\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Standardize exception handling \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **Create a model class for error**

**package** net.guides.springboot2.springboot2jpacrudexample.exception;

**import** java.util.Date;

**public** **class** ErrorDetails {

**private** Date timestamp;

**private** String message;

**private** String details;

**public** ErrorDetails(Date timestamp, String message, String details) {

**super**();

**this**.timestamp = timestamp;

**this**.message = message;

**this**.details = details;

}

**public** Date getTimestamp() {

**return** timestamp;

}

**public** String getMessage() {

**return** message;

}

**public** String getDetails() {

**return** details;

}

}

1. **Create @ControllerAdvice class for all exceptions**.

package net.guides.springboot2.springboot2jpacrudexample.exception;

**@ControllerAdvice**

public class GlobalExceptionHandler extends **ResponseEntityExceptionHandler** {

@ExceptionHandler(**ResourceNotFoundException.class**)

public ResponseEntity<?> resourceNotFoundException(ResourceNotFoundException ex, WebRequest request) {

ErrorDetails errorDetails = new ErrorDetails(new Date(), ex.getMessage(), request.getDescription(false));

return new ResponseEntity<>(errorDetails, HttpStatus.NOT\_FOUND);

}

@ExceptionHandler(**Exception.class**)

public ResponseEntity<?> globleExcpetionHandler(Exception ex, WebRequest request) {

ErrorDetails errorDetails = new ErrorDetails(new Date(), ex.getMessage(), request.getDescription(false));

return new ResponseEntity<>(errorDetails, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

1. **Create custom exception class also.**

package net.guides.springboot2.springboot2jpacrudexample.exception;

@ResponseStatus(value = HttpStatus.NOT\_FOUND)

**public class ResourceNotFoundException extends Exception{**

private static final long serialVersionUID = 1L;

public ResourceNotFoundException(String message){

super(message);

}

}

1. **Throw exception on conditions.**

Employee employee = employeeRepository.findById(employeeId)

.orElseThrow(() -> **new** ResourceNotFoundException("Employee not found for this id :: " + employeeId));

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Standardize exception handling \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*