


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

Códigos de respuesta HTTP

Resumen

- Repasaremos los posibles códigos de respuesta HTTP
- Crearemos un *endpoint* que devuelve un 404
- Veremos cómo inspeccionarlo con el navegador

Descripción

Los servidores HTTP entregan respuestas con las siguientes partes:

- Línea de respuesta
- Cabeceras
- Cuerpo

Hasta ahora sólo nos hemos preocupado por el cuerpo, mandando un JSON ó HTML.

En la línea de respuesta viaja algo esencial, el **código HTTP de respuesta**.

Se dividen en:

- 1xx: Informativos
- 2xx: De éxito
- 3xx: De redirección
- 4xx: Error de cliente
- 5xx: Error de servidor

¿Y por qué son esenciales?

HTTP es clave en el mundo de Internet porque agiliza los intercambios de información.

Por ejemplo, si una respuesta es **200 OK**, puede que un *intermediario* la *cachee* un tiempo. Esa **caché** servirá para que la respuesta se entregue **más rápido** en el futuro.

En cambio, si es un **500 Internal Server Error**, **no** se *cacheará*, pues *quizá* el servidor funcione bien en cualquier momento.

En resumen

El correcto uso de HTTP mejora las comunicaciones de red y la escalabilidad de tu aplicación.

Hazlo bien, y funcionará bien.

Hazlo mal...

Vale, ¿y cómo juego con códigos de respuesta en Django?

Basta con especificar el parámetro `status` ¹ en la `JsonResponse`.

¿Por ejemplo?

Añade esta nueva función a `endpoints.py`:

```
def my_not_found_endpoint(request):
    json_dictionary = {
        "message": "No encontrado"
    }
    return JsonResponse(json_dictionary, status=404)
```

Y **mapéala** en `urls.py`:

```
urlpatterns = [
    # ...
    path('api/v1/not_found_example', endpoints.my_not_found_endpoint),
```

```
] ]
```

¿Qué ves al visitar http://localhost:8000/api/v1/not_found_example ?

Un JSON

...y nada más.

El **código HTTP** es algo interno.

Pero podemos inspeccionarlo:

¿Cómo?

Click derecho en el navegador > Inspeccionar > Red


Refresca la página con F5.

¿Ves la indicación de **404** en alguna parte?

Por último

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

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

-
1. En Java, los parámetros son **posicionales**. Es decir, al invocar la función los escribimos *en el mismo orden* que están declarados. En Python, también. Pero **a mayores**, podemos indicar parámetros usando la **keyword** con la que están declarados y así desordenarlos. Por ejemplo, `JsonResponse([], status=404, safe=False)` es igual de válido que `JsonResponse([], safe=False, status=404)` 
-



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