CORS HEADERS

Dev: Letícia Lima

Para configurar os Cors Headers precisamos instalar uma biblioteca.

https://pypi.org/project/django-cors-headers/

O CORS (Cross-Origin Resource Sharing) é um mecanismo de segurança do navegador que impede que um site acesse recursos em outro site por padrão. Isso é feito para proteger os usuários de possíveis vulnerabilidades de segurança.

No entanto, às vezes é necessário permitir que um site acesse recursos em outro site. É aí que o CORS Headers entra em jogo. O CORS Headers é uma biblioteca do Django que permite que você configure seu servidor para permitir que um site acesse seus recursos.

Ao instalar o CORS Headers, você pode adicionar uma configuração simples no seu arquivo de configuração (settings.py) para permitir que um site específico acesse seus recursos. Por exemplo, se você tiver um site hospedado em "**http://meusite.com**" e quiser permitir que ele acesse seus recursos, você pode adicionar o seguinte ao seu **settings.py**:

Instalar a Biblioteca na nossa aplicação

`pip install django-cors-headers`

```
from corsheaders.defaults import default_headers
```

```
# Adicionar no settings.py
INSTALLED_APPS = [
    ...,
    "corsheaders",
    ...,
]
```

```
MIDDLEWARE = [
    ...,
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.common.CommonMiddleware",
    ...,
]
```

```
ALLOWED_HOSTS = [
    'localhost',
    '127.0.0.1',
]

CORS_ALLOW_HEADERS = list(default_headers) + [
      'X-Register',
]

# CORS Config
CORS_ORIGIN_ALLOW_ALL = True
# CORS_ORIGIN_ALLOW_ALL como True, o que permite que qualquer site acesse seus recursos.
# Defina como False e adicione o site no CORS_ORIGIN_WHITELIST onde somente o site da lista acesse os seus recursos.

CORS_ALLOW_CREDENTIALS = False

CORS_ORIGIN_WHITELIST = ['http://meusite.com',] # Lista.
```

SSL and Cookies Vamos deixar configurado tambem. No final do video vamos fazer deploy.
doc: https://docs.djangoproject.com/en/4.1/ref/settings/

```
if not DEBUG:
  SECURE_SSL_REDIRECT = True
  ADMINS = [(os.getenv('SUPER_USER'), os.getenv('EMAIL'))]
  SESSION_COOKIE_SECURE = True
  CSRF_COOKIE_SECURE = True
```

`if not DEBUG:` verifica se a aplicação está sendo executada em modo de depuração (`DEBUG=True`). Se `DEBUG` for `False`, isso significa que a aplicação está sendo executada em um ambiente de produção, portanto, as configurações de segurança devem ser aplicadas.

`SECURE_SSL_REDIRECT` direciona todas as solicitações HTTP para HTTPS.

`ADMINS` é uma lista de tuplas que contêm informações sobre os administradores do site. Se ocorrer um erro no site, um email será enviado para os endereços listados em `ADMINS`.

`SESSION_COOKIE_SECURE` garante que os cookies de sessão sejam definidos apenas em conexões HTTPS.

`CSRF_COOKIE_SECURE` garante que os cookies CSRF sejam definidos apenas em conexões HTTPS.

Essas configurações ajudam a proteger a aplicação contra ataques de interceptação e garantem que as informações confidenciais do usuário sejam mantidas seguras.

Com essas configurações, você permitirá que o site "**http://meusite.com**" acesse seus recursos. É importante lembrar que, para que isso funcione, o site que está acessando seus recursos também deve ter a configuração CORS correta.