

Modelo de respuesta para un error

Desarrollo de un API REST con Spring Boot

Mensaje de error por defecto

- Lo “fabrica” Spring
- Contiene
 - Fecha
 - Estado
 - Nombre del error
 - Mensaje
 - Ruta en la que se ha producido.

Error por defecto

- Instancia de DefaultErrorAttributes

```
{  
  "timestamp": "2019-10-05T16:38:25.227+0000",  
  "status": 404,  
  "error": "Not Found",  
  "message": "No se puede encontrar el producto con la ID: 1",  
  "path": "/producto/1"  
}
```

Puede que nos interese *customizarlo*

- Por ejemplo, para quitar el campo status (lo podemos obtener de la propia respuesta)
- Personalizar errores (por ejemplo, varios tipos diferentes de BAD_REQUEST)
- Añadir campos nuevos.
- Agregar varios errores (por ejemplo de validación) en uno solo
- ...

Modelo para el error

- Clase POJO
- Podemos usar lombok si es necesario
- Debe incluir aquellos campos que estimemos necesarios.

Modelo básico

- Estado (HttpStatus)
- Fecha (LocalDateTime)
- Mensaje (String)

Modelo más complejo

- Estado (HttpStatus)
- Fecha (LocalDateTime)
- Mensajes (List<String>)
 - Nos permite indicar más de un mensaje a la vez.

Modelo anidado

- Estado (HttpStatus)
- Fecha (LocalDateTime)
- Mensaje (String)
- MensajeDebug (String)
 - Mensaje del sistema, con mayor grado de descripción del error.
- List<SubError>
 - Lista de pequeños errores contenidos en este error.

SubError

- Podemos definir un *SubError* genérico
- O bien un *SubError* abstracto y crear subtipos
 - Por ejemplo, para validación, incluyendo
 - Objeto sobre el que hay un error
 - Campo sobre el que hay un error
 - Valor que provoca el error
 - Mensaje de validación

En nuestro ejemplo

- Comenzaremos por el modelo básico