山东大学 计算机科学与技术 学院

现代软件开发技术 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：2020001300 | 姓名： | | 班级： |
| 实验题目：自定义Starter，Spring Boot web\_Thymeleaf模板 | | | |
| 实验学时：2学时 | | 实验日期： 2023-10-24 | |
| 实验目的：  1.能够自定义和使用 Starter  2.能够使用Spring Boot web前端模板Thymeleaf的语法、属性表达视图网页。  声明、引入div片段；使用$(#内置对象...)操作数据；表单数据绑定 等 | | | |
| 硬件环境： | | | |
| 软件环境：  操作系统：windows 10  开发平台与工具：IDEA开发平台、jdk-11.0.1、apache-tomcat-9.0.37 | | | |
| 实验内容与问题回答：  **1.1 自定义Starter，**  **打开项目spring\_boot\_mystarters，分析其中的代码，**  **Q: pom.xml中增加的依赖？**  **Q: MyProperties 与 MyService的关系？**  **Q: MyAutoConfiguration中的执行逻辑？**  **Q: MyAutoConfiguration是如何注册的？**   1. **pom.xml中增加的依赖有以下几个：**          1. **MyProperties 与 MyService的关系**   MyProperties 类负责读取应用程序配置属性，并提供了 msg 属性，其值可以从应用程序的配置文件中获取。MyService 类包含了一个 msg 属性，它通过 setMsg(String msg) 方法可以被设置为来自 MyProperties 类的 msg 属性值。然后，sayMsg() 方法可以使用这个属性值返回包含 "my " 前缀的字符串。   1. **MyAutoConfiguration中的执行逻辑**   MyAutoConfiguration是一个自动配置类，它用于根据条件自动配置 MyService 的 Bean，并将属性值从 MyProperties 中注入到 MyService 中。以下是该代码的执行逻辑：@Configuration 注解标志着这是一个配置类，用于配置 Spring 容器中的 Bean。@EnableConfigurationProperties(MyProperties.class) 注解用于开启属性配置类 MyProperties，允许通过属性配置文件读取属性值。@ConditionalOnClass(MyService.class) 注解表示只有当类加载器中存在 MyService 类时，这个自动配置类才会生效。@ConditionalOnProperty(prefix = "my", value = "enabled", matchIfMissing = true) 注解表示只有当应用环境中的属性 my.enabled 存在，并且其值为 "enabled"，或者如果该属性不存在时，自动配置才会生效。在 MyAutoConfiguration 类中，通过 @Autowired 注解注入了 MyProperties 对象，用于获取属性值。   1. **MyAutoConfiguration是如何注册的**   在src/main/resources中新建META-INF文件夹，其中新建spring.factories文件，注册MyAutoConfiguration。在spring.factories文件中写入一下语句：  org.springframework.boot.autoconfigure.EnableAutoConfiguration=\  com.ch.spring\_boot\_mystarters.MyAutoConfiguration  **1.2.使用自定义的starters，ch4\_2**  **打开、分析ch4\_2 代码**  **Q：pom.xml文件添加的依赖？**  **Q：解释启动类Ch42Application的运行结果，**  **运行项目后，测试：http://localhost:8080/testStarters**  **修改apllication.properties文件，其中存在my.msg=XXX 与否对程序运行结果有什么影响？**   1. **pom.xml文件中添加的依赖如下：**     除了spring-boot-starter-web和spring-boot-starter-test两个依赖以外还添加了问题1.1中的spring\_boot\_mystarters项目。  运行项目后，测试：    修改application.properties文件，注释掉my.msg = starter pom之后，重新运行项目，结果如下图：    **-------------------------------------------**  **2. 参照Spring Boot 从入门到实战实验指导书： 实验六 自定义条件与Starters**  **对应项目代码shiyan6\_1、shiyan6\_2、spring\_boot\_addstarters。**  **3. Spring Boot的Web（包含Thymeleaf模板引擎）项目的建立**  **用STS4打开课本项目ch5\_1，分析代码，**  **Q:该项目与之前的项目，如ch2中的web项目和ch4\_2有什么不同？如：**  **项目结构；**  **pom.xml文件中依赖；**  **src/main/java下的代码文件；**  **src/main/resources/下的各种文件；**  **Maven Dependencies下的依赖包；**  **...等。**   1. **ch5\_1的pom.xml文件中增加了spring-boot-starter-thymeleaf这个依赖。**      1. **ch5\_1的src/main/java文件夹下多了controller层和model层。**      1. **ch5\_1的src/main/resources文件夹下多了i18n文件夹，里面多了一些.properties配置文件；**   **并且多了templates文件夹，该文件夹下多了一些.html的视图文件。**  **4.练习前端模板Thymeleaf的语法**  **打开ch5\_1项目src/main/resources/templates下的各种视图文件和com.ch.ch5\_1.controller包中的java文件，分析其中的代码。**  **学习前端模板Thymeleaf的基础语法。**  **（1）练习声明、引入div片段**  **分析src/main/resources/templates下的index.html和footer.html视图文件**  **Q: footer.html中声明了哪两个div片段？**  **Q: index.html中怎样引入了上面两个fragment?**  **运行com.ch.ch5\_1包中的 Ch51Application.java，run as application。**  **在浏览器地址栏输入 http://localhost:8080/ch5\_1/ 测试默认主页index.html**   1. footer.html中声明了div片段有这两个：   <div th:fragment="content">：这是一个名为 "content" 的 Thymeleaf 片段，用于表示页面的主体内容。  <div th:fragment="copy">：这是一个名为 "copy" 的 Thymeleaf 片段，用于表示版权信息，其中包含了 "©清华大学出版社" 的文本内容。   1. index.html使用了 Thymeleaf 的属性来引入上一个问题中的两个片段（fragment）：   <div th:include="footer::content"></div>：这行代码使用了 th:include 属性，它告诉 Thymeleaf 去引入名为 "content" 的片段，而这个片段来自一个名为 "footer" 的片段定义。这会将 "content" 片段的内容插入到当前页面的这个 <div> 元素中。  <div th:replace="footer::copy"></div>：这行代码使用了 th:replace 属性，它告诉 Thymeleaf 去引入名为 "copy" 的片段，同样这个片段来自一个名为 "footer" 的片段定义。不同于 th:include，th:replace 会完全替换当前 <div> 元素的内容为 "copy" 片段的内容。  这样，页面中的两个片段被引入并插入到当前页面中的不同 <div> 元素中，实现了内容的组合和替换。  运行com.ch.ch5\_1包中的 Ch51Application.java，run as application。  在浏览器地址栏输入 http://localhost:8080/ch5\_1/    **（2）练习使用$(#内置对象...)操作数据**  **分析com.ch.ch5\_1.controller包中的TestThymeleafController.java和src/main/resources/templates下showObject.html**  **Q:TestThymeleafController.java中@RequestMapping("/testObject")注解下的方法中，创建并向model中添加了哪些属性变量？**   1. nowTime：当前系统时间，类型为 Date。 2. nowCalendar：当前系统日历对象，类型为 Calendar。 3. myMoney：一个 BigDecimal 对象，值为 2019.613。 4. str：一个字符串，值为 "Test strings"。 5. bool：一个布尔值，值为 false。 6. mya：一个整数数组，包含元素 1, 2, 3, 4, 5。   **Q:showObject.html中使用了哪些内置对象的方法？其中哪个对象方法执行了排序、求平均值、求和？**  **测试showObject.html，http://localhost:8080/ch5\_1/testObject**  **showObject.html中使用了以下内置对象方法：**  #dates.format(nowTime, 'yyyy/MM/dd')：用于格式化日期对象的日期。  #dates.create(2019,6,13)：用于创建日期对象。  #calendars.format(nowCalendar, 'yyyy-MM-dd')：用于格式化日历对象的日期。  #numbers.formatInteger(myMoney,3)：用于格式化 BigDecimal 对象的整数。  #strings.length(str)：用于计算字符串的长度。  #objects.nullSafe(myMoney, 9999)：用于返回对象，如果传递的 BigDecimal 对象为空时，返回默认值 9999。  #bools.isFalse(bool)：用于判断布尔数据是否为 false。  #arrays.contains(mya, 5)：用于判断数组是否包含指定元素。  **执行了排序、求平均值、求和的对象方法如下：**  #lists.sort(myList1)：该方法对列表 myList1 中的数据执行排序操作。  #aggregates.sum(mya)：该方法用于将数组 mya 中的元素执行求和操作。  #aggregates.avg(mya)：该方法用于将数组 mya 中的元素执行求平均值操作。  **测试showObject.html，：**    **Q:显示页面最后一行，为何显示??msgKey\_zh\_CN?? ？**  在上述Thymeleaf模板中，<span th:text="${#messages.msg('msgKey')}"></span> 试图获取一个名为 msgKey 的消息，但未提供消息的具体内容。默认情决定显示一个默认的占位符，即 "??msgKey\_zh\_CN??"。这是因为没有为 msgKey 提供具体的消息文本。  **Q:将showObject.html中${#messages.msg('msgKey')中的msgKey改为chinese.key,重新运行Ch51Application.java，**  **再次测试，http://localhost:8080/ch5\_1/testObject，显示什么？**    再次运行，显示结果如上图所示。  **（3）练习表单提交及数据绑定的实现**  **分析com.ch.ch5\_1.model包中的LoginBean和com.ch.ch5\_1.controller包中的LoginController.java文件，**  **Q:LoginBean.java文件中两个属性是？**  uname urole：  **Q:LoginController.java中，**  **@RequestMapping("/toLogin")注解下的方法，向model中添加的变量是？**  @RequestMapping("/toLogin") 注解下的方法 toLogin 向 model 中添加的变量是一个名为 "loginBean" 的对象，其类型为 LoginBean。  **Q：分析src/main/resources/templates下的login.html文件，其中如何实现了表单与loginBean对象实例的绑定？**  **当点击“submit”按钮后，表单发出了什么URL请求？**  表单与 loginBean 对象实例的绑定是通过 Thymeleaf 的 th:field 属性实现的。在表单中的输入字段（例如，用户名和角色）上使用 th:field 属性，它的值应该是 "\*{属性名}"，其中属性名应该与 loginBean 对象的属性名相对应。这样，Thymeleaf 将能够将表单字段的值与 loginBean 对象的属性进行绑定。  **Q: 分析src/main/resources/templates下的result.html文件，其中如何显示loginBean对象的属性的？**  **测试：http://localhost:8080/ch5\_1/toLogin**  **http://localhost:8080/ch5\_1/testValidator 测试testValidator.html**  在浏览器中输入**<http://localhost:8080/ch5_1/toLogin>，结果如下图所示：**    在浏览器中输入[**http://localhost:8080/ch5\_1/testValidator**](http://localhost:8080/ch5_1/testValidator)**，结果如下图所示：** | | | |
| 思考题解答：  **1.SpringBoot的web项目与之前的SpringMVC项目相比有什么不同？**   1. 自动配置：Spring Boot的Web项目通过自动配置机制简化了项目的配置过程。它会根据应用程序的依赖和环境自动配置Web相关的组件，如Servlet容器、DispatcherServlet、静态资源处理等。您无需手动配置这些组件，可以快速搭建一个可运行的Web应用程序。 2. 内嵌服务器：Spring Boot默认使用内嵌的Servlet容器，如Tomcat、Jetty或Undertow。这意味着您无需手动部署WAR文件到外部的Servlet容器，而是直接通过运行Spring Boot应用程序来启动嵌入的服务器。这简化了部署和运行的过程。 3. 自动扫描：Spring Boot会自动扫描应用程序中特定的包，例如主应用程序类所在的包及其子包。这使得您无需显式配置组件扫描，Spring Boot会自动将带有特定注解的类（如@Controller、@Service、@Repository等）识别为Spring组件。 4. Starter依赖：Spring Boot提供了一系列的Starter依赖，这些依赖简化了对常见功能的引入和配置。例如，Spring Boot提供了spring-boot-starter-web依赖，它包含了Web开发所需的常用库和配置。您只需在项目中引入相应的Starter依赖，就能快速启动和配置相关功能。   **2.SpringBoot的前端模板Thymeleaf提供哪些功能？**  1. 表达式语言：Thymeleaf提供了强大的表达式语言（Expression Language，简称EL）支持，可以在模板中使用表达式来访问和操作后端数据。这使得在模板中动态地呈现数据变得非常方便。  2. 模板布局：Thymeleaf支持模板布局，可以将页面的不同部分（如页眉、页脚、导航栏等）定义为可重用的模板片段。通过使用Thymeleaf的布局功能，可以更加灵活地管理和组织页面结构。  3. 条件判断和循环：Thymeleaf提供了条件判断和循环功能，可以根据特定条件显示或隐藏页面的部分内容，或者对集合进行迭代和展示。  4. 表单处理：Thymeleaf提供了方便的表单处理功能，可以简化表单数据的绑定、验证和提交。它支持在模板中使用表达式来动态生成表单字段，并提供了验证和错误处理的支持。  5. 国际化和本地化：Thymeleaf提供了国际化和本地化的支持，可以根据不同的语言和地区显示不同的文本、日期和数字格式等。 | | | |
|  | | | |