PRESENTATION

29주차 주제 CORS

SOP와 CORS

by mun

CORS

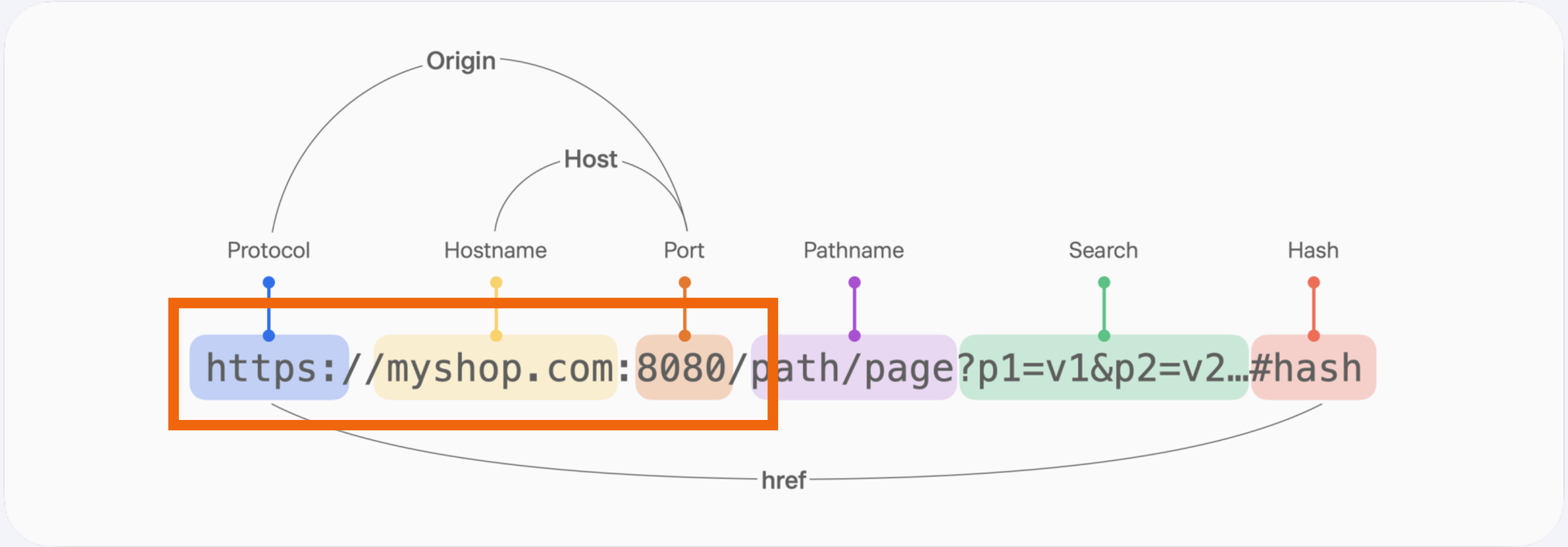
Cross-Origin Resource Sharing

- 교차 출처 리소스 공유
- 웹 페이지 상의 제한된 리소스를 최초 자원이
  서비스된 도메인 밖의 다른 도메인으로부터 요청할 수 있게 허용하는 구조
- 출처가 다른 서버 간의 리소스 공유를 허용하는 정책

무슨 소리지??

SOP 동일 출처 정책

Same-Origin Policy : 출처가 동일한 경우에만 리소스를 공유할 수 있게 제한하는 정책



출처(origin) = protocol + hostname (domain) + port

출처 비교 예시

출처(origin) = protocol + hostname (domain) + port

- 동일한 경우
  http://example.com/app1/index.html vs http://example.com/app2/index.html
  http://example.com:80/app1/index.html vs http://example.com/app2/index.html
  https://example.com:8080/app1/index.html vs https://example.com:8080/app2/index.html
- protocol 다른 경우
  http://example.com vs https://example.com
- host (domain) 다른 경우
  https://www.example.com vs https://hello.example.com
- port가 다른 경우
  https://example.com:8080 vs https://example.com

포트 생략시 기본 포트 사용
HTTP: 80 / HTTPS: 443

왜 필요한가?

기본적으로 브라우저는 토큰이나 쿠키 등과 같이 사용자의 정보와 관련된 데이터를 저장

개발자 도구를 열어 DOM, 통신하고 있는 네트워크 서버, 소스 코드 등 쉽게 확인 가능

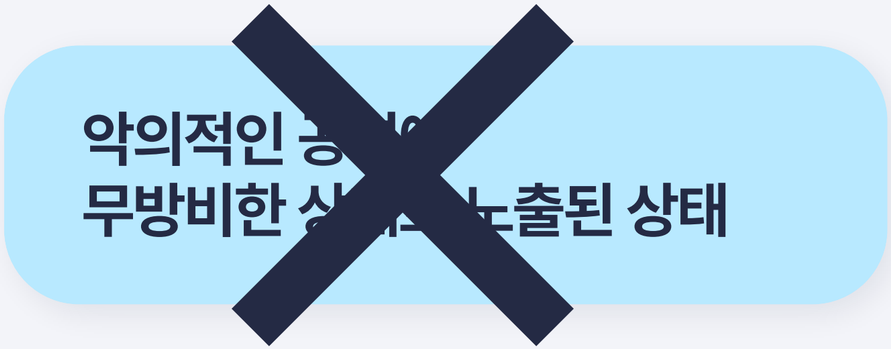
왜 필요한가?

기본적으로 브라우저는 토큰이나 쿠키 등과 같이 사용자의 정보와 관련된 데이터를 저장

개발자 도구를 열어 DOM, 통신하고 있는 네트워크 서버, 소스 코드 등 실제 확인 가능

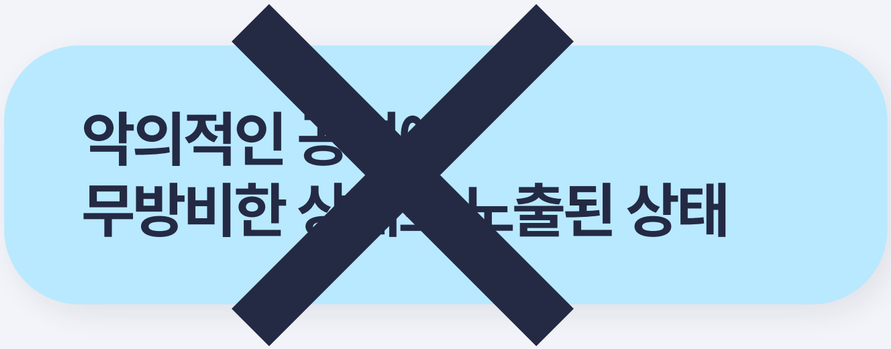
악의적인 공격에
무방비한 상태로 노출된 상태

SOP 를 사용하면



악의적인 사용자가 다른 사이트의 정보를 읽을 수 없다
브라우저는 강제적으로 SOP를 기본으로 한다

SOP 를 사용하면



악의적인 사용자가 다른 사이트의 정보를 읽을 수 없다
브라우저는 강제적으로 SOP를 기본으로 한다

BUT...
웹 환경은 개방되어 있음
→ 출처가 서로 다른 곳에서 리소스를 가져와서 사용하는 행위 빈번하게 일어남
→ 모두 막아버리면 웹 애플리케이션이 원활히 동작하기 힘들다 !

SOP 를 사용하면

악의적인 공[illegible]
무방비한 상[illegible]노출된 상태

악의적인 사용자가 다른 사이트의 정보를 읽을 수 없다
브라우저는 강제적으로 SOP를 기본으로 한다

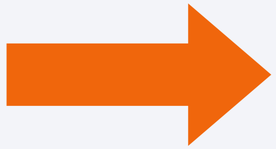
BUT...
웹 환경은 개방되어 있음
→ 출처가 서로 다른 곳에서 리소스를 가져와서 사용하는 행위 빈번하게 일어남
→ 모두 막아버리면 웹 애플리케이션이 원활히 동작하기 힘들다 !

출처가 다른 리소스를 사용할 수 있는
예외 조항이 존재

CORS

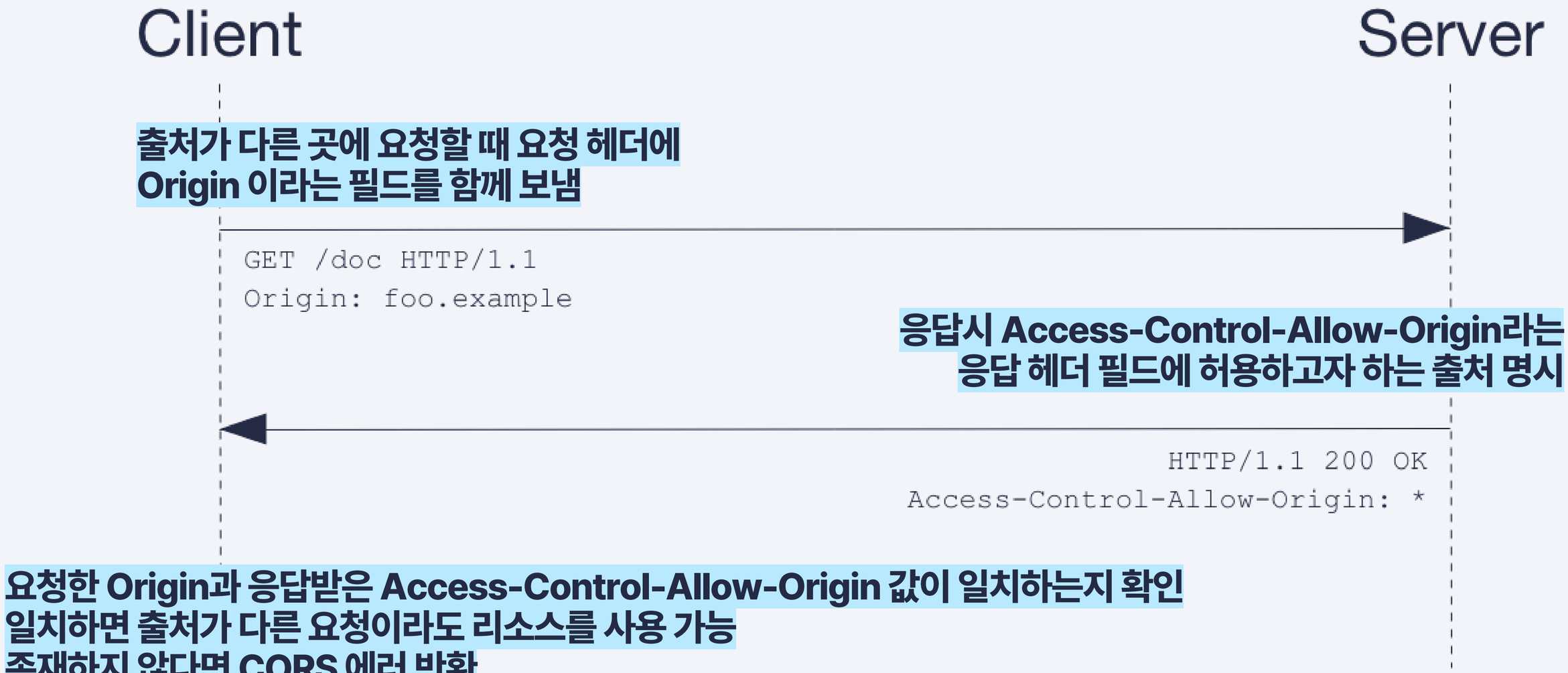
Cross-Origin Resource Sharing

- 교차 출처 리소스 공유
- 웹 페이지 상의 제한된 리소스를 최초 자원이
  서비스된 도메인 밖의 다른 도메인으로부터 요청할 수 있게 허용하는 구조
- 출처가 다른 서버 간의 리소스 공유를 허용하는 정책



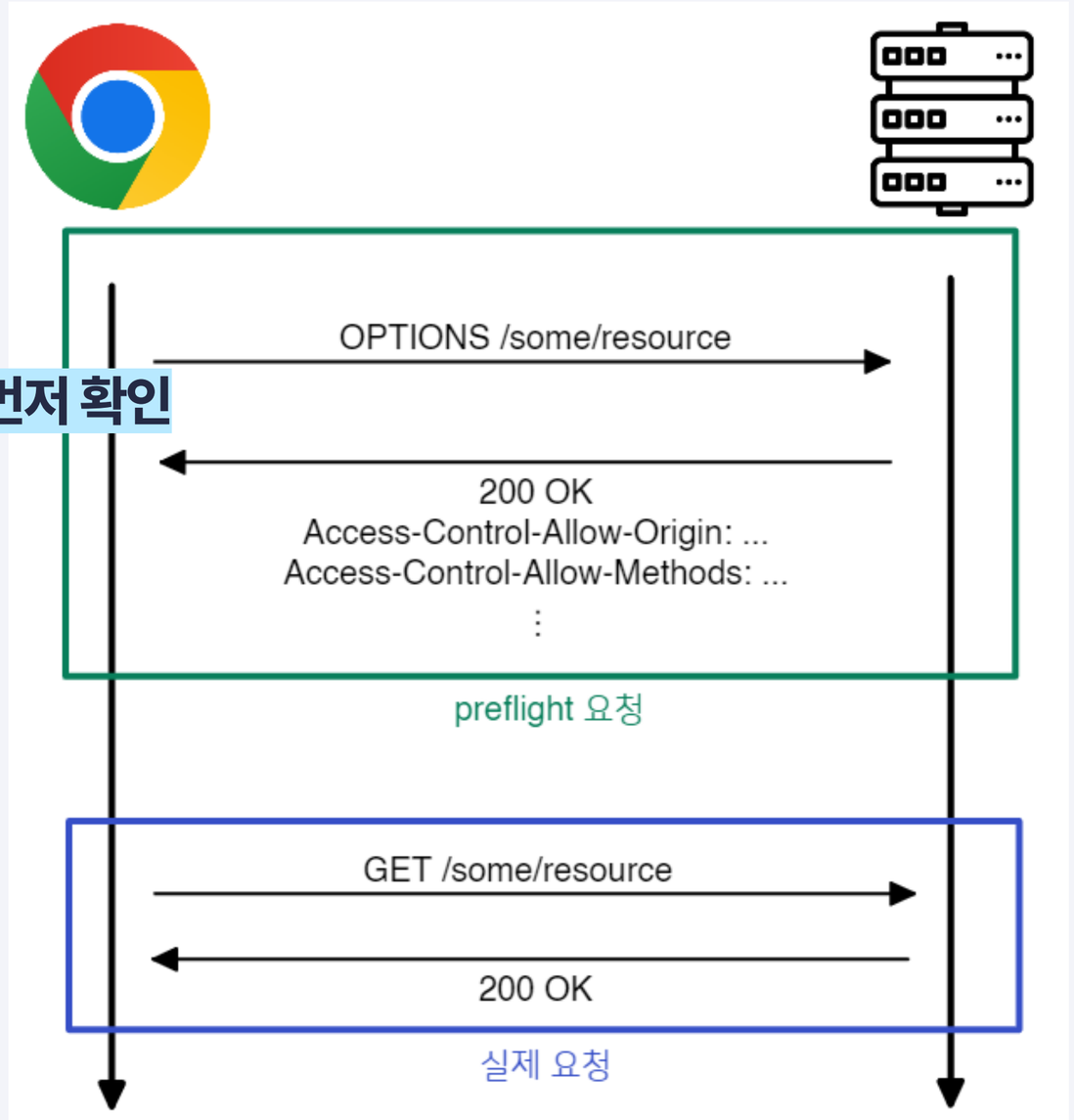
**CORS 정책을 지키면 SOP에서 기본적으로 제한하고 있는
행위들의 적용을 받지 않게 된다**

CORS 동작 방식


Client
Server
출처가 다른 곳에 요청할 때 요청 헤더에
Origin 이라는 필드를 함께 보냄
GET /doc HTTP/1.1
Origin: foo.example
응답시 Access-Control-Allow-Origin라는
응답 헤더 필드에 허용하고자 하는 출처 명시
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
요청한 Origin과 응답받은 Access-Control-Allow-Origin 값이 일치하는지 확인
일치하면 출처가 다른 요청이라도 리소스를 사용 가능
존재하지 않다면 CORS 에러 반환

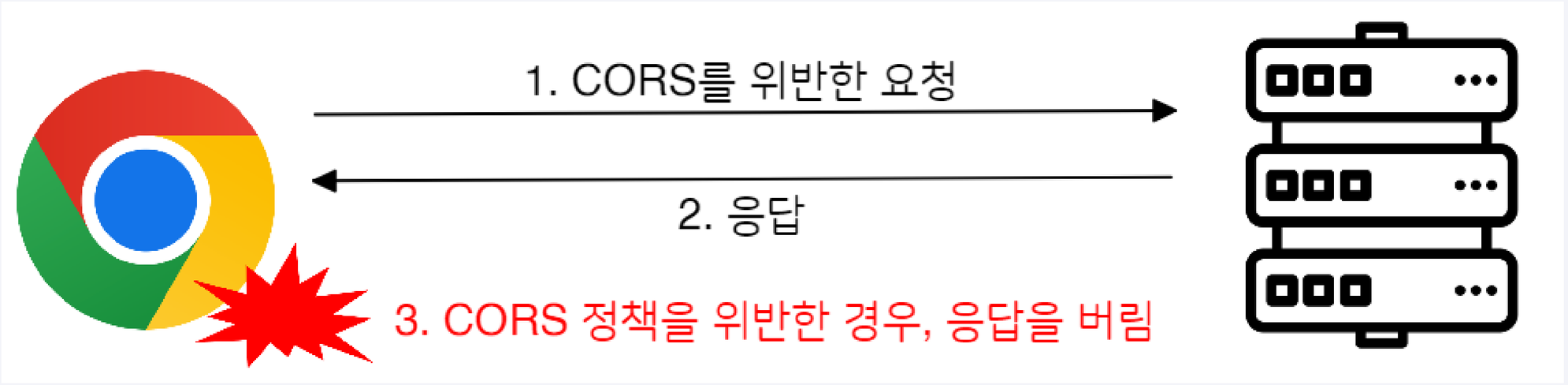
CORS 동작 방식 - 복잡한 요청

진짜 요청을 보내기 전에
preflight 요청을 보내 먼저 확인



알아야 할 점

SOP, CORS 모두 브라우저의 정책
-> 다른 출처의 리소스를 사용하는 것을 제한하는 것은 서버가 아니라 브라우저
-> 서버에 CORS 정책을 위반한 요청을 보냈을 때 이를 차단하는 것은 서버가 아니라 브라우저
-> Postman 등 브라우저가 아닌 환경에선 통신이 원활하게 이뤄질 수 있다



Spring 프로젝트에서의 설정

```
@Configuration
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
                .allowedOrigins("*")
                .allowedMethods("GET", "POST", "PUT", "DELETE")
                .allowedHeaders("Authorization", "Content-Type")
                .exposedHeaders("Custom-Header")
                .allowCredentials(true)
                .maxAge(3600);
    }
}
```

자원 공유를 허락할 Origin을 지정

허용할 HTTP method를 지정

CORS 요청에 허용되는 헤더를 지정

클라이언트측 응답에서 노출되는 헤더를 지정

클라이언트 측에 대한 응답에 credentials(예: 쿠키, 인증 헤더)를 포함할 수 있는지 여부를 지정

원하는 시간만큼 preflight 리퀘스트를 캐싱

원활한 웹 사용을 위해
CORS 정책을 적용해보아요!

감사합니다

참고 : https://docs.tosspayments.com/resources/glossary/cors#cors-에러-대응하기,
https://yoo11052.tistory.com/139, https://www.osckorea.com/post/msa-micro-service-architecture-neun-mueosimyeo-wae-pilyohalggayo,https://jaehyeon48.github.io/web/sop-and-cors/
https://velog.io/@yoonuk/Spring-Boot-CORS-%EC%84%A4%EC%A0%95%ED%95%98%EA%B8%B0