异常处理

SpringMVC通过HandlerExceptionResolver（接口）处理程序的异常，包括Handler映射、数据绑定以及目标方法执行时发生的异常。

常用的实现类：

* + **★ExceptionHandlerExceptionResolver**
  + **DefaultHandlerExceptionResolver**
  + **ResponseStatusExceptionResolver**
  + **SimpleMappingExceptionResolver**

使用了<mvc:annotation-driven>配置会自动添加三个实现类

ExceptionHandlerExceptionResolver，DefaultHandlerExceptionResolver

，ResponseStatusExceptionResolver

1. **ExceptionHandlerExceptionResolver**

主要处理@ExceptionHandler注解标记的方法

使用方法：

1. 写一个异常方法，加上@ExceptionHandler注解

@ExceptionHandler({ArithmeticException.**class**})

**public** ModelAndView testExceptionHandler(Exception e) {

ModelAndView modelAndView = **new** ModelAndView("error");

modelAndView.addObject("message", e.getMessage());

**return** modelAndView;

}

**注意：**

**① 可以添加一个exception入参获取那个异常对象**

**② 入参中不能有map，只能用ModelAndView做返回值来添加域对象属性**

**③ @ExceptionHandler注解中可以指定需要处理的异常类型**

**④ @ExceptionHandler注解标记的异常有优先级的问题，越接近的类型(异常父类中)越先被匹配，如果标记了这个异常直接被匹配**

**⑤这个异常处理方法只能在此类中使用**

**可以直接创建一个异常处理类，类前面加上一个注解@ControllerAdvice**

@ControllerAdvice

**public** **class** HandlerException {

@ExceptionHandler({ ArithmeticException.**class** })

**public** ModelAndView testExceptionHandler(Exception e) {

ModelAndView modelAndView = **new** ModelAndView("error");

modelAndView.addObject("message", e.getMessage());

**return** modelAndView;

}

}

这个类就可以专门用来处理异常了，**这个类中的所有异常处理方法可以处理所有handler类的异常，如果handler内有异常处理方法，优先使用**

1. **ResponseStatusExceptionResolver**

处理@ResponseStatus注解标记的异常类获取方法，**可以指定错误码（例404）和错误信息到错误页面。**

* 被@ResponseStatus注解标记的方法，如果执行成功，会被转到指定的错误页面，如果有异常，则转向异常对应的错误页面。
* 被@ResponseStatus注解标记的异常类，如果被抛出，会显示注解中指定的错误码和错误信息

例：@ResponseStatus(reason="测试",value=HttpStatus.***NOT\_FOUND***)

1. **DefaultHandlerExceptionResolver**

处理SpringMVC特定的一些异常类，如请求方式不对应

1. **SimpleMappingExceptionResolver**

可以在SpringMVC中简单配置，出现那个异常转向哪个页面

<bean class=*"org.springframework.web.servlet.handler.SimpleMappingExceptionResolver"*>

**该异常处理器会自动把处理的异常放到request域中，属性名默认为exception，可以通过设置exceptionAttribute来自定义**

<property name=*"exceptionAttribute"* value=*"exception"*></property>

**配置需要转向某个页面的异常类型，如果key为空，则所有异常全部转向这个页面**

<property name=*"exceptionMappings"*>

<props>

<prop key=*"java.lang.ArrayIndexOutOfBoundsException"*>error</prop>

</props>

</property>

</bean>