


Open · Opened 4 days ago by  **Rubén Montero**

## ¿safe=False?

### Resumen

- Crearemos un *endpoint* que serializa un JSON lista ( `[ ]` )
- Explicaremos que, al no ser un diccionario, debemos usar `safe=False` en `JsonResponse`

### Descripción

Los documentos JSON generalmente contienen *objetos* o *diccionarios*, que se indican con *llaves* ( `{ }` ):

```
{
  "ejemplo": "Ser buena persona",
  "valor": 10
}
```

Y un *tipo de dato* que se admite son listas de elementos encerrados en *corchetes* ( `[ ]` ):

```
{
  "ejemplo": "Ser buena persona",
  "valor": 10,
  "recordatorios": ["¡Trabaja!", "¡Defiéndete!", "¡Ayuda al prójimo!"]
}
```

```
{
  "ejemplo": "Ser buena persona",
  "valor": 10,
  "recordatorios": [
    {
      "mensaje": "¡Trabaja!",
      "importante": true
    },
    {
      "mensaje": "¡Defiéndete!",
      "importante": true
    },
    {
      "mensaje": "¡Ayuda al prójimo!",
      "importante": true
    }
  ]
}
```

### Una lista como elemento raíz

**También** es un JSON válido:

Por eso, en Restlet, también considerábamos `JSONArray` vs. `JSONObject` .

El siguiente es un ejemplo de algo que *mapearíamos* usando `JSONArray` en Restlet:

```
[
  {
    "mensaje": "¡Trabaja!",
    "importante": true
  },
  {
    "mensaje": "¡Defiéndete!",
    "importante": true
  },
  {
    "mensaje": "¡Ayuda al prójimo!",

```

```
    "importante": true
  }
]
```

## ¿Podemos devolver un JSON lista con Django?

¡Hagámoslo!

**Crea** una nueva función en `endpoints.py`:

```
def my_list_endpoint(request):
    animals = [
        "Focha común (Fulica atra)",
        "Lagartija carpetana (Iberolacerta cyreni)",
        "Avispón europeo (Vespa crabro)"
    ]
    return JsonResponse(animals, safe=False)
```

Como ves, devuelve una lista con tres [animales autóctonos de España](#).

Nos vemos forzados a pasar un *parámetro* muy curioso a `JsonResponse`

**`safe=False`**

Es necesario cuando enviamos como respuesta cualquier cosa que **no** sea un diccionario Python.

Por ejemplo, una **lista**.

Si te preguntas *por qué*, la [documentación](#) lo aclara:

### Warning

Before the 5th edition of ECMAScript it was possible to poison the JavaScript Array constructor. For this reason, Django does not allow passing non-dict objects to the JsonResponse constructor by default. However, most modern browsers implement ECMAScript 5 which removes this attack vector. Therefore it is possible to disable this security precaution.

**Y ahora...**

**`urls.py`, ¿no?**

¡Sí!

Mapeemos nuestro nuevo *endpoint* a la URL: `/api/v1/animals`

**¡Tarea completada!**

**Por último**

Verifica que tu código pasa el *test* asociado a la tarea.

Haz `commit` y `push` para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 3](#) 4 days ago