

自定义全局异常处理器

一、主要知识点

1、springmvc提供一个HandlerExceptionResolver接口，自定义全局异常处理器必须要实现这个接口

2、全局异常处理器处理思路：

- 解析出异常类型
- 如果该异常类型是系统自定义的异常，直接取出异常信息，在错误页面展示
- 如果该异常类型不是系统自定义的异常，构造一个自定义的异常类型（信息为“未知错误”）

二、实现步骤

1、自定义一个异常类

```
public class SysException extends Exception{  
    // 存储提示信息的  
    private String message;  
    public String getMessage() {  
        return message;  
    }  
    public void setMessage(String message) {  
        this.message = message;  
    }  
    public SysException(String message) {  
        this.message = message;  
    }  
}
```

2、配置异常处理器

```
public class SysExceptionResolver implements HandlerExceptionResolver{
```

```

public ModelAndView resolveException(HttpServletRequest request,
HttpServletResponse response, Object handler, Exception ex) {
    // 获取到异常对象
    SysException e = null;

    //如果抛出的是java系统自定义的异常则直接转换
    if(ex instanceof SysException){
        e = (SysException)ex;
    }else{
        //如果抛出的不是系统自定义的异常，则重新构造一个未知错误异常
        //这里我就也有SysException 省事了，实际中应该要再定义一个新的异常
        e = new SysException("系统未知错误");
    }
    // 创建ModelAndView对象，向前台返回错误信息
    ModelAndView mv = new ModelAndView();
    mv.addObject("errorMsg",e.getMessage());
    mv.setViewName("error"); //error.jsp
    return mv;
}
}

```

3、在springmvc.xml中配置自定义的异常处理器：

```

<!-- 开启注解扫描 -->
<context:component-scan base-package="cn.itcast"/>

<!-- 视图解析器对象 -->
<bean id="internalResourceViewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/pages/" />
    <property name="suffix" value=".jsp" />
</bean>

<!--前端控制器，哪些静态资源不拦截-->
<mvc:resources location="/css/" mapping="/css/**"/>
<mvc:resources location="/images/" mapping="/images/**"/>
<mvc:resources location="/js/" mapping="/js/**"/>

<!--配置异常处理器-->
<bean id="sysExceptionResolver"
class="cn.itcast.exception.SysExceptionResolver"/>

<!-- 开启SpringMVC框架注解的支持 -->
<mvc:annotation-driven />

```

4、配置error.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java"
isELIgnored="false" %>
<html>
<head>
    <title>Title</title>
</head>
<body>

    ${errorMsg}

</body>
</html>
```

5、测试

```
@Controller
@RequestMapping("/user")
public class UserController {
    @RequestMapping("/testException")
    public String testException() throws SysException{
        System.out.println("testException执行了...");
        try {
            // 模拟异常
            int a = 10/0;
        } catch (Exception e) {
            // 打印异常信息
            e.printStackTrace();
            // 抛出自定义异常信息
            throw new SysException("查询所有用户出现错误了...");
        }

        return "success";
    }
}
```