

## 一、SpringBoot整合全局捕获异常

使用AOP技术，采用异常通知

`@ExceptionHandler` 表示拦截异常

`@ControllerAdvice(basePackages = "com.example.wxgz.conntroller")` //捕获异常的包，是Controller的一个辅助类，最常用的就称为全局异常处理的切面

`@ExceptionHandler(RuntimeException.class)` //运行时异常捕获

`@ResponseBody` //进行json转换的返回方法

创建一个用于拦截异常的类，定义异常后的返回处理

```
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseBody;

import java.util.HashMap;
import java.util.Map;

/*
 * 全局捕获异常类
 */
@ControllerAdvice(basePackages = "com.example.wxgz.conntroller") //捕获异常的包
public class GlobExceptionHandler {

    @ExceptionHandler(RuntimeException.class) //运行时异常捕获
    @ResponseBody //进行json转换的返回方法
    public Map<String, Object> errResult(){
        //如果捕获到异常返回错误信息
        Map<String, Object> map = new HashMap<>();
        map.put("errorCode", "10011");
        map.put("errorMsg", "系统错误");
        return map;
    }
}
```

`@ExceptionHandler` 表示拦截异常。

`@ControllerAdvice` 是 controller 的一个辅助类，最常用的就是作为全局异常处理的切面

`@ControllerAdvice` 可以指定扫描范围。

`@ControllerAdvice` 约定了几种可行的返回值，如果是直接返回 model 类的话，需要使用

`@ResponseBody` 进行 json 转换。

返回 String，表示跳到某个 view。

返回 `modelAndView`。

返回 `model + @ResponseBody`。

## 二、SpringBoot整合log4j日志记录

1、准备 log4j.properties 配置文件

2、引入log4j的依赖

<!-- logf4j日志 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-log4j</artifactId>

<version>1.3.8.RELEASE</version>

</dependency>

3、普通使用（直接在Controller类中引用）

```
private static final Logger logger = LoggerFactory.getLogger(当前Controller类.class);
```

在方法中

```
logger.info(".....要输出的日志信息.....");
```

```
@Controller
public class JspController {
    private static final Logger logger = LoggerFactory.getLogger(JspController.class);

    @RequestMapping("/jspIndex")
    public String jspIndex() {
        logger.info("springboot 集成logger 日志成功!!!");
        return "jspIndex";
    }
}
```

普通方式基础日志

## 三、使用AOP统一处理Web请求日志

1、引入AOP依赖

<!--aop整合logf4j -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-aop</artifactId>

</dependency>

2、创建一个WebLog的日志类，详细见附件 [WebLogAspect.java](#)





