

课程汇报

课题：Spring 框架的实践

实践平台：LAMP；Apache+SpringMVC+MySql

项目总数：1+11

Linux+Apache+Mysql+php:

环境配置完毕，由于时间限制，对 php 语言仍未熟练掌握，使用了 discuz 的平台，建立了一个基础论坛，但是基本未写代码，实践意义不如 SpringMVC 大，只能算是会使用，因此不将代码上传，只上传配置环境成功截图。

SpringMVC:

使用 Eclipse，通过 jar 包建立与数据库的连接，通过 apache 服务器配合程序实现简单功能。

1 test

准备条件：MySql-connector-java-5.1.38-bin.jar

测试目的：测试与本地数据库的连接是否正常

测试过程：安装 MySql5.7 程序；在 MySql 的 workbench 中手动建立一个 student 数据库；通过 jar 尝试访问数据库内容，若成功访问则在控制台打印完成，若出错则表示连接有问题

原理：新建链接 url，使用用户名和密码请求服务器和数据库名，获取对其中的表格的访问和修改权

问题：MySql 服务启动出错

解决方案：在 MySql 的 Notifier 中点击 Managed Monitored Items 手动删除已经建立的 Services 和 instances，再手动添加本地 sql 服务，重启即可。

2 TestTomcat1

准备条件：Apache7.0

测试目的：测试与本地服务器的连接是否正常

测试过程：安装 Apache7.0 服务；新建 Dynamic Web Project；在 WebContent 目录下新建一个 jsp；将项目部署至服务器；开启服务器；通过网页进行访问，若访问成功则在网页中显示字符串，若出错则表示连接有问题

原理：从本地服务器访问文件

问题：书写错误

解决方法：确认书写

3 SpringMVC1

准备条件：spring-framework-4.2.5.RELEASE-dist 的 jar 包，commons-logging.jar，jstl.jar

测试目的：测试 Apache 服务器和 SpringMVC 配合是否能正常传输数据

测试过程：配置 Spring.xml 和 web.xml；新建数据组；传递到服务器；服务器将数据打印在 jsp 页面，若成功则显示数据，若出错则表示代码有问题

原理：web.xml 中设置让 springMVC 操控；springMVC 设置操控的文件路径，处理方法；Controller 设置接收到信号后调用的文件名；关键文件中将数据打印至页面

问题：catalina 错误

解决方案：增加两个 jar 包

#### 4 SpringMVC

准备条件: spring-framework-4.2.5.RELEASE-dist 的 jar 包, commons-logging.jar, jstl.jar

测试目的: 测试 SpringMVC 能否处理多种类的事件

测试过程: 在 handler 中增加一个监听器

原理: 同 SpringMVC1 (3)

#### 5 SpringMVC-1

准备条件: spring-framework-4.2.5.RELEASE-dist 的 jar 包, commons-logging.jar, jstl.jar, jquery-1.12.3.min.js

测试目的: 使用 SpringMVC 读取静态文件

测试过程: 修改 springAnnotation-servlet.xml; 使用 src 方法请求文件(js, img 文件); SpringMVC

对特定请求设计相应方式; 资源显示在页面上

原理: 使用相对路径访问静态文件

问题: 资源无法正常显示

解决方案: 修改资源路径

#### 6 SpringMVC-2

准备条件: spring-framework-4.2.5.RELEASE-dist 的 jar 包, commons-logging.jar, jstl.jar, jquery-1.12.3.min.js

测试目的: SpringMVC 使用注解访问

测试过程: 使用注解扫描包; 修改 springAnnotation-servlet.xml, 开启注释; 将调用的方法名打印在页面上

原理: SpringMVC 支持注解, 使用注解对数据进行监听和访问

问题: 注释格式错误

解决方案: 查询注释格式

#### 7 SpringMVC-3

准备条件: spring-framework-4.2.5.RELEASE-dist 的 jar 包, commons-logging.jar, jstl.jar, jquery-1.12.3.min.js

测试目的: SpringMVC 注解优化

测试过程: 优化注解, UserController2.java

原理: 同 SpringMVC-2

问题: 同 SpringMVC-2

解决方案: 同 SpringMVC-2

#### 8 SpringMVC-4

准备条件: spring-framework-4.2.5.RELEASE-dist 的 jar 包, commons-logging.jar, jstl.jar, jquery-1.12.3.min.js

测试目的: SpringMVC 传递参数

测试过程: 网页获取用户名和年龄; Apache 将数据传递至 SpringMVC; SpringMVC 将数据返回到页面并显示

原理: request.setAttribute

问题: 乱码

解决方案: 1 修改 Apache 默认编码方式为 UTF-8 2 强制转码 (此 project 使用后者)

## 9 SpringMVC-5

准备条件：spring-framework-4.2.5.RELEASE-dist 的 jar 包，commons-logging.jar，jstl.jar，jquery-1.12.3.min.js

测试目的：SpringMVC 使用 json 传递参数

测试过程：创建 User 对象，设置 set，get 方法；页面通过 ajax 接收数据，传递到 SpringMVC；通过对数据的拆分重组，营造出 json 类型数据，并返回给页面

原理：数据组合

问题：乱码

解决方案：强制转码

## 10 SpringMVC-6

准备条件：spring-framework-4.2.5.RELEASE-dist 的 jar 包，commons-logging.jar，jstl.jar，jquery-1.12.3.min.js，com.springsource.org.apache.commons.fileupload-1.2.0.jar，com.springsource.org.apache.commons.io-1.4.0.jar

测试目的：SpringMVC 传输文件

测试过程：选择文件，使用 java 自带文件传输流方法将文件写到指定位置

原理：文件流的操作

问题：传输数据慢

解决方案：SpringMVC-7

## 11 SpringMVC-7

准备条件：spring-framework-4.2.5.RELEASE-dist 的 jar 包，commons-logging.jar，jstl.jar，jquery-1.12.3.min.js，com.springsource.org.apache.commons.fileupload-1.2.0.jar，com.springsource.org.apache.commons.io-1.4.0.jar

测试目的：SpringMVC 快速传输文件

测试过程：新建 interface，使用 SpringMVC 自带的文件传输方法将文件写到指定的位置

原理：Spring 文件操作

问题：暂无

其他问题和理解：

模型-视图-控制器（MVC）是一个众所周知的以设计界面应用程序为基础的设计模式。它主要通过分离模型、视图及控制器在应用程序中的角色将业务逻辑从界面中解耦。通常，模型负责封装应用程序数据在视图层展示。视图仅仅只是展示这些数据，不包含任何业务逻辑。控制器负责接收来自用户的请求，并调用后台服务（manager 或者 dao）来处理业务逻辑。处理后，后台业务层可能会返回了一些数据在视图层展示。控制器收集这些数据及准备模型在视图层展示。MVC 模式的核心思想是将业务逻辑从界面中分离出来，允许它们单独改变而不会相互影响。SpringMVC 和 Struts2 十分类似，两者可以共同合作让项目开发更加轻松，但是 SpringMVC 比 Struts2 更加容易上手，而且更为简洁，在一般场合完全可以取代 struts2。

Spring 是一个开源框架，Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架，由 Rod Johnson 在其著作 Expert One-On-One J2EE Development and Design 中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许使用者选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成

的框架。**Spring** 使用基本的 **JavaBean** 来完成以前只可能由 **EJB** 完成的事情。然而，**Spring** 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 **Java** 应用都可以从 **Spring** 中受益。**Spring** 的核心是控制反转（**IoC**）和面向切面（**AOP**）。简单来说，**Spring** 是一个分层的 **JavaSE/EEfull-stack**(一站式) 轻量级开源框架。

**Spring** 注解将给工作带来些轻松，利用反射机制原理出现的注解能够解决配置大量的配置问题。

模块之间必定会有这样或那样的依赖关系，不要幻想消除所有依赖。但是，过强的耦合关系（如一个模块的变化会造成一个或多个其他模块也同时发生变化的依赖关系）会对软件系统的质量造成很大的危害。特别是当需求发生变化时，代码的维护成本将非常高。所以，我们必须想尽办法来控制 and 消解不必要的耦合，特别是那种会导致其它模块发生不可控变化的依赖关系。依赖倒置、控制反转、依赖注入等原则就是因此不断产生和发展起来的。

查阅资料得，**Spring** 有如下优点：

- 1.低侵入式设计，代码污染极低
  - 2.独立于各种应用服务器，基于 **Spring** 框架的应用，可以真正实现 **Write Once,Run Anywhere** 的承诺
  - 3.**Spring** 的 **DI** 机制降低了业务对象替换的复杂性，提高了组件之间的解耦
  - 4.**Spring** 的 **AOP** 支持允许将一些通用任务如安全、事务、日志等进行集中式管理，从而提供了更好的复用
  - 5.**Spring** 的 **ORM** 和 **DAO** 提供了与第三方持久层框架的良好整合，并简化了底层的数据库访问
  - 6.**Spring** 并不强制应用完全依赖于 **Spring**，开发者可自由选用 **Spring** 框架的部分或全部
- 这些优点在本次大量的工程中有非常明显的体现，联系去年的课程项目，加深了我对于 **Spring** 框架的理解，同时也能渐渐感受到他带来的改变。