

2024. 01. 31

REST API RESTful

Kang sinbeom

목차

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et
dolore magna aliqua.

a. REST의 등장 배경

- 초기의 웹에 비해 발전되는 환경
- 통일된 통신 규약의 필요성

b. REST란 무엇인가

Sed do eiusmod tempor incididunt.

c. REST API은 무엇인가?

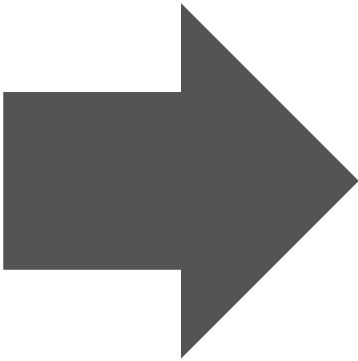
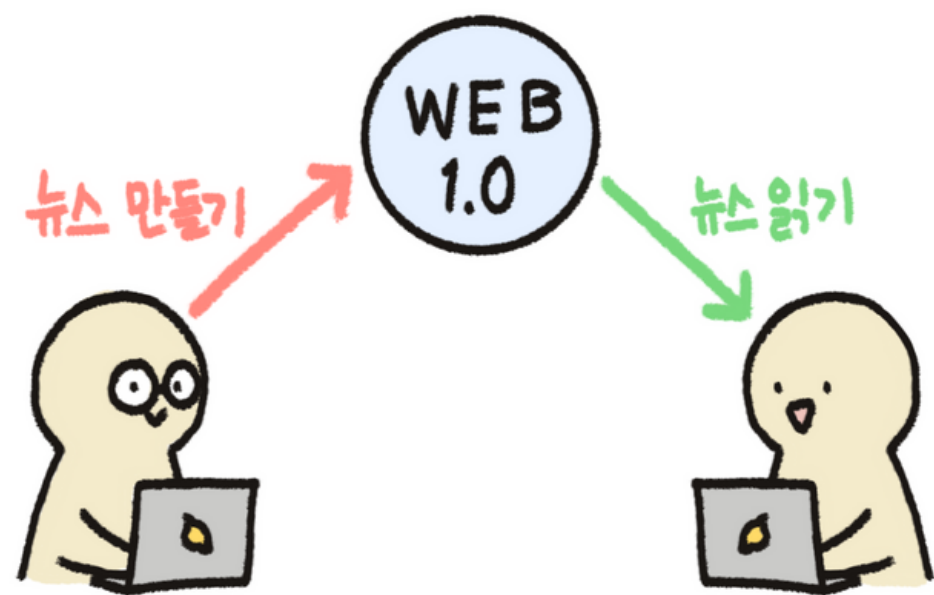
Sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.

d. RESTful은 무엇인가?

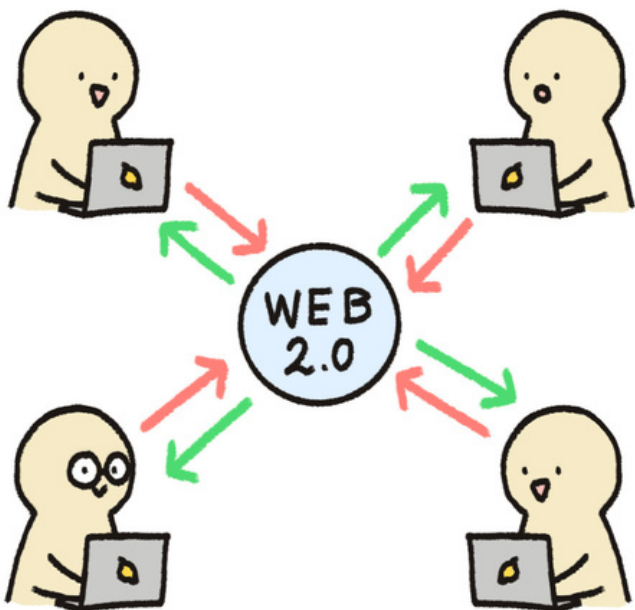
Sed do eiusmod tempor incididunt.

REST 등장 배경

단순한 읽기 쓰기만 가능한 초기의 웹

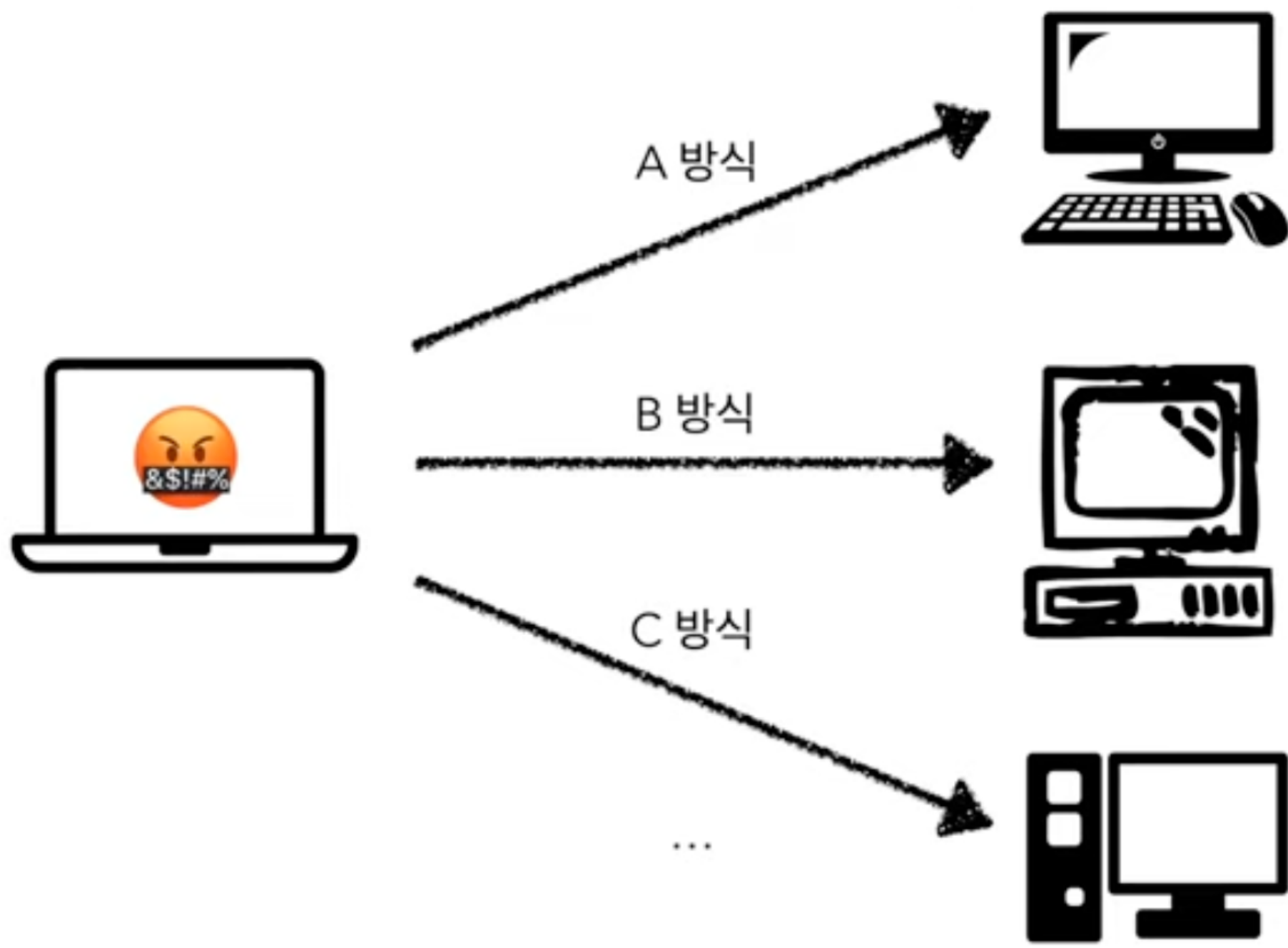


점점 복잡해지는 웹의 구조
 많아지는 통신 횟수 및 기능



REST 등장 배경

일관된 통신 방식의 필요성 대두



REST 등장 배경



SOAP
(Simple Object Access Protocol)

어렵다

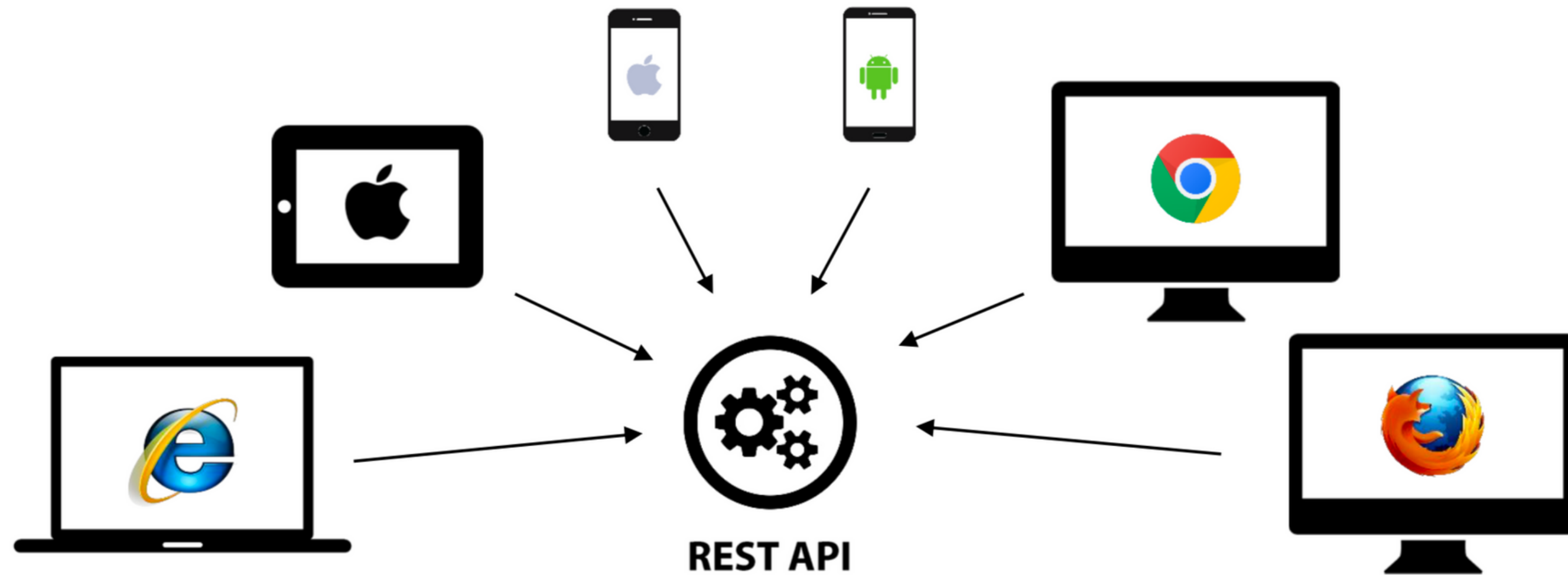
복잡하다

느리다

사용하지 않는다

{ REST }

REST란 무엇인가



REST (REpresentational State Transfer)

“REST”

GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

- 무상태 (Stateless)
- 캐시 처리 기능
- 계층화
- 서버, 클라이언트 구조
- ☒인터페이스 일관성

REST (REpresentational State TreaNfer)

“REST”

GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

- Roy Fielding, 2000년대, 180p
- 서버와 클라이언트의 통신 방식
- 자원을 이름(자원의 표현)으로 구분해 해당 자원의 상태를 주고 받는 모든 것

- 자원 - 문서, 그림, 데이터 등 소프트웨어가 관리하는 모든 것
- 자원의 표현 - 학생정보(데이터), students (지칭하는 말)
- 자원의 상태 - 네트워크를 통해 주고 받는 데이터 그 자체

이름으로 구분하여

- URI(Uniform Resource Identifiers)
- Singleton and Collection Resource
- Collection and Sub-collection Resource

Example: Web service called **User Management**

HTTP Method	URI	Operation
GET	<u>/api/users</u>	returns a list of users
GET	<u>/api/users/1</u>	returns the user with ID 1
POST	<u>/api/users</u>	creates a new user
PUT	<u>/api/users/3</u>	updates the user with ID 3
DELETE	<u>/api/users/4</u>	deletes the user with ID 4
DELETE	<u>/api/users</u>	deletes all the users

- 자원을 식별하기 위한 문자열의 구성
- 네이밍을 잘 할수록 API가 직관적이다.

이름으로 구분하여

- URI(Uniform Resource Identifiers)
- Singleton and Collection Resource
- Collection and Sub-collection Resource

특정 변수로 선언된 singleton

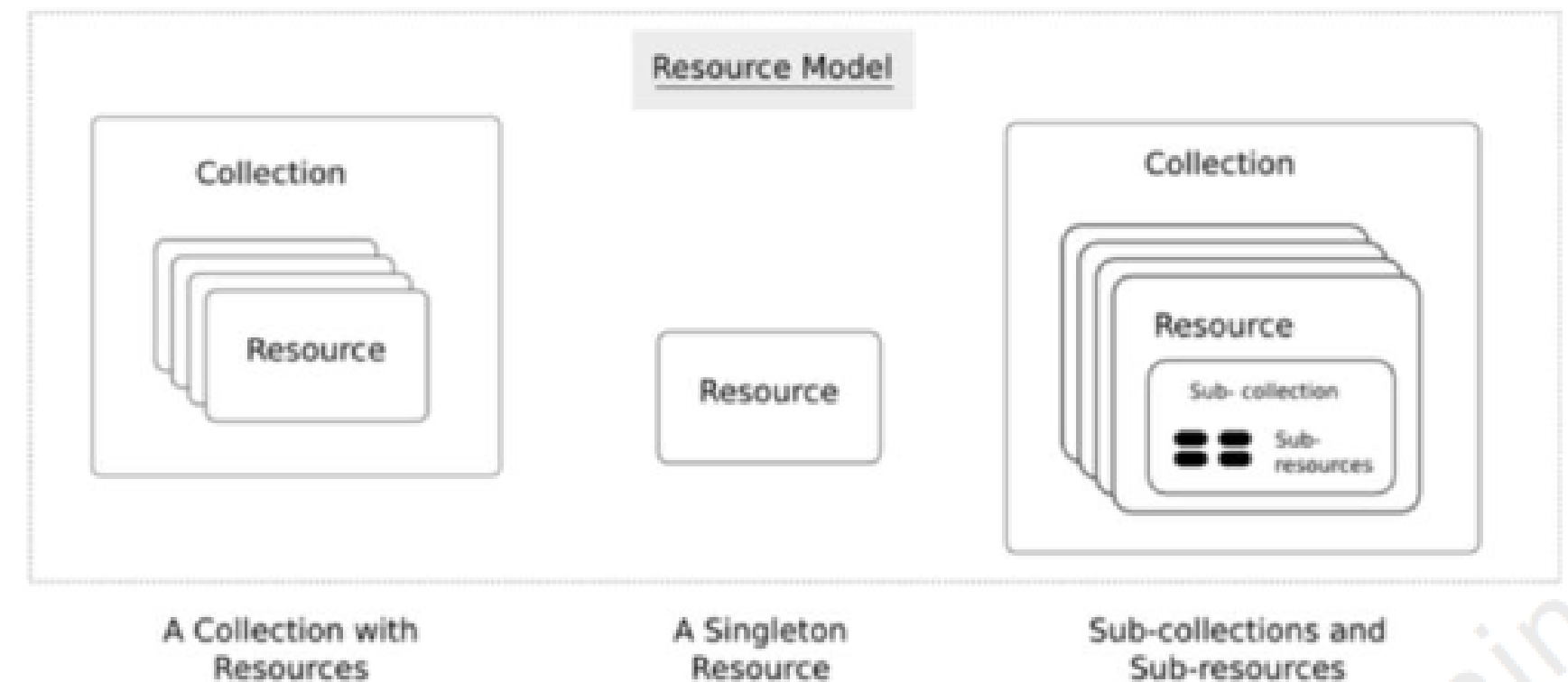
- /customer

docs의 의미인 collection

- /customers

서브 컬렉션 포함 가능

- /customers/{customerId}/account
- /customers?hot=false

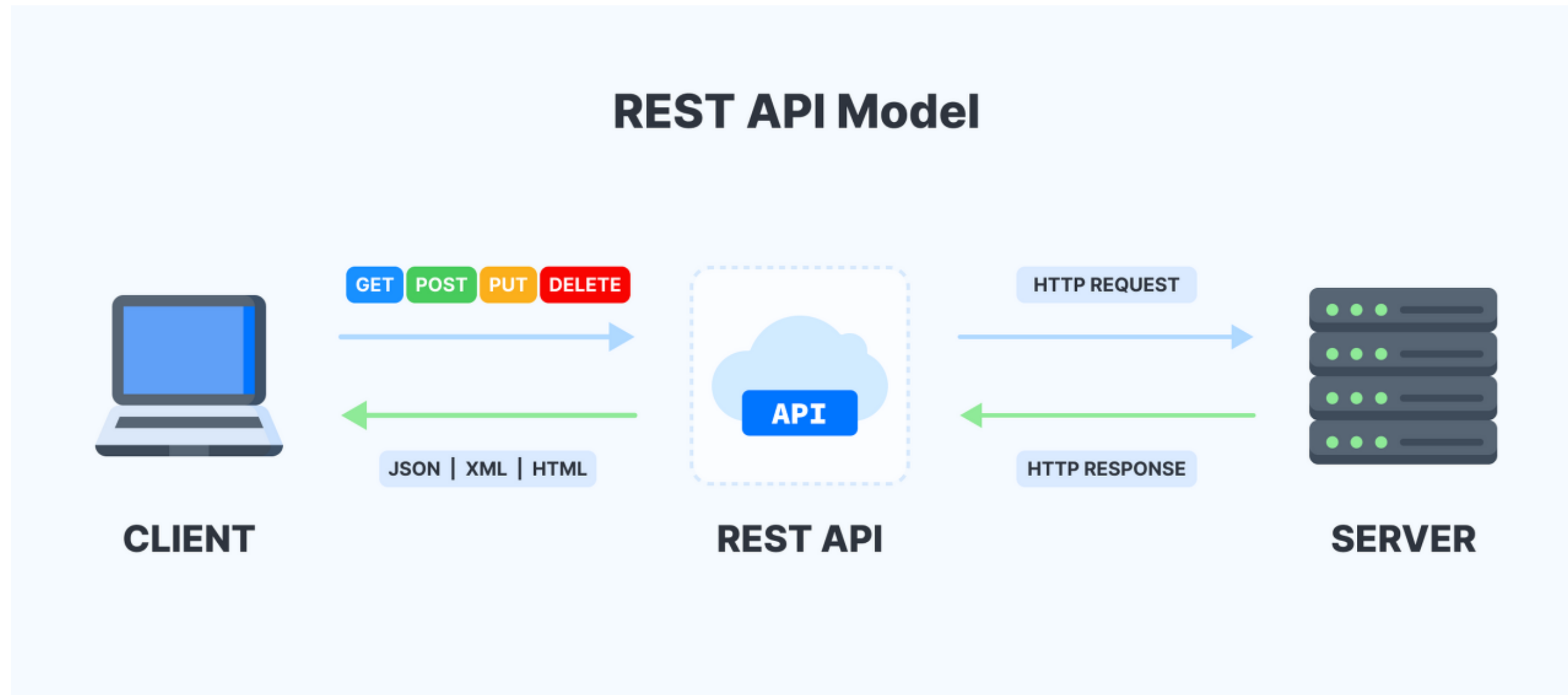


☒ URI 네이밍 규칙

칙

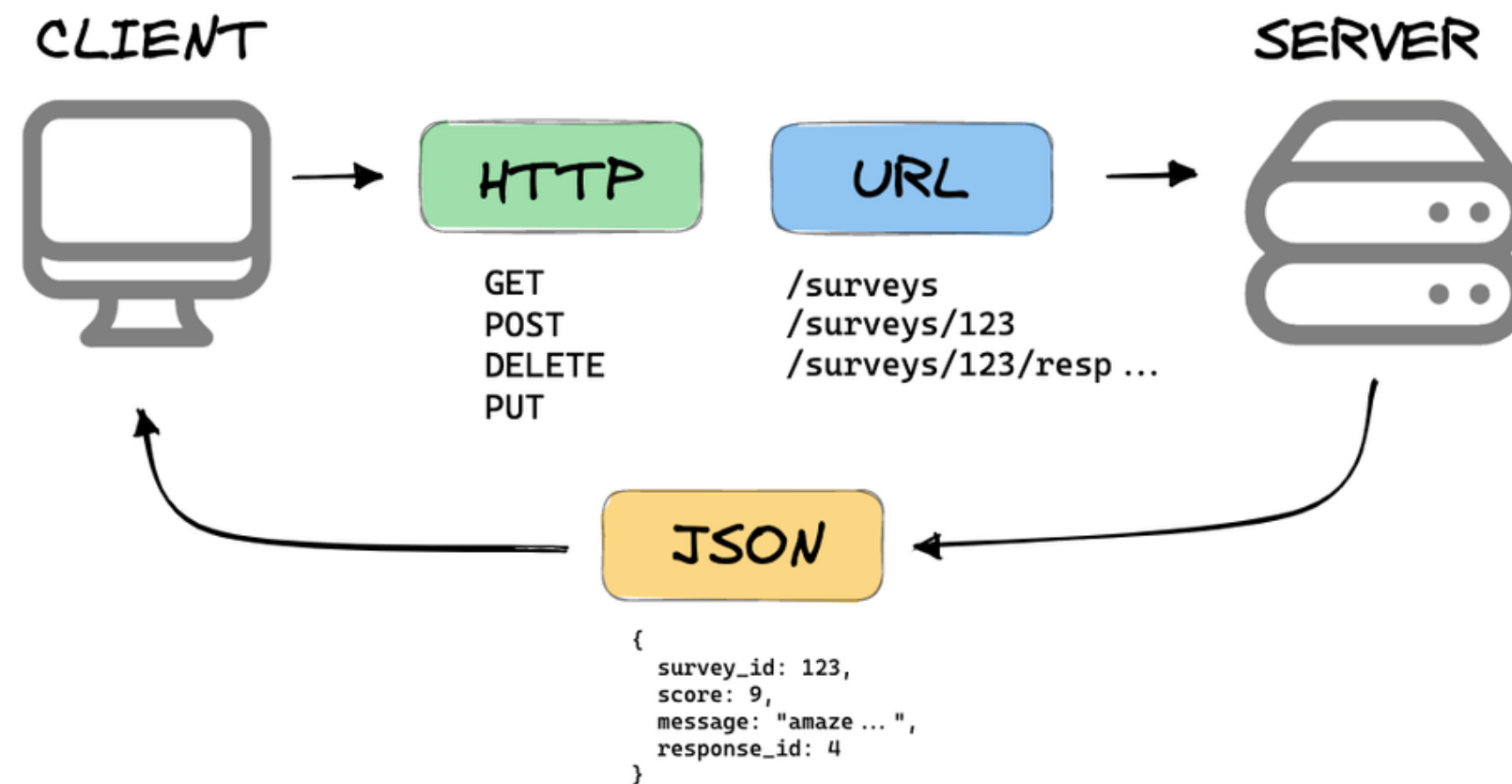
- 명사를 사용해서 자원을 표현하자
- 하이픈 기호를 사용해서 가독성을 높이자
- 마지막에는 / 를 붙이지 않는다
- 계층 관계를 표현하기 위해 / 를 사용하자
- 밑줄은 가급적 사용하지 않는다
- CRUD 함수를 이름에 사용하지 말자
- 필터링을 위해서는 Query string을 사용하자
- 소문자만을 사용하고, 파일 확장자를 표시하지 말자

자원의 상태를 주고 받는



REST API란 무엇인가

WHAT IS A REST API?



REST 기반으로 서비스 API를 구현한 것

- 쉽게 추론할 수 있다
- 재사용성 및 유지 보수가 용이함
- HTTP 프로토콜을 그대로 사용

- API - 서로 정보를 교환 가능하도록 하는 것

REST API란 무엇인가

예시

CRUD	HTTP verbs	Route
resource들의 목록을 표시	GET	<i>/resource</i>
resource 하나의 내용을 표시	GET	<i>/resource/:id</i>
resource를 생성	POST	<i>/resource</i>
resource를 수정	PUT	<i>/resource/:id</i>
resource를 삭제	DELETE	<i>/resource/:id</i>

REST API란 무엇인가

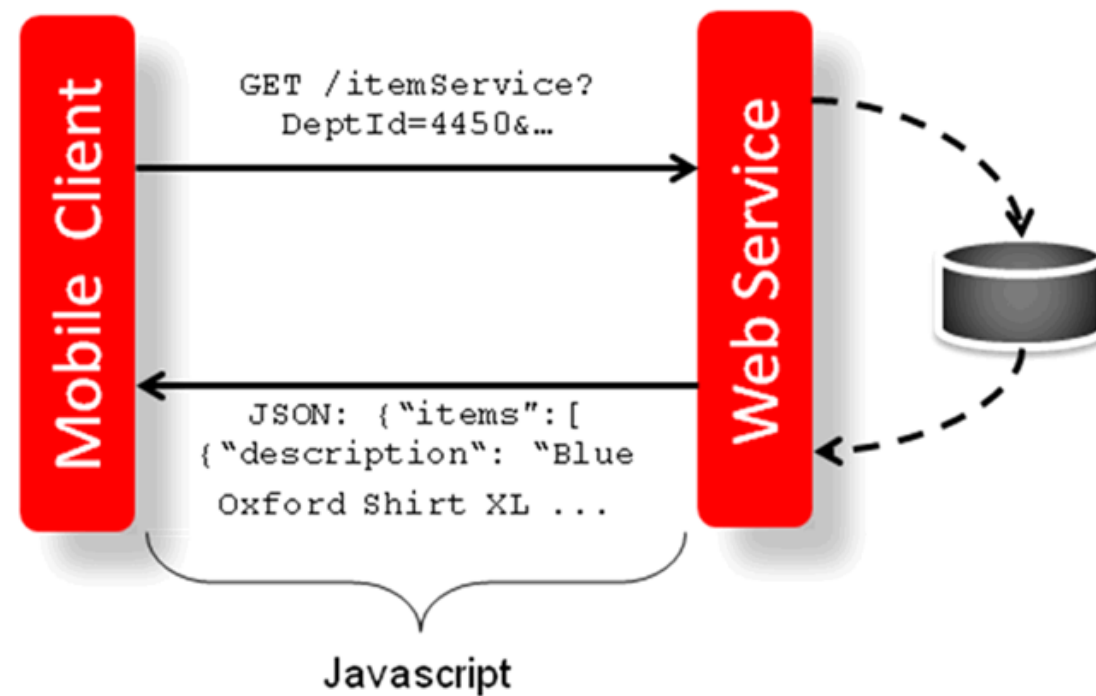
상태 코드

클라이언트가 보낸 요청의 처리 상태를 응답에서 알려주는 기능

- 1xx (Informational): 요청이 수신되어 처리중
- 2xx (Successful): 요청 정상 처리
- 3xx (Redirection): 요청을 완료하려면 추가 행동이 필요
- 4xx (Client Error): 클라이언트 오류, 잘못된 문법등으로 서버가 요청을 수행할 수 없음
- 5xx (Server Error): 서버 오류, 서버가 정상 요청을 처리하지 못함

RESTful

REST API를 제공하는 웹 서비스는 RESTful하다!



“RESTful”

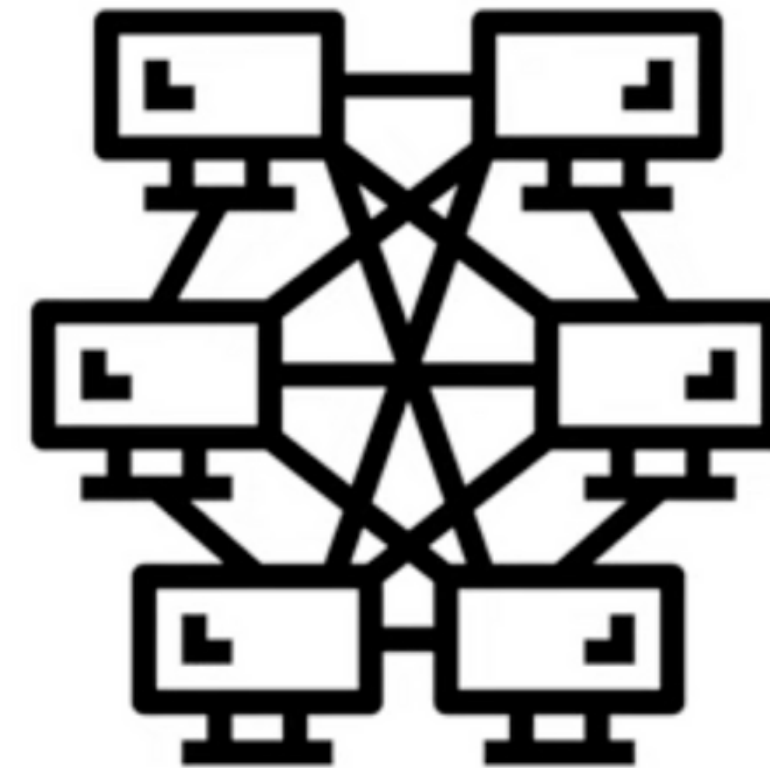
목적

- 이해하기 쉬운 REST API
- 일관된 컨벤션을 통한 이해도 및 호환성 상승

RESTful 하지 못한 경우

- CRUD 기능을 모두 POST로만 처리하는 경우
- /student/updateName (x) /student?updateName (o)

- 사실 규칙을 지키지 않아도 되지만..!
 - A : /보여주세요/크루들의정보
 - B : /showmetherecruinfo
 - C : /get/recruinfo
 - D : /i_want_toKnowcREWinfo
 - E : /asdkfjhasjdkfhjkqwjkfqwbkjbkj
- 일관성 X -> URI의 의도 파악이 어렵다



단점

- 제한된 형태의 메소드

- 표준의 부재