)
   6. Uniform Interface(인터페이스 일관성)
5. REST의 장점
   1. HTTP 프로토콜의 인프라를 그대로 사용하므로 REST API 사용을 위한 별도의 인프라 구축이 필요없다.
   2. HTTP 프로토콜의 표준을 최대한 활용하여 여러 추가적인 장점 사용 가능.
   3. HTTP 표준 프로토콜에 따르는 모든 플랫폼에서 사용이 가능.
   4. REST API 메시지가 의도하는 바를 쉽게 파악할 수 있다.
   5. 여러가지 서비스 디자인에서 생길 수 있는 문제를 최소화한다.
   6. 서버와 클라이언트의 역할을 명확히 분리
6. REST의 단점
   1. 표준이 존재하지 않다.
   2. 사용할 수 있는 메소드가 4가지 밖에 없다.
   3. HTTP Method 형태가 제한적이다.
   4. 구형 브라우저가 지원해주지 못하는 부분이 존재 (PUT, DELETE, pushState)
7. RSET가 필요한 시점
   1. 애플리케이션 분리 및 통합
   2. 다양한 클라이언트
   3. 다양한 플랫폼에서도 통신 가능한 서버 프로그램이 필요할 때
   4. 멀티 플랫폼에 대한 지원을 위해 서비스 자원에 대한 아키텍처가 필요할 때

REST API

REST 기반으로 서비스 API를 구현한 것을 REST API라고 한다.

OpenAPI(누구나 사용할 수 있도록 공개된 API: 구글 맵, 공공 데이터 등), 마이크로 서비스(하나의 큰 애플리케이션을 여러 개의 작은 애플리케이션으로 쪼개어 변경과 조합이 가능하도록 만든 아키텍처)등을 제공하는 업체 대부분 REST API를 제공

1. REST API의 특징
   1. 사내 시스템들도 REST 기반으로 시스템을 분산해 확장성과 재사용성을 높여 유지보수 및 운용을 편리하게 할 수 있다.
   2. REST는 HTTP 표준을 기반으로 구현하므로, HTTP를 지원하는 프로그램 언어로 클라이언트, 서버를 구현할 수 있다.

HTTP 통신

HTTP(Hyper Text Transfer Protocol) 란 한마디로 HTML(웹문서를 만들기 위한 언어) 문서를 주고 받는데 쓰이는 통신프로토콜(통신규약)이며, TCP 와 UDP 를 사용하여 통신하며 80번 포트를 사용하는 통신프로토콜(통신규약)이다.

1. HTTP 통신과정
   1. Client가 Server에 요청을 보내면 패킷에 데이터를 담아서 응답한다.

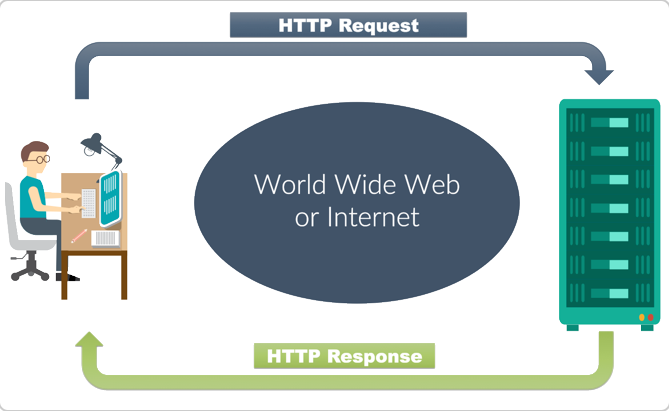


그림 HTTP 통신 과정

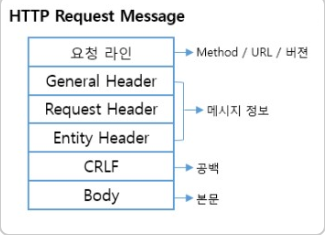


그림 HTTP Request 과정

참고문서

RESTful API 설계 가이드: <https://sanghaklee.tistory.com/57>

RSET, REST API, RESTful 개념: <https://blog.naver.com/dktmrorl/222072210416>

HTTP 통신 기본 : https://korshika.tistory.com/49