

Rapport sur le problème CORS entre Django et Angular

1 Problème rencontré

En tentant d'accéder à mon backend Django depuis mon frontend Angular, hébergé sur un domaine différent, j'ai rencontré un problème : les navigateurs bloquent ces requêtes pour des raisons de sécurité. Après quelques recherches, j'ai découvert que ce blocage était dû à la politique CORS.

2 Qu'est-ce que CORS ?

CORS (Cross-Origin Resource Sharing) est un mécanisme de sécurité qui empêche un frontend (comme Angular) d'accéder à un backend (comme Django) hébergé sur un domaine différent. Par défaut, les navigateurs bloquent ces requêtes pour des raisons de sécurité. Pour résoudre ce problème, il faut configurer le backend afin d'autoriser l'accès depuis l'origine du frontend.

3 Solution apportée

Pour résoudre ce problème, nous avons configuré Django afin d'autoriser les requêtes cross-origin.

Étapes:

1. Installer la bibliothèque nécessaire dans Django :

```
pip install django-cors-headers
```

2. Ajouter 'corsheaders' dans les INSTALLED_APPS de settings.py.

3. Ajouter le middleware CorsMiddleware dans MIDDLEWARE :

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

4. Configurer les domaines autorisés dans settings.py :

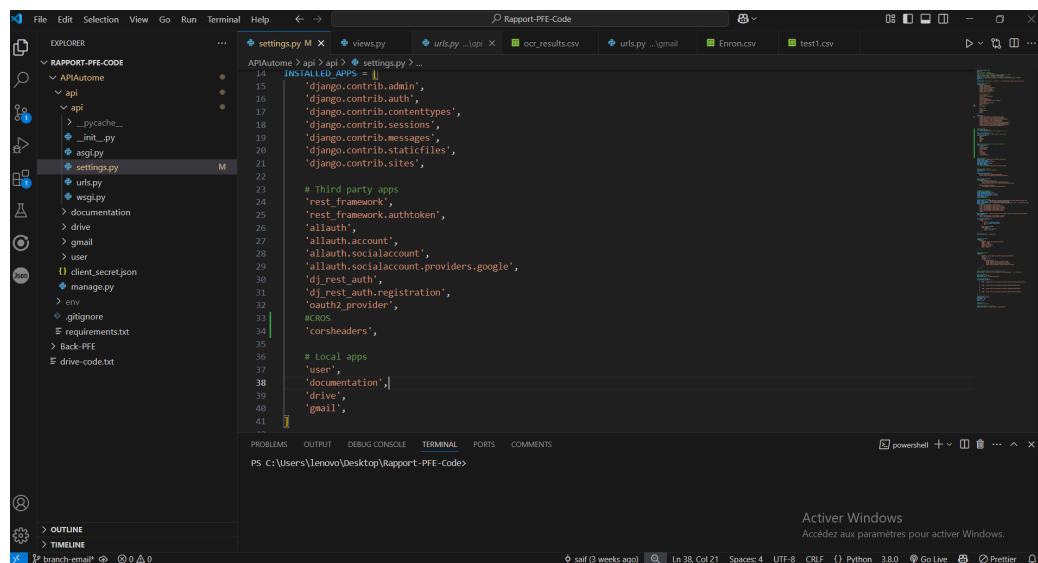
```
CORS_ALLOWED_ORIGINS = [  
    "http://localhost:4200",  
]
```

5. Redémarrer le serveur Django pour appliquer les modifications :

```
python manage.py runserver
```

4 Code ajouté dans Django

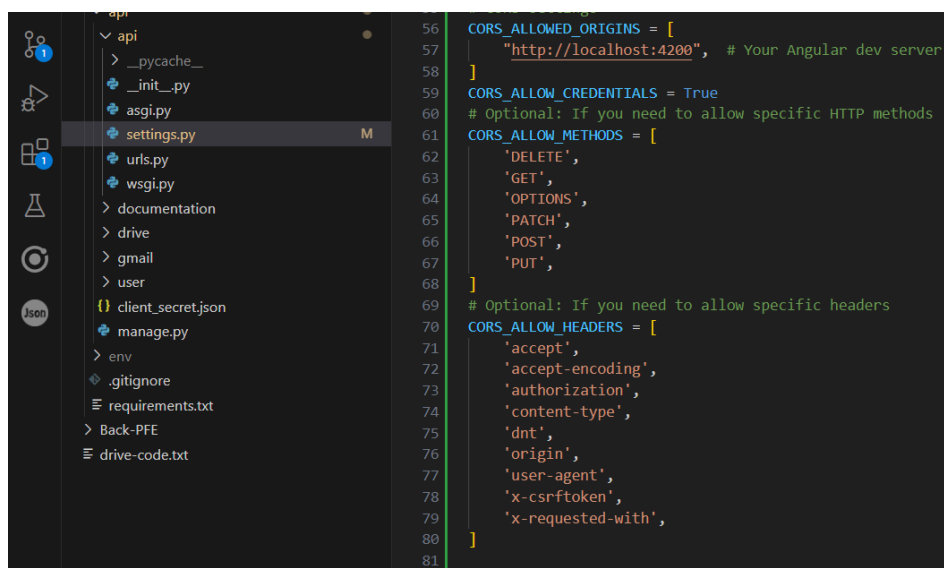
4.1 'corsheaders' dans les INSTALLED APPS



4.2 CorsMiddleware dans MIDDLEWARE

```
43 MIDDLEWARE = [  
44     'corsheaders.middleware.CorsMiddleware', #CROS  
45     'django.middleware.security.SecurityMiddleware',  
46     'django.contrib.sessions.middleware.SessionMiddleware',  
47     'django.middleware.common.CommonMiddleware',  
48     'django.middleware.csrf.CsrfViewMiddleware',  
49     'django.contrib.auth.middleware.AuthenticationMiddleware',  
50     'django.contrib.messages.middleware.MessageMiddleware',  
51     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
52     'allauth.account.middleware.AccountMiddleware',  
53 ]
```

4.3 Configurer les domaines autorisés dans settings.py

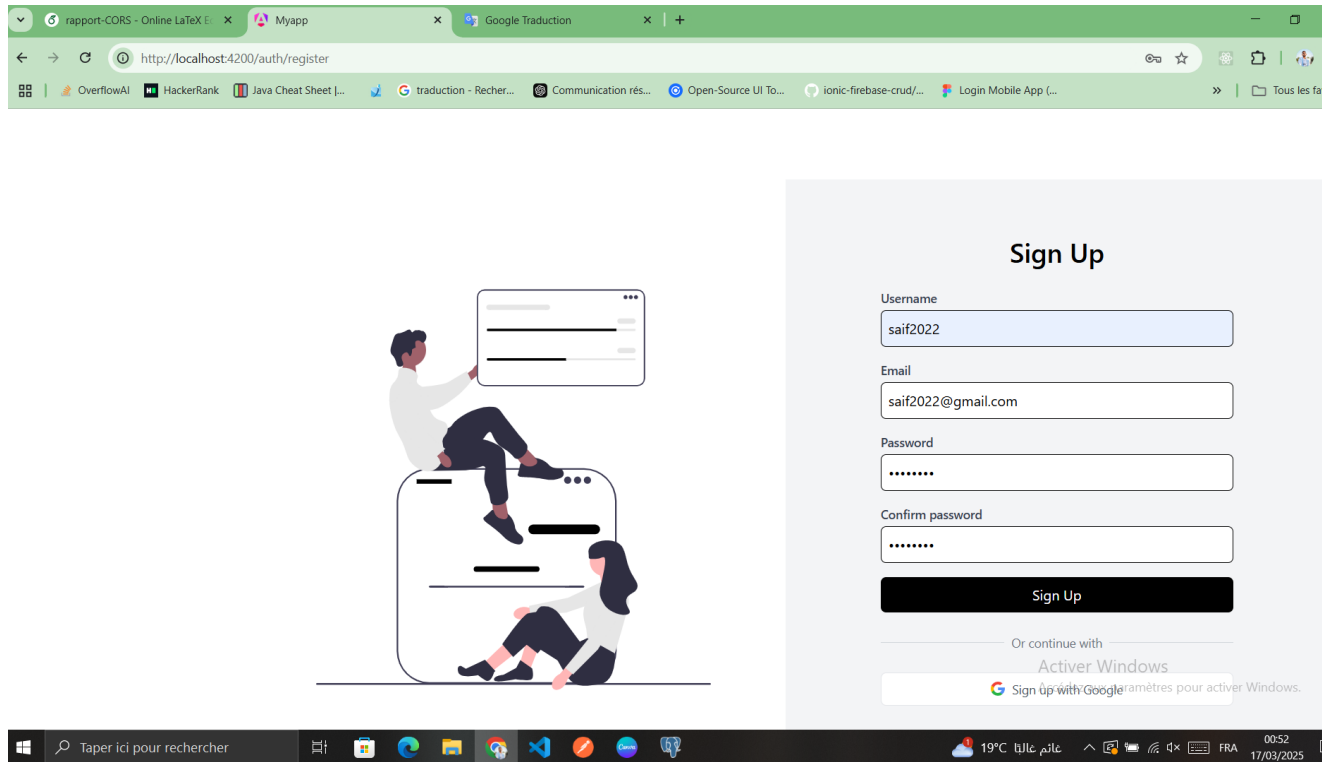


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'api' containing files like 'urls.py', 'wsgi.py', and 'settings.py'. The 'settings.py' file is selected. The code editor shows the following configuration:

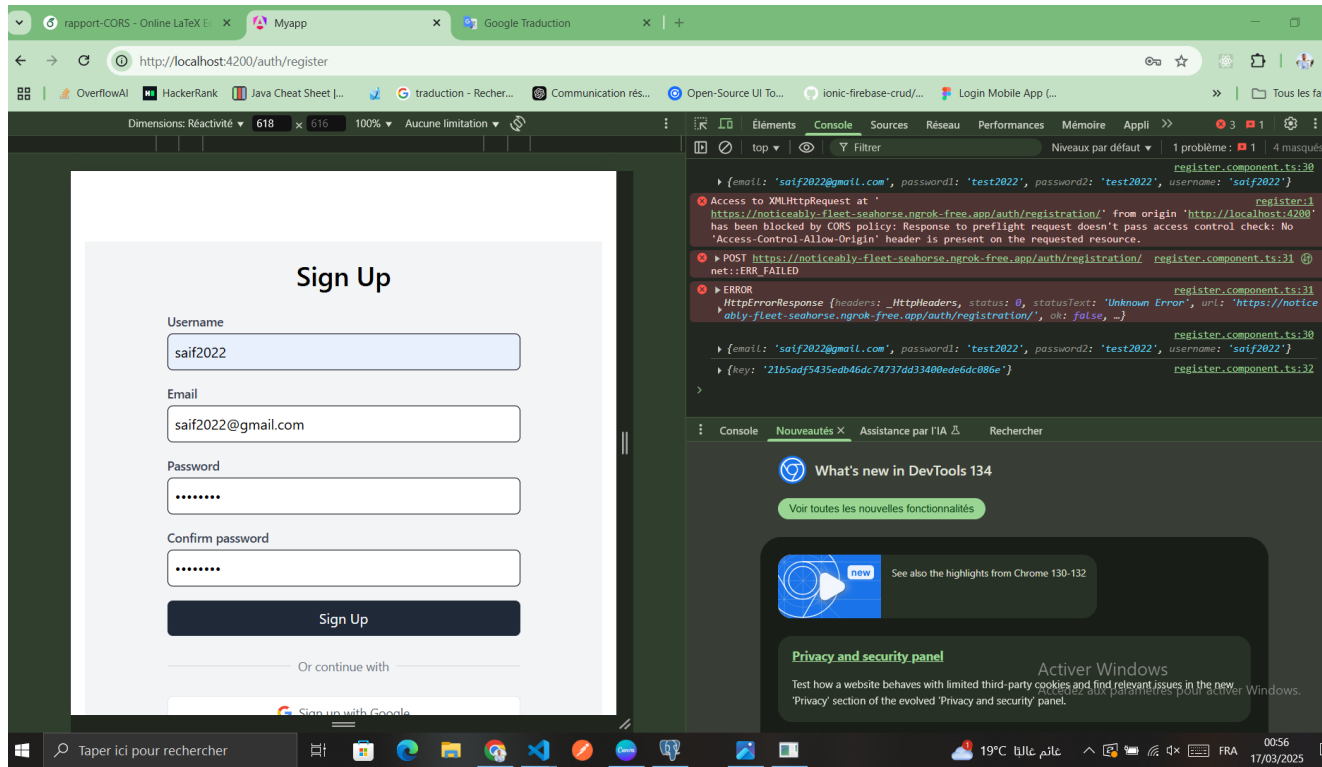
```
56 CORS_ALLOWED_ORIGINS = [  
57     "http://localhost:4200", # Your Angular dev server  
58 ]  
59 CORS_ALLOW_CREDENTIALS = True  
60 # Optional: If you need to allow specific HTTP methods  
61 CORS_ALLOW_METHODS = [  
62     'DELETE',  
63     'GET',  
64     'OPTIONS',  
65     'PATCH',  
66     'POST',  
67     'PUT',  
68 ]  
69 # Optional: If you need to allow specific headers  
70 CORS_ALLOW_HEADERS = [  
71     'accept',  
72     'accept-encoding',  
73     'authorization',  
74     'content-type',  
75     'dnt',  
76     'origin',  
77     'user-agent',  
78     'x-csrftoken',  
79     'x-requested-with',  
80 ]  
81
```

5 Résultat après correction

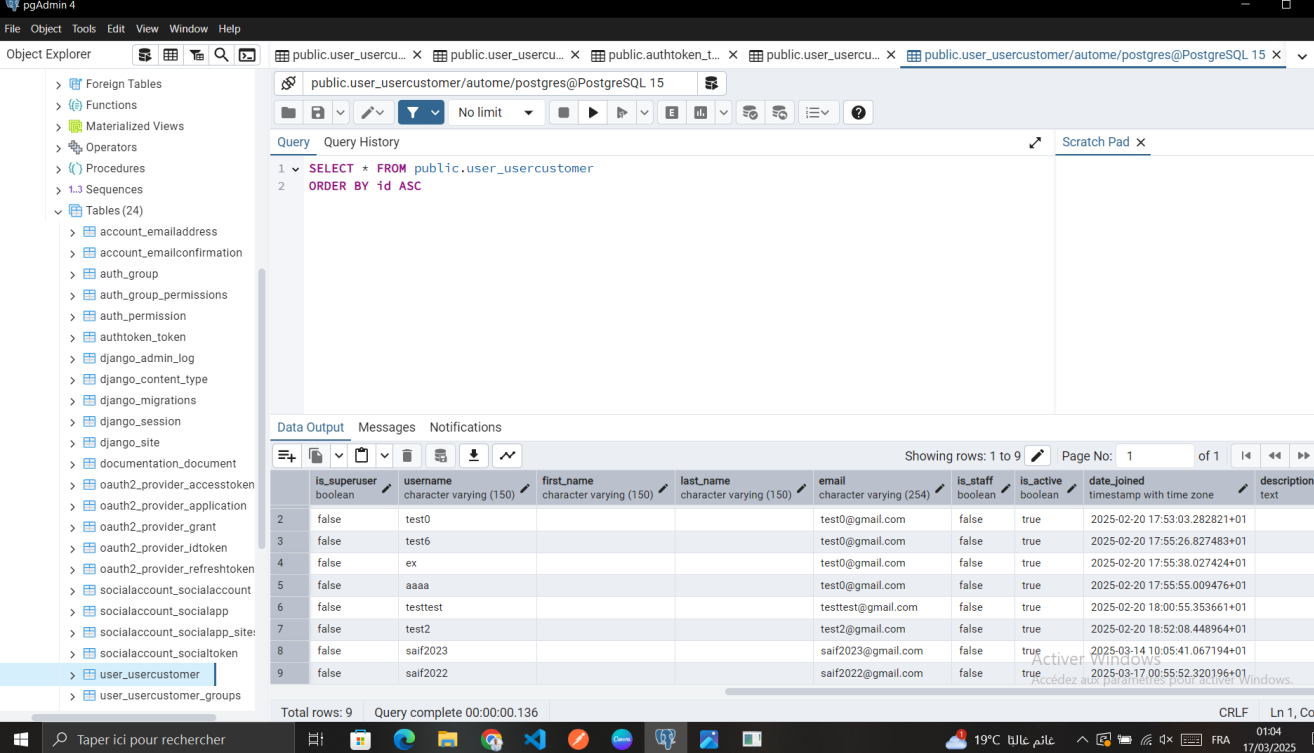
5.1 Créer un compte:



5.2 Résultat de création du compte(Console)



5.3 Résultat de création du compte(Base de données)



The screenshot displays the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, with 'Tables (24)' expanded. The main pane shows a SQL query: `SELECT * FROM public.user_usercustomer ORDER BY id ASC`. The query result is displayed in a table with the following columns: `is_superuser` (boolean), `username` (character varying (150)), `first_name` (character varying (150)), `last_name` (character varying (150)), `email` (character varying (254)), `is_staff` (boolean), `is_active` (boolean), `date_joined` (timestamp with time zone), and `description` (text). The table contains 9 rows of data.

	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined	description
2	false	test0			test0@gmail.com	false	true	2025-02-20 17:53:03.282821+01	
3	false	test6			test0@gmail.com	false	true	2025-02-20 17:55:26.827483+01	
4	false	ex			test0@gmail.com	false	true	2025-02-20 17:55:38.027424+01	
5	false	aaaa			test0@gmail.com	false	true	2025-02-20 17:55:55.009476+01	
6	false	testtest			testtest@gmail.com	false	true	2025-02-20 18:00:55.353661+01	
7	false	test2			test2@gmail.com	false	true	2025-02-20 18:52:08.448964+01	
8	false	saif2023			saif2023@gmail.com	false	true	2025-03-14 10:05:41.067194+01	
9	false	saif2022			saif2022@gmail.com	false	true	2025-03-17 00:55:52.320196+01	

Total rows: 9 Query complete 00:00:00.136