

Manejo de errores con *@ExceptionHandler*

Desarrollo de un API REST con Spring Boot

Modelo de error

```
@Setter
@Getter
public class ApiError {

    private HttpStatus estado;
    @JsonFormat(shape = Shape.STRING, pattern = "dd/MM/yyyy
hh:mm:ss")
    private LocalDateTime fecha;
    private String mensaje;

}
```

¿Qué buscamos?

- Que nuestros errores sean ahora del tipo

```
{
```

```
  "estado": "NOT_FOUND",
```

```
  "fecha": "05/10/2019 08:08:56",
```

```
  "mensaje": "No se puede encontrar el producto con la ID: 1"
```

```
}
```

Unificar excepciones y personalización del mensaje

- Lo hacemos a través de la anotación *@ExceptionHandler*
- Acepta como parámetro una o varias excepciones
 - *@ExceptionHandler(TheException.class)*
 - *@ExceptionHandler({ OneException.class, TwoException.class })*
- Una primer aproximación nos permite ubicarla en un método de cualquier controlador.
- Manejará las excepciones de uno o varios tipos que se produzcan en dicho controlador.

@ExceptionHandler

- Anotando un método del controlador
- Dicho método puede tener una firma variopinta.
- Como argumento puede recibir:
 - Una instancia de una excepción
 - Objetos Request, Response
 - Objetos de sesión (HttpSession)
 - Locale
 -

@ExceptionHandler

- Como tipo de retorno, puede devolver:
 - void
 - *Model*, *ModelAndView*, *Map* (motor de plantillas)
 - String
 - @ResponseBody
 - HttpEntity, ResponseEntity

¿Qué vamos a añadir?

- Método *handleProductoNotFoundException*
 - Para manejar excepciones de productos no encontrados
- Método *handleJsonMappingException*
 - Para manejar los errores de parseo de valores al crear/editar un producto (*JsonMappingException*)