

Crear una API REST por medio de DJANGO REST FRAMEWORK

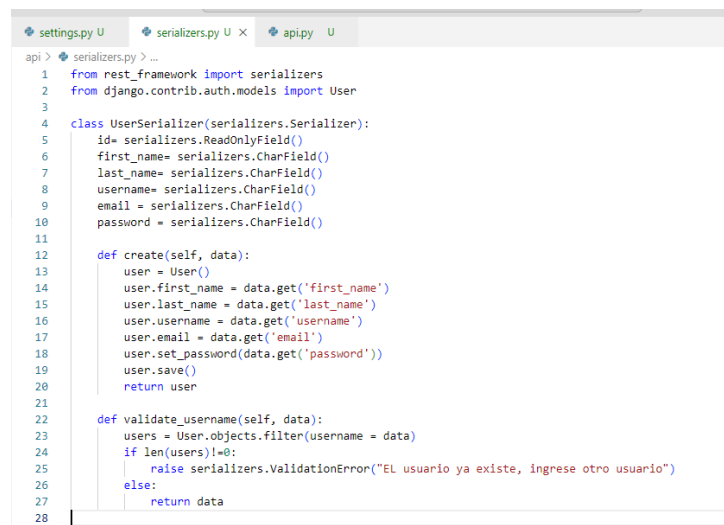
Los pasos necesarios son los siguientes:

1. Instalar django-rest: `pip install djangorestframework`
2. En el archivo `settings.py` registramos `rest_framework`.
3. Creamos una aplicación: `py manage.py startapp api`
4. Registramos la aplicación en `settings.py`

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

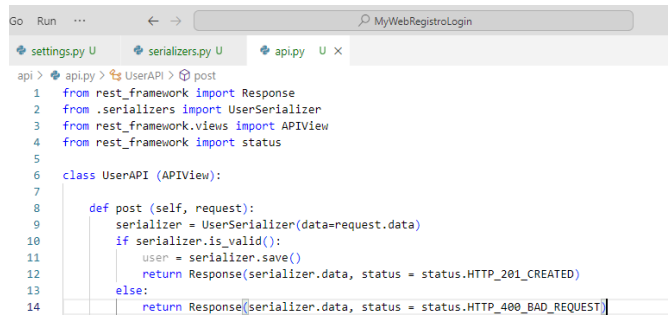
Nota: si hubiera problema de versiones, entre `django` y `djangorestframework`, actualizar ambas aplicaciones: `pip install --upgrade django djangorestframework`

5. Creamos en la aplicación `api` un archivo llamado: `serializers.py`. Un archivo de tipo `serializers.py` permite transformar un diccionario Python en un archivo json. Ingresamos el siguiente código:



```
api > serializers.py > ...
1 from rest_framework import serializers
2 from django.contrib.auth.models import User
3
4 class UserSerializer(serializers.Serializer):
5     id = serializers.ReadOnlyField()
6     first_name = serializers.CharField()
7     last_name = serializers.CharField()
8     username = serializers.CharField()
9     email = serializers.CharField()
10    password = serializers.CharField()
11
12    def create(self, data):
13        user = User()
14        user.first_name = data.get('first_name')
15        user.last_name = data.get('last_name')
16        user.username = data.get('username')
17        user.email = data.get('email')
18        user.set_password(data.get('password'))
19        user.save()
20        return user
21
22    def validate_username(self, data):
23        users = User.objects.filter(username = data)
24        if len(users) != 0:
25            raise serializers.ValidationError("El usuario ya existe, ingrese otro usuario")
26        else:
27            return data
28
```

6. Creamos un archivo llamado `api.py` en la app `api`. Este archivo permitirá administrar las peticiones para un api rest.

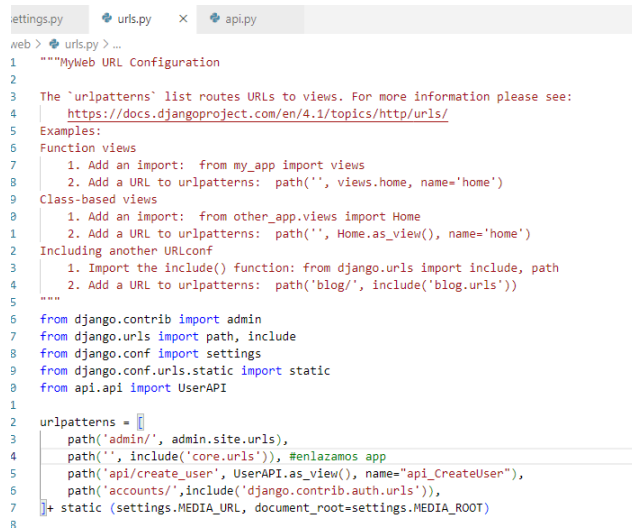


```

1 from rest_framework import Response
2 from .serializers import UserSerializer
3 from rest_framework.views import APIView
4 from rest_framework import status
5
6 class UserAPI (APIView):
7
8     def post (self, request):
9         serializer = UserSerializer(data=request.data)
10        if serializer.is_valid():
11            user = serializer.save()
12            return Response(serializer.data, status = status.HTTP_201_CREATED)
13        else:
14            return Response(serializer.data, status = status.HTTP_400_BAD_REQUEST)

```

7. Agregamos el path de nuestra aplicación al archivo urls.py de nuestro proyecto:



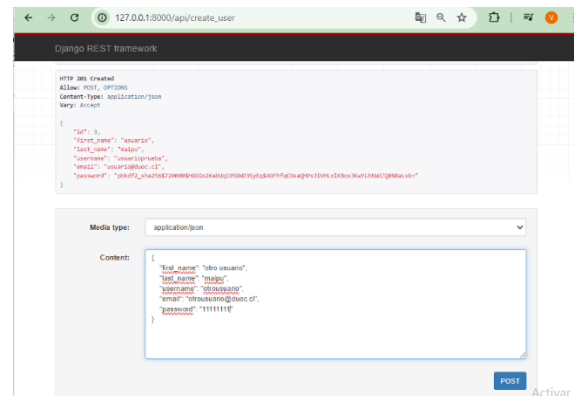
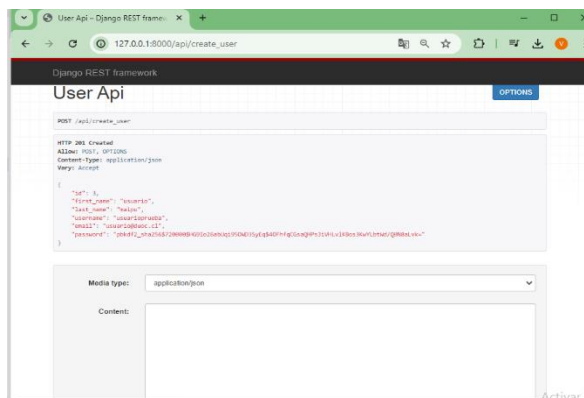
```

1 """MyWeb URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/4.1/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18 from django.conf import settings
19 from django.conf.urls.static import static
20 from api.api import UserAPI
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('', include('core.urls')), #enlazamos app
25     path('api/create_user', UserAPI.as_view(), name='api_CreateUser'),
26     path('accounts/', include('django.contrib.auth.urls')),
27 ] + static (settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
28

```

8. Levantamos el servidor e ingresamos al siguiente path:

http://127.0.0.1:8000/api/create_user



Los usuarios creados por medio de rest_framework serán visualizados en el Admin de django.