


Open Opened 2 days ago by  [Rubén Montero](#)

Otra pequeña mejora: Tamaño del resultado

Resumen

- Añadiremos soporte para el *query param* `?size=N` en el *endpoint* `/entries`
- Si viene, devolveremos sólo las primeras `N` *Entradas*

Descripción

Si **todo el mundo** va a enviar *Entradas* a nuestro **muro**... ¡Va a acabar bastante **lleno**!

Es previsible que la tabla tenga un tamaño enorme, y en ese caso `Entry.objects.order_by("-publication_date")` devolverá muchos resultados y las peticiones HTTP se volverán voluminosas. Todo funcionará más lento hasta que llegue un punto en que nadie quiera usar nuestro API REST.

¿Y qué hacemos?

Vamos a empezar permitiendo el *query param* `?size=N` en el *endpoint* `/entries`.

- Si viene, sólo devolvemos las `N` primeras *Entradas* en la respuesta JSON

¿Cómo?

Lo único que debemos recordar es cómo acceder a *parte* de un *array* en Python.

Si tenemos:

```
guitarist = ["Paco", "de", "Lucía"]
```

...entonces:

```
# Imprime ["Paco", "de"]
print(guitarist[:2])
```

¿Y se puede usar el operador `[:N]` en un `QuerySet[]` de Django?

¡Sí!

Modificamos `all_entries`

Vayamos a dicha función en `endpoints.py` y centrémonos en la parte que procesa el `"GET"`:

```
# ...
if request.method == "GET":
    all_rows = Entry.objects.order_by("-publication_date")
    json_response = []
    for row in all_rows:
        json_response.append(row.to_json())
    return JsonResponse(json_response, safe=False)
```

Lo que nos interesa es que si el *query param* vale `N`, usemos `Entry.objects.order_by("-publication_date")[:N]`.

Primero hay que saber si han enviado `?size`

Ya sabemos cómo recuperar un *query param*:

```
# ...
if request.method == "GET":
    size = request.GET.get("size", None) # Obtenemos query_param ?size con valor por defecto None
    # ...
    all_rows = Entry.objects.order_by("-publication_date")
    json_response = []
    for row in all_rows:
```

```
        json_response.append(row.to_json())
    return JsonResponse(json_response, safe=False)
```

Antes que nada, vamos a intentar **convertirlo a `int`**.

Recuerda que los *query param* son de tipo `string` por defecto, y tenemos que validarlos. Es decir, si alguien envía `?size=big` ... Deberíamos entregar un `400 Bad Request`.

Así:

```
if request.method == "GET":
    size = request.GET.get("size", None)
    if size is not None:
        try:
            size = int(size)
        except ValueError:
            return JsonResponse({"error": "Wrong size parameter"}, status=400)
    # ...
    all_rows = Entry.objects.order_by("-publication_date")
    json_response = []
    for row in all_rows:
        json_response.append(row.to_json())
    return JsonResponse(json_response, safe=False)
```

`size` ya está validado

Ahora, o bien es `None`, o es un **número** válido.

Un simple `if` es lo que necesitamos:

```
- # ...
- all_rows = Entry.objects.order_by("-publication_date")
+ if size is None:
+     all_rows = Entry.objects.order_by("-publication_date")
+ else:
+     all_rows = Entry.objects.order_by("-publication_date")[:size]
```

¡Listo!

Ya puedes verificar algún ejemplo:

- <http://localhost:8000/entries?size=1>

Por último

Comprueba 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 4](#) 2 days ago