

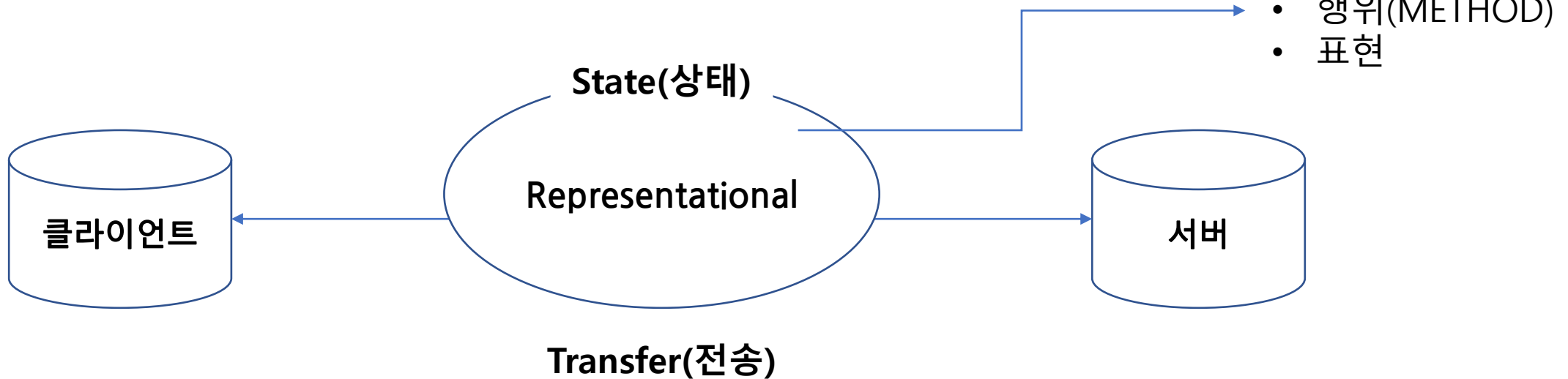
REST API

REST란?

REpresentational State Transfer

- 자원을 이름으로 구분해 해당 자원의 상태를 전송하여 주고 받는 것

WEB 기술 + HTTP 프로토콜을 활용한 아키텍처 스타일 방식



HTTP 프로토콜

Request

```
GET /index.html HTTP/1.1
user-agent: MSIE 6.0; Window NT 5.0
accept: test/html; */*
cookie:name=value
refere: http://abc.com
host: www.abc.com
```

요청 URL 정보 + HTTP 버전
웹 브라우저 종류
데이터 타입
쿠키 정보
요청 도메인 및 경유지



Response

```
HTTP/1.1 200 OK
Server: Apache
Content-type: text/html
Content-length : 1593
<html> <head> .....
```

HTTP 버전 및 응답 코드
웹 서버 종류
데이터 타입
HTTP BODY 사이즈
HTTP BODY 콘텐츠

HTTP 응답코드

- **1xx (정보전송 임시 응답)** : 전송 프로토콜 수준의 정보 교환 (요청 후 프로세스 계속 진행)
- **2xx (성공)** : 클라이언트 요청이 성공적으로 수행됨
- **3xx (리다이렉트)** : 클라이언트는 요청을 완료하기 위해 추가적인 행동을 취해야 함(변경된 url)
- **4xx (클라이언트 에러)** : 클라이언트의 잘못된 요청
- **5xx (서버 에러)** : 서버쪽 오류로 인한 상태코드

500 Internal Server Error

Sorry, something went wrong.

A team of highly trained monkeys has been dispatched to deal with this situation.

If you see them, send them this information as text (screenshots frighten them):

```
APkpgMWh1ZDfqpUbbMqGVMXUaR3363zKv4rqo1SDGDTGe3zez-EeK3K1  
G6Ti3Cz4jwBRw0EIdVBAYfMvw1KxZXId92isD-hGMZzz4rc3WFOsj05y  
y3ZBzHC67_iYqUFh5wldzXn41BKAK6d0W9_xZ-V1zRUbHS1qFUYwUU4g  
02mqMrfqfg9vEv8P9AoP1_14KLpU7gCjKJgti1-wA1_a-z3I-vZpo8tUm
```



404. That's an error.

The requested URL `/a_cool_website` was not found on this server. That's all we know.



HTTP 메소드

HTTP Method	설명	Idempotent
GET	요청된 URI 정보 응답	O
POST	요청된 자원 생성	X
PUT	요청된 자원 수정(자원 전체 갱신)	O
DELETE	요청된 자원 삭제	O
HEAD	요청된 URI 정보 응답(BODY 없고 HEAD만)	O
PATCH	요청된 자원 수정(해당 자원만 갱신)	X (제약 때문)
CONNECT	프록시 기능 요청시 사용 (클라이언트-서버 SSL 핸드셰이크)	X
TRACE	루프백 메시지 호출 시 사용	O
OPTIONS	메소드 종류 확인 시 사용	O

※ 서비스 시 지원해야 할 메소드

- GET, POST, HEAD, OPTIONS

나머지 메소드는 보안 취약점 야기
가능성 존재하므로 비활성화 필요

- **PUT** : Web Shell을 통한 시스템
침투가 가능(파일 업로드)

- **DELETE** : 클라이언트에서 웹 서
버 파일 삭제 가능

- **CONNECT** : HTTP 프록시 악용
가능

- **TRACE** : XST 공격으로 세션 탈
취 가능

※ Idempotent : 멱등 (한 번 수행했을 때, 여러 번 수행했을 때 결과가 같은가)

REST 특징

1. Server-Client

Server : 자원 존재 / Client : 자원 요청

클라이언트와 서버의 의존성 감소

2. Stateless

Client의 context를 Server에 저장하지 않음 → 구현 단순

Server는 모든 요청을 각자 별개의 것으로 인식하고 처리

3. Cacheable

HTTP의 캐싱 기능 활용 가능 → 응답 시간, 성능 빨라짐

4. Uniform Interface

특정 언어에 종속되지 않고 HTTP 프로토콜을 따르면 모든 플랫폼에 적용 가능

(Loosely Coupling)

REST API 설계

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

[설계 규칙]

- '/'는 계층 관계를 나타냄
- URL 마지막 문자에는 '/' 포함 X
- URL이 길어지면 '-'(하이픈)으로 가독성 높임
- '_'(밑줄)은 URL에 사용 X
- URL 경로는 소문자 사용
- 파일 확장자는 URI에 포함 X → Accept Header 활용

[Resource]

- 동사보다 명사, 대문자보다 소문자
- 객체 인스턴스, DB : 단수 명사
- 서버 및 클라이언트 리소스 : 복수 명사

요약

REST API가 무엇인가?

: 웹 상에서 REST는 HTTP URI와 Method로 자원들을 구분하여 표현한 상태를 클라이언트와 서버가 서로 전송을 주고 받아 CRUD 작업을 진행하는 것이다. 이런 인터페이스를 사용자가 활용할 수 있도록 구축해둔 것이 REST API라고 할 수 있다.

RESTful하다는 게 무슨 뜻인가?

: REST API를 제공하는 웹 서비스를 RESTful하다고 말할 수 있다. 설계 규칙에 맞게 통용되는 일관된 컨벤션을 유지하며 API의 이해도를 높여주는 것이 중요하다.

REST에 사용하는 HTTP 메소드는 무엇이 있는가?

: GET, HEAD, POST, PUT, PATCH, CONNECT, TRACE, OPTIONS

REST가 필요한 이유는? 장단점은 뭔지

: 이젠 웹 서버에서 웹 브라우저, 안드로이드, IOS 등 다양한 멀티 플랫폼에서 통신을 지원할 수 있어야 한다. 이에 필요한 아키텍처로 REST가 통용되기 시작했으며, 다양한 클라이언트를 대응할 수 있게 되었다. 장점은 HTTP 프로토콜 인프라를 사용하여 추가 인프라 구축이 필요 없고, API 메시지 의미가 명확하므로 이해도가 높아지며, 클라이언트와 서버의 의존도를 낮출 수 있다.