

Package exception

Class GlobalExceptionHandler

java.lang.Object
exception.GlobalExceptionHandler

@ControllerAdvice
public class **GlobalExceptionHandler**
extends Object

Manejador global de excepciones Intercepta excepciones y devuelve respuestas HTTP apropiadas con códigos estándar Códigos HTTP manejados: - 400 Bad Request: Errores de validación de entrada - 401 Unauthorized: Token inválido o expirado - 403 Forbidden: Sin permisos para acceder al recurso - 404 Not Found: Recurso no encontrado - 409 Conflict: Recurso duplicado - 422 Unprocessable Entity: Validación de negocio fallida - 500 Internal Server Error: Error no manejado en el servidor

Constructor Summary

Constructors

Constructor

Description

GlobalExceptionHandler()

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type

Method

Description

org.springframework.http.ResponseEntity<ErrorResponse>

handleAccessDeniedException(org.springframework.security.access.AccessDeniedException ex,
org.springframework.web.context.request.WebRequest request)

Maneja AccessDeniedException → 403 Forbidden Se lanza cuando Spring Security rechaza el acceso por falta de rol/permisos
Ejemplo: Usuario sin rol ADMIN intenta acceder a /api/admin/*

org.springframework.http.ResponseEntity<ErrorResponse>

handleDuplicateResourceException(DuplicateResourceException ex,
org.springframework.web.context.request.WebRequest request)

Maneja DuplicateResourceException → 409 Conflict Se lanza cuando se intenta crear un recurso que ya existe (duplicado)
Ejemplo: Registrar usuario con username que ya existe

org.springframework.http.ResponseEntity<ErrorResponse>

handleForbiddenException(ForbiddenException ex, org.springframework.web.context.request.WebRequest request)

Maneja ForbiddenException → 403 Forbidden Se lanza cuando el usuario no tiene permisos para realizar la acción Ejemplo:
USER intenta editar lección creada por otro ADMIN

org.springframework.http.ResponseEntity<ErrorResponse>

handleGlobalException(Exception ex, org.springframework.web.context.request.WebRequest request)

Maneja excepciones genéricas → 500 Internal Server Error Se lanza para cualquier excepción no manejada explícitamente
Incluye: - NullPointerException - DatabaseException - Errores inesperados del servidor Acción: Se registra el error completo en logs para debugging

org.springframework.http.ResponseEntity<ErrorResponse>

handleIllegalArgumentException(IllegalArgumentException ex,
org.springframework.web.context.request.WebRequest request)

Maneja IllegalArgumentException → 400 Bad Request Se lanza cuando se proporcionan argumentos inválidos Ejemplo:
Parámetro de página negativo

org.springframework.http.ResponseEntity<ErrorResponse>

handleMethodArgumentTypeMismatchException

```
(org.springframework.web.method.annotation.MethodArgumentTypeMismatchException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja MethodArgumentTypeMismatchException → 400 Bad Request Se lanza cuando un parámetro tiene tipo incorrecto
Ejemplo: GET /api/lessons/abc (esperaba un Long, recibió String)

```
org.springframework.http.ResponseEntity<ErrorResponse>
```

```
handleNoHandlerFoundException(org.springframework.web.servlet.NoHandlerFoundException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja NoHandlerNotFoundException → 404 Not Found Se lanza cuando se accede a un endpoint que no existe Ejemplo: GET /api/no-existe-este-endpoint

```
org.springframework.http.ResponseEntity<ErrorResponse>
```

```
handleResourceNotFoundException(ResourceNotFoundException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja ResourceNotFoundException → 404 Not Found Se lanza cuando se intenta acceder a un recurso que no existe
Ejemplo: GET /api/lessons/999 (lección no existe)

```
org.springframework.http.ResponseEntity<ErrorResponse>
```

```
handleUnauthorizedException(UnauthorizedException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja UnauthorizedException → 401 Unauthorized Se lanza cuando el token es inválido, expirado o no está presente
Ejemplo: Token JWT expirado o malformado

```
org.springframework.http.ResponseEntity<ErrorResponse>
```

```
handleUnprocessableEntityException(UnprocessableEntityException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja UnprocessableEntityException → 422 Unprocessable Entity Se lanza cuando la validación de negocio falla Los datos son válidos sintácticamente pero violan reglas de negocio Ejemplos: - Intentar publicar lección sin pasos - Intentar marcar lección como completada si no está publicada - Valores inconsistentes o conflictivos

```
org.springframework.http.ResponseEntity<ErrorResponse>
```

```
handleValidationException(org.springframework.web.bind.MethodArgumentNotValidException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja validación de DTOs → 400 Bad Request Se lanza cuando los parámetros de entrada no cumplen validaciones @Valid
Ejemplo: Email inválido en POST /api/auth/register

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Details

GlobalExceptionHandler

```
public GlobalExceptionHandler()
```

Method Details

handleResourceNotFoundException

```
@ExceptionHandler(ResourceNotFoundException.class)
public org.springframework.http.ResponseEntity<ErrorResponse> handleResourceNotFoundException
(ResourceNotFoundException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja ResourceNotFoundException → 404 Not Found Se lanza cuando se intenta acceder a un recurso que no existe
Ejemplo: GET /api/lessons/999 (lección no existe)

handleDuplicateResourceException

```
@ExceptionHandler(DuplicateResourceException.class)
```

```
public org.springframework.http.ResponseEntity<ErrorResponse> handleDuplicateResourceException  
(DuplicateResourceException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja DuplicateResourceException → 409 Conflict Se lanza cuando se intenta crear un recurso que ya existe (duplicado)
Ejemplo: Registrar usuario con username que ya existe

handleForbiddenException

```
@ExceptionHandler(ForbiddenException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleForbiddenException  
(ForbiddenException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja ForbiddenException → 403 Forbidden Se lanza cuando el usuario no tiene permisos para realizar la acción Ejemplo:
USER intenta editar lección creada por otro ADMIN

handleUnauthorizedException

```
@ExceptionHandler(UnauthorizedException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleUnauthorizedException  
(UnauthorizedException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja UnauthorizedException → 401 Unauthorized Se lanza cuando el token es inválido, expirado o no está presente
Ejemplo: Token JWT expirado o malformado

handleAccessDeniedException

```
@ExceptionHandler(org.springframework.security.access.AccessDeniedException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleAccessDeniedException  
(org.springframework.security.access.AccessDeniedException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja AccessDeniedException → 403 Forbidden Se lanza cuando Spring Security rechaza el acceso por falta de rol/permisos
Ejemplo: Usuario sin rol ADMIN intenta acceder a /api/admin/*

handleUnprocessableEntityException

```
@ExceptionHandler(UnprocessableEntityException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleUnprocessableEntityException  
(UnprocessableEntityException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja UnprocessableEntityException → 422 Unprocessable Entity Se lanza cuando la validación de negocio falla Los datos
son válidos sintácticamente pero violan reglas de negocio Ejemplos: - Intentar publicar lección sin pasos - Intentar marcar
lección como completada si no está publicada - Valores inconsistentes o conflictivos

handleValidationException

```
@ExceptionHandler(org.springframework.web.bind.MethodArgumentNotValidException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleValidationException  
(org.springframework.web.bind.MethodArgumentNotValidException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja validación de DTOs → 400 Bad Request Se lanza cuando los parámetros de entrada no cumplen validaciones @Valid
Ejemplo: Email inválido en POST /api/auth/register

handleIllegalArgumentException

```
@ExceptionHandler(java.lang.IllegalArgumentException.class)  
public org.springframework.http.ResponseEntity<ErrorResponse> handleIllegalArgumentException  
(IllegalArgumentException ex,  
org.springframework.web.context.request.WebRequest request)
```

Maneja IllegalArgumentException → 400 Bad Request Se lanza cuando se proporcionan argumentos inválidos Ejemplo:
Parámetro de página negativo

handleMethodArgumentTypeMismatchException

```
@ExceptionHandler(org.springframework.web.method.annotation.MethodArgumentTypeMismatchException.class)
public org.springframework.http.ResponseEntity<ErrorResponse> handleMethodArgumentTypeMismatchException
(org.springframework.web.method.annotation.MethodArgumentTypeMismatchException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja MethodArgumentTypeMismatchException → 400 Bad Request Se lanza cuando un parámetro tiene tipo incorrecto
Ejemplo: GET /api/lessons/abc (esperaba un Long, recibió String)

handleNoHandlerFoundException

```
@ExceptionHandler(org.springframework.web.servlet.NoHandlerFoundException.class)
public org.springframework.http.ResponseEntity<ErrorResponse> handleNoHandlerFoundException
(org.springframework.web.servlet.NoHandlerFoundException ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja NoHandlerFoundException → 404 Not Found Se lanza cuando se accede a un endpoint que no existe Ejemplo: GET /api/no-existe-este-endpoint

handleGlobalException

```
@ExceptionHandler(java.lang.Exception.class)
public org.springframework.http.ResponseEntity<ErrorResponse> handleGlobalException
(Exception ex,
org.springframework.web.context.request.WebRequest request)
```

Maneja excepciones genéricas → 500 Internal Server Error Se lanza para cualquier excepción no manejada explícitamente
Incluye: - NullPointerException - DatabaseException - Errores inesperados del servidor Acción: Se registra el error completo en logs para debugging