### **团队项目——团队总结及‘站立’会议**

博客地址：https://blog.csdn.net/weixin\_42200027/article/details/90608103#\_122

* [团队成员介绍](https://mp.csdn.net/mdeditor/90608103" \l "_2)
  + [胡茜文的总结](https://mp.csdn.net/mdeditor/90608103" \l "_10)
  + [孙俊杰的总结](https://mp.csdn.net/mdeditor/90608103" \l "_35)
  + [任静的总结](https://mp.csdn.net/mdeditor/90608103" \l "_53)
  + [王诚昱的总结](https://mp.csdn.net/mdeditor/90608103" \l "_67)
  + [黄政杰的总结](https://mp.csdn.net/mdeditor/90608103" \l "_107)
* [“站立”会议](https://mp.csdn.net/mdeditor/90608103" \l "_122)
  + [会议照片](https://mp.csdn.net/mdeditor/90608103" \l "_123)
  + [会议内容](https://mp.csdn.net/mdeditor/90608103" \l "_131)

# **团队成员介绍**

\*\*胡茜文（组长）

孙俊杰

任静

王诚昱

黄政杰\*\*

## **胡茜文的总结**

总结按照以下几方面进行介绍，1、做了什么，2、自己认为做的怎么样，3、最终的总结

做了什么：

1. 管理员页面的增删改
2. 管理员页面用户的搜索、分页
3. 提出页面修改意见
4. 部分地方的优化
5. 协助同学进行代码编写
6. 统一整理项目文档
7. 编写团队博客

总结：

1. 在写管理员页面的增删改过程中，能够快速地编写代码，但在网上听实战课程的过程中发现，问题还较多地存在。比如，当select数据库中的数据时，习惯性地写‘select \* from table’这样的语句。这样对于后面继续编写其实是不好的，其实只需要拿出自己的数据即可，如果别人在其他模块中，增加了数据库中的属性，那么select \* 选择到的数据就更多了，但是有很多数据其实是不需要的。  
   因为我们现在的数据库中量级较小，影响较小，但如果出现多部门合作时，很有可能数据库很庞大，select到许多没用的数据会极大地影响我们程序的速度
2. 分页时，出现了非常细微的错误，PageHelper.startPage();这个方法必须写在最前面，否则将无法使用。这样的错误很小，不太容易发现，以致于浪费了许多的时间。
3. 在合页面时，发现了数据量增多会使前端页面出现一些显著的变化，一些提示由于合的过程中，没有注意，导致了没有加入到页面中去，页面删除和修改时，不会询问是否删除。
4. 帮助同学进行编写，给出意见，表达自己的想法。这个方式对于一个团队来说，我觉得是一个比较好的方式，相互监督，当团队的其他人在做事的时候，其他人也不方便玩耍，形成了互相促进的局面，有助于开发进度。
5. 编写文档，在我看来其实是一件比较麻烦的事情，一方面来说，因为第一次接触团队开发的经验不足，文档的某些部分让人感觉很难描述。编写的过程也需要及时的改动和修正一些之前存在的不足之处。
6. 整理任务，分配到每个人之后，需要每隔一段时间对相应负责的人进行验收和询问。否则会出现拖延提交项目的时间，这样会延缓开发的进度。定时查看组员的任务进度是非常重要的事情。
7. 前后端分离，可能是因为经验不足的原因，导致后台写完了功能代码之后，一直在等待前台给出页面进行测试。总感觉等待时间会比较长，在一定程度上影响了开发的效率。

## **孙俊杰的总结**

我们将从任务、前期感悟、时间历程、后期感悟四个方面来作出如下总结。

任务

* 在本次团队项目中承担后端的角色，完成了登录注册（其中注册使用邮箱发送短信）、用户个人管理也就是用户个人信息修改。同时完成用户关注的部分。

前期感悟

* 之前写后端代码都是利用SSM（Spring+SpringMVC+Mybatis）框架进行写，同时前端后端代码结合在jsp页面中。这一次团队项目首先尝试前后端分离。因此在写代码前期，存在很大的困难。我们团队的前端开发人员之前用过前后端分离开发的经验，写了初步的接口文档。给与了我挺大的帮助。在前后端分离的过程中，发现自己与前端存在一些分歧，很多后端和前端都能处理的东西，都想要交给对方。除此以外，以前前端只是负责前端的渲染，进行精美页面的制作，现在也逐渐承担了处理数据的任务。后端主要负责处理逻辑然后将处理好的数据通过JSONObject 传给前端。

时间历程

* 从5.9开始，我们进行了第一次团队讨论，初步进行了分工。在接下来的两周内，完成了除了头像以外的基本功能，但是在第二周的时候，和代码的时间基本跟写代码的时间是一样的。在线下进行前后端分离的时候，是把我的电脑当做服务器，前后端的电脑连接同一个手机热点，然后后端的电脑关闭电脑的防火墙，通过Ipv4的ip地址就能进行访问了。
* 在5.4-5.19这将近两周的时间，我每天都拿出一定的时间写我负责部分的代码。完成了除了用户头像上传和修改的部分。在5.20-5.26这一时间段，完成了剩余部分的头像上传和修改，这个时候团队项目也进入了尾声，开始完善一些细节部分。

后期感悟

* 写了用户端的一些基本功能，在项目中得到了不断的成长，从之前前后端合jsp页面页面，到能够实现前后端的分离。不说这个项目做得如何，但是我在这其中学到了挺多。这才是我在这次团队项目中的最大的收获。其次，从之前的两人结对项目，到现在5人合作一个项目。分工很重要，每个人都有自己擅长的部分。其次，感觉这次团队项目中由于大家的代码风格不一样，也可能是默契度不够，在做项目的时候，总有一种对队友的小小的不信任，后端成员之间总要弄明白对方写的代码，前端后端合代码时候，后端要亲眼看到自己的代码运行成功才罢休。希望自己在以后的编码过程中，再接再厉，不断在项目中学习提高。

## **任静的总结**

我分别从做了什么、及相应感悟、总结三个方面阐述：

* 编写网站后台管理员界面前端部分用户部分

收获：对自己关于前端的内容有了更好的把握，利用bootstrap时的某些样式很方便

* 编写文章管理界面

收获：用bootstrap的table标签很方便，很简朴也不需要自己再调整格式，它会实现数据的自适应，觉得很方便。

* 编写管理员部分

收获：觉得要经常的编写程序，才能不断的熟练起来。

* 修改之前有错误的部分

收获：要想做到一个比较好的地步，要不断的去修改改正，尽力做到比较好的状态。

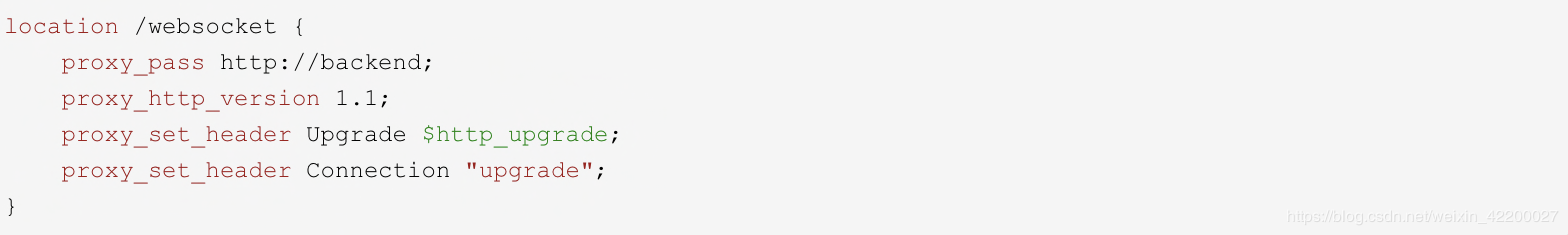
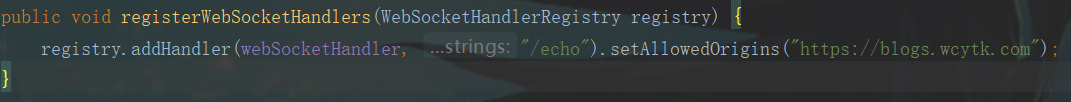
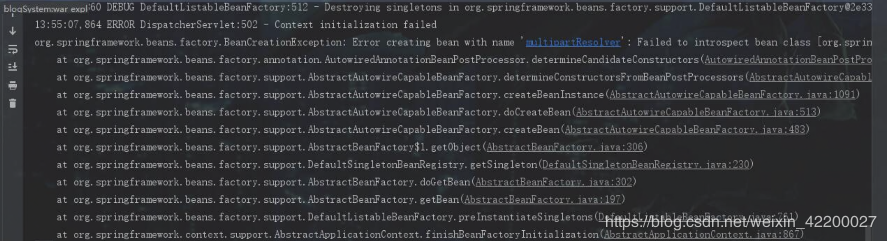
*****总结*****

* 在本次的项目中，由于之前的敲代码能力不佳，因此承担的是较为简单的页面的编写。由小组的另一位成员黄同学负责大部分的前端页面的编写。个人感觉一个项目组，合作是很重要的，其中少不了摩擦，但这是变得更好的一个过程。本次收获有很多，包括在用框架的时候，现有的样式确实能给自己省很大的力气去编写。另外一点是，我觉得老师给我们提出的所用的制度虽然很繁琐，但是可能将来到公司中就会这样子，优胜劣汰，这样子才能打造一个精尖的队伍。最后一点是，这次的项目告诉我要好好学习，这样子才能不给同伴拖后腿，我也认识到了我的更多不足，在今后的学习生活中要好好的修正不足。

## **王诚昱的总结**

以下以遇到的问题和总结两个方面做出介绍

本次项目遇到的主要问题:

1.session在本地能取到，在服务器上取不到。  
可能原因;  
① 跨域没有处理好,(主要是前端的问题,此问题出现的原因较多,也比较常见)  
② nginx没有配置好，location后的路径没有对应上里面代理的proxy的路径  
③ ajax中取的data不对,可能Response里有值但是没取到值  
2.js中的websocket的路径不对(connection refused)。  
可能原因:  
① websocket的路径写的是测试时的本地路径，但是应该是服务器的地址。  
② nginx没有配置升级支持websocket，要在配置中加上四句内容。  
  
③ 站点开启了https://，websocket禁止访问。ws://不支持https，改为wss://  
④ 改用wss:// 后前端js的websokcet又一直报403错误  
原因是我的js请求的不是http://localhost,而Spring会默认开一个OriginHandshakeInterceptor的拦截器  
解决方法,在注册handler的时候加一句setAllowedOrigins  
  
4.一开始在使用文件上传的时候报错  
  
后面查找资料发现是缺少了一个jar包  
  
5.如何设置文件上传的相对路径  
总结:

这次使用了前后端分离的形式，在体会到其带来的方便的同时,也遇到了很多的困难。

优点:  
① 不需要合jsp，减轻后端的压力。  
② 后端成了接口的形式，提高扩展性。

缺点:  
① 测试困难，特别是在部署到服务器上后如果后端代码修改了，需要重新上传和重启。  
② 前后端参数不统一，没有按照统一的接口文裆来写在合项目的时候，大部分的问题都出在参数上。

所以以后都要先在本地测试通过后，再传到服务器上，同时，前后端要使用统一的接口文裆。

总而言之，这次的项目主要是在重造车轮，除了一些Mybatis的操作外，没有学到很多新的东西,调bug的能力倒是提高了不少

## **黄政杰的总结**

以下总结分为四点：前端架构分析总结、前后分离开发总结、失误点总结以及感悟

前后架构分析总结

* 本次项目前端采用分层架构+模块化来进行开发。我使用了自制的前端项目架构脚手架实现了视图层–逻辑层–数据层的分层开发，另外还有一个工具层封装了部分通用类工具。在视图层，实现了前端组件化；在逻辑层，通过操作对象来实现层层解耦；在数据层，统一按模块封装了各类ajax请求。在模块化方面，选用的是es6模块化。在构建工具方面，选用了webpack4来进行打包编译，分离开发环境与生产环境。

前后分离开发总结

* 这是第一次与组员进行前后分离开发，采用的是比较广泛的RESTful API方案。我们的前后端开发组员在开发前共同编写了初始的接口文档，也在后面的开发过程中不断将其完善。此外，我们约定了统一的接口信息格式:数据状态码，数据和反馈信息。

失误点总结

* 第一次前后端联调测试的时候，前端无法获取后端的返回数据，花费了较长时间去检查，发现是后端的接口信息没有按照约定的接口信息格式返回。
* 在后端将代码部署到服务器上时，我们发现在登录后无法保持登录状态。寻找原因，得出的结果是前端在处理跨域时访问后端服务器接口的路径有问题。  
  感悟
* 这是我第一次与组员进行前后分离开发，遇到了不少问题，但最终还是一步一步地解决了。这次我做的前端方面，缺点在于性能优化做得较少，页面加载速度没有达到预期。但总的来说，这是一次很好的尝试，还是获得了不错的经验。

# **“站立”会议**

## **会议照片**

只放三张图片，图片太多会使页面过长

## **会议内容**

会议内容仅挑重要问题、矛盾和事件来叙述

1.第一次会议

* 最开始，确定以何种方式来进行项目，在是使用jsp还是使用前后端分离技术进行了一番争论。
* 以胡茜文为代表的一方，觉得使用jsp比较熟练，前后端分离技术在之前的学习中没有接触，设想遇到问题之后无法解决，不敢迎难而上。
* 以黄政杰为代表的一方坚持使用前后分离技术，觉得使用jsp在合项目时同样很麻烦，而且容易弄错，并坚持说之后的项目中前后分离技术是一定得会的。于是最终选择了前后分离技术。

2.第二次会议

* 在后台的基础功能代码基本完成后，由于前段写得较慢，代码部分没有合在一块儿，让写后端的同学感到十分着急，后端同学不知道自己之后该做什么（因为看不到前台的效果），同时接口文档只写了一部分，于是关于使用jsp还是前后分离的争论又开始了。于是召开了紧急会议做最终的决定
* 以胡茜文为代表的jsp派认为前后分离让其不能及时看到效果，遂提出重新使用jsp。
* 但王诚昱同学说之前的功能已经写好，return 的对象为JSON格式，要是再改成使用jsp，也会浪费掉不少时间，况且前端的框架已基本搭好，要是换方式，得不偿失。
* 最终结果：肯定了前后分离的方针。

3.第三次会议

* 第三次会议，开始了第一次合项目。针对合项目的问题进行了再一次会议。
* 在合项目的过程中，前端写的ajax不能处理传来的Json对象，但后端的格式的确是按照接口文档所写的，大家在一起讨论并解决这个问题，在前端改动了之前写的统一的Ajax文件之后，程序能够正常地跑起来了。
* 同时也发现了后端传输的一些问题，例如：
* {"msg":"获取粉丝列表成功","code":0,"data":{"list":[{"id":19,"isLogin":0,"username":"huxiaoxi"}{"id":25,"isLogin":0,"username":"huxiaoxi4"}]}} 这样的 json是正确的。但是后端传值时传成了 {"msg":"获取粉丝列表成功","code":0,"data":{"[{"id":19,"isLogin":0,"username":"huxiaoxi"}{"id":25,"isLogin":0,"username":"huxiaoxi4"}]}}
* 这样也导致了错误
* 这一次会议主要解决了JSON传输格式的规范化。不再随意而为

4.第四次会议

* 后段代码合在一台计算机上。
* 在这个时候，我们意识到了最开始敲代码的时候，团队没有把项目上传到git上，导致后来合项目的时候，每一次在自己电脑上改动之后，合到另外一台电脑上时，不知道什么地方改动了，加之每个人的代码习惯和配置不一样，加大了后端代码合在一块儿的难度，极大影响了项目进度，严重影响了合项目同学的心情，多次更改，导致会出现新的bug与错误，重新找出bug并改正的过程浪费了许多时间。
* 本次会议，总结了经验教训，在下一次实验的团队项目、结对项目中，应该将项目上传到git上。防止类似问题的出现。

5.第五次会议

* 最终项目基本成形，商议如何讲？谁去讲？怎么讲？同时解决遗留bug，在一些必须改正的地方和有重大问题的地方进行更正。
* 最终决定团队整体上台，自己讲自己做的一部分，并完善相应文档。同时写下最终的项目心得和经验总结。