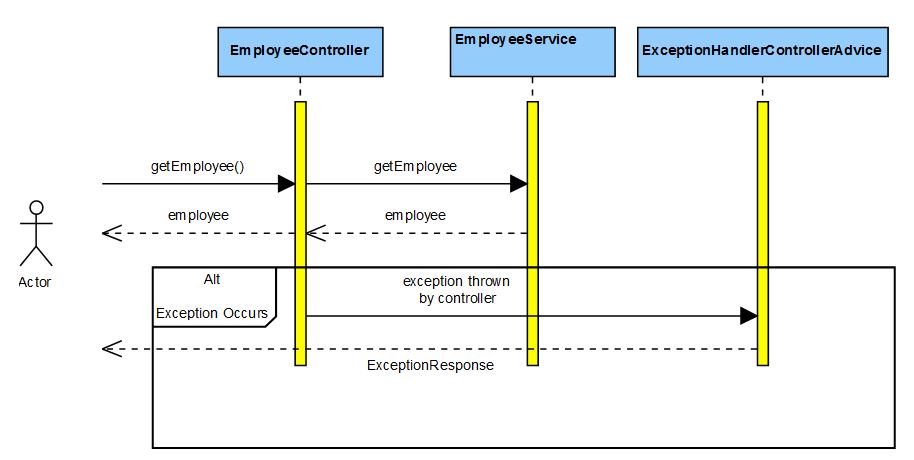
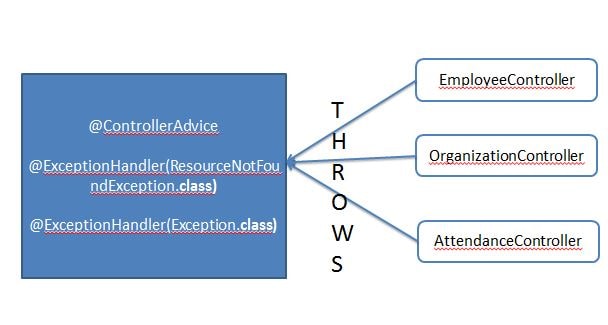
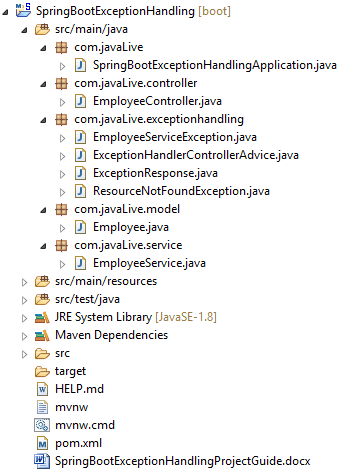
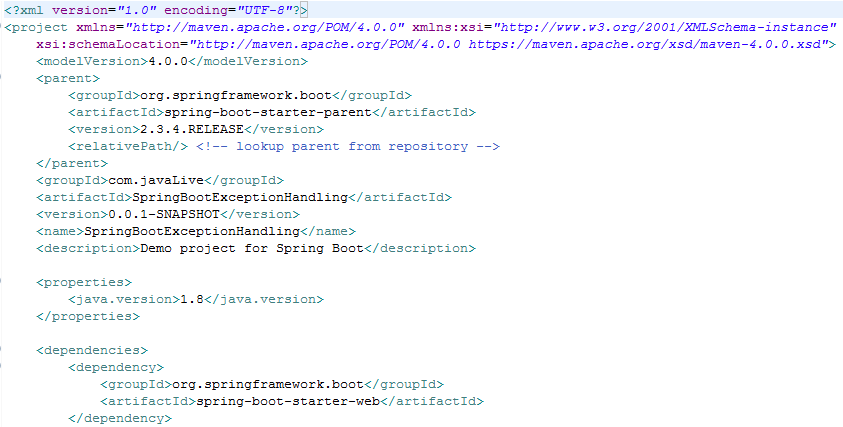
Spring Boot Exception Handling Using ControllerAdvice

Spring provides a very useful way to handle exceptions using ControllerAdvice.  
We will be implementing a ControlerAdvice class which will handle all exceptions thrown by the controller class.  
  
Exceptions thrown by a Controller method is mapped to the ControllerAdvice method using @ExceptionHandler annotations.  
Lets Begin-

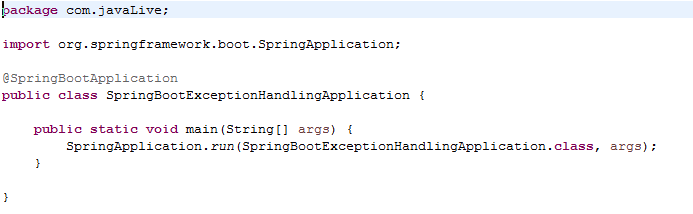
Maven Project will be as follows-



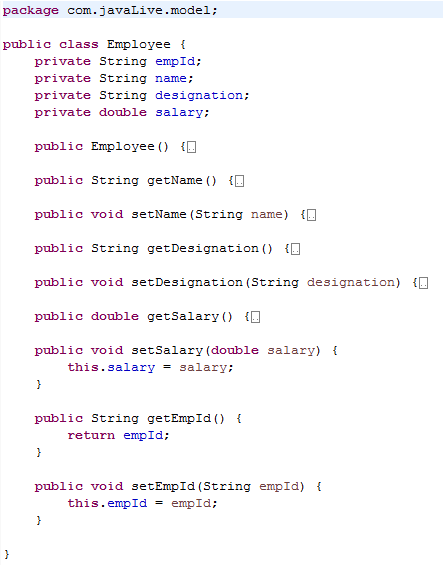
In the Maven we need the spring boot web dependency.Maven will be as follows-



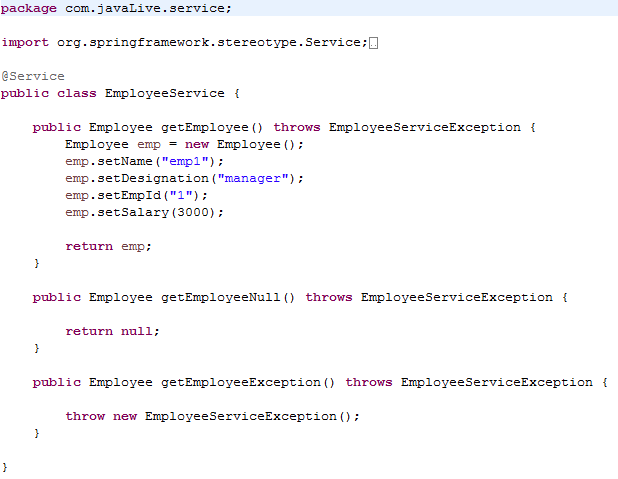
Create the SpringBootHelloWorldApplication.java as below-



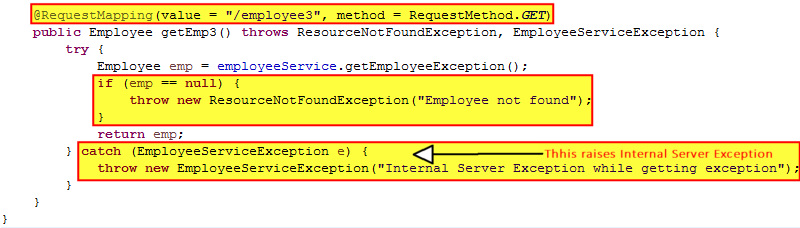
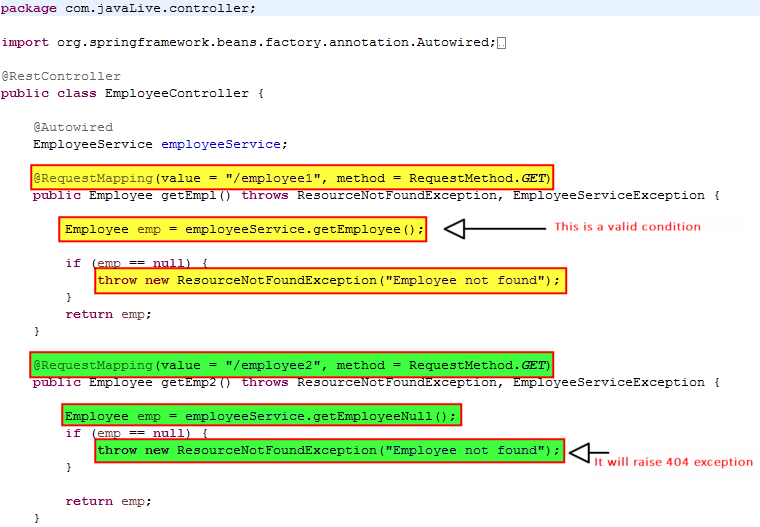
Create the Employee model class as follows-



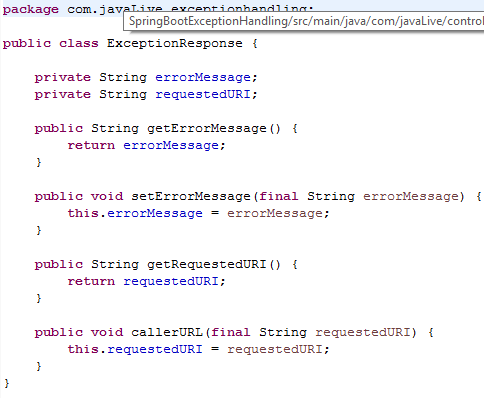
We first define the classes in the service package.  
Define the service class. We have defined three methods in it for explaining 3 scenarios.



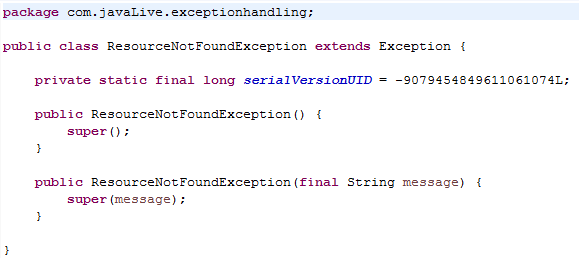
Now we define the Controller package classes  
@RequestMapping maps /employee request to return an employee object. We have two other requestmappings, one returns a null employee and the other throws an exception. Also the EmployeeController has the EmployeeService instance autowired.



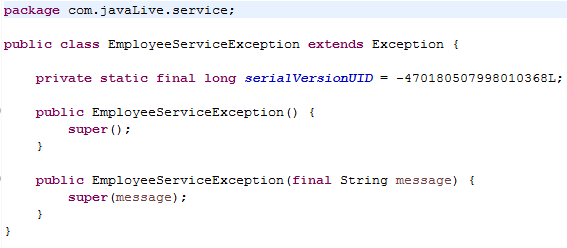
Define the POJO class which will be sent as response by the ControllerAdvice class viz. com.javaLive.exceptionhandling.ExceptionResponse



Define the com.javaLive.exceptionhandling.ResourceNotFoundException class. This Exception will be thrown by the controller when no resource i.e. employee to be returned in our case is found.



Define the EmployeeServiceException to thrown by the EmployeeService class in case of exceptions

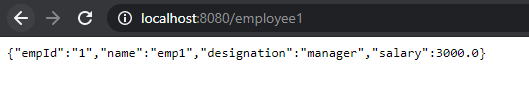


The ExceptionHandlerControllerAdvice will catch the exceptions thrown by the controller method, and we send more appropriate response to the caller. For example if an exception occurs we don't want the caller to get the entire stack trace involving technical details. **Also we can return the appropriate response status depending on the business logic.**

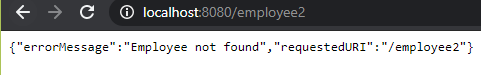
****

Compile and the run the SpringBootHelloWorldApplication.java as a Java application.

**localhost:8080/employee1**

****

**localhost:8080/employee2** Employee is not found, so ResourceNotFoundException is thrown. **Response code is sent as 404.**

****

**localhost:8080/employee3** EmployeeServiceException is thrown. **Response code is sent as 500.**

