程序员之家网站整体框架结构介绍

1. 前言：

网站整体使用前后端分离的思想，即前端只负责View和Controller层，也就是页面的展示以及数据的显示。后端只负责Model层，业务处理/数据的存取等，这样一来有利于以后的扩展，二来可以进行明确的分工，同时能够很好地进行维护（[详情](http://frontenddev.org/link/full-stack-development-with-nodejs-1.html)）

1. 总体架构：
2. 前端部分：

主体结构：

1. 前端框架：Vue.js
2. vue.js配套的UI框架：Element ui
3. 前后端数据交互：axios
4. 前端自动化构建工具：WebPack
5. 开发工具：WebStorm
6. 项目托管平台：GitHub

详细介绍：

1. Vue.js配合Element UI主要作为展示层，即MVC模式中的V
2. 使用Vue router作为控制层，即MVC模式中的C
3. 因为前后端进行了分离，所以前后端交互就会出现问题，可以使用axios进行前后端数据的请求与转发
4. 使用WebPack作为项目构建工具，负责前端项目的资源的模块化管理于打包处理
5. 后端部分：

主体结构：

1. 后端框架：Spring Boot + MyBatis
2. 数据库：MySql、Redis
3. 项目构建工具：gradle
4. 开发工具：IDEA
5. 项目托管平台：GitHub

详细介绍：

1. Spring Boot + MyBatis主要用于模型层，即MVC中的M（model），以及对数据库的访问操作。同时二者相互结合可以实现纯注解，无需配置文件，提高开发效率（[详情](http://www.ityouknow.com/springboot/2016/11/06/springboot(%E5%85%AD)-%E5%A6%82%E4%BD%95%E4%BC%98%E9%9B%85%E7%9A%84%E4%BD%BF%E7%94%A8mybatis.html)）。MyBatis再配合redis可以实现二级缓存，提高后台整体运行速度（[详情](http://www.jianshu.com/p/ac3ef835cc13)）。
2. MySql用于存储数据
3. Gradle作为项目的构建工具以及单元测试
4. 学习资源：
5. 前端：

* 理论：

1. <https://cn.vuejs.org/v2/guide/index.html>
2. <https://zhuanlan.zhihu.com/p/23134551>
3. <http://www.jianshu.com/p/76e6a3ae6ab7>
4. <http://www.imooc.com/article/17891>

* Node.js(vue.js需要node的一些基础):

1. <http://www.cnblogs.com/zhongweiv/p/nodejs.html>
2. <http://www.jianshu.com/p/729c4fe7f93c>
3. <http://www.jikexueyuan.com/path/nodejs/>
4. 《Node.js实战》学校图书馆有

* Vue.js：
  1. <https://www.qcloud.com/community/article/430630001490779316?fromSource=gwzcw.84867.84867.84867>
  2. <http://www.jianshu.com/c/edec9ac6d9a7>
  3. <https://www.bilibili.com/video/av12739476/?from=search&seid=16386538748131347346>
  4. <http://www.rxshc.com/59.html>
* Vue.cli：

1. <http://www.cnblogs.com/wisewrong/p/6255817.html>
2. <http://www.jianshu.com/p/2769efeaa10a>

* WebPack：

1. <https://segmentfault.com/a/1190000006178770>
2. <https://doc.webpack-china.org/concepts>

* Axios：

1. <http://coderlt.coding.me/2017/03/21/axios-api-md/>
2. <https://ykloveyxk.github.io/2017/02/25/axios%E5%85%A8%E6%94%BB%E7%95%A5/>

* Element UI：

1. <http://element-cn.eleme.io/#/zh-CN>
2. <https://www.codecasts.com/series/use-element-ui-tool-kit>

* 前端项目Demo：

1. <http://www.jianshu.com/p/d87d7140944e>
2. <https://github.com/kenberkeley/vue-demo>
3. <https://github.com/varHarrie/Dawn-Blossoms/issues/7>
4. <https://segmentfault.com/a/1190000009978596>
5. 前后端分离接口实现理论（RESTful API）：
6. <http://www.ruanyifeng.com/blog/2014/05/restful_api.html>
7. <http://blog.jobbole.com/112006/?utm_source=blog.jobbole.com&utm_medium=relatedPosts>
8. [https://tw.twincl.com/programming/\*641y](https://tw.twincl.com/programming/*641y)
9. 后端：

* Spring（学习spring boot 需要有spring的基础）：

1. <http://edu.51cto.com/course/1956.html>（后面与Hibernate整合就可以跳了）

* Spring Boot：

1. <http://www.ityouknow.com/spring-boot>
2. <https://segmentfault.com/a/1190000008539153>

* MyBatis：

1. <http://www.imooc.com/learn/154>
2. <http://www.imooc.com/learn/260>

* Redis：

1. <http://www.jianshu.com/p/7e22ad3a9061>
2. <https://code.ziqiangxuetang.com/redis/redis-intro.html>

* Gradle：

1. <http://www.imooc.com/learn/833>

* 前后端结合Demo：

1. <http://blog.csdn.net/dream_broken/article/details/77451639>
2. <https://segmentfault.com/a/1190000008968295>
3. <https://segmentfault.com/a/1190000007021883>
4. <https://github.com/boylegu/SpringBoot-vue>
5. 各类工具：

* GitHub：

1. <http://www.imooc.com/learn/390>

* IDEA、WebStorm：

1. <https://github.com/judasn/IntelliJ-IDEA-Tutorial>