MIRI COMPANY

내용은 하기 나름!

CORS 이겨내자 !

CROSS ORIGIN RESOURCE SHARING

Praesent molestie, orci a dictum elementum, tortor molestie, orci a dictum elemvelit quis ligula. Nam non libero ut diamelementum sagittis non id odio. molestie, orci a dictum elementum.

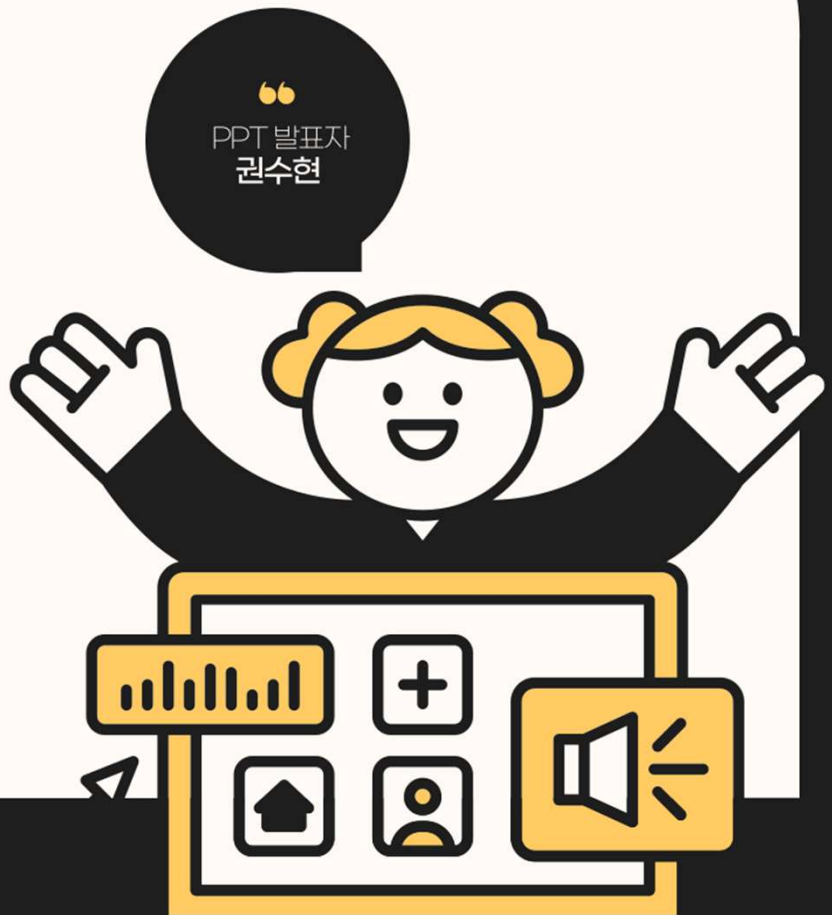
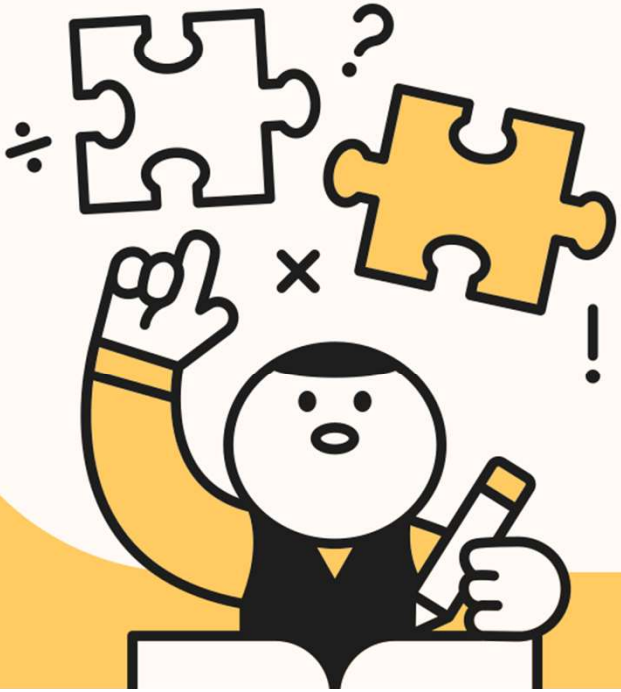


Table of Contents

목차보기

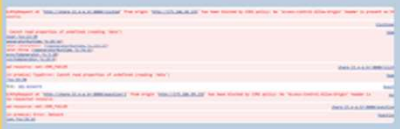


MIRI COMPANY

```
MLHttpRequest at 'http://share-it.p-e.kr:8080/visited' from origin 'http://175.106.99.235' has bee blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on th
esource.                                                                                                                                                    VisitUser

: Cannot read properties of undefined (reading 'data')                                                                                                      Home
:User.jsx:13:30
egeneratorRuntime.js:45:16)
rator.<anonymous> (regeneratorRuntime.js:133:17)
rator.throw (regeneratorRuntime.js:74:21)

                                                                                                                                                            share-it.p-e.kr:8080/visit
erties of undefined (reading 'data')                                                                                                                        Home

                                                                                                                                                            Quest
:8080/question/1' from origin 'http://175.106.99.235' has be n blocked by CORS policy: No 'Access-Control-Allow-Origin' header is                          que

                                                                                                                                                            share-it.p-e.kr:8080/question
                                                                                                                                                            Questior
```

@권수현 살아있니?

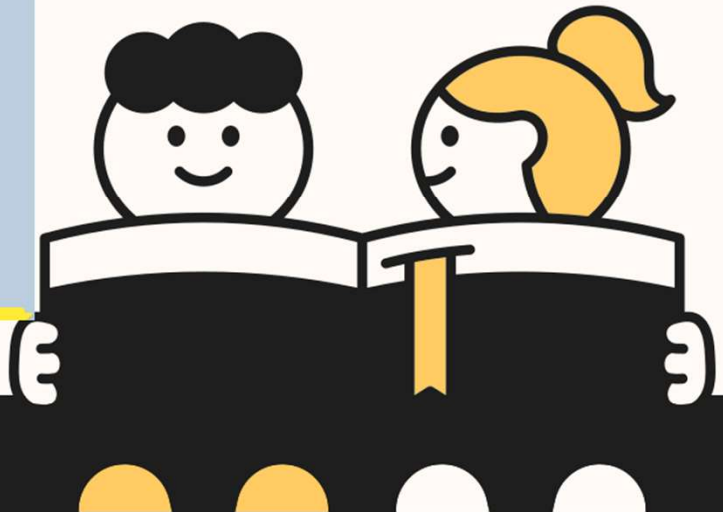
우리 배포 되긴했는데 오전 12:50



오전 12:50

CORS뜨는데 ㅋㅋㅋ 오전 12:50

오전 12:50



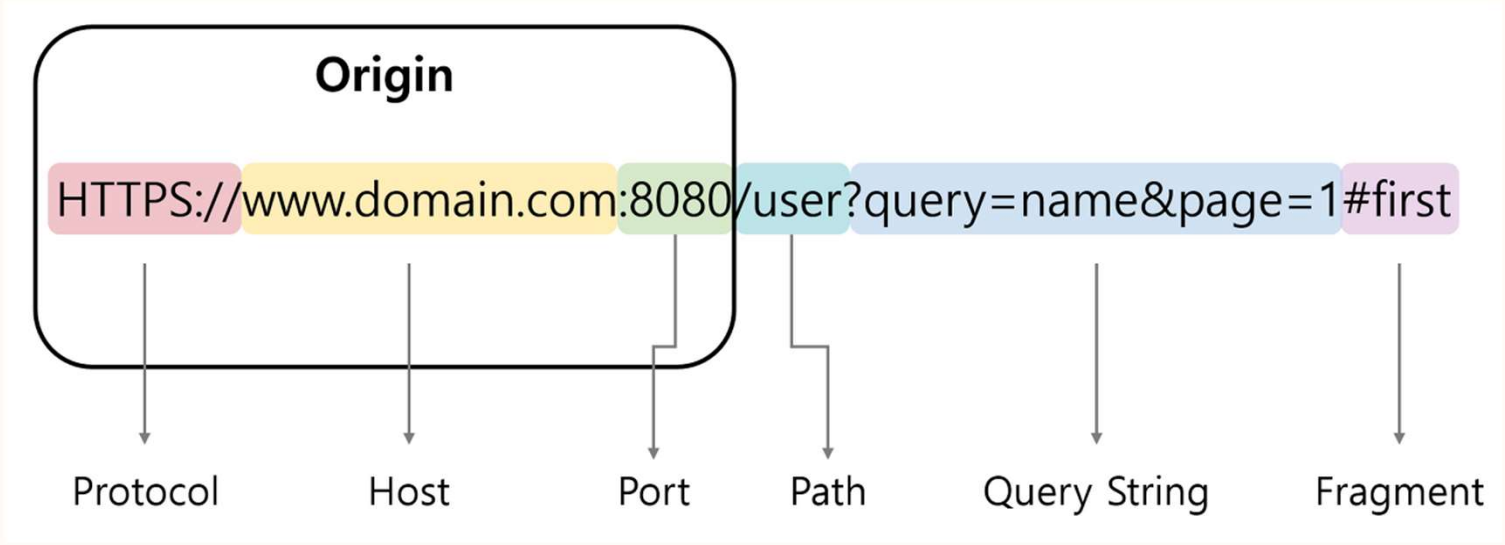
CORS

Cross-Origin Resource Sharing

교차 출처 리소스 공유 정책

: 다른 출처 사이에 리소스 공유가 가능 / 불가능

출처 Origin


Origin
HTTPS://www.domain.com:8080/user?query=name&page=1#first
Protocol
Host
Port
Path
Query String
Fragment

same-origin

동일 출처 정책 (Same-Origin Policy)

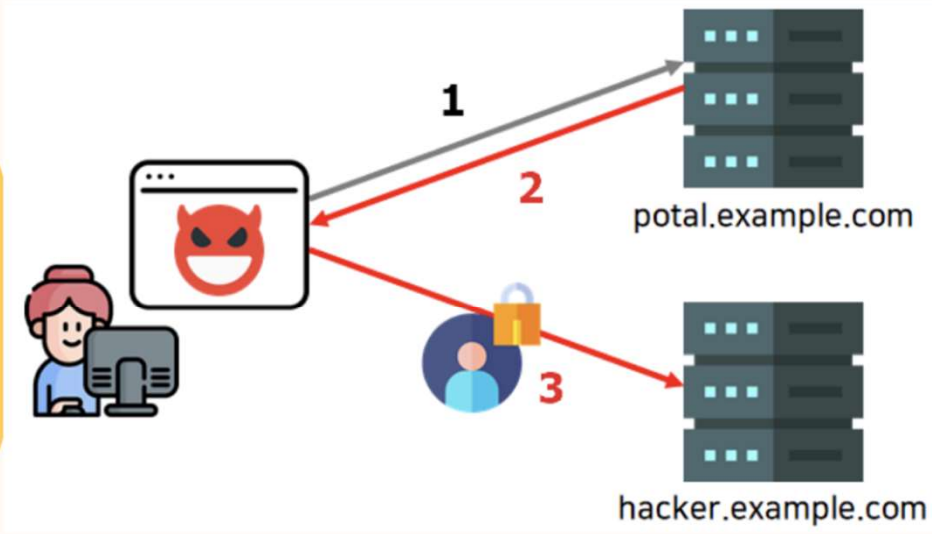
동일한 출처에서만 리소스를 공유할 수 있다.

즉, 다른 출처(Cross-Origin)에 있는
리소스는 상호작용이 불가능

실제 사례에는 보통 Javascript를 통한 AJAX통신과정에서 보통 발생

cross-origin

왜 필요할까 ?



다른 서버의 내용까지 읽을 수 있게된다면, 사용자의 키보드 입력을 가로채는 스크립트를 다른 페이지에 심을 수도 있음.

해커가 CSRF(Cross-Site Request Forgery)나 XSS(Cross-Site Scripting) 등의 방법을 이용해서 어플리케이션에서 해커가 심어놓은 코드가 실행되어 개인 정보를 가로챌 수 있음.

https://www.domain.com:8080

https://www.domain.com:8080/about	O
https://www.domain.com:8080/about?username=inpa	O
http://www.domain.com:8080	X
https://www.another.co.kr:8080	X
https://www.domain.com:8888	X
https://www.domain.com	X

요청 헤더 소스 보기

Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Access-Control-Request-Headers: content-type
Access-Control-Request-Method: POST
Connection: keep-alive
Host: localhost:4000
Origin: http://localhost:3000
Referer: http://localhost:3000/

응답 헤더 소스 보기

Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: content-type
Access-Control-Allow-Methods: GET,POST,OPTIONS,DELETE,PUT,PATCH
Access-Control-Allow-Origin: http://localhost:3000
Connection: keep-alive

1. 클라이언트에서 요청헤더에 Origin을 담아 보낸다.

2. 서버는 응답헤더에 Access-Control-Allow-Origin을 담아 클라이언트에 전달한다.

3. 클라이언트에서 Origin과 서버가 보내준 Access-Control-Allow-Origin을 비교한다.

해결방법

1. Chrome확장 프로그램 이용

2. 프록시 서버 이용

3. 서버에서 Access-Control-Allow-Origin헤더 세팅

3. 서버에서 Access-Control-Allow-Origin헤더 세팅

```
@Configuration
public class WebConfig implements WebMvcConfigurer {
    kwonssshyeon *
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping( pathPattern: "/**")
                .allowedOrigins("http://share-it.p-e.kr","http://localhost:3000") // 허용할 출처
                .allowedMethods("*")
                .allowCredentials(true);
    }
}
```

내가 겪은 CORS오류 상황

쿠키에 세션 정보를 담아 매번 요청헤더에 포함해야하는 상황

Same-Site 속성

다른 도메인 간의 통신에 대한 보안에 대한 설정

내가 겪은 CORS 해결방법

- Same-Site속성을 None → Secure을 None으로 설정

  → https 프로토콜로 바꿔줘야함

- 도메인을 똑같이 맞춰주는 방법

끝! 입니다

"감사합니당!"

Praesent molestie, orci a dictum elementum, tortor molestie, orci a dictum elemvelit quis ligula. Nam non libero ut diamelementum sagittis non id odio. quis ligula. Nam non libero ut diamelementum sagittis non id odio.
molestie, orci a dictum elementum,