출처 별 CORS/쿠키 설정

장고 웹 서버와 웹 프론트엔드 서버 간의 주소 관계에 따라 필요한 CORS 및 쿠키 설정

#1. 같은 호스트/포트

장고
http://mydj.com/api/

웹 프론트엔드 서버 (리액트 등)
http://mydj.com/

✅ 동일 출처라서 CORS 설정 필요없음

✅ 세션 쿠키가 기본 공유됨. 별도 설정없이 fetch 요청 만으로 **세션 쿠키가 자동으로 전달**

✅ API 서버 HOST 지정 필요없음

실서비스에서 추천 (장고/리액트가 같은 서버 인프라에서 리버스 프록시 서버(Nginx, Caddy2, HAProxy 등)를 활용)

```
const url = "/api/profile/";
await fetch(url);
```

```
FRONT_HOST_WITH_SCHEME = "http://mydj.com:3000"
CSRF_TRUSTED_ORIGINS = [FRONT_HOST_WITH_SCHEME]
CORS_ALLOW_CREDENTIALS = True
CORS_ALLOWED_ORIGINS = [FRONT_HOST_WITH_SCHEME]
```

#2. 같은 호스트, 다른 포트

장고
http://mydj.com:8000

웹 프론트엔드 서버 (리액트 등)
http://mydj.com:3000

🚫 포트가 달라 다른 출처이기에 장고 서버에서 **CORS 설정이 필요**

✅ 세션 쿠키가 기본 공유됨. fetch 요청보낼 때 **credentials: "include" 설정**하면 해당 주소의 쿠키를 자동 전송

주로 개발환경에서의 상황

```
const url = "http://mydj.com:8000/api/profile/";
await fetch(url, {
  credentials: "include",
});
```

```
FRONT_HOST_WITH_SCHEME = "http://mydj.com"
CSRF_TRUSTED_ORIGINS = [FRONT_HOST_WITH_SCHEME]
CORS_ALLOW_CREDENTIALS = True
CORS_ALLOWED_ORIGINS = [FRONT_HOST_WITH_SCHEME]
```

#3. 같은 루트 도메인

장고
http://api.mydj.com

웹 프론트엔드 서버 (리액트 등)
http://mydj.com

🚫 서브 도메인이 달라 다른 출처이기에 장고 서버에서 **CORS 설정이 필요**

🚫 세션 쿠키가 기본 공유되지 않음.
SESSION_COOKIE_DOMAIN = ".mydj.com" 설정을 적용하면, 서브 도메인 내 페이지에서는 세션 쿠키가 공유됨. fetch 요청보낼 때 **credentials: "include" 설정**하면 해당 주소의 쿠키를 자동 전송

실서비스에서 추천
(장고/리액트가 다른 서버 인프라)

```
const url = "http://api.mydj.com/api/profile/";
await fetch(url, {
  credentials: "include",
});
```

```
FRONT_HOST_WITH_SCHEME = "http://myreact.com"
CSRF_TRUSTED_ORIGINS = [FRONT_HOST_WITH_SCHEME]
CORS_ALLOW_CREDENTIALS = True
CORS_ALLOWED_ORIGINS = [FRONT_HOST_WITH_SCHEME]

# 추가로 JWT 등의 설정과 추가 개발이 필요
```

#4. 다른 도메인

장고
http://mydj.com

웹 프론트엔드 서버 (리액트 등)
http://myreact.com

🚫 도메인이 달라 다른 출처이기에 장고 서버에서 **CORS 설정이 필요**

🚫 세션 쿠키가 공유되지 않음.
다른 도메인에 대해서는 세션 쿠키 공유 불가.
세션인증 외에 다른 인증방법 (JWT 등) 검토가 필요
(도메인을 맞춰, 세션 인증을 권장)

장고/리액트가 하나의 서비스라면,
가급적 #1/#3 상황으로 운영하시길 추천

```
const url = "http://mydj.com/api/profile/";

# 웹에서는 XSS 취약점을 통해 token 탈취의 위험. 앱에서는 안전한 저장소가 있어요.
# 웹은 X. JWT 쿠키를 쓸 것이라면, 세션 쿠키를 쓰세요.
const token = localStorage.getItem("token");
await fetch(url, {
  headers: {
    "Authorization": "Bearer " + token
  }
});
```