STRAND
COSMETICS EUROPE

Erreurs rencontrés et resolues

django - vuejs - python

description du pb :

> lors de l'appel de api , erreur du Navigateur :
> origin 'http://127.0.0.1:8080' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.

How to Fix CORS Issue with Django Rest Framework (Content ...

- install papckage django-cors-headers

`sh $ pip install django-cors-headers `

- ajouter appli dans settings.apps

`py INSTALLED_APPS = ( ... 'corsheaders', ... ) `

- ensuite dans les middleware ajouter ces lignes

`py MIDDLEWARE_CLASSES = (... 'corsheaders.middleware.CorsMiddleware',
'django.middleware.common.CommonMiddleware', ...) `

> You can do by using a custom middleware, even though knowing that the best option is using the tested approach of the package django-cors-headers.
> With that said, here is the solution:

create the following structure and files:

`py -- myapp/middleware/__init__.py `

` py
from corsMiddleware import corsMiddleware

-- myapp/middleware/corsMiddleware.py

class corsMiddleware(object):
def process_response(self, req, resp):
resp["Access-Control-Allow-Origin"] = "*"
return resp
add to settings.py the marked line:

MIDDLEWARE_CLASSES = (
"django.contrib.sessions.middleware.SessionMiddleware",
"django.middleware.common.CommonMiddleware",
"django.middleware.csrf.CsrfViewMiddleware",

```
# Now we add here our custom middleware
 'app_name.middleware.corsMiddleware' <---- this line
```

)

``

In case anyone is getting back to this question and deciding to write their own middleware,

this is a code sample for Django's new style middleware -

`` py
class CORSMiddleware(object):
def **init**(self, get_response):
self.get_response = get_response

```
def __call__(self, request):
    response = self.get_response(request)
    response["Access-Control-Allow-Origin"] = "*"
    return response
```

```

> For Django versions > 1.10, according to the documentation, a custom MIDDLEWARE can be written as a function,
> let's say in the file: yourproject/middleware.py (as a sibling of settings.py):
```

` py

def open_access_middleware(get_response):
def middleware(request):
response = get_response(request)
response["Access-Control-Allow-Origin"] = "" *response["Access-Control-Allow-Headers"] = ""*
return response
return middleware
and finally, add the python path of this function (w.r.t. the root of your project) to the
MIDDLEWARE list in your project's settings.py:

MIDDLEWARE = [
.
.
'django.middleware.clickjacking.XFrameOptionsMiddleware',
'yourproject.middleware.open_access_middleware'
]
`