

Serializers

Nuestros intérpretes

Serializers

- Los serializers se encargan de transformar los datos de una petición HTTP en un objeto y viceversa.
- REST Framework se encarga de transformar los datos en función de su formato (JSON, XML, etc.) en un diccionario que pasa al serializer.
- Se definen como los modelos de Django (con Fields)

Serializers

Hay que definir dos métodos para construir el objeto en base al diccionario con los datos validados en función de lo que vayamos a hacer:

```
create(self, validated_data)
```

```
update(self, instance, validated_data)
```

- `validated_data` es el diccionario con los datos validados
- `instance` es el objeto que queremos actualizar con los datos que vienen en el diccionario
- **Importante:** guardar el objeto si usamos un Django Model

Serializers

```
from rest_framework import serializers
```

```
class ContactSerializer(serializers.Serializer):
```

```
    email = serializers.EmailField()
```

```
    content = serializers.CharField()
```

```
def create(self, validated_data):
```

```
    return Contact(**validated_data)
```

```
def update(self, instance, validated_data):
```

```
    instance.email = validated_data.get('email', instance.email)
```

```
    instance.content = validated_data.get('content', instance.content)
```

```
    return instance
```

