

REST API

REST : 웹 (HTTP) 의 장점을 활용한 아키텍처

1. REST (Representational State Transfer) 기본

- REST의 요소

- Method

Method	의미	Idempotent
POST	Create	No
GET	Select	Yes
PUT	Update	Yes
DELETE	Delete	Yes

Idempotent : 한 번 수행하냐, 여러 번 수행했을 때 결과가 같냐?

- Resource

- http://myweb/users와 같은 URI
 - 모든 것을 Resource (명사)로 표현하고, 세부 Resource에는 id를 붙임

- Message

- 메시지 포맷이 존재

: JSON, XML 과 같은 형태가 있음 (최근에는 JSON 을 씀)

```
HTTP POST, http://myweb/users/  
{  
  "users" : {  
    "name" : "terry"  
  }  
}
```

- REST 특징

- Uniform Interface

- HTTP 표준만 맞는다면, 어떤 기술도 가능한 Interface 스타일

예) REST API 정의를 HTTP + JSON로 하였다면, C, Java, Python, IOS 플랫폼 등 특정 언어나 기술에 종속 받지 않고, 모든 플랫폼에 사용이 가능한 Loosely Coupling 구조

- 포함
 - Self-Descriptive Messages
 - API 메시지만 보고, API를 이해할 수 있는 구조 (Resource, Method를 이용해 무슨 행위를 하는지 직관적으로 이해할 수 있음)
 - HATEOAS(Hypermedia As The Engine Of Application State)
 - Application의 상태(State)는 Hyperlink를 통해 전이되어야 함.
 - 서버는 현재 이용 가능한 다른 작업에 대한 하이퍼링크를 포함하여 응답해야 함.
 - Resource Identification In Requests
 - Resource Manipulation Through Representations
- Statelessness
 - 즉, HTTP Session과 같은 컨텍스트 저장소에 **상태 정보 저장 안함**
 - ****Request만 Message로 처리****하면 되고, 컨텍스트 정보를 신경쓰지 않아도 되므로, **구현이 단순해짐.**
 - 따라서, REST API 실행중 실패가 발생한 경우, Transaction 복구를 위해 기존의 상태를 저장할 필요가 있다. (POST Method 제외)
- Resource 지향 아키텍처 (ROA : Resource Oriented Architecture)
 - Resource 기반의 복수형 명사 형태의 정의를 권장.
- Client-Server Architecture
- Cache Ability
- Layered System
- Code On Demand(Optional)