

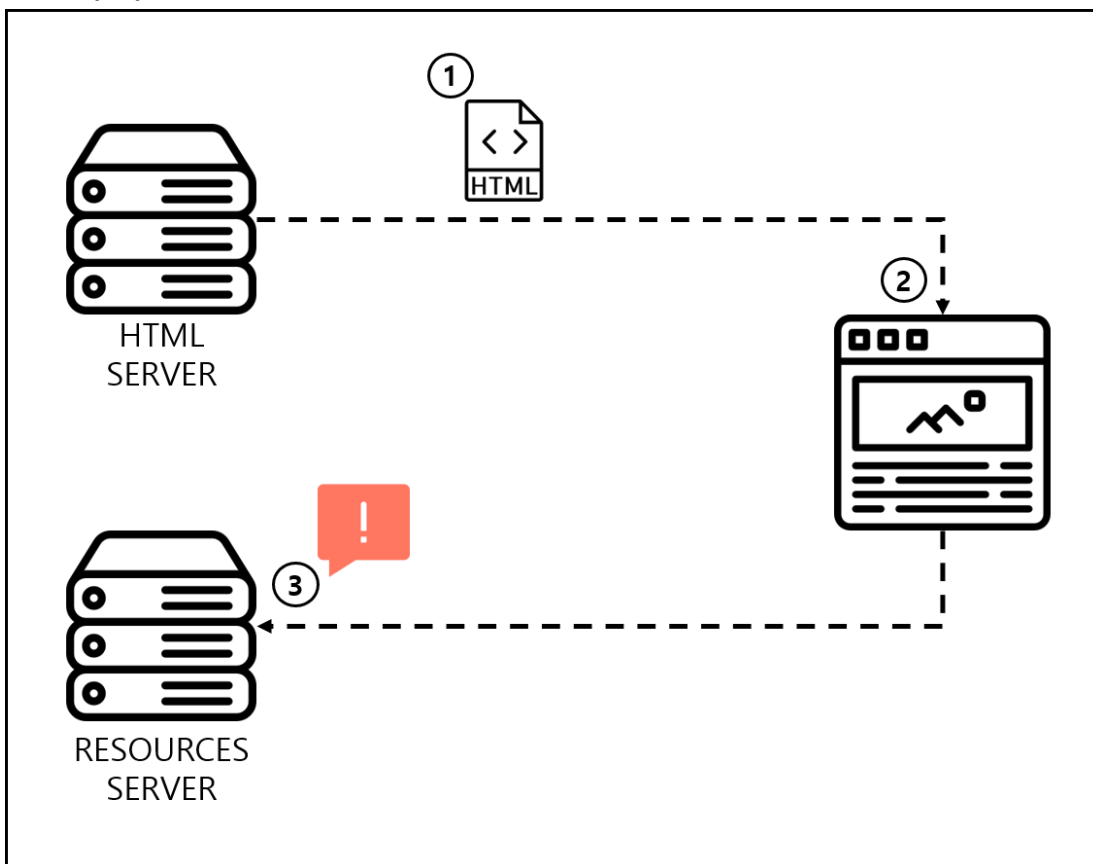
CORS

개요

- CORS

- Cross Origin Resource Sharing의 약자
- 즉, 현재 도메인과 다른 도메인으로 리소스가 요청될 경우를 말한다.
- 예를 들어 <http://A.com>에서 HTML을 끌어오고 HTML내 이미지(jpg, png 등)는 <http://B.com>에서 끌어올 시 해당 리소스(이미지)는 cross-origin HTTP 요청이 수행된다.
하지만, 브라우저는 보안상의 이유로 CORS를 제한한다.

- CORS의 예



- ① 웹 브라우저가 HTML 서버(<http://A.com>)에 HTML 문서를 요청
- ② 웹 브라우저가 HTML 문서 내에 들어갈 자원을 확인
- ③ 웹 브라우저가 RESOURCES 서버(<http://B.com>)로 필요한 자원을 cross-origin HTTP 요청
하지만, HTML 서버와 주소가 다른 RESOURCES 서버로 cross-origin HTTP 요청을 수행해 웹 브라우저에서 보안상 이유로 CORS 에러를 발생시킴

해결방법

- 응답 헤더 설정

- 리소스 및 API 서버의 응답 헤더 중 Access-Control-Allow-Origin라는 프로퍼티에 CORS를 허용해 줄 도메인을 입력하면 된다.

- 모든 도메인을 허용
Ex) Access-Control-Allow-Origin : *
- 특정 도메인만을 허용
Ex) Access-Control-Allow-Origin : <http://A.com>, <http://B.com>, ...

Node.js Express CORS 설정

1. CORS 모듈 설치
\$ npm install cors --save
2. 미들웨어에 CORS 모듈 설정

```
var express = require('express');
var cors = require('cors');
var app = express();

// CORS 설정
app.use(cors());

app.get('/products/:id', function (req, res, next) {
  res.json({msg: 'This is CORS-enabled for all origins!'})
});

app.listen(80, function () {
  console.log('CORS-enabled web server listening on port 80')
});
```

참고문헌

- 전현준 (2017). <http://guswnsxodlf.github.io/enable-CORS-on-express>
- 림준호 (2020). <https://limjunho.github.io/2020/11/12/What-is-SOP,-CORS.html>