

基于 SpringBoot 的 Web 快速开发框架

喻佳, 吴丹新

(华东交通大学信息工程学院, 南昌 330013)

摘要: 基于 J2EE 的 Web 框架和 J2EE 核心模式的基础之上, 提出了 SpringBoot 为基础的 Web 快速开发框架——RDWFBS (Rapid Development Web Framework based on SpringBoot)。使用 SpringBoot 集成框架, 围绕 Web 进行架构, 框架采用了表现层、业务逻辑层、持久化层三层体系。在表现层中和业务逻辑层使用了 SpringMVC 和 Spring 的开发框架; 在持久化层中, 集成了 Spring Data JPA 框架。另外, 对于用户接口的开发, 引入了 Thymeleaf 框架。同时, 具体介绍框架的整合和具体实现。利用这一框架实现一个博客应用系统, 系统的运行体现改进型框架的正确性和有效性。

关键词: Web 框架; 构件技术; SpringBoot 框架; 博客系统

DOI:10.16184/j.cnki.comprg.2021.09.013

1 概述

提出了基于 SpringBoot 的 Web 快速开发框架 (以下简称 RDWFBS), 通过集成各种优秀、开源的 J2EE Web 框架, 不仅实现了各层的松散耦合, 还可创建独立的应用程序, 同时大量减少构建配置。另外, 通过添加 Thymeleaf 可以轻松地与 Spring MVC 等 Web 框架进行集成, 作为 Web 应用的模板引擎, 极大地解决了前、后端开发不一致的问题, 使得前、后端分离的开发形式成为可能。

2 基于 SpringBoot 的 Web 快速开发框架

RDWFBS 整体结构图如图 1 所示。

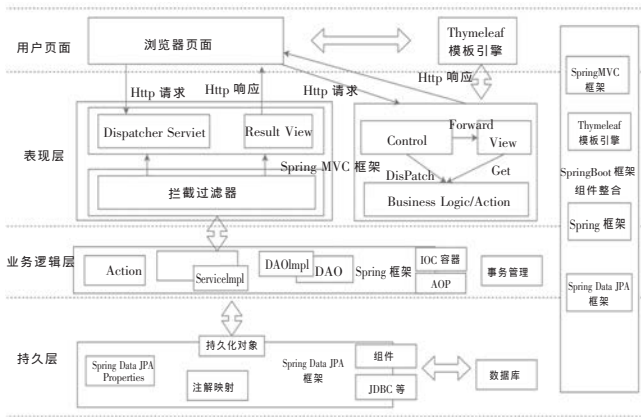


图 1 框架整体结构图

2.1 用户页面

为了方便后端逻辑的独立开发, 在用户页面的开发上嵌入了 Thymeleaf 的模板引擎。Thymeleaf 是 Web 和独立环境的现代服务器端 Java 模板引擎。

Thymeleaf 提供一种可维护的创建模板的方式。为实现这一点, 它建立在自然模板的概念上, 将其逻辑注入到模板文件中, 不会影响模板被用作设计原型。这改

善了设计的沟通, 弥补了设计和开发团队之间的差距。而它从一开始就设计了 Web 标准特别是 HTML5 允许创建完全验证的模板。

2.2 RDWFBS

为了避免繁多的配置和复杂的部署流程以及降低第三方技术集成难度, 使用 SpringBoot 作为 RDWFBS 的基石, 简化 Spring 应用初始搭建以及开发过程。由于 SpringBoot 使用简化配置的理念能够帮助项目快速运行起来。使用 RDWFBS 就会非常容易创建一个独立运行准生产级别的框架项目, 并且使用 RDWFBS 你可以不用或者只需要很少的 Spring 配置, 在组件的搭配上已经趋于固定, 无需考虑不同组件之间的冲突问题。所以, 相对于 SpringBoot 来说更加能够提高开发效率。

RDWFBS 相对于其他以往的一些开发框架来说, RDWFBS 集成了它们大部分的优点, 而且提高了效率。另外, RDWFBS 还有良好的拓展性, 对于目前流行的框架组件都能够兼容。RDWFBS 不但使用简单, 而且功能更加丰富, 性能更加稳定而健壮。使用 RDWFBS 开发框架, 不仅能提高开发速度, 增强生产效率, 一定意义上, 可以说是解放了程序员的劳动, 一种新技术的使用, 更能增强系统的稳定性和扩展系统的性能指标。

2.3 表现层

框架的表现层能够响应用户的请求, 并且发送相应数据给后台, 使用服务将对数据持久化等操作, 最后将执行结果返回用户页面。在框架的表现层中包含一个 SpringMVC 框架中, 它的功能实现主要依赖于前端控制

基金项目: 江西省教育厅科学技术研究项目 (GJJ190316)。

作者简介: 喻佳 (1981-), 女, 副教授, 研究方向: Web 应用开发、数据库技术等。



器,直接接收用户的一切服务请求,再通过请求的形式判断调用的处理器类型。

另外,在 RDWFBS 中,还继承了 Servlet 组件技术,可以担任拦截过滤器和 Web 页面间消息转发器的角色。因此,在表现层还有另一个作用,对用户的请求到达控制层或者视图之前进行一些额外的操作,一般有预处理和后处理的逻辑,例如权限控制、字符编码的转换等。此外,拦截过滤器在运行时根据不同的请求进行动态调用,实现控制的集中化。

2.4 业务逻辑层

业务逻辑层在该体系架构中处于数据访问层与表示层中,包含领域对象模型、领域实体、业务规则、验证规则和业务流程。业务逻辑层起了数据交换中承上启下的作用。由于层是一种弱耦合结构,层与层之间的依赖是向下的,底层对于上层而言是“无知”的,改变上层的设计对于其调用的底层而言没有任何影响。所以,可以使得应用程序的各部分和业务逻辑的各个层面之间相互独立。如果在分层设计时,遵循了面向接口设计的思想,那么这种向下依赖也应该是一种弱依赖关系。

2.5 持久层

持久层的工作就是从数据库中取出或者存入数据,并且将关系表中的数据映射到持久化的对象中去,一般包括一个实体模型。在目前一般使用到的是 JDBC 等技术,虽说可以对数据库进行持久化操作,但其不具备良好的扩展性,反复的进行数据库连接资源的获取和关闭,及批量操作频繁与数据库的交互,都容易造成效率的下降。此外,对结果集的解析繁琐,这种方式没有做到面向对象编程,使开发者的思考仍然停留在 SQL 语句上。

基于以上的分析,选择使用 Spring Data JPA 作为数据持久层的框架。从框架的设计理念上来看, Spring Data JPA 是一种面向对象,它对 JDBC 进行了非常轻量级的对象封装,它将 POJO 与数据库表建立映射关系,是一个全自动的 ORM 框架,并且可以自动生成 SQL 语句,自动执行,另外, Spring Data JPA 还能够方便开发者去关注代码本身,而不用关注数据库,通过操作映射表的实体对象,此外,使用极简的代码实现对数据库的访问和操作一定程度上简化了开发,提高了开发效率。

3 应用实例

为了实现人们资源共享的需求,及网络的人际交流,以个人为中心的信息摘选和知识管理和以个人为中心的传播出版资源库。为实现网络交流与信息交换,数

据传播,进行了博客系统的开发,而此系统采用了 RD-

WFBS 进行软件的架构设计。

经过研究将整个系统在逻辑上分为 3 个层次:业务逻辑层、表现层和持久层。持久层主要是对应映射的 POJO 将数据持久化到数据库。业务层主要是针对用户所发送的请求,对数据进行处理和加工,最后返回到页面,显示给用户页面。而表现层则是通过路径接收用户的请求及将实例化的对象返回到页面,将页面渲染。

此外,在解决各种不同用户权限的验证上,使用了依赖于 SpringMVC 的拦截器,在每一次用户页面的请求处理之前,都会进入并且对用户身份进行鉴别,对于相应的用户则开启不同的权限。这样,大部分的用户操作都得接收权限检查。当然,这种权限检查,还需要两个方面的支撑,即“用户具有什么权限”和“当前用户需要什么权限”。解决这两个问题,在用户登录的时候,将用户所能访问的权限标记一同存入 Session 中去,在 Thymeleaf 的模板页面中就可以判断当前请求是否在用户的权限集合中了,反之,则提交请求进行业务逻辑处理。

因为以往的项目构建,大量的 XML 文件存在于项目之中,导致 JavaEE 变得越来越笨重,同时降低了开发和部署效率。所以,框架选择以 Spring Boot 作为基础, Spring Boot 并不是用来替代 Spring,而是和 Spring 框架紧密结合以提升 Spring 开发者体验的工具。它集成了大量常用的第三方库配置, Spring Boot 应用中这些第三方库几乎可以是零配置的开箱即用,大部分的 Spring Boot 应用都只需要非常少的基于 Java 配置代码,从而使得开发者能专注于业务逻辑研究。

另外,框架中的 ORM 使用了 Spring Data JPA, SpringData 是用来简化基于 Spring 构建的应用程序的持久层框架的框架。可以理解为简化了持久层方案的方案。SpringData 有很多子模块,其中 SpringDataJPA 就是其中一个很不错的子模块。简化了开发者的持久层开发工作,持久层不用去书写具体实现代码,直接写一个继承 Repository 接口的 Dao 接口,随后开启 JPA 的注解扫描就可以进行持久层开发。当然, SpringDataJPA 也提供了一些常用的 ACID 抽象方法,开发者可以直接拿来使用,在这些抽象方法不够使用的时候,还提供了属性表达式、命名查询、Query 注解等方式的使用。Spring-DataJPA 简化业务处理,通过 Repository 接口实现了 PagingAndSortingRepository 接口,开发者只是将相关数据封装到 PagingAndSortingRepository 接口中的 findAll() 方法中,并自动返回 Page 对象,然后针对返回的 Page 对象根据需求自动解析并封装成不同需要的 JSON 数

据。这样看来 Spring Data JPA 框架就是一个提高开发人员效率的优秀框架。

Thymeleaf 框架则是前端模板, Thymeleaf 是为了方便前端和后端的独立开发, 基于它写的模版文件可以直接在浏览器打开, 效果接近高保真原型。但 Thymeleaf 又提供了类似与 JSP include 的用法, 使用模版片段的方式可以简化模版代码。所以有利于前后端 UI 组件化的合作方式, 对于一些无法组件化的 UI 效果, 可以前端先做 HTML 页面, 后台在其 HTML 标签上添加 Thymeleaf 标签属性填充动态数据。

RDWFBS 是以简化配置和方便解耦的思想渐长的, 对于开发人员提高效率来说无疑是一个非常好用的框架。

另外, RDWFBS 设计的博客系统已在服务器上稳定运行, 目前符合设计性能达到的预期。

4 结语

在实际应用的开发过程中, 使用 RDWFBS 能够针对 J2EE 的 Web 架构进行良好的支撑, 并且对于每一个

(上接第 16 页)

颜色也随着变换, 如图 4 所示。

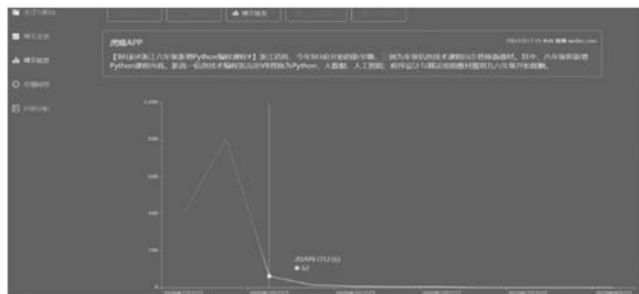


图 4 每日博文转发数量图

5 结语

近年来, 自然语言处理处于快速发展阶段。互联网与移动互联网和世界经济一体化的潮流对自然语言处理技术的迫切需求, 为自然语言处理研究发展提供了强大的市场动力。

系统根据微博的用户的相关博客内容, 提取关键词, 分析情感倾向, 主题聚类等, 对微博上数据的分析和可视化。通过主题分析和情感分析, 对微博用户行为数据进行分析, 有效把控舆情, 营造出和谐的网络社会环境。系统为 snownlp 对博文进行情感分析网民情绪, 可以看出微博用户对事件持的态度。通过微博内容分析功能判断舆情是消极还是积极, 通过传播网络路径功能

工作层次都嵌入了优异的组件, 很好地实现了系统中多层次组件的交互。随之从系统的拓展性、系统的性能和移植性的角度来评价此框架的特点。拓展性, 系统的开发是采用了多层次的结构设计, 每个层次都相互协调, 但又互不影响。系统的性能, RDWFBS 对于不同的组件集成都有良好的支持效果, 在性能上对于其他框架来说也是优化不少。移植性, 基于 Java 语言的系统就能实现平台无关, 可以在不同的系统或者平台运行。

参考文献

- [1] 曾亮, 齐欢, 王小平, 陈迎春. 基于 J2EE 核心模式的组合 Web 框架研究 [J]. 华中科技大学学报 (自然科学版), 2007, (06): 43-46.
- [2] 张峰. 应用 SpringBoot 改变 Web 应用开发模式 [J]. 科技创新与应用, 2017, (23): 193-194.
- [3] 刘志鹏, 卫晨. Spring Data 在 JavaEE 系统中的应用与研究 [J]. 电子世界, 2014, (16): 278.

对消极的舆情进行有效控制。

参考文献

- [1] 丁忠祥, 杨彦红, 杜彦明. 基于 Scrapy 框架影视信息爬取的设计与实现 [C]. 北京: 北京印刷学院, 2018, (9).
- [2] 郭京飞, 王瞰. 新冠肺炎网络舆情的生成机理与路径引导 [J]. 贵州大学学报, 2020, 38 (5): 18-19
- [3] 梁山清, 李恩宁, 王学萍, 等. 基于 Python 的企业员工舆情分析 [J]. 计算机与网络, 2021, (5): 42
- [4] 盖赞, 宿培成. 基于机器学习的网络舆情分析 [J]. 科学智囊, 2020, (9): 69
- [5] 滕靖, 刘韶杰, 龚越, 等. 交通事件网络舆情分析方法 [J]. 交通信息与安全, 2019, (6): 142
- [6] 汪明亮. 基于微博数据的网络舆情分析 [D]. 北京: 首都经济贸易大学, 2017.
- [7] 王高飞, 李润之, 李梅. 微博网络舆情研究综述 [J]. 今传媒 (学术版), 2021, (1): 28.
- [8] 赵志升, 靳晓松, 温童童. 基于 Python-Snownlp 的新闻评论数据分析 [J]. 科技传播, 2018, (18): 105.
- [9] 唐琳, 董依萌, 何天宇. 基于 Python 的网络爬虫技术的关键性问题探索 [J]. 电子世界, 2018, (14): 32.
- [10] 霍柄良. 基于 Python 的网络爬虫技术探析 [J]. 数码世界, 2020, (4): 73.