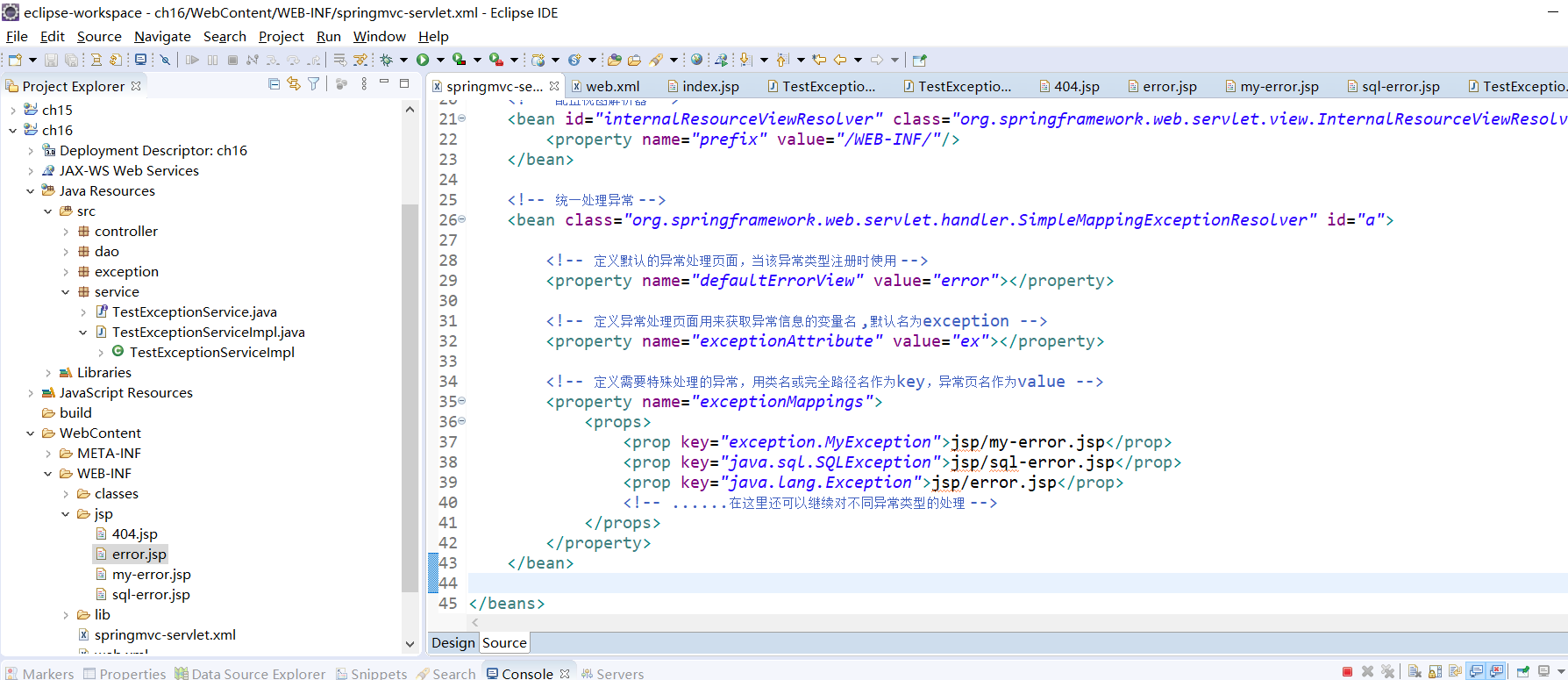
统一异常处理简介：

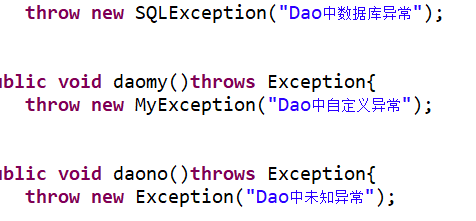
1. 在Spring MVC应用开发中，不管是对底层数据库操作，还是业务层或控制层操作， 都会不可避免的遇到可预知的、不可预知的异常需要处理。而如果对每个过程的异 常都单独处理，那么系统的代码耦合度高、工作量大且不好统一，以后维护的工作 量也大。
2. 如果能将所有类型的异常处理从各层中解耦出来，这样既保证了相关处理过程的功 能单一，又实现了异常信息的统一处理和维护。幸运的是，Spring MVC提供了这样 的功能。
3. Spring MVC提供了三种解决统一异常处理的方案：
4. Spring MVC提供的简单异常处理器SimpleMappingExceptionResolver
5. 通过实现Spring的异常处理接口HandlerExceptionResolver自定义自己的 异常处理器。
6. 使用@ExceptionHandler注解实现异常处理

**SimpleMappingExceptionResolver类：**

使用org.springframework.web.servlet.handler.SimpleMappingExceptionResolver类统一 处理异常时需要在Spring MVC配置文件中配置异常类和View的对应关系。如下：

（需要注意的是所有的错误页要使用isErrorPage属性确定为错误页）

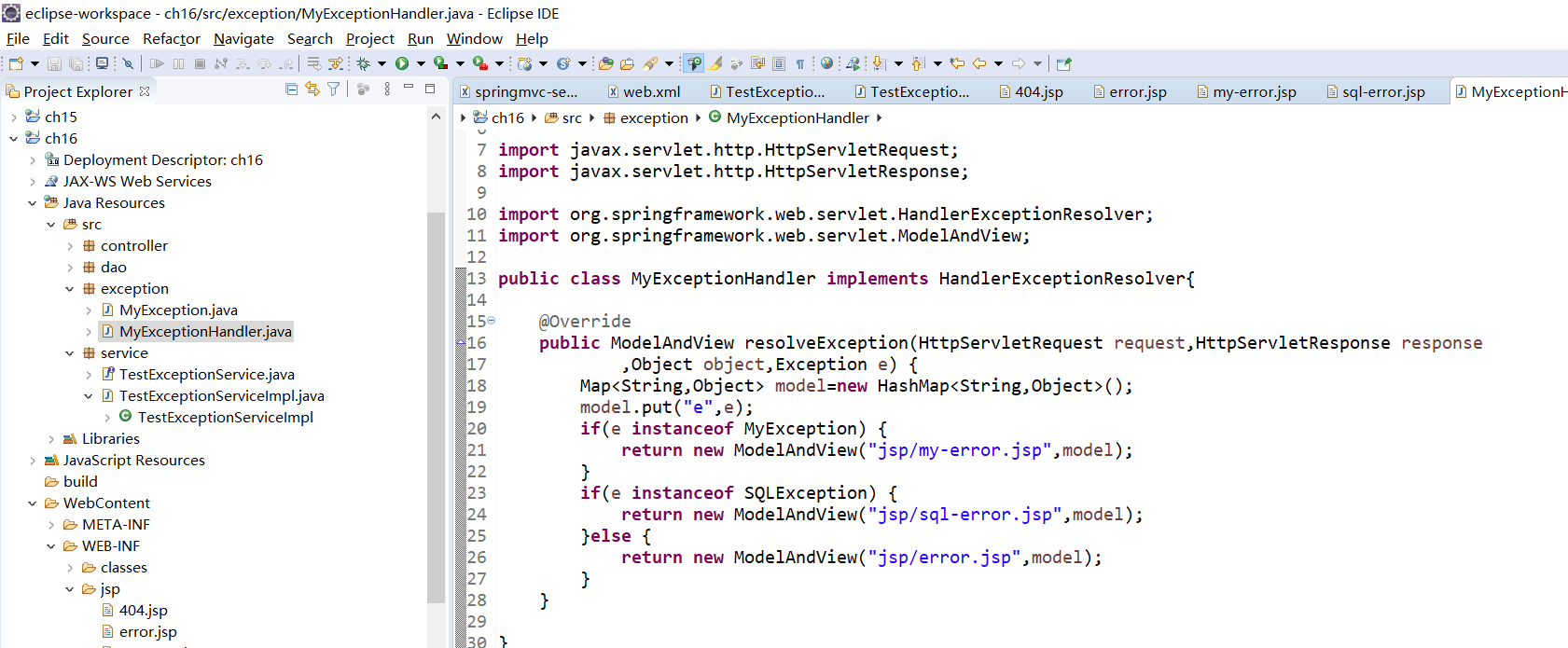






**HandlerExceptionResolver接口：**

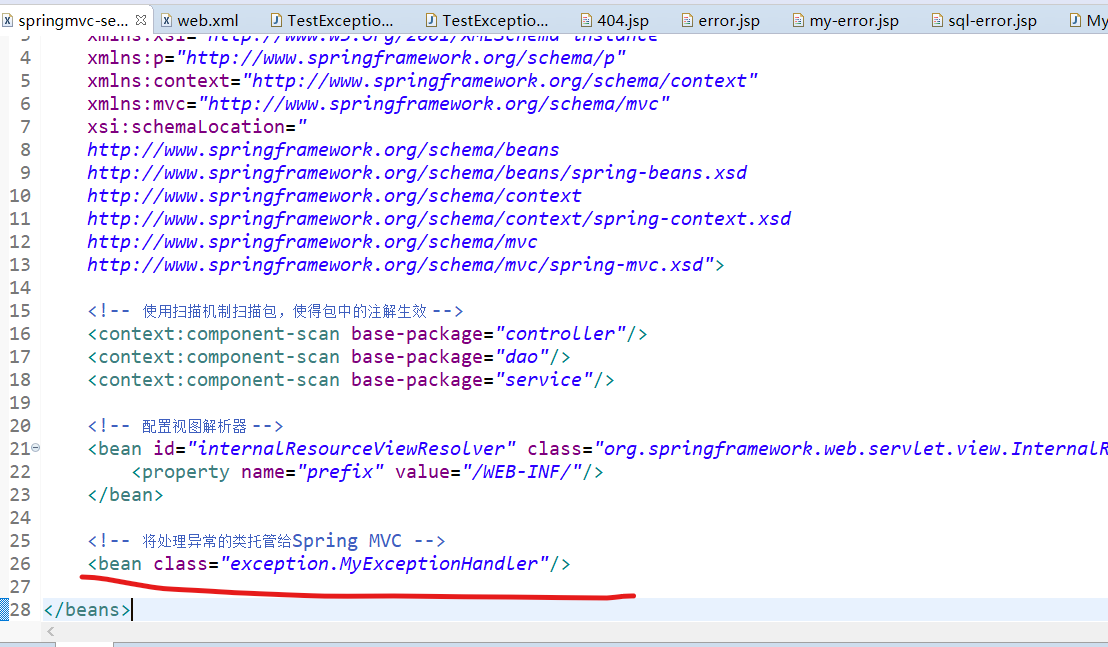
org.springframework.web.servlet.handlerExceptionResolver接口用于解析请求处理过程中 所产生的的异常。开发者可以开发该接口的实现类进行Spring MVC应用的异常统一处 理。



开发完处理异常的类后，还需要在Spring MVC配置文件中将其托管给Spring MVC框架 才能进行异常的统一处理，配置代码为：

<bean class=”异常处理类的包类路径”/>

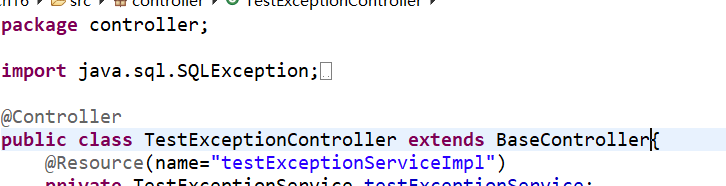
如：



**@ExceptionHandler注解：**

使用@ExceptionHandler注解统一处理异常时不需要在Spring MVC配置文件中声明任何 关于异常处理的代码，而是定义一个类（抽象类更好，因为它只需要处理异常，不需要 实例化），让所有的控制器都继承这个类。并在这个类中定义一个方法完成统一的异常 处理。如：





**总结：**

可以得出在Spring MVC中对异常进行统一的处理有3种方法

1. 一是使用SimpleMappingExceptionResolver类来完成。采用这种方法只需要在

Spring MVC的配置文件中进行配置即可。

1. 二是使用HandlerExceptionResolver接口。采用这种方式需要编写统一异常处理 类，还需要在Spring MVC配置文件中进行配置，将统一处理异常的类托管给 Spring MVC。
2. 三是使用@ExceptionHandler注解。采用这种方式也需要编写一个统一异常处理 类，在处理类中编写一个方法用于处理异常，该方法需要使用 @ExceptionHandler注解进行标注。同时，各个控制器都需要继承这个类。不需要 在Spring MVC配置文件中进行配置。

开发者可以根据实际情况和个人风格选择适合的方式来进行统一处理。另外，在进 行统一异常处理时，一般也需要借助Web服务器来辅助统一异常处理，例如配置 404全局错误页；其实统一异常处理也可 以单独靠Web服务器来完成，通过 web.xml描述符来配置全局错误。