**ErrorResponse**

ErrorResponse es una entidad utilizada para representar la respuesta de error devuelta por el controlador en caso de que se produzca una excepción. Esta entidad tiene tres campos:

HttpStatus: el estado HTTP que se devolverá en la respuesta. En este ejemplo, se establece a HttpStatus.BAD\_REQUEST.

message: un mensaje de error general para la respuesta. En este ejemplo, se establece a "Validacion Fallida".

errors: una lista de mensajes de error específicos. En este ejemplo, se utiliza para almacenar los mensajes de error de validación.

**@ControllerAdvice**

@ControllerAdvice es una anotación utilizada para definir una clase que maneja excepciones para todos los controladores en una aplicación. Cuando se produce una excepción en un controlador, la clase anotada con @ControllerAdvice puede interceptar la excepción y proporcionar una respuesta de error personalizada.

**@ExceptionHandler**

@ExceptionHandler es una anotación utilizada para definir un método que maneja una excepción específica. Cuando se produce la excepción especificada, Spring llama al método anotado con @ExceptionHandler para manejar la excepción.

En el ejemplo proporcionado, hay dos métodos @ExceptionHandler. Uno maneja MethodArgumentNotValidException y otro maneja ConstraintViolationException.

**MethodArgumentNotValidException**

MethodArgumentNotValidException es una excepción que se produce cuando se intenta vincular un objeto de solicitud a un método de controlador, pero la validación del objeto falla. Por ejemplo, si se requiere que un campo de un objeto de solicitud tenga una longitud mínima de 5 caracteres, pero se proporciona una cadena de solo 3 caracteres, se producirá una excepción MethodArgumentNotValidException.

**ConstraintViolationException**

ConstraintViolationException es una excepción que se produce cuando se viola una restricción de validación. Las restricciones de validación son anotaciones que se pueden agregar a los campos de un objeto de solicitud para especificar reglas de validación. Por ejemplo, la anotación @NotNull se puede utilizar para especificar que un campo no puede ser nulo. Si un campo anotado con @NotNull es nulo, se producirá una excepción ConstraintViolationException.