# 📘 Global Exception Handling in Spring Boot

## 🧾 Overview

This document explains the centralized exception handling mechanism implemented in our Spring Boot application. It ensures that all runtime errors are handled uniformly, logged properly, and returned to the client in a consistent structure.

## 📌 Objectives

- Centralize all exception responses.

- Ensure proper logging with traceability.

- Return user-friendly and developer-informative error responses.

- Avoid boilerplate try-catch in every controller.

- Support custom business exceptions (like record not found, duplicate, etc.).

## 🔧 Key Components

|  |  |
| --- | --- |
| Component | Description |
| GlobalExceptionHandler | A @ControllerAdvice class that catches and processes exceptions globally. |
| ApiCustomException | Custom exception for application-specific errors. |
| ErrorMessage | POJO representing the error response. |
| ValidationDetails | Encapsulates field-specific error details. |
| ErrorConstants | Central store of error code keys. |
| ErrorHandlingUtil | Utility class to fetch messages and build validations. |

## 📁 Exception Flow Diagram

[Request] → [Controller] → [Exception Thrown] → [GlobalExceptionHandler] → [ErrorMessage Response]

## 🚨 Exception Types Covered

|  |  |  |  |
| --- | --- | --- | --- |
| Exception Type | HTTP Status | Handler Method | Use Case Example |
| RuntimeException | 500 Internal Error | handleAnyException | Unexpected backend failure |
| HttpRequestMethodNotSupportedException | 405 Not Allowed | handleMethodTypeException | GET instead of POST |
| HttpMediaTypeNotSupportedException | 415 Unsupported | handleMediaTypeException | Missing/invalid content-type |
| MethodArgumentNotValidException | 400 Bad Request | handleMethodArgumentNotValidException | Invalid input validation |
| ApiCustomException | Custom (e.g. 404) | handleApiCustomException | Record not found, duplicate, etc. |

## 🔧 Sample Error Response Format

{  
 "refId": "2ac2e320-abc1-4e2a-9a2e-9ab1c721c86a",  
 "code": "ERR\_404",  
 "message": "No record found for the given ID.",  
 "status": 404,  
 "severity": "ERROR",  
 "errorDetails": [  
 {  
 "field": "recordId",  
 "message": "Record not found for ID: 99"  
 }  
 ]  
}

## ✅ Benefits

- Cleaner controller code.

- Easy debugging with structured logs.

- Consistent error response for all clients (mobile/web).

- Extensible for new custom exceptions.