

















文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图 福利浏览器 保存到云笔记

粘贴 指向选区 复制

宋体 中文字体 - 三号 - A+ A- Aa 字体

段落

导航

搜索文档

标题 页面 结果

- 一、Spring介绍
  - 1.1. Spring简介
  - 1.2. 系统要求
- 二、快速入门
  - 2.1. 创建一个Maven工程
  - 2.2. pom文件引入依赖
  - 2.3. 编写HelloWorld服务
  - 2.4. @RestController
  - 2.5. @EnableAutoConfiguration...
  - 2.6 SpringApplication.run方法...
  - 2.7. SpringBoot启动方式1
  - 2.8. SpringBoot启动方式2
- 三、Web开发
  - 3.1. 静态资源访问
  - 3.2. 全局异常处理
    - 3.2.1. 演示全局异常
    - 3.2.2. 演示Web页面
  - 3.4. 使用Freemarker模板引擎
    - 3.4.1. pom文件引入...
    - 3.4.2. 后台代码
    - 3.4.3. 前台代码
    - 3.4.4. Freemarker其他用法
    - 3.4.5. Freemarker配置
  - 3.5. 使用Thymeleaf和JSTL
    - 3.5.1. pom文件引入以下依赖
    - 3.5.2. 在application.properties中添加
    - 3.5.3. 后台代码
- 四、数据访问
  - 4.1. springboot整合使用MyBatis
  - 4.2. springboot整合使用Redis
  - 4.3. springboot整合使用SpringCloud
  - 4.4. springboot整合多数据源
- 五、事件总线
  - 5.1. SpringBoot整合事物管理
  - 5.2. SpringBoot分布式事物...

第4章\_共29页\_6/4035个字\_第1页\_中文(简)

**3.3、渲染 Web 页面**

```

@ControllerAdvice
public class GlobalExceptionHandler {
    @ExceptionHandler(RuntimeException.class)
    @ResponseBody
    public Map<String, Object> exceptionHandler() {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put("errorCode", "101");
        map.put("errorMsg", "系统错误!");
        return map;
    }
}

```

