

16주차 발표\_민서연

---

# REST

---

# 01 REST

## REST

Representational State Trasfer

HTTP URL를 통해 자원을 명시하고 메서드를 통해  
해당 자원에 대한 CRUD Operation을 적용하는 것

# 01 REST

## REST

Representational State Trasfer

자원(Resource): HTTP URI

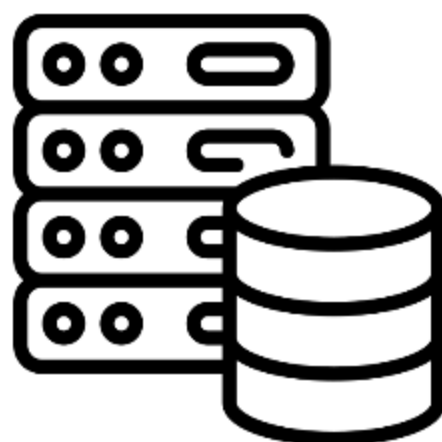
자원에 대한 행위(Verb) : HTTP Method

자원에 대한 행위의 내용(Representations): HTTP Message Pay Load

# 01 REST

## 특징

서버 클라이언트 구조 (Server-Client)



SERVER



CLIENT

- REST Server: API를 제공하고 비즈니스 로직 처리 및 저장
- Client: 사용자 인증이나 context(세션, 로그인 정보) 등을 직접 관리하고 책임

# 01 REST

## 특징

### 무상태 (Stateless)

작업을 위한 상태 정보를 따로 저장하고 관리하지 않음

stateful

: 클라이언트와 서버의 동작, 상태 정보를 저장하는 형태  
세션상태에 따라 서버의 응답이 달라짐

# 01 REST

## 특징

### 무상태 (Stateless)

작업을 위한 상태 정보를 따로 저장하고 관리하지 않음

#### ➡ rest가 stateless 해야하는 이유

- stateless가 시스템 확장에 유리
- 사용 가능한 모든 인스턴스를 활용 가능

# 01 REST

## 특징

### 캐시 처리 기능(Cacheable)

웹 표준 HTTP 프로토콜을 그대로 사용하기 때문에  
웹에서 사용하는 기존 인프라를 그대로 사용 가능

즉, HTTP가 가진 캐싱 기능 사용 가능하다 (LAST-Modified태그, E-Tag 이용)

캐시 사용을 하면 응답 시간이 빨라지며, 트랜잭션이 발생하지 않아 전체 응답시간, 성능, 서버의 이용률 향상시킴

# 01 REST

## 특징

### 계층화 (LayeredSystem)

client는 REST API Server만 호출하고 REST Server는 다중 계층으로 구성될 수 있음

다중 계층으로 구성되기 때문에 보안 로드 밸런싱, 암호화 계층을 추가해서 구조상의 유연성을 둘 수 있고 네트워크 기반 중간 매체(PROXY, 게이트웨이)를 사용할 수 있게 함

로드 밸런싱: 서버가 처리해야 할 업무나 요청을 여러대의 서버로 분배해서 처리

PROXY: 보안상의 문제로 직접 통신 할 수 없을 때 중계 해주는 서버

게이트웨이: 어느 네트워크에서 다른 네트워크로 이동할 때 거치는 관문



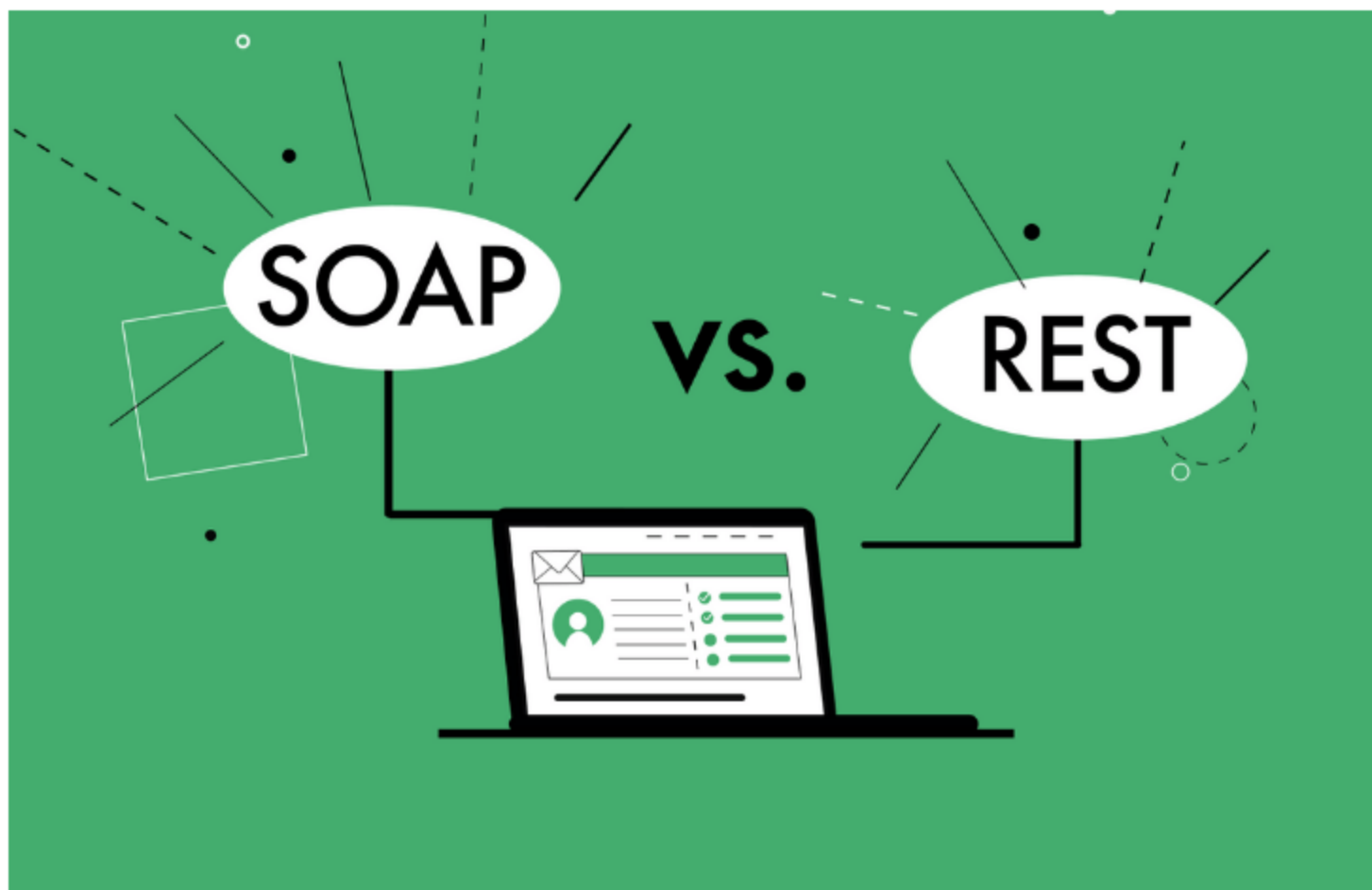
# 01 REST

## 특징

### 인터페이스 일관성 (Uniform Interface)

URI로 지정한 리소스에 대한 조작을 통일되고 한정적인 인터페이스로 수행하는 아키텍처 스타일  
특정 언어나 기술에 종속되지 않음

## 02 REST와 SOAP의 비교

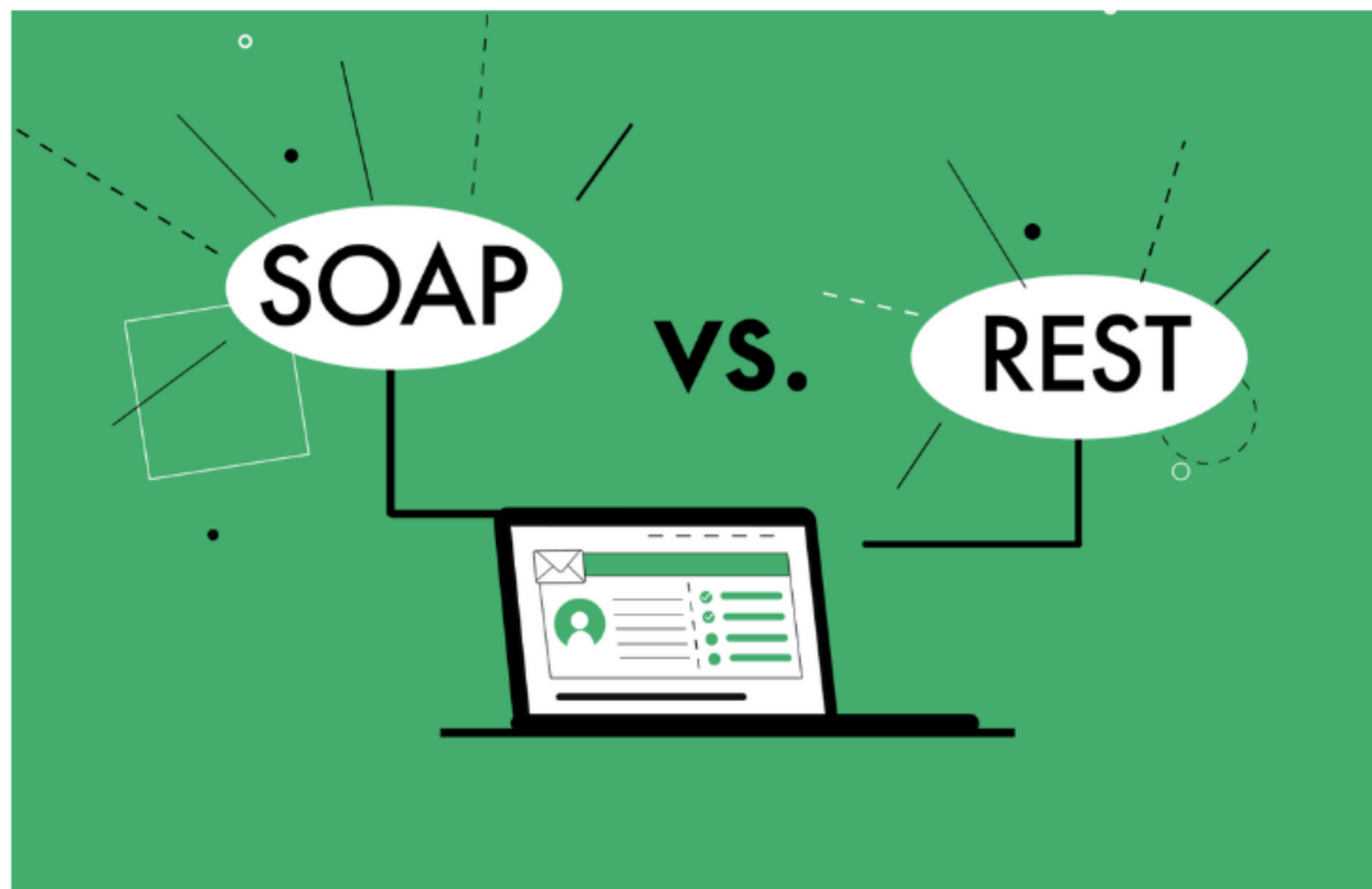


### SOAP(Simple Object Access Protocol)

서로 다른 언어로 다른 플랫폼에서 빌드된 애플리케이션이 통신 할 수 있도록 설계된 최초의 표준 프로토콜

HTTP, HTTPS, SMTP 등을 통해 XML 기반의 메시지를 컴퓨터 네트워크 상에서 교환하는 프로토콜

## 02 REST와 SOAP의 비교



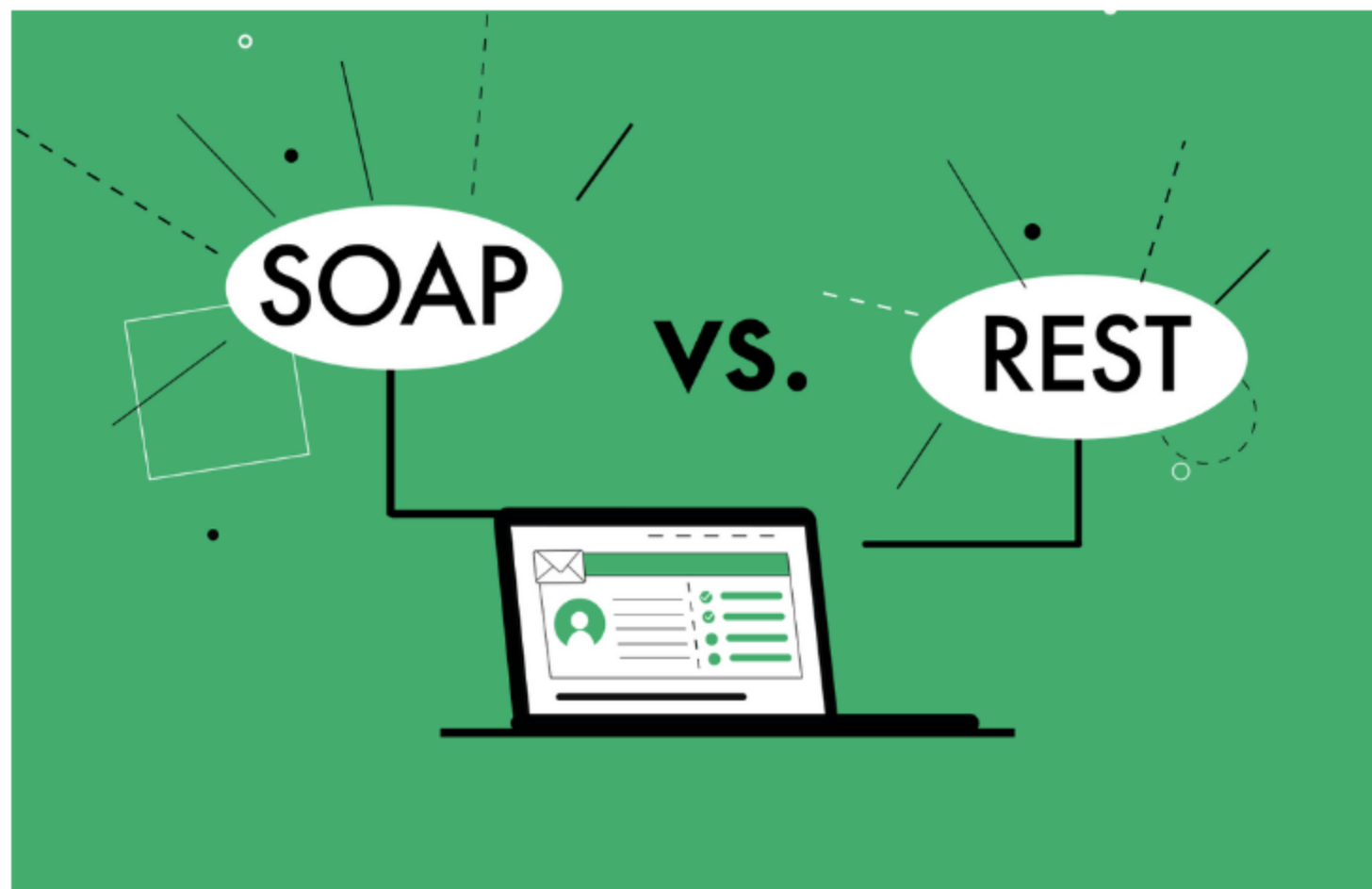
### REST와 SOAP 차이

REST는 CSV, JSON, RSS의 데이터를 활용해서 전송  
SOAP는 오직 XML에만 의존해서 전송

REST는 아키텍처 원칙 세트이고  
SOAP는 W3C(World Wide Web Consortium) 에서 유지관리하는 프로토콜

REST는 GET, POST, PUT, DELETE 등의 메서드를 활용  
SOAP는 기본적으로 POST만을 이용해서 CRUD 처리

## 02 REST와 SOAP의 비교



### 그래서 REST와 SOAP 중 좋은건?

전반적으로 REST는 이미 존재하는 웹을 활용하고 자유도가 제한되어 있어서 더 간단하다. (선택할 항목이 적기 때문에)

SOAP는 여러가지 대안을 제공하고 개발도 약간 더 어렵지만 더 많은 대안과 작업 영역을 제공한다.

본인이 사용하는 프로젝트에 따라 맞는 것을 고르면 됨!

## 03 출처

<https://velog.io/@gomuzom/RESTful-API>

<https://velog.io/@yhlee9753/REST-API-%EC%99%80-HTTP-Stateless-%EC%97%90-%EB%8C%80%ED%95%9C-%EA%B3%A0%EC%B0%B0>

<https://khj93.tistory.com/entry/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-REST-API%EB%9E%80-REST-RESTful%EC%9D%B4%EB%9E%80>

<https://www.redhat.com/ko/topics/integration/whats-the-difference-between-soap-rest>

<https://restfulapi.net/soap-vs-rest-apis/>

<https://sabarada.tistory.com/30>