


Open   Opened 2 days ago by  **Rubén Montero**

# Una pequeña mejora: Ordenando el resultado

## Resumen

- Sustituiremos `Entry.objects.all()` por otra alternativa que ordena las *Entradas*

## Descripción

Ahora mismo, cada vez que alguien añade una *Entrada* a la base de datos, va a parar **al final**.

Título	Fecha
Primera publicación	Lunes
Otra publicación	Martes
¡Llenando el muro!	Jueves

Debemos saber que `Entry.objects.all()` devuelve **todos** los datos, pero no sabemos en *qué* orden.

Predeciblemente, lo que le resulte más comodo a la base de datos. Es decir, **empezando por el más antiguo**.

## Un orden un poco raro

Siendo así, el usuario tendrá que mirar **todas** las *Entradas* hasta poder llegar a la más reciente.

¿No es común ordenar de *más reciente a más antiguo*? (e.g.: *Timeline* de Twitter, bandeja del correo electrónico,...)

## Podemos hacerlo

En Django, podemos sustituir:

```
all_rows = Entry.objects.all()
```

...por:

```
all_rows = Entry.objects.order_by("-publication_date")
```

- El `-` significa orden *descendente*. Si lo quisiéramos al revés, podríamos usar `+`
- A continuación, va el nombre de un *atributo* de `Entry`

## ¡Pues hagámoslo!

Debe ser en la parte que procesa el `"GET"` de la función `all_entries` en `endpoints.py` :

```
@csrf_exempt
def all_entries(request):
    if request.method == "GET":
        - all_rows = Entry.objects.all()
        + 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)
    # ...
```

(Modifica **sólo** este endpoint. **No** alteres el de los Comentarios o el test romperá)

## 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