



It's Your Life

with





REST

API가 뭔가요!?



그란데 말입니다

그란데 말입니다

API는 뭐였죠!?

뭐의 줄임말이죠!?





A

Application

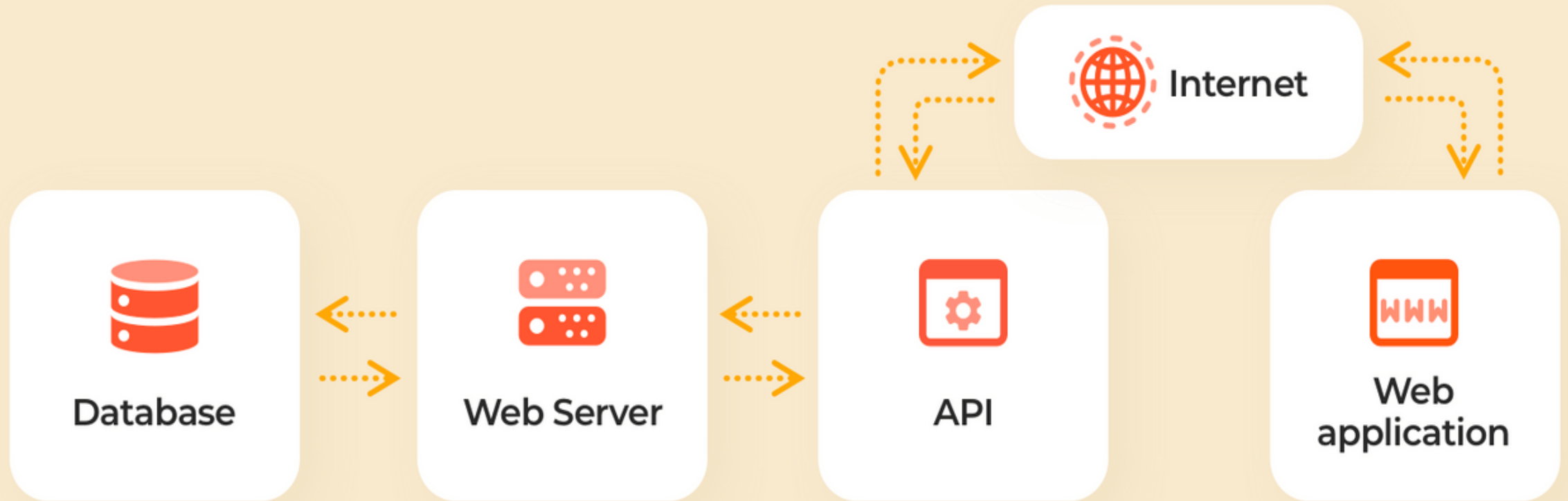
P

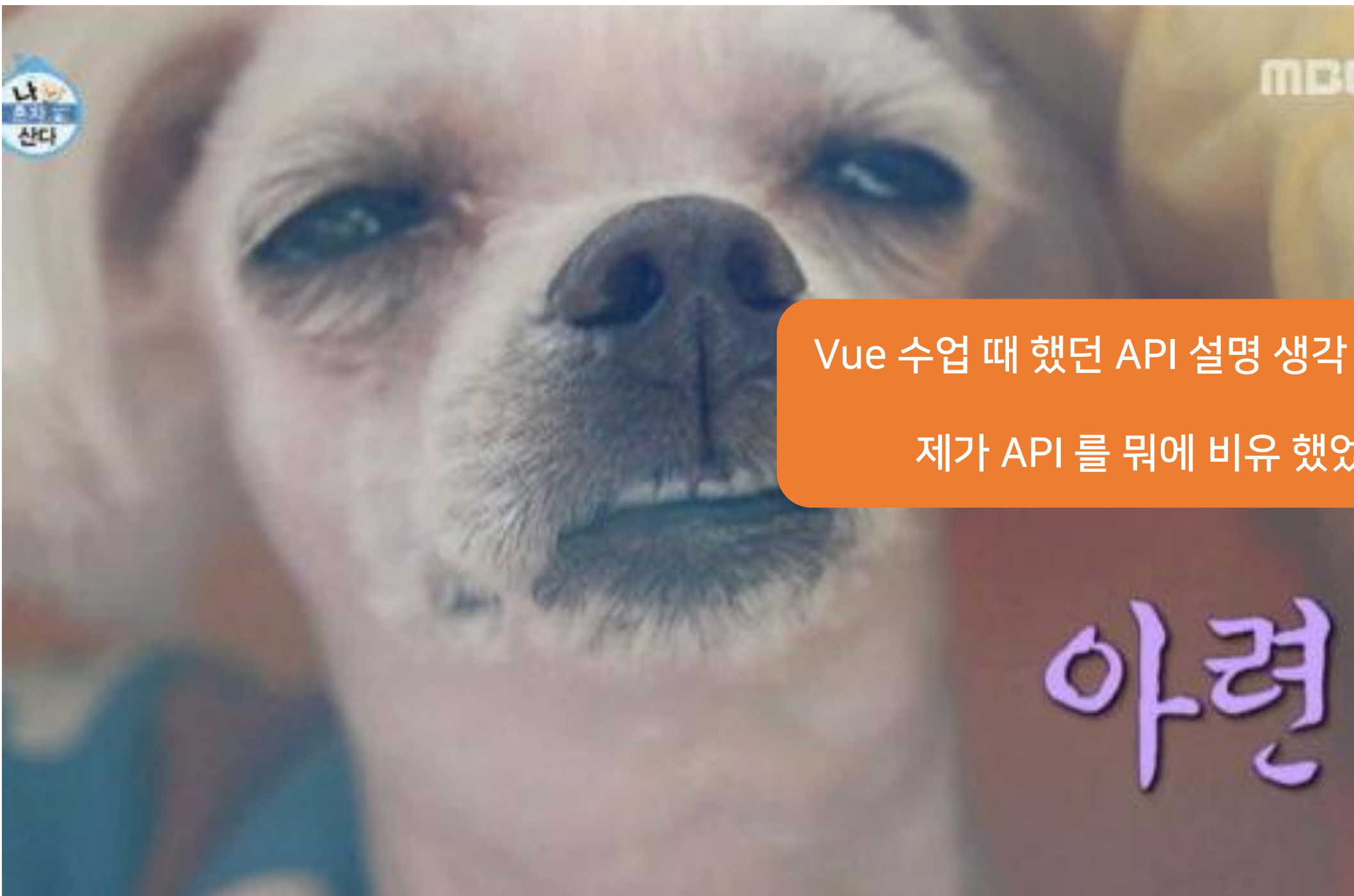
Programming

I

Interface

What is an API?



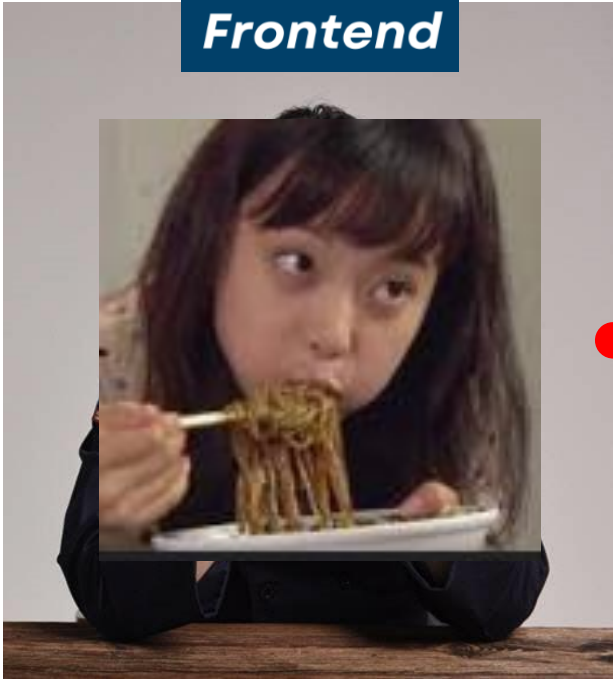


Vue 수업 때 했던 API 설명 생각 나시는 분?

제가 API 를 뭐에 비유 했었죠!?

아련





짜장면
주세요





그란데 말입니다

그럼 API 면 API 지
REST API 는 뭘까요!?

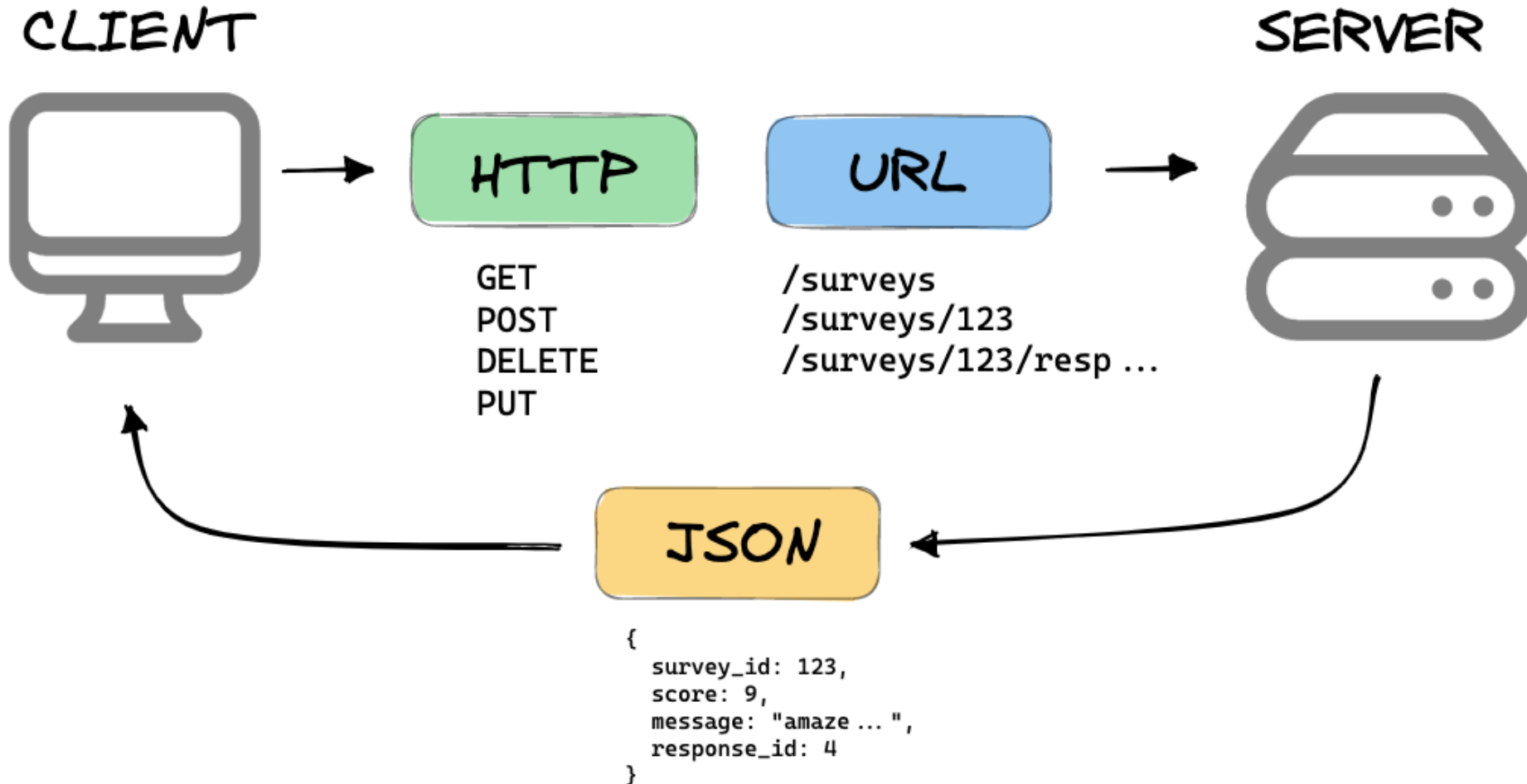


REpresentational State Transfer



Application Programming Interface

WHAT IS A REST API?





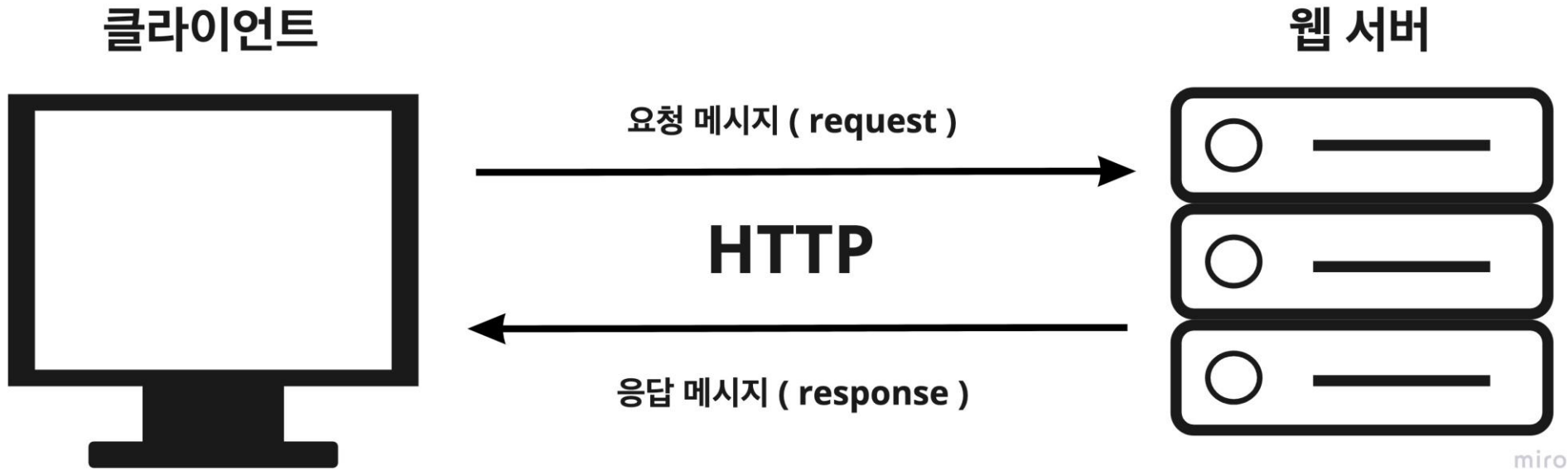
REST

API의 주요 특징

클라이언트 서버 구조



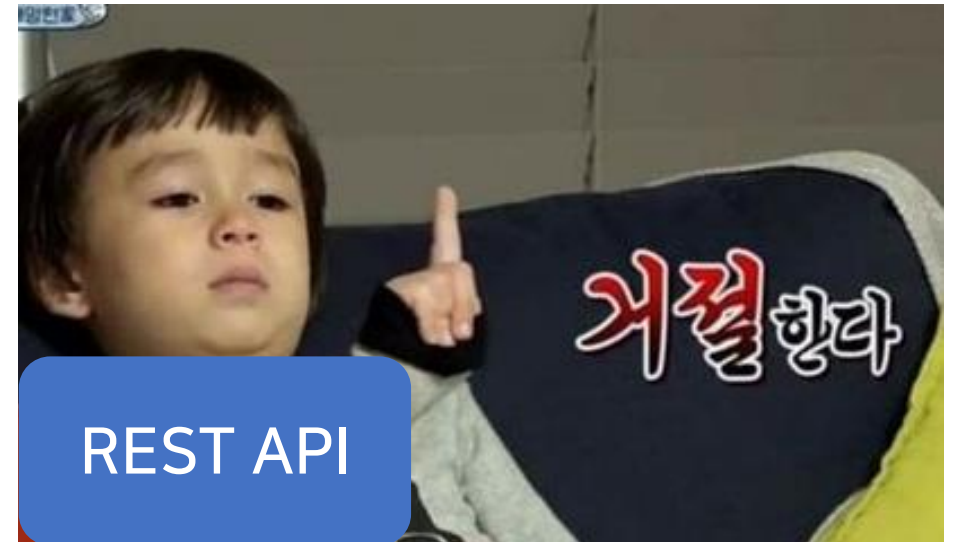
- 기본적으로 요청(Request)과 응답(Response) 구조를 가진다!



무상태(Stateless)



- 각 요청은 독립적이며, 상태가 없이 요청과 응답만 존재 합니다!



통합 인터페이스(Uniform Interface)



- 요청에 대한 구분은 오직 요청 주소(=URI) 만으로만 구분 합니다!

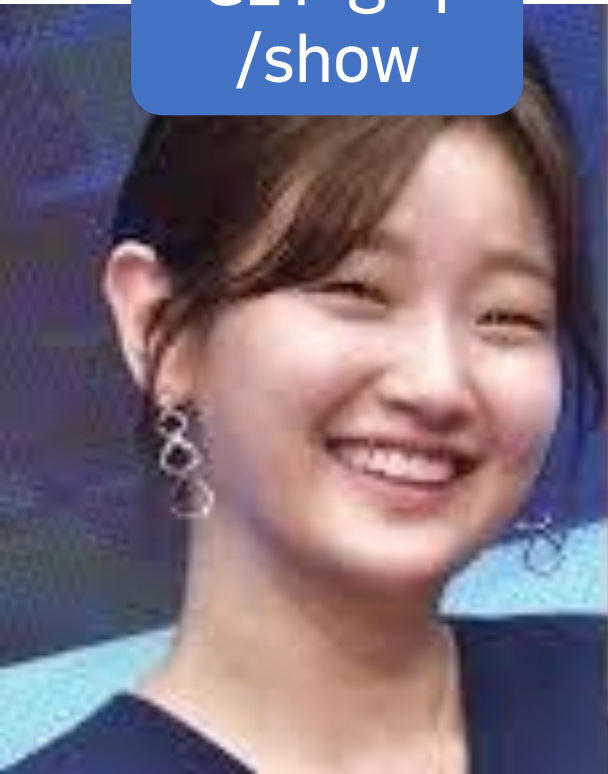


통합 인터페이스(Uniform Interface)



- 요청에 대한 구분은 오직 요청 주소(=URI) 와 Method 만으로만 구분

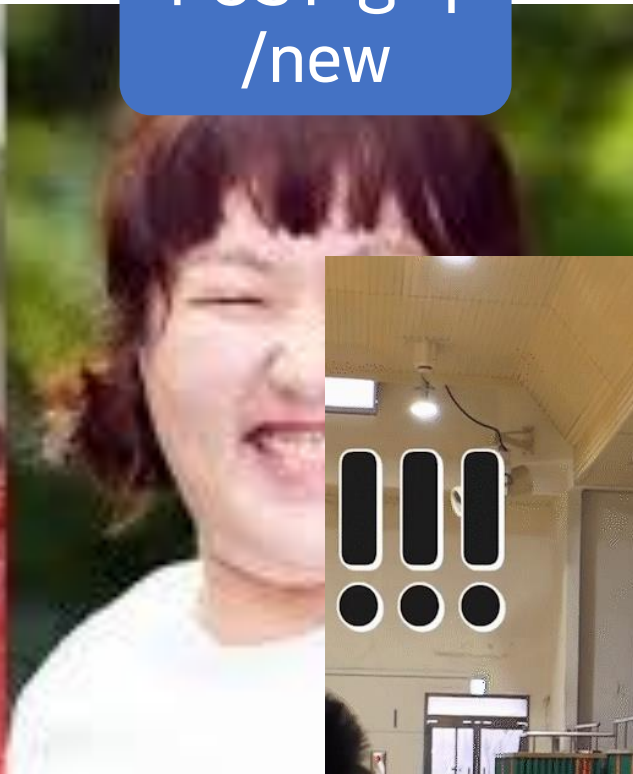
GET 방식
/show



GET 방식
/new



POST 방식
/new



공통된 인터페이스를 제공



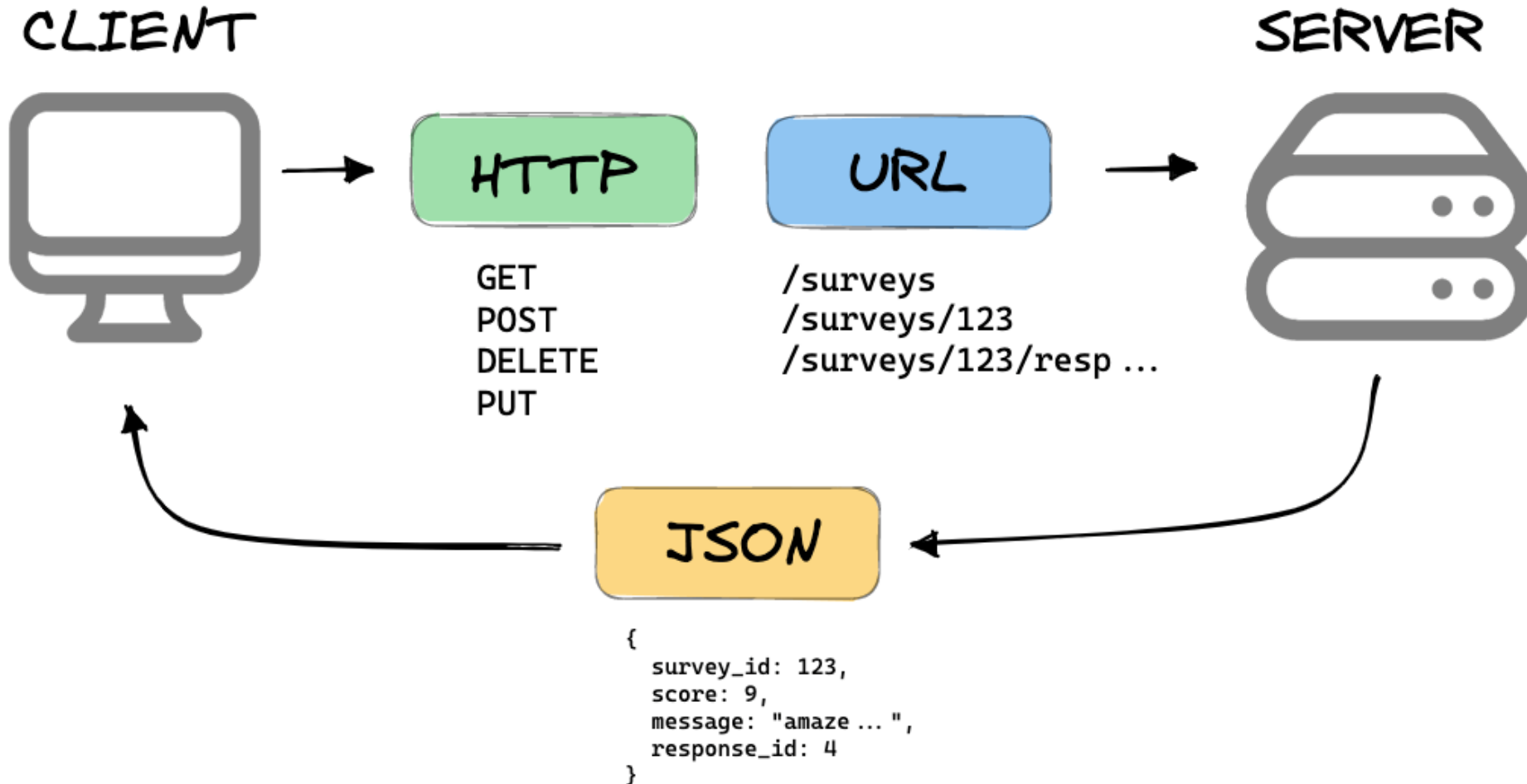
- 모든 요청에 대한 응답은 "JSON" 형태를 통해 전달 됩니다!
- 따라서, 어느 서비스에서나 활용 가능





1. 좋은 글이네요
2. 하지만 읽지는 않았습시다
3. 세 줄 요약 좀

WHAT IS A REST API?





아~ 완벽히 이해했어!

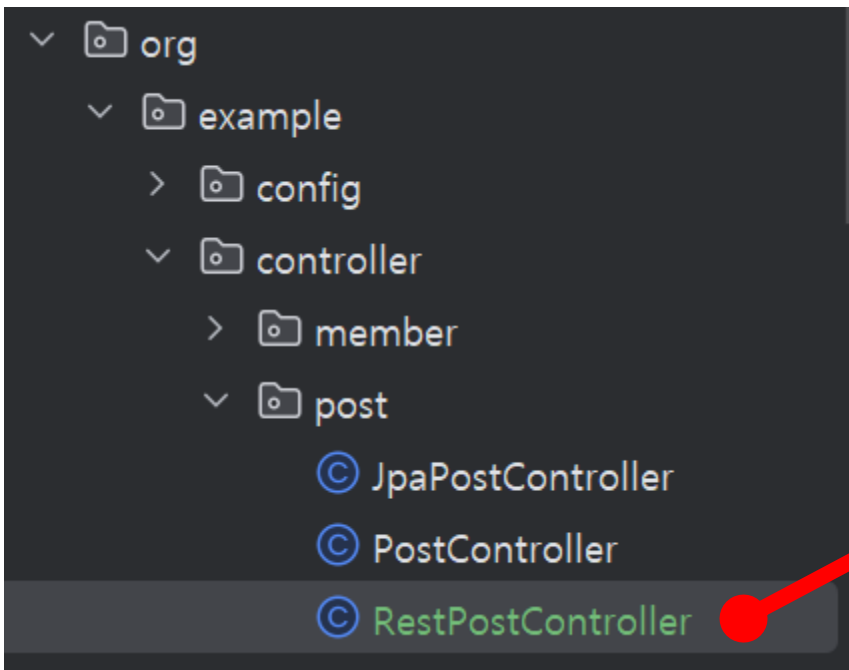


이해못했음

대뜸 SPRING 서버에



REST API 적용하기



REST 방식을 만족하는
API 를 만들기 위해
RestPostController 작성



이제 컨트롤러의 응답은 REST 규칙을 지키는
JSON 형태로 나가야 하므로
@RestController 어노테이션을 사용

```
@RestController new *
@Slf4j
@RequiredArgsConstructor
@RequestMapping(🌐📄"/post/v1/rest")
public class RestPostController {
    private final PostRepository postRepository;
    | private String context = "/post/v1"; 4 usages
```

기존 MVC 패턴의 요청과는
구분하기 위해서
최초 컨텍스트로
/post/v1/rest 를 지정

기존 MVC 패턴에서는 view 폴더의 jsp 파일명을 문자열로 리턴하면 되었지만 이제는 직접 데이터를 JSON 형태로 변환해서 전달해야 하므로 메서드의 리턴 타입을 변경 합니다!

```
// 게시글 목록
@GetMapping("/show") new *
public List<PostDto> postList(HttpServletRequest request, Model model) {
    log.info("=====> 게시글 목록 페이지 호출, " + request.getRequestURI());

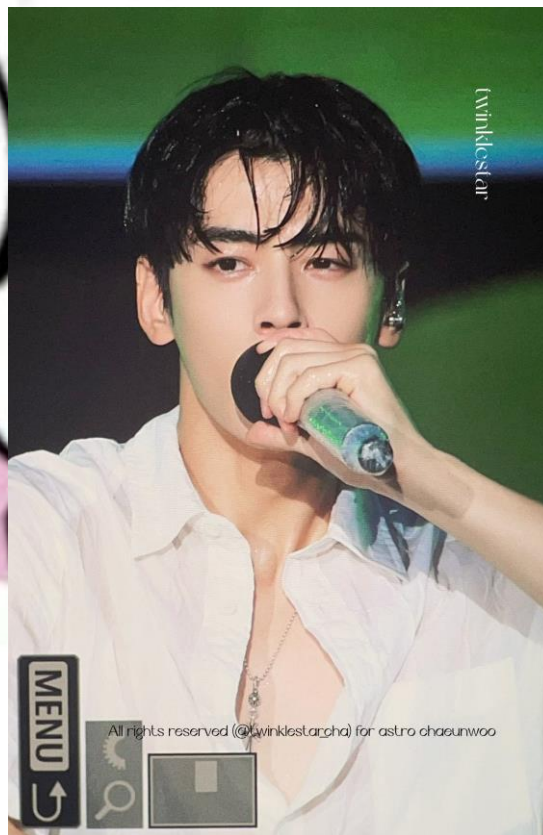
    return postRepository.findAll();
}
```

해당 컨트롤러는 @RestController 어노테이션에 의해서 모든 return 값이 JSON 으로 자동 변환되어 전송 되므로 게시글 List 를 그대로 리턴하면 끝입니다!



설마 이게 다인가.....







REST API에 요청 보내보기

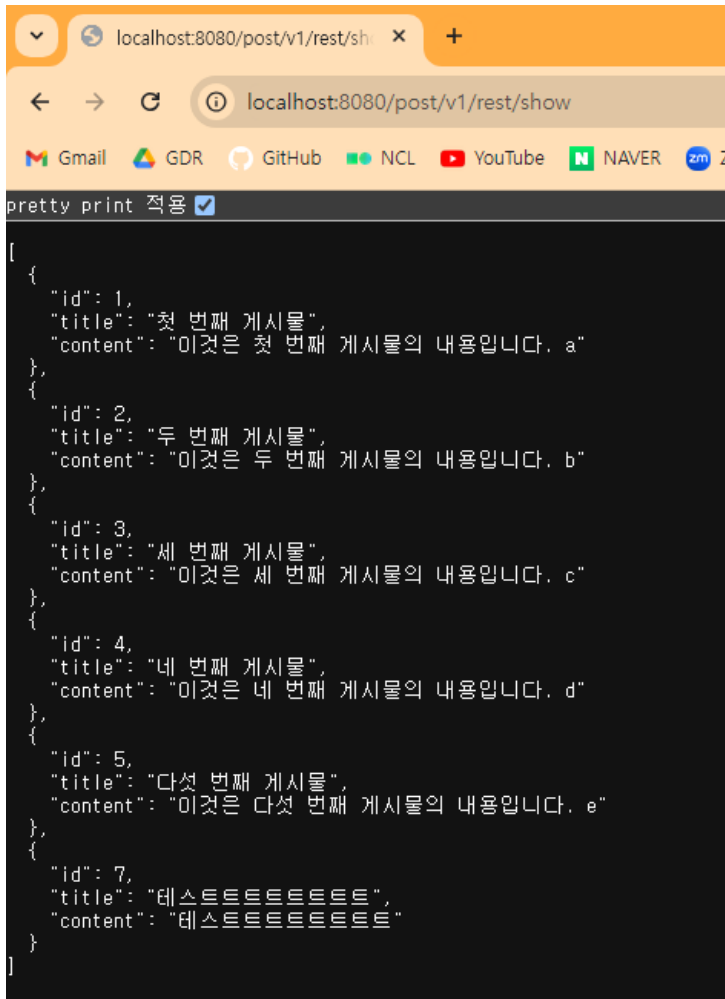
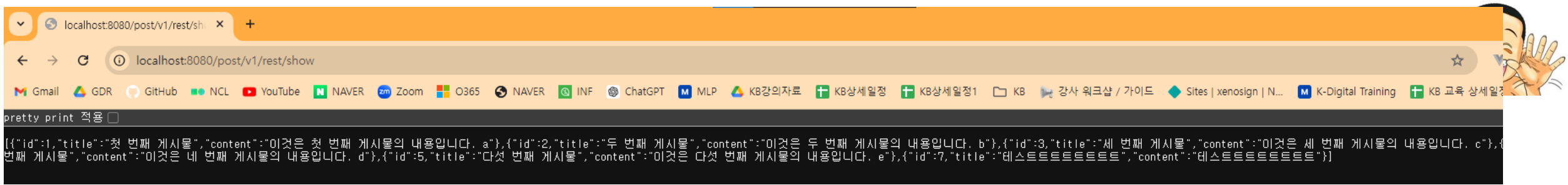
```
@RequestMapping(🌐 "/post/v1/rest")
public class RestPostController {
    private final PostRepository postRepo
    private String context = "/post/v1";

    // 게시글 목록
    @GetMapping(🌐 "/show") new *
```

우리가 요청을 보내야 할 주소는

localhost:8080
+ /post/v1/rest
+ /show 입니다!!







조건 검색 기능도

REST API 로!

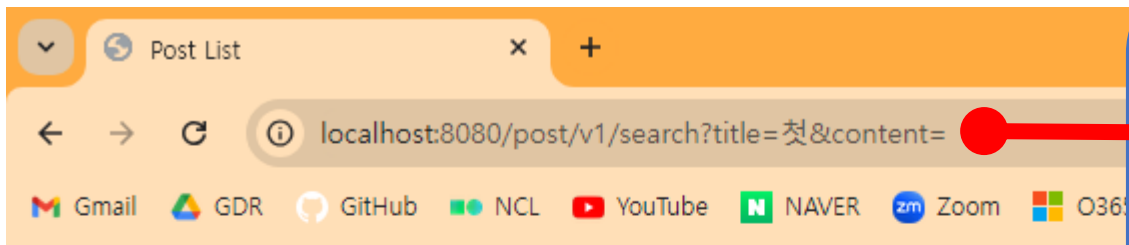


일단 파일명이 아닌
게시글 배열을 리턴해야 하므로
메서드의 리턴 타입을 변경!

```
// 게시글 검색
@GetMapping("/search") new *
public List<PostDto> postSearch(
    @RequestParam("title") String title,
    @RequestParam("content") String content,
    HttpServletRequest request,
    Model model
) {
    log.info("=====> 게시글 검색 기능 호출, " + request.getRequestURI());

    return postRepository.findByCond(title, content);
}
```

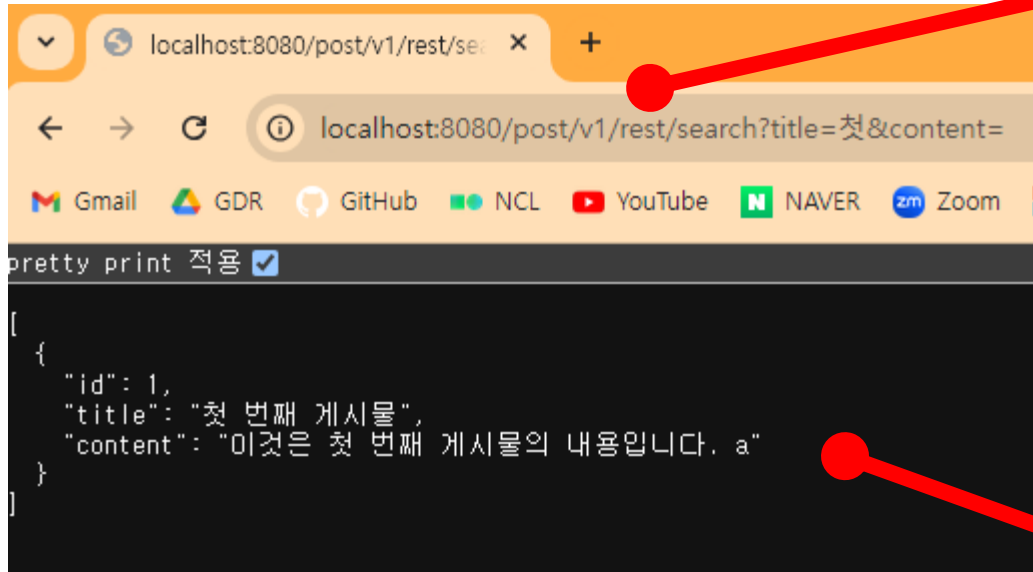
배열 데이터를 그냥 바로 리턴!
스프링이 알아서 JSON 으로 변환해 줍니다!



실제로 검색을 해보면 이러한
주소 요청이 들어가는데요!

우리는 v1 뒤에 rest 만 붙이면
다른 요청이 들어가므로 rest 를 붙여서 요청!

주소에 rest 를 붙여서 방금 만든
RestPostController 에 요청!

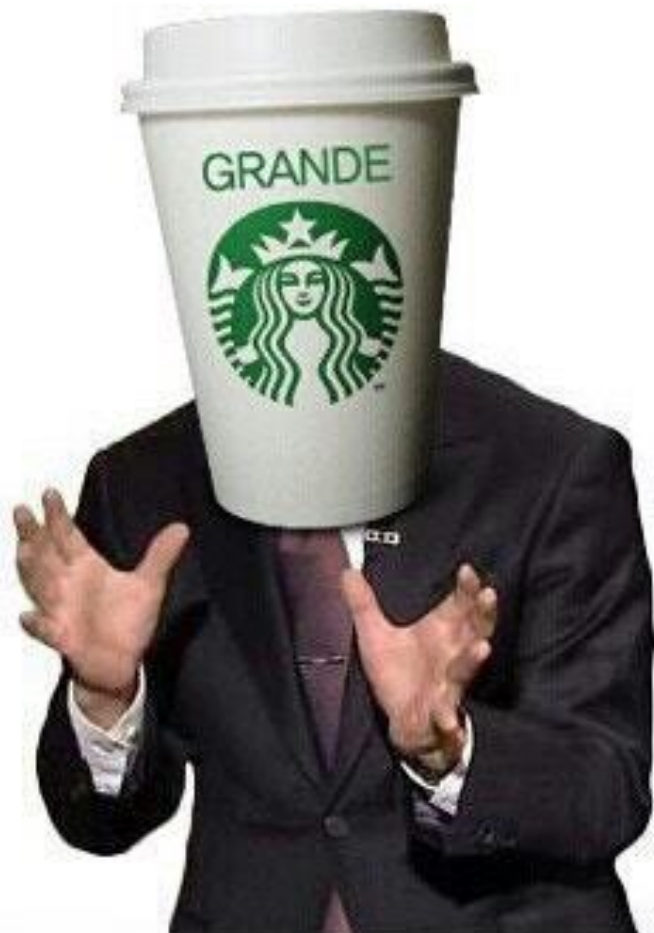


The screenshot shows a web browser window with the address bar displaying `localhost:8080/post/v1/rest/search?title=첫&content=`. Below the address bar, there are several icons for Gmail, GDR, GitHub, NCL, YouTube, NAVER, and Zoom. The main content area of the browser shows a JSON response in a dark-themed editor with a 'pretty print' checkbox checked. The JSON data is as follows:

```
[
  {
    "id": 1,
    "title": "첫 번째 게시물",
    "content": "이것은 첫 번째 게시물의 내용입니다. a"
  }
]
```

Two red arrows point from the text boxes to the browser window. One arrow points to the address bar, and the other points to the JSON response area.

검색 결과 데이터가 잘 들어오는
것을 확인 할 수 있습니다!



그란데 말입니다

그란데 말입니다

매번 이렇게 테스트를 하면
정신 건강에 좋을까요!?





腾讯视频



이 몸, 등장

API 테스트

전통의 강자

POST MAN!



POSTMAN

<https://www.postman.com/>

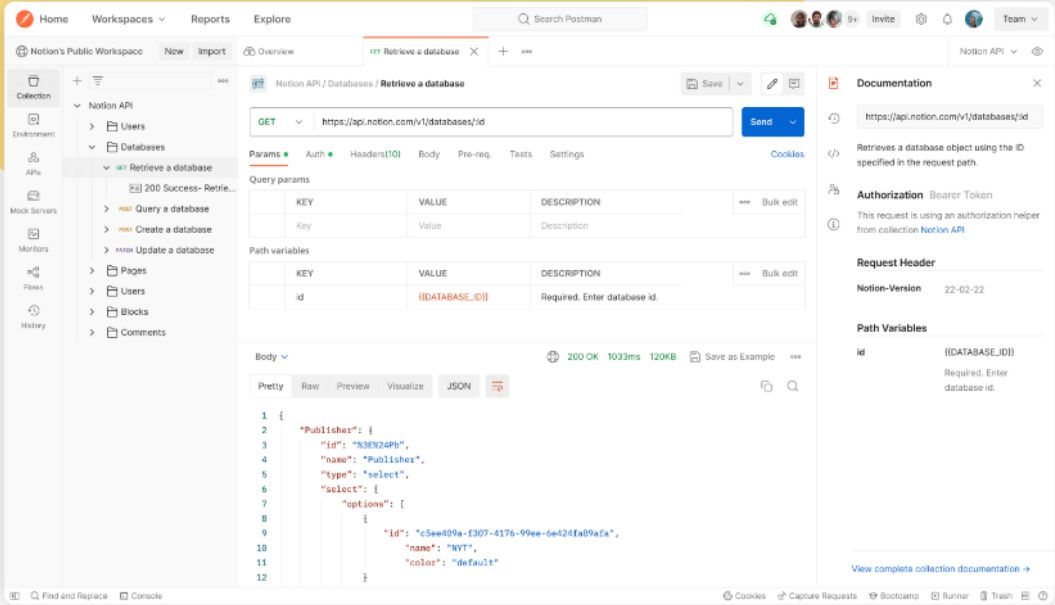


Build APIs together

Over 30 million developers use Postman. Get started by signing up or downloading the desktop app.

Sign Up for Free

Download the desktop app for



[Product ^](#)[Pricing](#)[Enterprise](#)[Resources and](#)

Getting Started

[What is Postman?](#)[Customer Stories](#)[Download Postman →](#)

API Platform Features

[Collections](#)[Workspaces](#)[Postbot](#)[API Client](#)[Browse API Tools](#)

but

Download



Download the app to get started using the Po
browser experience, you can try

The Postman app

Download the app to get started with the Postman API Platform.


 Windows 64-bit



By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes →](#)

Home

 Notion's



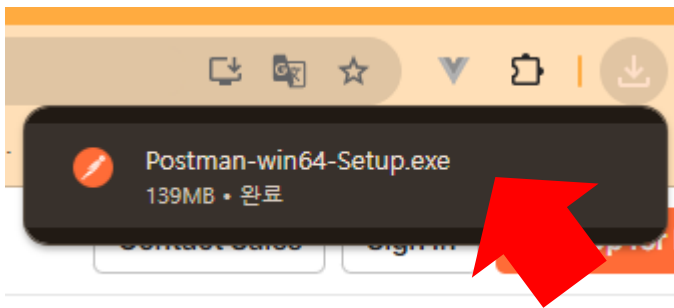
Collection



Environment



History



설치!
진행 시켜!!





Join 30 Million developers who use Postman

Simplify your API design, development and testing workflows with Postman.

- ✦ Save and organize your work in collections and workspaces
- ✦ Share your work with clients and collaborate with teammates
- ✦ Integrate with tools like VS Code, GitHub and BitBucket



나중에 엔간해선 쓰게 될 것이기 때문에
이 참에 가입 하세요!!

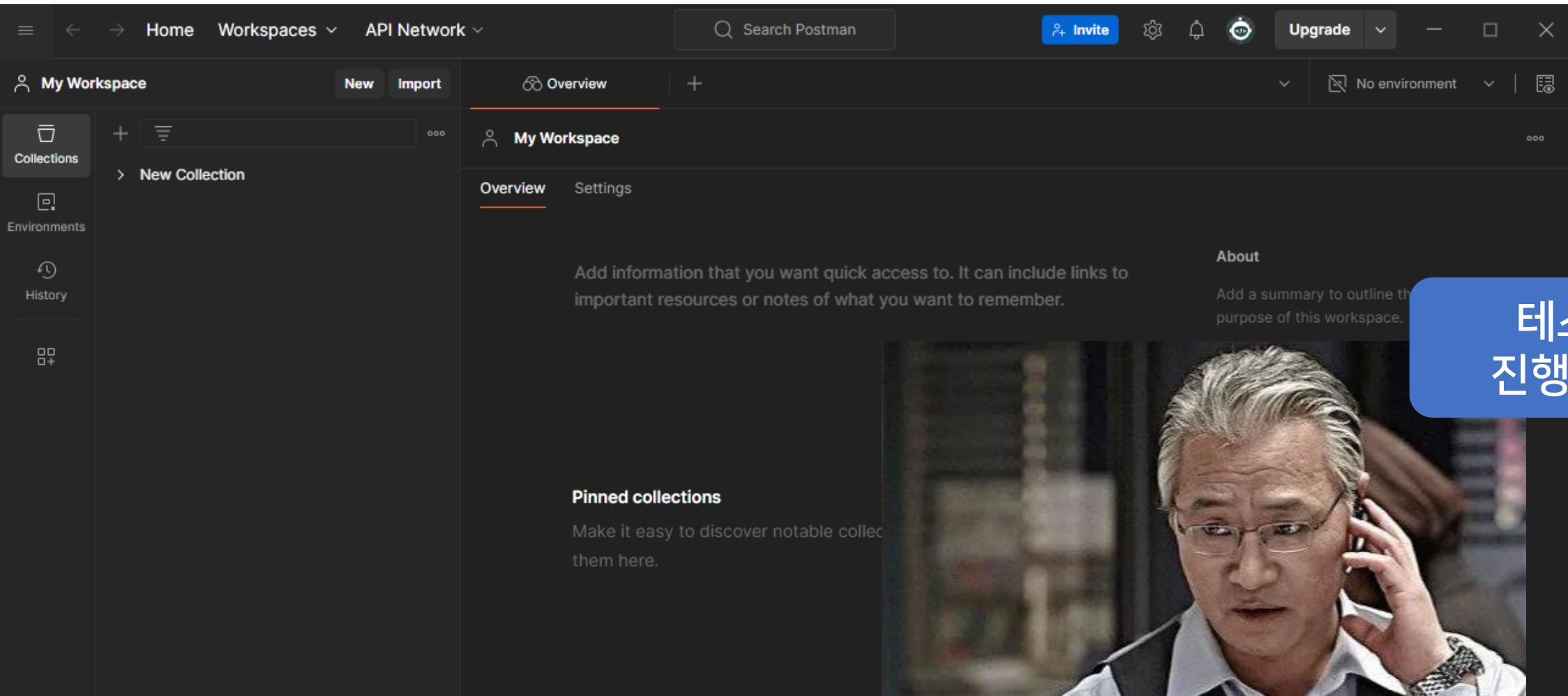
Create a free Postman account

Work email

Create Free Account

Already have an account? [Log In](#)

Or continue with the [lightweight API client](#).



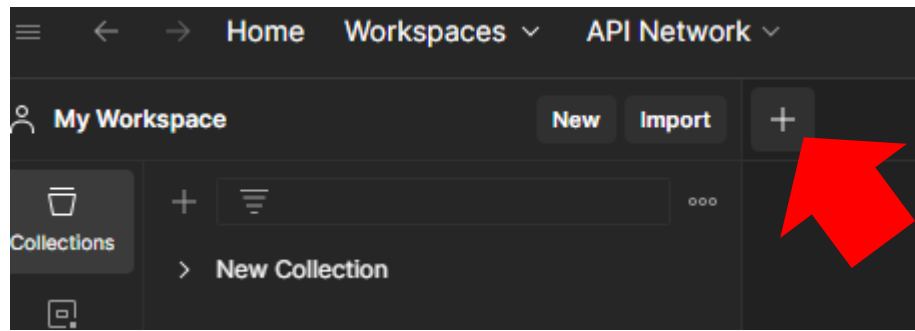
테스트!
진행 시켜!





POST MAN 으로

API 테스트!





GET localhost:8080/post/v1/ +

No environment

HTTP localhost:8080/post/v1/rest/show Save Share

GET localhost:8080/post/v1/rest/show Send

Params Authorization Headers (6) Body Scripts Test Settings Cookies

Query Params

Key	Value
-----	-------

메서드 및 주소를 잘 입력하고 Send 를 누르면!

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

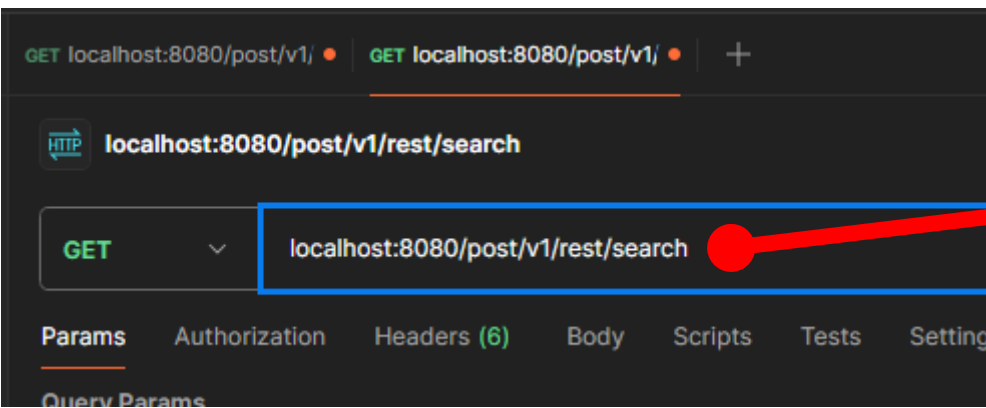
```
1  [
2    {
3      "id": 1,
4      "title": "첫 번째 게시물",
5      "content": "이것은 첫 번째 게시물의 내용입니다. a"
6    },
7    {
8      "id": 2,
9      "title": "두 번째 게시물",
10     "content": "이것은 두 번째 게시물의 내용입니다. b"
11   },
12   {
13     "id": 3,
14     "title": "세 번째 게시물",
15     "content": "이것은 세 번째 게시물의 내용입니다. c"
16   },
17   {
18     "id": 4,
19     "title": "네 번째 게시물",
20     "content": "이것은 네 번째 게시물의 내용입니다. d"
21   },
22   ]
```





POST MAN 으로

검색 API 테스트!



검색 기능은 URL 요청 주소가
/post/v1/rest/search 이므로 요청 주소를 변경

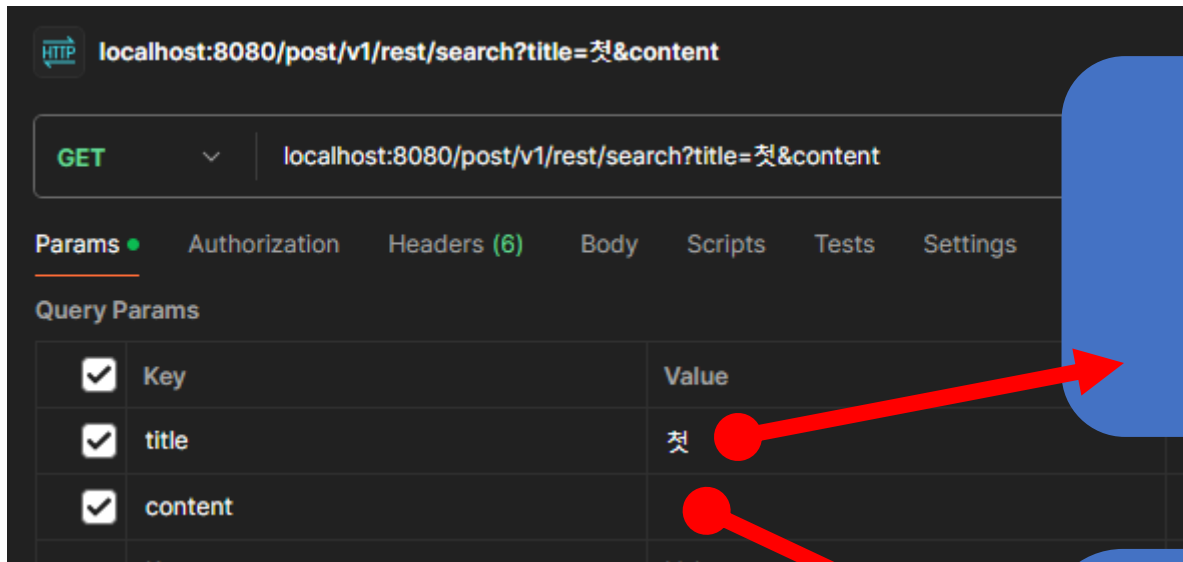


그란데 말입니다

그란데 말입니다

검색 조건인 쿼리 스트링을 직접
우리가 또 적어줘야 할까요!?





쿼리는 직접 작성할 필요 없이
하단의 Query Params 에
key 와 value 를 적절히 넣으시면
POST MAN 이 자동으로 완성!

다만 우리의 컨트롤러 및 Repository 는
2개의 파라미터가 일단 들어오는 것을 가정
따라서, content 도
일단 빈 값을 전달해야 합니다!



개발자

못하겠다...



CSR 과

SSR



그란데 말입니다

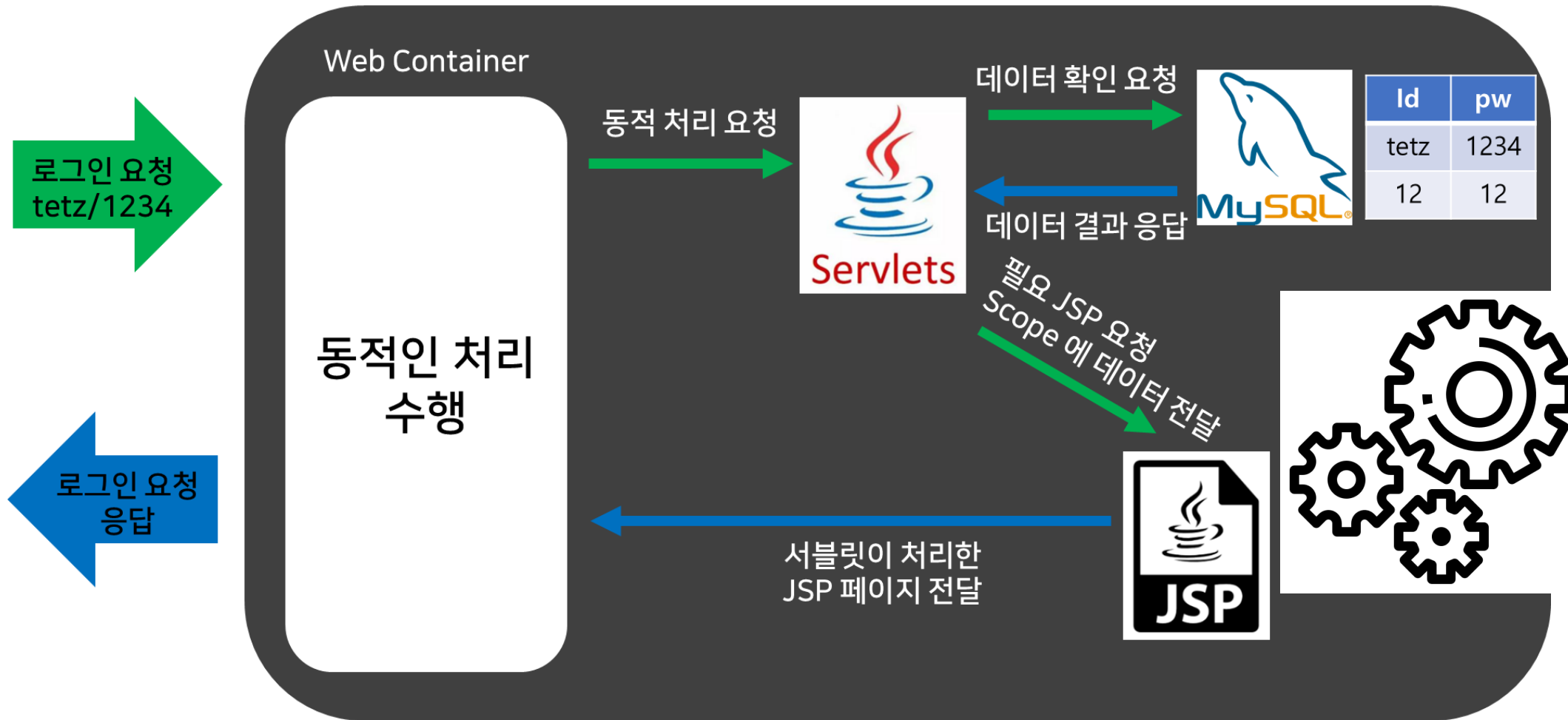
CSR 과 SSR 설명 가능 하신분!?



요 구조는 프론트엔드도 누가 처리를 하고 있죠!?



Tomcat Server







FRONT-END



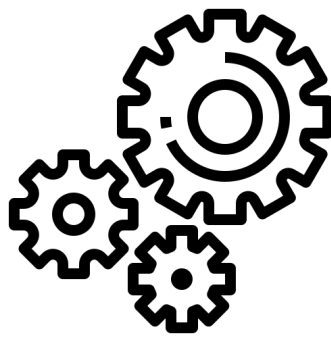
희망나눔 상금
060-700-1212

참... 좋다!

오늘부터



Client



다양한 요청

요청에 대한 응답

Tomcat
Server





FRONT-END

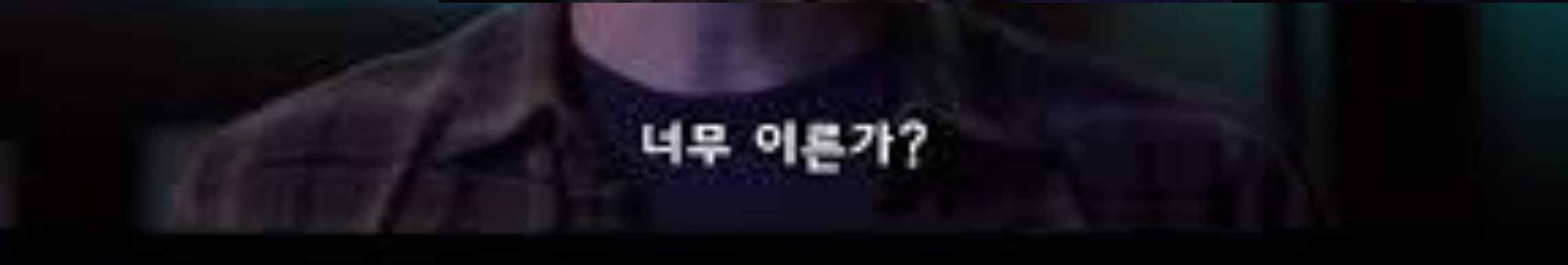


이런 건 직접 해봐야 아는거시구만유!!

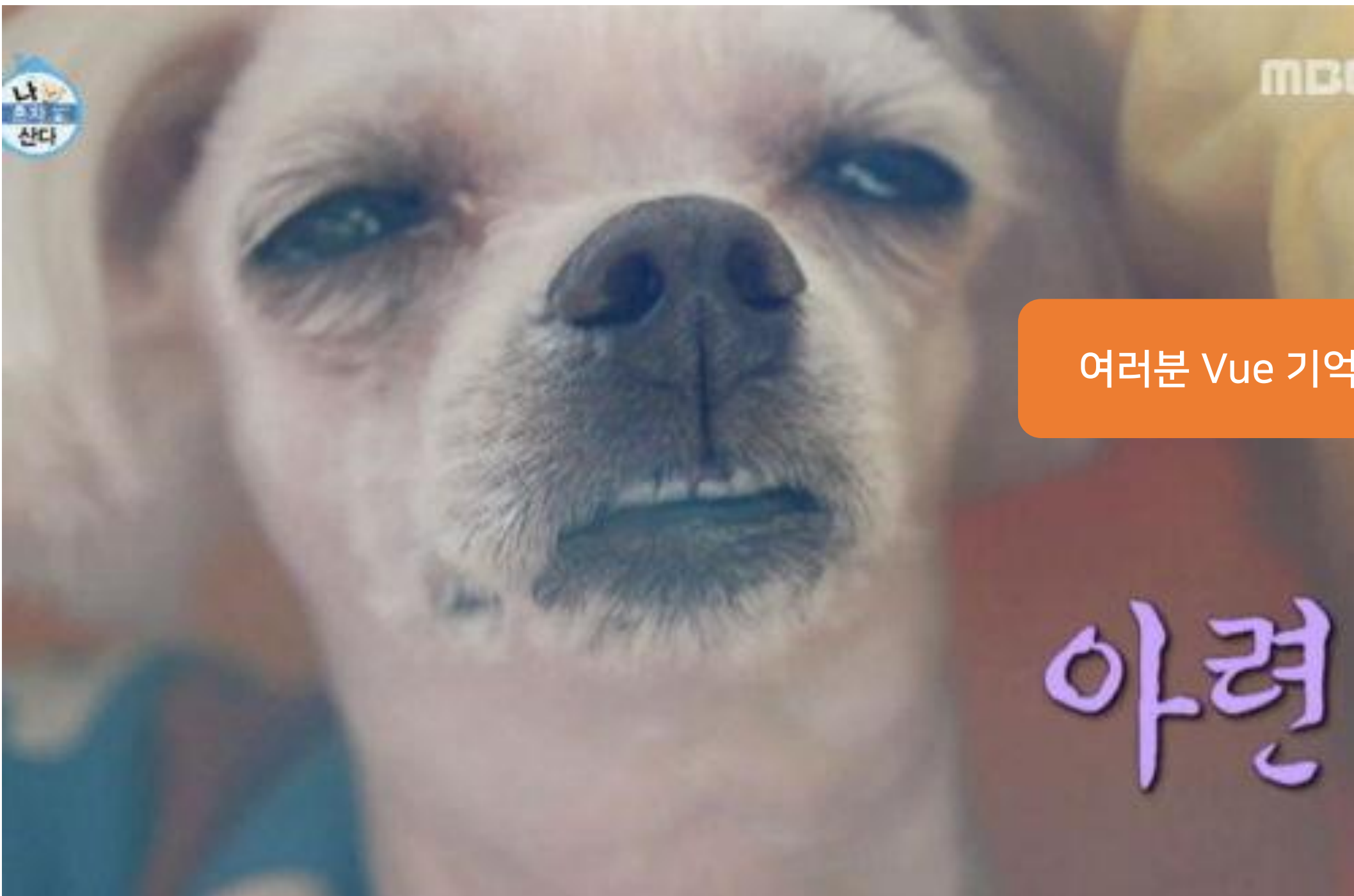




당장 하자



너무 이른가?



여러분 Vue 기억 하나요!?

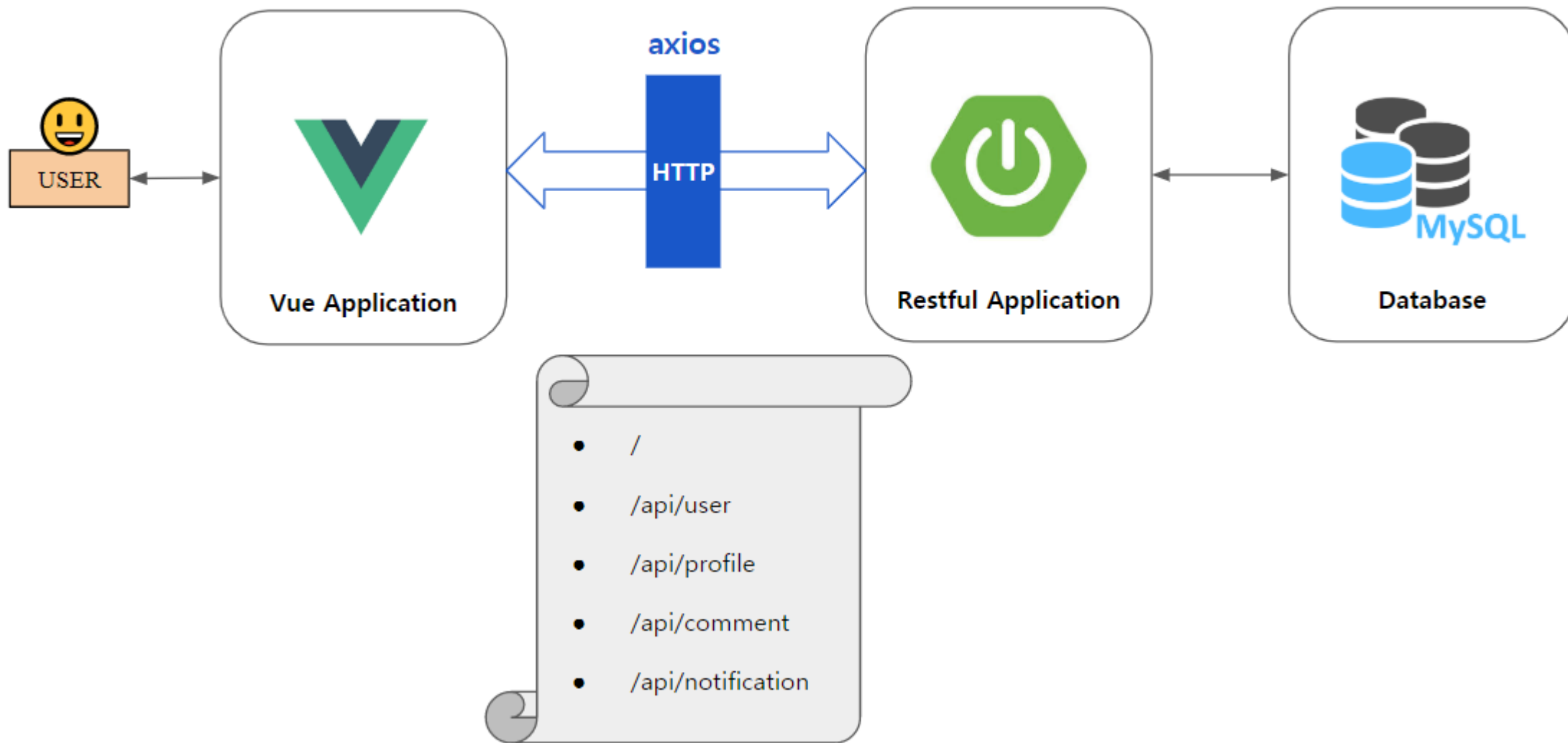
아련

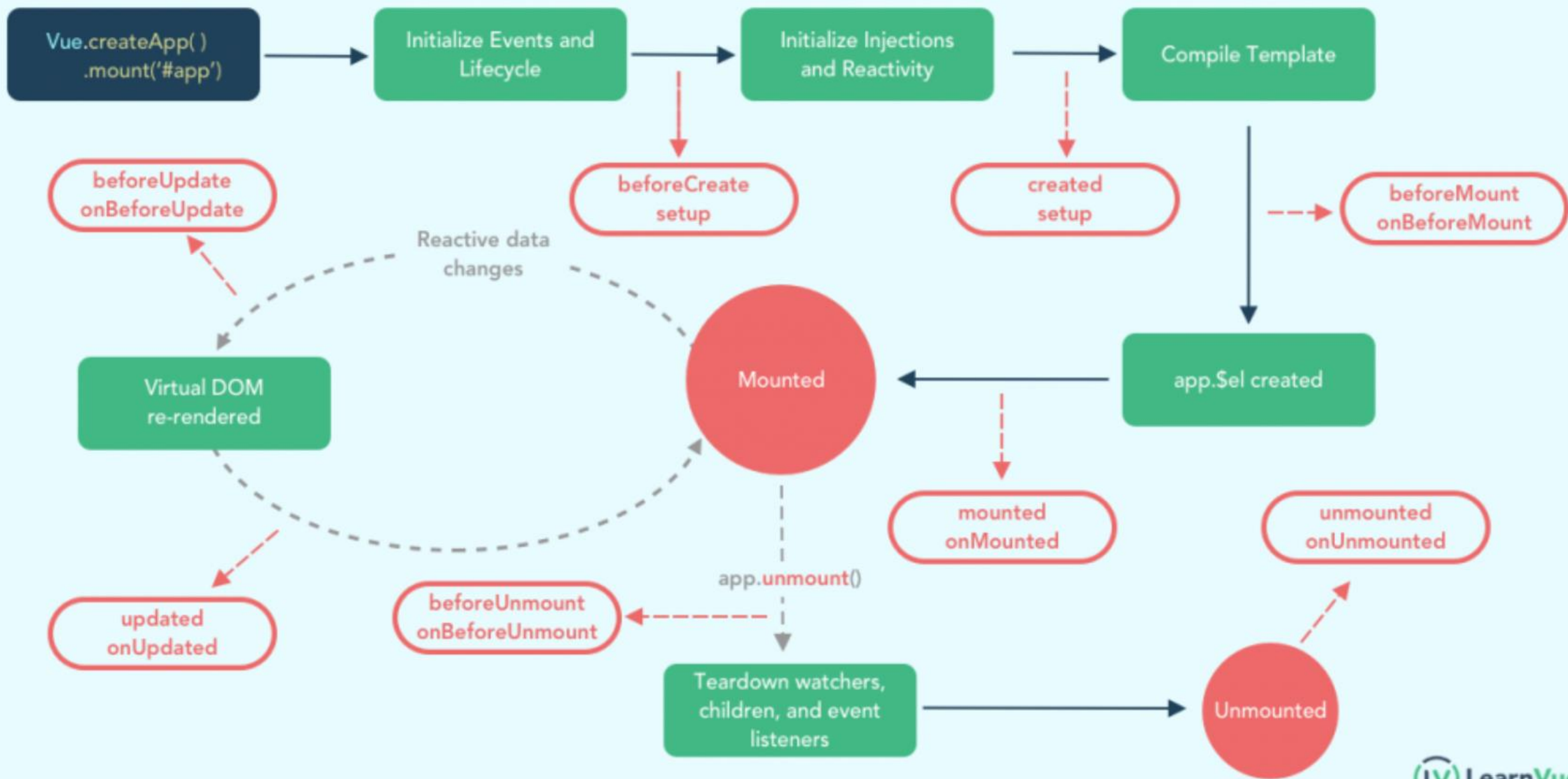


Vue 로

게시판 만들기!

Vue + Spring Boot 기본 구성





게시판

데이터 요청

첫 번째 게시물

이것은 첫 번째 게시물의 내용입니다. a

두 번째 게시물

이것은 두 번째 게시물의 내용입니다. b

세 번째 게시물

이것은 세 번째 게시물의 내용입니다. c

네 번째 게시물

이것은 네 번째 게시물의 내용입니다. d

다섯 번째 게시물

이것은 다섯 번째 게시물의 내용입니다. e

테스트테스트테스트

테스트테스트테스트

게시판

첫 번째 게시물

이것은 첫 번째 게시물의 내용입니다. a

두 번째 게시물

이것은 두 번째 게시물의 내용입니다. b

세 번째 게시물

이것은 세 번째 게시물의 내용입니다. c

네 번째 게시물

이것은 네 번째 게시물의 내용입니다. d

다섯 번째 게시물

이것은 다섯 번째 게시물의 내용입니다. e

테스트테스트테스트

테스트테스트테스트



**좋아!
가는거야!**



Tetz



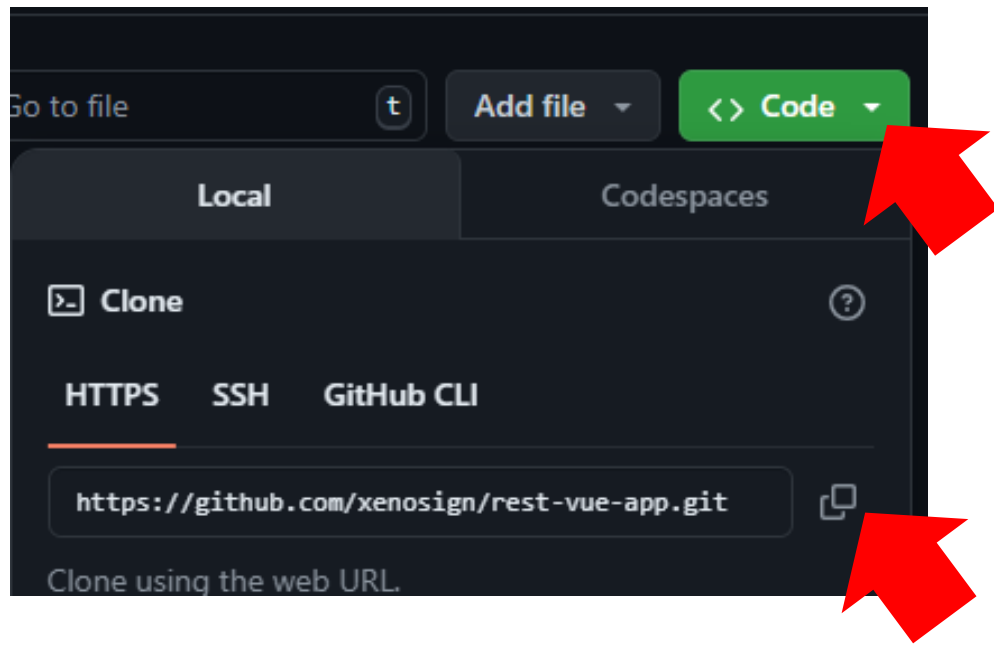
<https://github.com/xenosign/rest-vue-app>

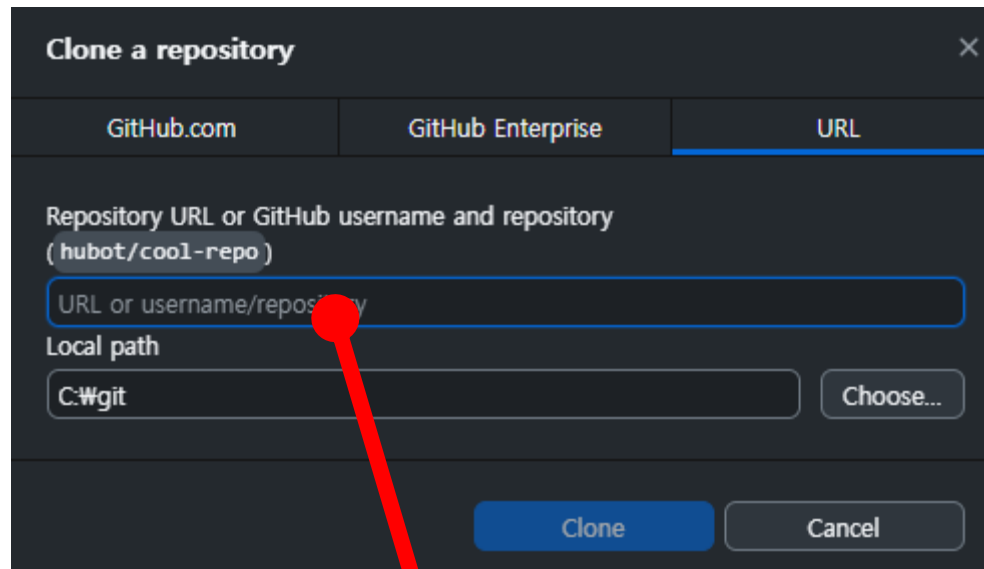
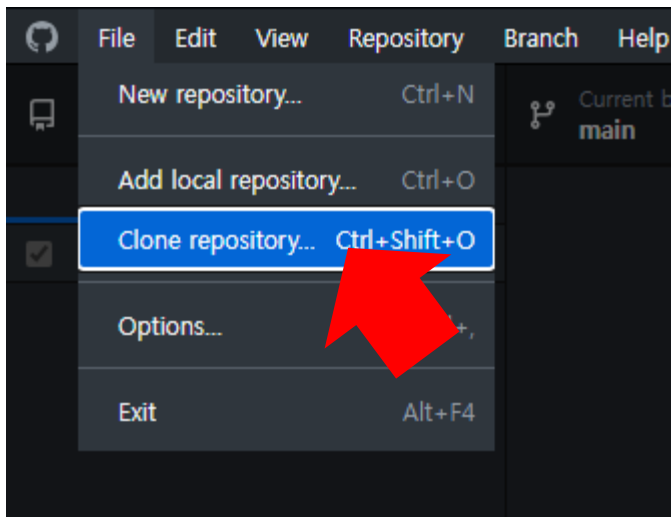




Vue 코드

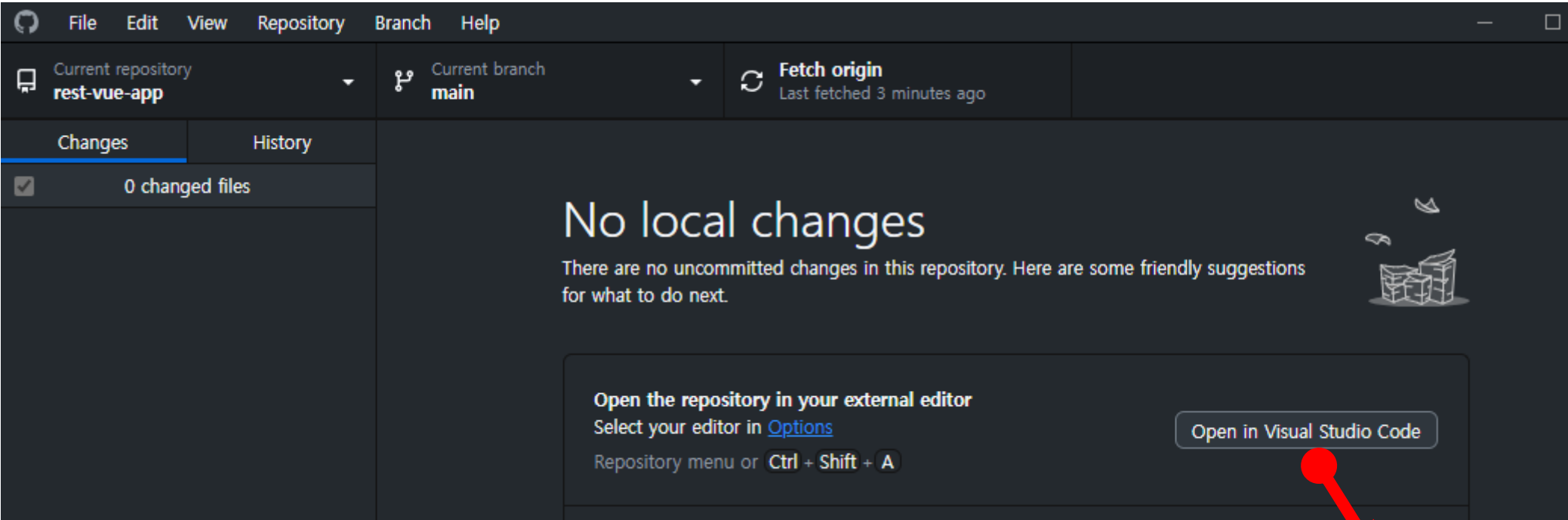
클론 & 실행





복사한 주소를 복붙 → Clone





오~~~~~랜만에
VSCode 로 열어 봅시다!!



오~~~~~랜만에
터미널도 열어 주세요!

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)

탐색기

REST-VUE-APP

.vscode

node_modules

public

src

components

OnClickBoard.vue

OnMountedBoard.vue

App.vue

main.js

.gitattributes

.gitignore

index.html

jsconfig.json

package-lock.json

package.json

README.md

vite.config.js

OnClickBoard.vue

OnMountedBoard.vue

main.js

src > JS main.js

1 import { createApp } from 'vue';

2 import App from './App.vue';

3

4 createApp(App).mount('#app');

5

문제 출력 디버그 콘솔 터미널 포트

student@M503 MINGW64 /c/git/rest-vue-app (main)

\$



```
student@M503 MINGW64 /c/git/rest-vue-app (main)  
$ npm i
```

npm i 로 의존성이 있는 라이브러리를 설치

→ 이 때 Vue 프레임 워크와 Axios 등이
설치 됩니다!

```
student@M503 MINGW64 /c/git/rest-vue-app (main)  
$ npm run dev
```

설치가 완료 되면
npm run dev 로 Vue App 구동!

문제 출력 디버그 콘솔 터미널 포트

VITE v5.4.0 ready in 460 ms

→ Local: <http://localhost:5173/>
→ Network: use --host to expose
→ press h + enter to show help



게시판

데이터 요청

지금은 데이터를 받아오지 못했습니다

외알되?



게시판

Failed to load the posts.



검사 창을

봅시다!

✖ Access to XMLHttpRequest at '<http://localhost:8080/post/v1/rest/show>' from origin '<http://localhost:5173>' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.



✖ Failed to load resource: net::ERR_FAILED [:8080/post/v1/rest/show:1](#) 

has been blocked by CORS policy:

하지만
CORS 이슈가
출동하면 어떨까?

C

S



O

R



그란데 말입니다

CORS(Cross Origin Resource Sharing) 에
대해서 설명 가능하신분!?







CORS

알아보기



이게 약간 비유를 하자면
지역 감정 비슷한 것 입니다!!



<http://localhost:8080>

→ 여기를 경상도라 해봅시다!





입덕직캠

GOT the beat

원터

<http://localhost:8080/winter>

원터네 집(양산)
<http://localhost:8080/winter>



호동이네(마산)
<http://localhost:8080/hodong>



마산



<http://localhost:8080/winter>

<http://localhost:8080/hodong>

클라이언트

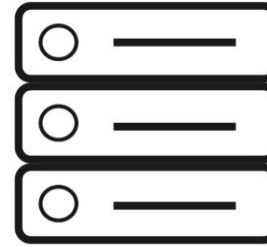
웹 서버



요청 메시지 (request)

HTTP

응답 메시지 (response)



miro



아는형님 전학생 | 에스파



개사도 아이가아~
원터

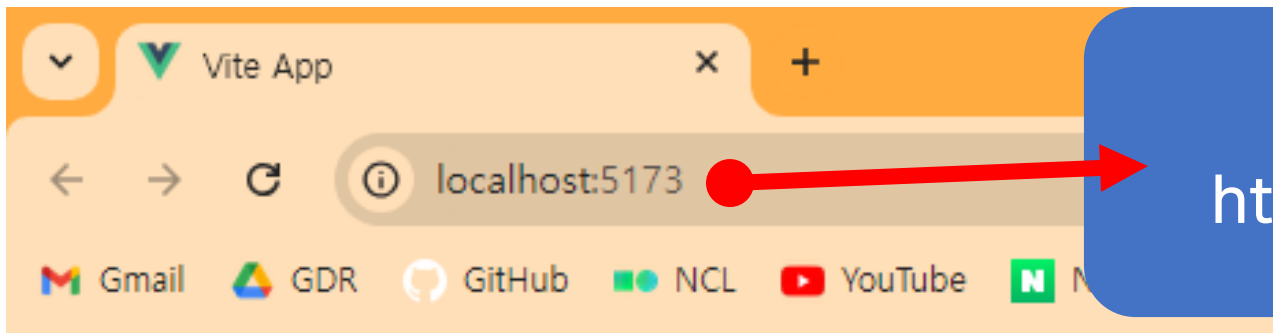
jtbc



여러분의 Vue 어플리케이션은
지금 어느 주소를 가지고 있죠!?



그란데 말입니다



Vue App 은
http://localhost:5173 에 있습니다!





원터네 집(양산)
<http://localhost:8080/winter>



Vue 네(광주)
<http://localhost:5173/Vue>



아는



<http://localhost:8080/winter>

VICODIN for DTS V



<http://localhost:5173/Vue>





경상도가 아니네

<http://localhost:8080/winter>

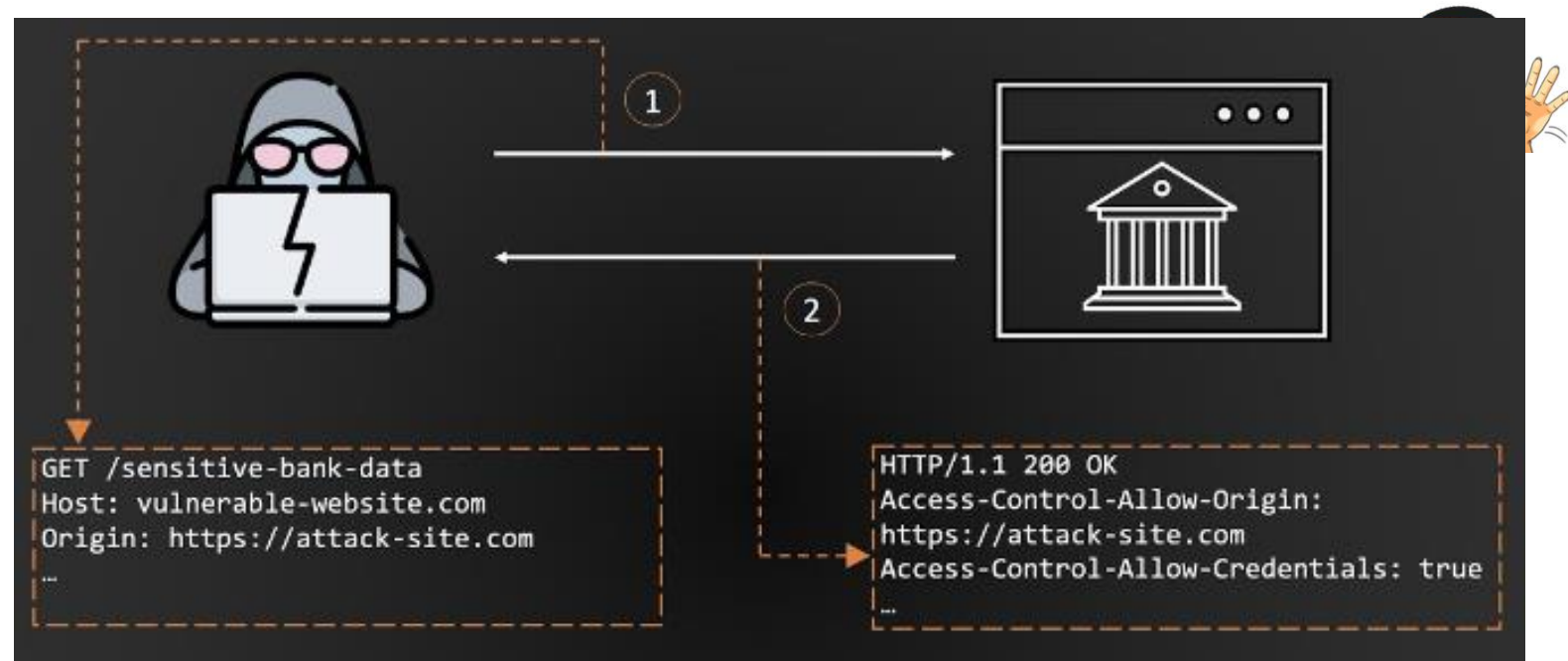


<http://localhost:5173/Vue>





- Cross
- Origin
- Resource
- Sharing





CORS

우회하기



이 안에 뭐가 들었는지 존나 뻔하다
딱 봐도 아이스크림이겠지



으이구 뻔해





개발자

238

세상 귀찮음...



컨트롤러에

CORS 어노테이션 추가



```
© RestPostController.java × © PostController.java  
1 package org.example.controller.post;  
2
```

```
3 > import ...
```

```
13  
14 ∨ @RestController new *  
15 @Slf4j  
16 @RequiredArgsConstructor  
17 @CrossOrigin(origins = "*")  
18 @RequestMapping(🌐"/post/v1/rest")  
19 🚫 public class RestPostController {  
20 🚫 private final PostRepository postRe
```

@CrossOrigin 어노테이션을 붙이면
CORS 이슈 우회가 가능합니다!

```
@CrossOrigin(origins = "*")
```

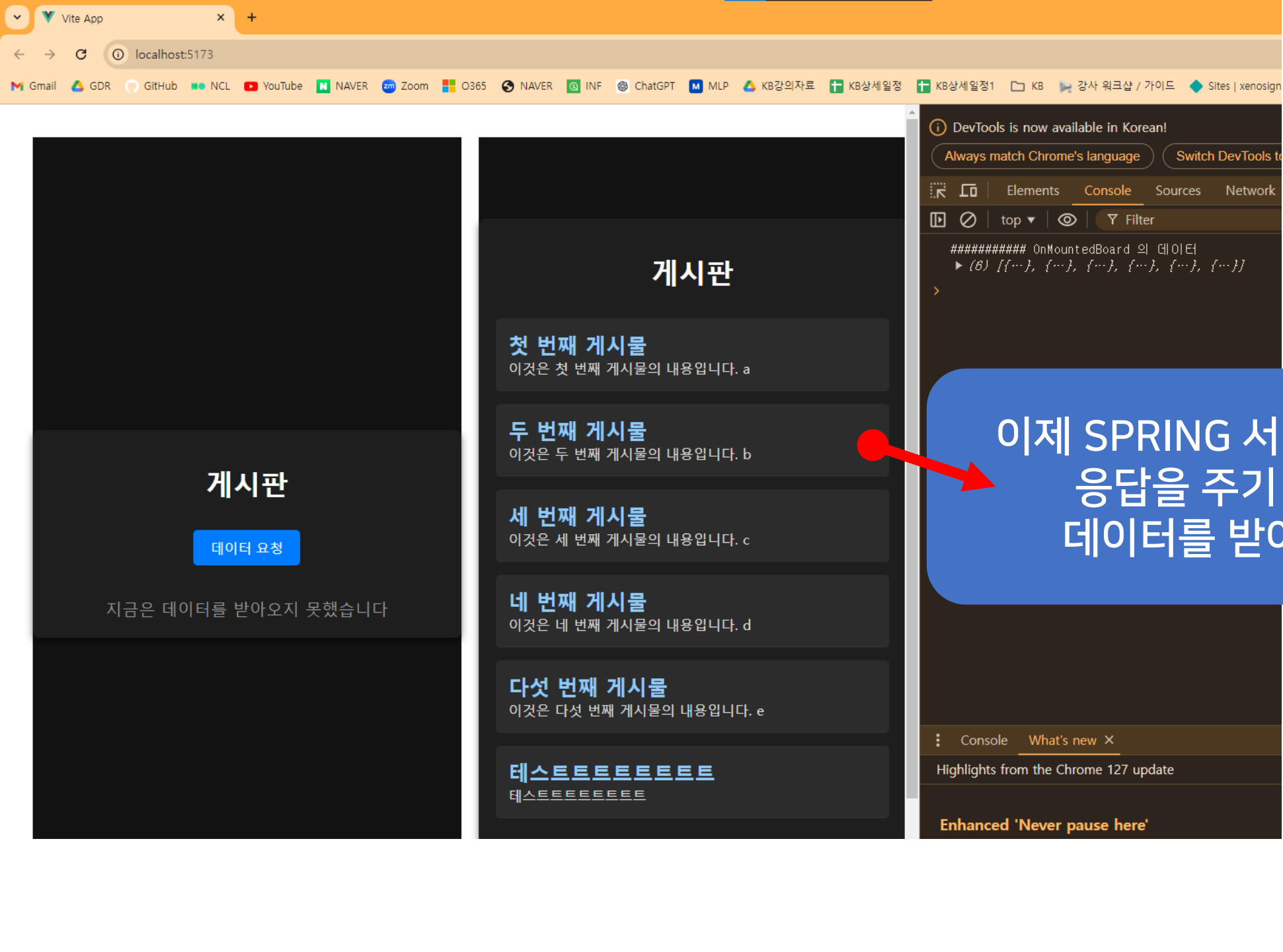


해당 패턴의 주소로 들어오는 요청은
CORS 를 위반해도 응답을 해주겠다는
어노테이션 입니다!

* 을 붙이면 모든 요청에 응답을
하겠다는 의미!



실제로 게시판이
작동하는지 봅시다

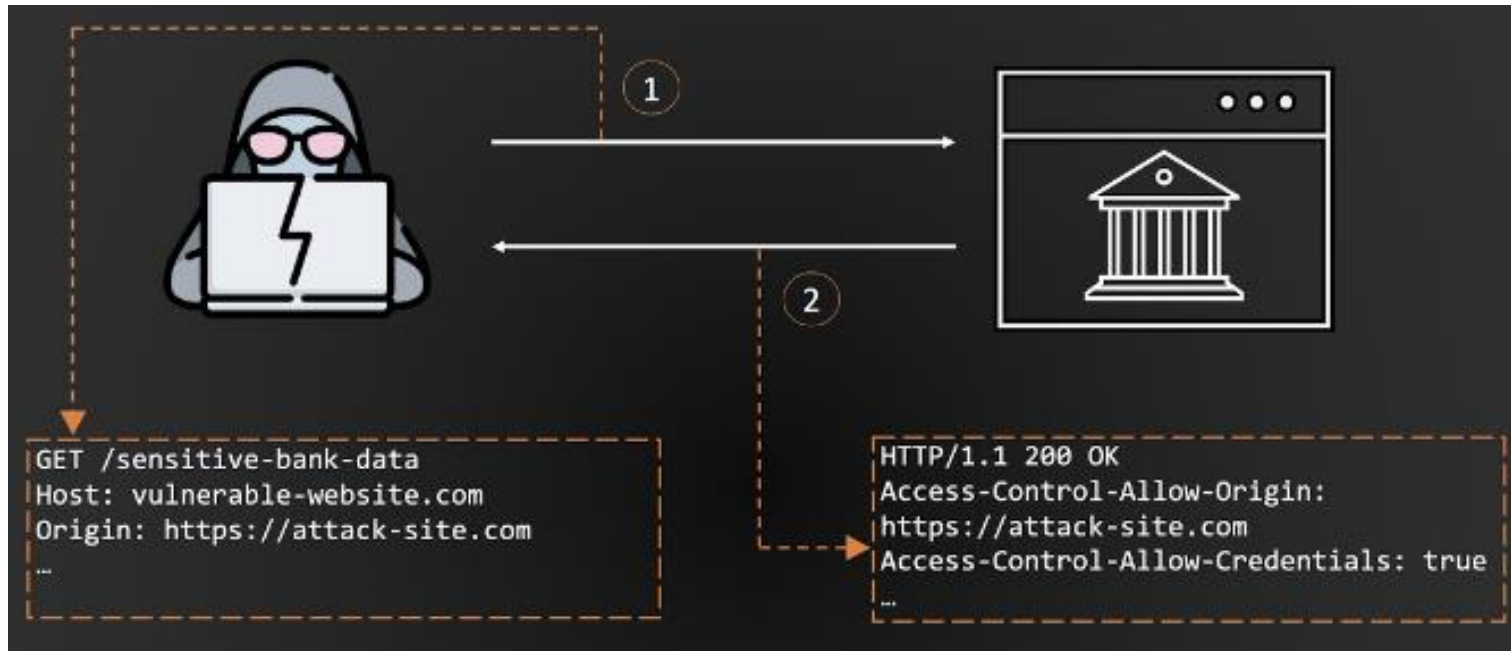


이제 SPRING 서버는 모든 요청에 대해
응답을 주기 때문에 문제 없이
데이터를 받아 올 수 있습니다!!





주의 사항!





```
@RestController new *  
@Slf4j  
@RequiredArgsConstructor  
@CrossOrigin(origins = "http://localhost:5173")  
@RequestMapping("/post/v1/rest")  
public class RestPostController {  
    private final PostRepository postRepository;  
    private String context = "/post/v1"; 2 usages
```

모든 주소의 요청에 서버가 열려 있으면
다양한 공격, 해킹의 위험, 트래픽 증가
등등 많은 위험이 있으므로

특정 주소만 열어두는 것이 안전 합니다!

