CORS(Cross-Origin Resource Sharing)

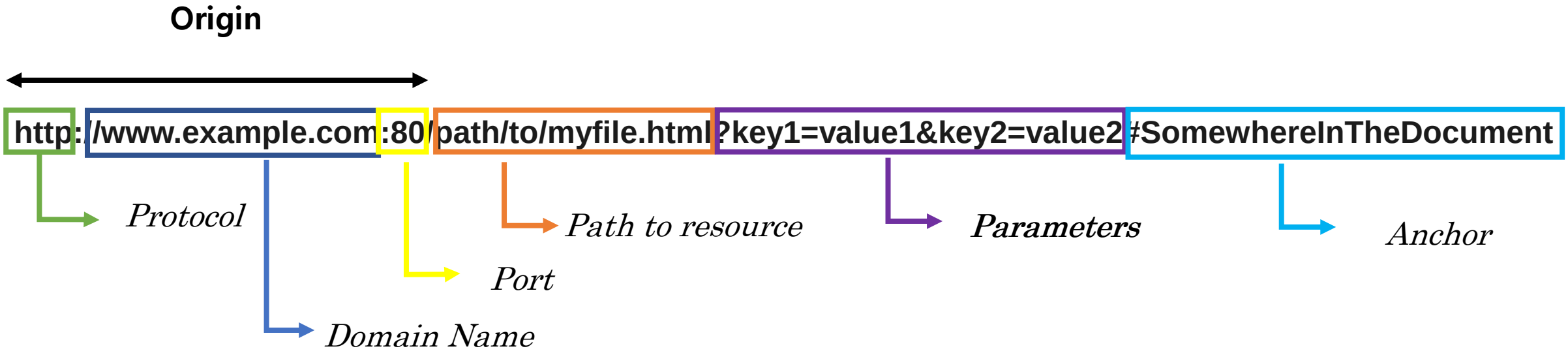
프로젝트 중 발생한 에러,
개발자라면 한 번쯤은 마주칠 수 있다.

CORS(Cross-Origin Resource Sharing) : 교차 출처 리소스 공유

➡ HTTP 헤더를 사용하여, 한 출처에서 실행 중인 웹 애플리케이션이 다른 출처의 선택한 자원에 접근할 수 있는 권한을 부여하도록 브라우저에 알려주는 체제로, 웹 애플리케이션은 리소스가 자신의 출처(도메인, 프로토콜, 포트)와 다를 때 교차 출처 HTTP 요청을 실행한다.

출처??

URL의 구조



출처(origin)는 접근할 때 사용하는 URL의 스킴(프로토콜), 호스트(도메인), 포트로 정의된다.
두 객체의 스킴, 호스트, 포트가 모두 일치하는 경우 같은 출처를 가졌다고 말한다.

다른 출처의 예제

http://example.com/app1	https://example.com/app2	다른 스킴
http://example.com	http://www.example.com	http://myapp.example.com	다른 호스트
http://example.com	http://example.com:8080	다른 포트

CORS 동작 방식

Simple Request

1. 브라우저에서 HTTP Header에 Origin 속성에 요청을 보내는 Origin을 담아 서버에게 요청을 보낸다.

Origin: https://example.com

CORS 동작 방식

1. 서버가 이 요청에 대한 응답을 할 때, 응답 Header의 Access-Control-Allow-Origin에 '리소스를 접근하는 것이 허용된 Origin'를 담아 브라우저에게 응답을 보낸다. (도메인을 리턴하거나 다 허용하면 *)

2. 응답을 받은 브라우저는 자신이 보냈던 요청의 Origin과 서버가 보내준 응답의 Access-Control-Allow-Origin 값을 비교한다.

3. 두 개가 동일하다면 유용한 응답이라 판단한다.

CORS 처리하는 방법

1. **Simple Request (단순 요청)**

: Preflight 요청을 유발하지 않는 것인데, 다음 조건을 모두 충족해야 한다.

- GET, HEAD, 또는 POST method를 사용해야 한다.
- 헤더는 Accept, Accept-Language, Content-Language, Content-Type만 허용
- Concent-Type header는 application/x-www-form-urlencoded, multipart/form-data, text/plain 값만 허용한다.

CORS 처리하는 방법

2. Preflight Request

: 실제 요청을 보내기 전에 실제로 요청하려는 경로와 같은 URL에 대해 서버에 OPTIONS 메서드로 사전 요청을 보내고 요청을 할 수 있는 권한이 있는지 확인한다.

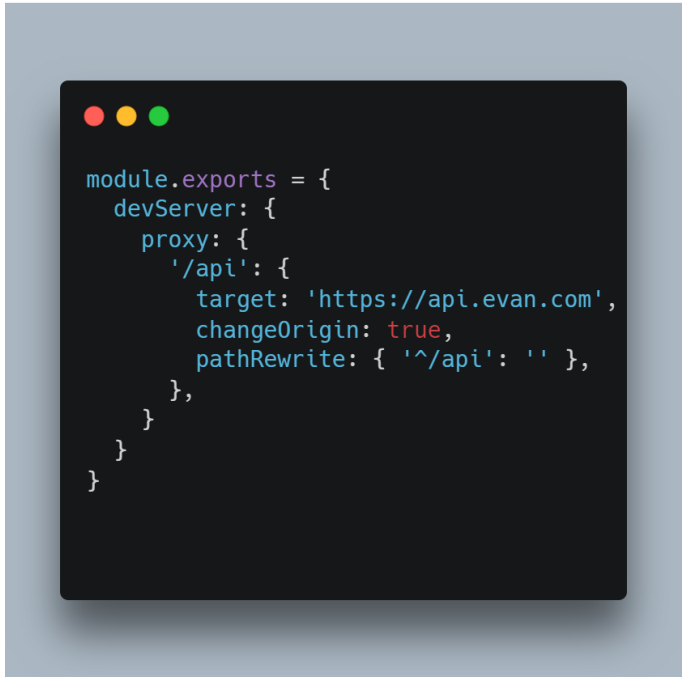
이때 **Response**로 **Access-Control-Allow-Origin**과 **Access-Control-Allow-Methods**가 넘어오는데 이는 서버에서 **어떤 origin**과 **어떤 method**를 허용하는지 브라우저에게 알려주는 역할을 한다.

브라우저가 결과를 성공적으로 확인하고 나면 **cross-origin** 요청을 보내서 그 이후 과정을 진행한다.

CORS 해결 방법

Access-Control-Allow-Origin 세팅하기
Access-Control-Allow-Origin: https://velog.io/@forwardyoung과 같이 출처를 명시하자.

Webpack Dev Server로 리버스 프록싱하기
: 프론트엔드 개발자는 대부분 웹팩과 webpack-dev-server를 사용하여 자신의 머신에 개발 환경을 구축하게 되는데,
이 라이브러리가 제공하는 프록시 기능을 사용하면
아주 편하게 CORS 정책을 우회할 수 있다.



```
module.exports = {
  devServer: {
    proxy: {
      '/api': {
        target: 'https://api.evan.com',
        changeOrigin: true,
        pathRewrite: { '^/api': '' },
      },
    }
  }
}
```