**REST란?**



REST(Representational State Transfer)의 약자로 자원을 이름으로 구분하여 해당 자원의 상태를 주고받는 모든 것을 의미합니다.

**즉 REST란**

1. HTTP URI(Uniform Resource Identifier)를 통해 자원(Resource)을 명시하고,
2. HTTP Method(POST, GET, PUT, DELETE, PATCH 등)를 통해
3. 해당 자원(URI)에 대한 CRUD Operation을 적용하는 것을 의미합니다.

**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.

Create : 데이터 생성(POST)  
Read : 데이터 조회(GET)  
Update : 데이터 수정(PUT, PATCH)  
Delete : 데이터 삭제(DELETE)

**REST 구성 요소**

REST는 다음과 같은 3가지로 구성이 되어있다.

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

**REST의 특징**

1. Server-Client(서버-클라이언트 구조)
2. Stateless(무상태)
3. Cacheable(캐시 처리 가능)
4. Layered System(계층화)
5. Uniform Interface(인터페이스 일관성)

**REST의 장단점**

장점

* HTTP 프로토콜의 인프라를 그대로 사용하므로 REST API 사용을 위한 별도의 인프라를 구출할 필요가 없다.
* HTTP 프로토콜의 표준을 최대한 활용하여 여러 추가적인 장점을 함께 가져갈 수 있게 해 준다.
* HTTP 표준 프로토콜에 따르는 모든 플랫폼에서 사용이 가능하다.
* Hypermedia API의 기본을 충실히 지키면서 범용성을 보장한다.
* REST API 메시지가 의도하는 바를 명확하게 나타내므로 의도하는 바를 쉽게 파악할 수 있다.
* 여러 가지 서비스 디자인에서 생길 수 있는 문제를 최소화한다.
* 서버와 클라이언트의 역할을 명확하게 분리한다.

단점

* 표준이 자체가 존재하지 않아 정의가 필요하다.
* HTTP Method 형태가 제한적이다.
* 브라우저를 통해 테스트할 일이 많은 서비스라면 쉽게 고칠 수 있는 URL보다 Header 정보의 값을 처리해야 하므로 전문성이 요구된다.
* 구형 브라우저에서 호환이 되지 않아 지원해주지 못하는 동작이 많다.(익스폴로어)

**REST API란?**



**RESPT API란 REST의 원리를 따르는 API를 의미합니다.**

**하지만 REST API를 올바르게 설계하기 위해서는 지켜야 하는 몇가지 규칙이 있으며 해당 규칙을 알아 보겠습니다.**

**REST API 설계 예시**

**1. URI는 동사보다는 명사를, 대문자보다는 소문자를 사용하여야 한다.**

**Bad Example**[http://khj93.com/Running/](http://khj93.com/test/) **Good Example**[http://khj93.com/run/](http://khj93.com/test/)

**2. 마지막에 슬래시 (/)를 포함하지 않는다.**

**Bad Example**<http://khj93.com/test/>**Good Example**[http://khj93.com/test](http://khj93.com/test/)

**3. 언더바 대신 하이폰을 사용한다.**

**Bad Example**[http://khj93.com/test\_blog](http://khj93.com/test/) **Good Example**[http://khj93.com/test-blog](http://khj93.com/test/)

**4. 파일확장자는 URI에 포함하지 않는다.**

**Bad Example**[http://khj93.com/photo.jpg](http://khj93.com/test/)**Good Example**[http://khj93.com/photo](http://khj93.com/test/)

**5. 행위를 포함하지 않는다.**

**Bad Example**[http://khj93.com/delete-post/1](http://khj93.com/test/)**Good Example**[http://khj93.com/post/1](http://khj93.com/test/)

**RESTful이란?**



**RESTFUL이란 REST의 원리를 따르는 시스템을 의미합니다. 하지만 REST를 사용했다 하여 모두가 RESTful 한 것은 아닙니다.  REST API의 설계 규칙을 올바르게 지킨 시스템을 RESTful하다 말할 수 있으며**

**모든 CRUD 기능을 POST로 처리 하는 API 혹은 URI 규칙을 올바르게 지키지 않은 API는 REST API의 설계 규칙을 올바르게 지키지 못한 시스템은 REST API를 사용하였지만 RESTful 하지 못한 시스템이라고 할 수 있습니다.**

References

* <https://gmlwjd9405.github.io/2018/09/21/rest-and-restful.html>
* [www.incodom.kr/REST](http://www.incodom.kr/REST)