ChatGPT

Django Backend Setup for Flutter Authentication

This guide documents the setup process for the Django backend used for token-based authentication with a Flutter frontend.

📦 Dependencies Installed

These packages were installed in your Django environment:

```
pip install django
django-admin startproject backend
pip install djangorestframework
pip install djangorestframework-simplejwt
pip install django-cors-headers
```

🛠 Project Structure (Sample)

```
backend/
├── backend/                # Django project root
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── manage.py
```

⚙ settings.py Configuration

Add apps to `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'corsheaders',
]
```

Add middleware:

```
MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Add this to allow requests from Flutter:

```
CORS_ALLOW_ALL_ORIGINS = True  # For development only
```

REST Framework & JWT Configuration:

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    )
}
```

Add allowed IP (for example, Flutter runs on add your local ip):

```
ALLOWED_HOSTS = [*]  # Add your IP here
```

Creating Superuser

```
python manage.py createsuperuser
```

Provide `username`, `email`, and `password` when prompted.

Example:

- Username: `admin`
- Email: (you can leave blank)
- Password: `admin123`

Use these credentials in your Flutter app or API testing.

URLs Configuration

In `urls.py`:

```
from django.contrib import admin
from django.urls import path
from rest_framework_simplejwt.views import (
    TokenObtainPairView,
    TokenRefreshView,
)

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('api/token/refresh/', TokenRefreshView.as_view(),
name='token_refresh'),
]
```

🗄 Database Setup and Server Run

Run the following commands to apply migrations, create the admin user, and start the development server:

```
python manage.py migrate
python manage.py createsuperuser
python manage.py runserver 0.0.0.0:8000
```

Use your machine's IP (e.g. `:8000`) to access from another device/browser.

Testing the Token API with `curl`

```
curl -X POST http://127.0.0.1:8000/api/token/ \
-H "Content-Type: application/json" \
-d '{"username": "admin", "password": "admin123"}'
```

Response:

```
{
  "access": "<access_token>",
  "refresh": "<refresh_token>"
}
```

Next Steps

- Connect with Flutter using `http.post`.
- Display the token on-screen or use it to access protected routes.
- Add `.env` file and keep IPs and secrets secure.
- Use `pip freeze > requirements.txt` and `.gitignore` before uploading to GitHub.