

Guía Profesional: Cómo Crear un eCommerce con Django REST + React

Módulo 4: Registro de Usuarios y Autenticación JWT

4.1 Instalar paquetes necesarios

Necesitamos un paquete adicional para autenticación JWT:

```
pip install djangorestframework-simplejwt
```

4.2 Configurar JWT en settings.py

En backend/settings.py, agregar:

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

4.3 Crear URLs para login y refresh

En tienda/urls.py:

```
from rest_framework_simplejwt.views import (  
    TokenObtainPairView,
```

```

    TokenRefreshView,
)

urlpatterns += [
    path('token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
]

```

Esto permite hacer login con usuario y contraseña para obtener un token JWT, y luego refrescarlo.

4.4 Crear vista de registro de usuario

En tienda/views.py:

```

from django.contrib.auth.models import User

from rest_framework import status

from rest_framework.decorators import api_view

from rest_framework.response import Response

from rest_framework.serializers import ModelSerializer


class RegisterSerializer(ModelSerializer):

    class Meta:

        model = User

        fields = ['username', 'password']

        extra_kwargs = {'password': {'write_only': True}}

```

```

def create(self, validated_data):

    user = User.objects.create_user(**validated_data)

    return user


@api_view(['POST'])
def register(request):

    serializer = RegisterSerializer(data=request.data)

    if serializer.is_valid():

        serializer.save()

        return Response({'message': 'Usuario creado correctamente'},
                        status=status.HTTP_201_CREATED)

    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

```

4.5 Agregar la ruta para registro

En tienda/urls.py:

```

from .views import register

urlpatterns += [

    path('register/', register, name='register'),

]

```

4.6 Probar autenticación

1. POST a /api/register/ para crear un nuevo usuario.

2. POST a `/api/token/` con los campos "username" y "password" para obtener un access y refresh token.
3. Usar el token en headers: `Authorization: Bearer <token>` para acceder a endpoints protegidos.

¡Ya tenés login, registro y autenticación JWT funcionando!