

你知道的越多，你不知道的越多

点赞再看，养成习惯

**GitHub** <https://github.com/JavaFamily>上已经收录有一线大厂面试点脑图、个人联系方式和人才交流群，欢迎Star和指教

## 前言

群里不知道多少次大家说自己的项目没亮点，在大学没项目出去后怕找不到工作，问敖丙我实习的时候是怎么拿的华为Offer。

其实大学的项目很简单不需要太大的难度，因为面试官知道你也没啥经验，不会很在意这个，但是你得有项目这是必要条件，再不济增删改查你要比别人6吧，不然找个0经验的人进去，不是给他自己挖坑？

而且如果你有个还不错的履历，或者你的项目还有一些亮点，那完美了，你的Offer率会拉开同行的仔一大截的。

丙帅我呢大学做过几个项目，前些天问了问学妹，还在学校稳定运行着，说实话我有点小骄傲了。



但是说实话，我拿华为Offer跟这些项目关系不大，大学时候完成的项目完全是兴趣使然，想起当初自己一个人在室友还没起床的时候去图书馆，在室友都睡觉的时候还没回去的时光，我的眼角又湿了。

**Tip：**怎么拿的Offer我会在《程序人生》系列写，过年前或者后面出个我大学到现在的心路历程哈哈。

当时在电子阅览室当管理员，甚至多次为了赶进度，在图书馆通宵达旦，好在最后还是完成了那个项目，自己一个人从前端到服务端，从研发到部署上线。

大家都知道学校选课用选课系统的时候，很多学校给第三方公司做的系统都很卡，敖丙做的项目就不卡，因为那个时候我就用到Redis了，现在想想设计思想还算前卫。

好了吹了这么多，其实就是为了引出今天的主题，**如何从0到1搭建一个可以外网访问的项目**。

很多小伙伴看到最后要说了，丙丙我要学的是**分布式**，才不要学垃圾的单机系统。

分布式不就是一个单机的服务构成的，你多起几个进行RPC通信不就好了？

## 正文

双十二阿里服务器推广，不买的小伙伴直接跳过这段，不过用来学习真的香，**比学生的9.9每月还便宜**。

我帮阿里云推广服务器89/年，229/3年，买来送自己，送女朋友马上过年再合适不过了，买了搭建个项目给面试官看也香，还可以熟悉技术栈，我明天会出一个服务器搭建个人项目的教程（老用户用家人的购买，我用我妈的😂）。扫码或者点击购买



我就用自己的服务器给大家举例，怎么从0到1搭建一个学生和新手可以用来面试的项目，老手也可以回忆一下自己逝去的青春。

可能有小伙伴疑惑，丙丙为啥要服务器，本地不行么？

可以，当然不是为了推广服务器哈，大家可以不买，我只是阐述一下自己的观点，因为大家以后出去难免遇到自己操作服务器的情况，有些公司没有集成发布系统，那就需要你亲自去机器上操作了。

然后你`rm -rf`，好了去财务室领下工资吧，年底了，提前回家过年可还行？



开玩笑的哈，只是很多操作大家不经常熟悉其实会都不知道，真正去操作或者面试官问你，熟悉各种操作么，你说不熟悉，好的回去等消息。

作为面试官，面试没多少经验的你，他更看重你思考的思路，还有你有没有实操过，务实么？

帅丙我大学的项目就是经常操作，所以后面很长一段时间的服务器操作，还有Linux命令啥的都还是不错的，技术栈搭建起来也是很熟悉，不过最近都是用的公司的工具，命令忘了好多了，需要反省一下了。

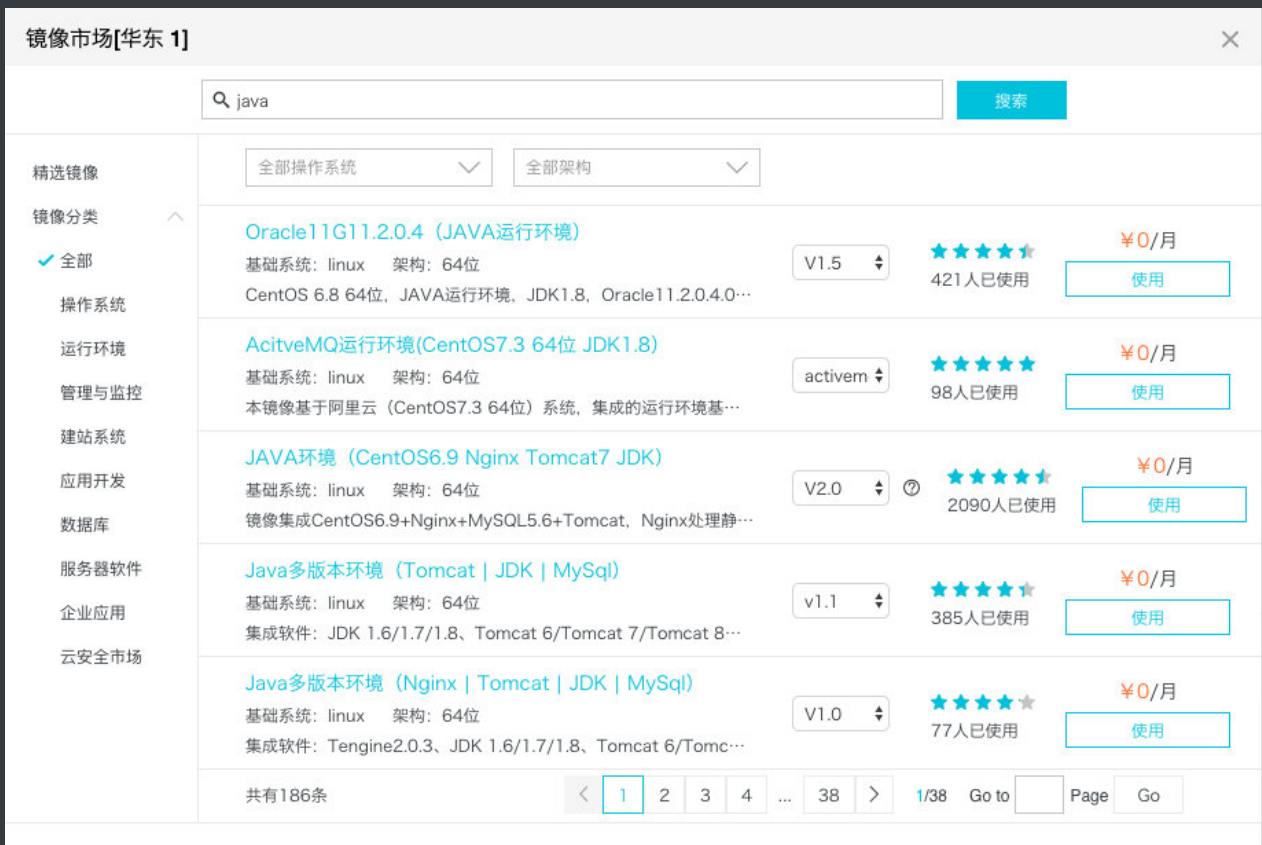
服务器在激活的时候会让你选系统，这个时候敖丙我个人建议，有点基础已经熟悉安装过程和环境配置的同学可以直接选择镜像，镜像市场里面有很多很适合的镜像，我选的就是带JDK8，Tomcat8，MySQL的镜像。

没搞过的同学可以尝试着选个空白机，然后下好这些东西，一步步去尝试，我想收获还是会有有的，跟你在自己电脑搭建的区别你会马上体会到，这就是为啥建议大家去服务器操作的原因之一。

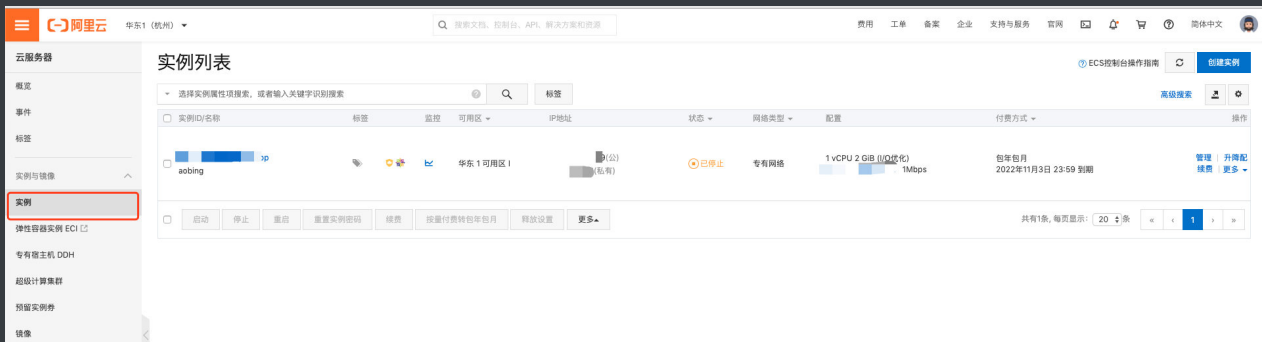
### 镜像选择

The screenshot shows a cloud image selection interface. At the top, there are four tabs: '公共镜像' (Public Images), '自定义镜像' (Custom Images), '共享镜像' (Shared Images), and '镜像市场' (Image Market). The '公共镜像' tab is selected and highlighted with a red box. Below the tabs, there are two dropdown menus: '请选择操作系统类别' (Please select operating system category) and '请选择版本' (Please select version). A red arrow points to the '请选择版本' dropdown with the text '空白镜像里面啥都没' (Nothing in the blank image). To the right of the dropdowns is a link '教我选择>>' (Teach me to choose >>). A red arrow points to this link with the text '可以选择自己需要的环境 如JDK8, Tomcat, MySQL' (You can choose the environment you need, such as JDK8, Tomcat, MySQL). Below the dropdowns, there is a checkbox labeled '安全加固' (Security Reinforcement) with a question mark icon. Further down, there are fields for '系统盘' (System Disk) with a dropdown set to '高效云盘' (High Performance Disk), a size field set to '40 GB', and 'IOPS' set to '2120'. Below these fields is a link '如何选择 SSD云盘 / 高效云盘 / 普通云盘, 请看 详细说明>' (How to choose SSD disk / High Performance disk / General disk, please see detailed explanation >). At the bottom, there are input fields for '登录密码' (Login Password) and '确认密码' (Confirm Password), with a note '8 - 30 个字符, 且同时包含三项 (大写字母、小写字母、数字、特殊符号)' (8 - 30 characters, and must contain three items (uppercase letters, lowercase letters, numbers, special characters)).

镜像记得选不要钱的，一般都是免费的。



选完你在**实例**就可以看到自己的服务器了，下面就是帅丙的服务器，居然跟我一样帅，天呐。



上面都是新搭建的小伙伴，已经有服务器的小伙伴，但是又想换成镜像怎么办？

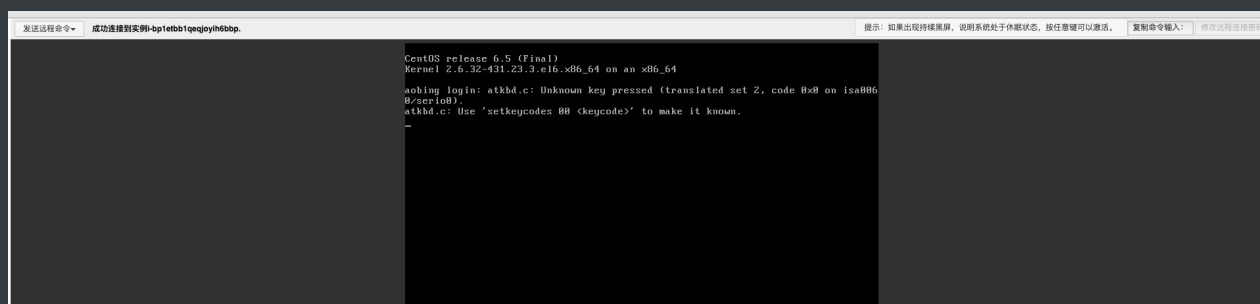
大家先把**服务停掉**然后点击右边的**更换操作系统**就好了，会让你去镜像市场选的，之前你选了什么系统都可以换。



## 登录服务器

可以直接在管理界面网页登录服务器远程链接，也可以自己本地用对应的控制台工具连接。

个人推荐本地的控制台连接会好点，每次要登录服务器都要去网页很麻烦的。



远程链接的代码，记住是大家的公网ip，管理界面看得到的，要记得你购买的时候配置的密码，因为这个时候要输入。

```
ssh root@123.14.123.8 -p 22 （后面的22是个端口，后面我介绍到）
```

## 实例列表

选择实例属性项搜索，或者输入关键字识别搜索						?	Q	标签
<input type="checkbox"/> 实例ID/名称	标签	监控	可用区	IP地址	状态			
<input type="checkbox"/> i-bp1etbb1qeajoyih6bbp aobing			华东 1 可用区 I	134.239(公) 119.8(私有)	运行			
<input type="checkbox"/>	启动	停止	重启	重置实例密码	续费	按量付费转包年包月	释放设置	更多

```
login: root
Password:
Last login: Sun Dec  1 11:44:28 from 101.68.79.178

Welcome to aliyun Elastic Compute Service!

[root@aobing ~]# atkbd.c: Unknown key pressed (translated set 2, code 0x0 on is
0060/serio0).
atkbd.c: Use 'setkeycodes 00 <keycode>' to make it known.
```

到这一步的时候，很多小伙伴直接登录成功，但是之前就已经有服务器的朋友是不发现，出现下面这个页面了？

```
aobingdeMacBook-Pro:~ root# ssh root@134.239.119.8 -p 22
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
SHA256:cuwIWWtWPni0TJ/ku/yDsIcI9z8/44eTKUHQTfLz06E.
Please contact your system administrator.
Add correct host key in /var/root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /var/root/.ssh/known_hosts:2
RSA host key for 134.239.119.8 has changed and you have requested strict checking.
Host key verification failed.
```

会出现这些信息是因为，第一次SSH连接时，会生成一个认证，储存在客户端（也就是用SSH连线其他电脑的那个，自己操作的那个）中的known\_hosts，但是如果服务器验证过了，认证资讯当然也会更改，服务器端与客户端不同时，就会跳出错误啦～因此，只要把电脑中的认证资讯删除，连线时重新生成，就一切完美啦～要删除很简单，只要在客户端输入一个指令

```
ssh-keygen -R +输入服务器的IP
```

接下来再次连接一次，会出现

```
Are you sure you want to continue connecting (yes/no)?
```

输入yes！ 就完成连接啦！同时，新的认证也生成了。



```
aobingdeMacBook-Pro:~ root# ssh root@ -p 22
root@116.62.134.239's password:
Last login: Sun Dec 1 12:03:17 2019

Welcome to aliyun Elastic Compute Service!

[root@aobing ~]#
```

进去后如果是镜像的小伙伴会发现本身他准备好的东西都在了，环境都搭建起来了，大家只管直接使用就好了。

但是如果没有选择镜像的朋友，就需要自己搭建一个FTP或者使用传输命令去把你下载好的Tomcat、MySQL等传输进来安装好。

```
[root@aobing ~]# more README.txt
+-----+
| YJCOM [ EASY CLOUD EASY WEBSITE]" >> $README
+-----+
| Copyright (c) 2015 http://yjcom.com All rights reserved.
+-----+

tomcat6          /var/tomcat/tomcat-6
tomcat7          /var/tomcat/tomcat-7
tomcat8          /var/tomcat/tomcat-8
mysql5.6         /var/lib/mysql

mysql:           service mysql (start|stop|restart)
vsftpd:          service vsftpd (start|stop|restart)

www ftp directory    /yjdata/www/www/

change tomcat version for the default site
/yjdata/www/change_tomcat_version.sh 6|7|8

change jdk version for the default site
/yjdata/www/change_jdk_version.sh 1.6|1.7|1.8
[root@aobing ~]#
```

大家可以看到还有一个**默认密码文件夹**，里面就有MySQL和FTP的账号密码什么的，不同的镜像可能有差异。

这个在之后大家自己的服务写代码的时候，配置连接本地的MySQL就是需要这个的。

```
[root@aobing ~]# more default.pass
+-----+
| YJCOM [ EASY CLOUD EASY WEBSITE]
+-----+
| Copyright (c) 2015 http://yjcom.com All rights reserved.
+-----+

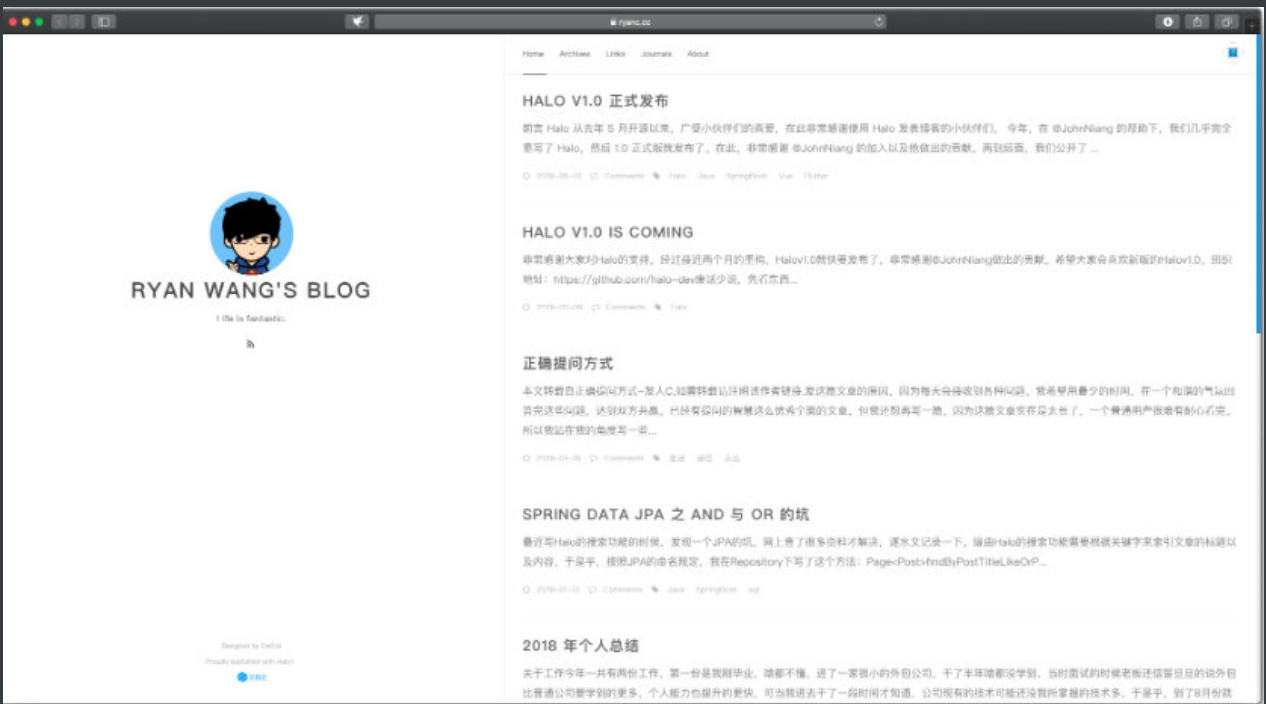
MySQL root password: 
MySQL database name: 
MySQL user: 
MySQL password: 
FTP account: 
FTP password: 
[root@aobing ~]#
```

博客网站

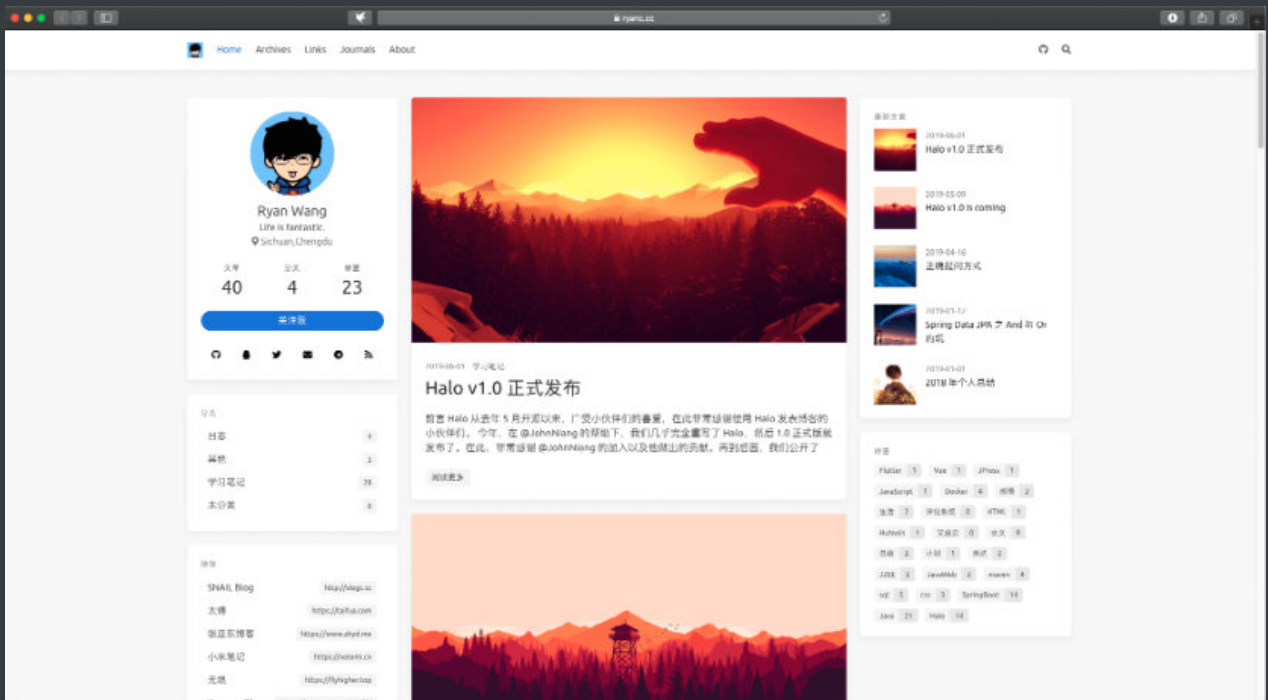
这次教大家搭建的项目说大不大，说小吧他啥都有，反正小白和学生肯定很适合，是个个人博客网站。

我们先看看博客的效果，看看大家作为大学的项目到底合格么？

我大学做的那个项目比起这个就差点意思了，当时要是有这么好的UI就很香了！







## 项目代码

以我这种直男审美都觉得UI很不错，你以为只是UI可以？那我们看看项目！

项目的Git地址 <https://github.com/halo-dev/halo.git> 大家克隆一下就好了。

这个项目本身是**Gradle**的，很多小伙伴就要说了，啊帅丙我不要，我就要**Maven**项目的。



我只想告诉大家，**技多不压身**，目前很多大厂都是Maven项目Gradle项目都占有很高的比重，你怎么知道你去的公司会用啥？

你会用进去**减少多少学习的成本**啊，Leader在旁边暗自给你比一个大拇指，暗自感叹：**帅丙的读者，有点东西**。

Mac很多小伙伴安装的时候发现没有**Homebrew** 就没办法用很方便的命令行去下载安装Gradle，那简单我们安装一下，用下面的命令。（Windows的朋友直接官网下载一个就好了

<https://gradle.org/install>)

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

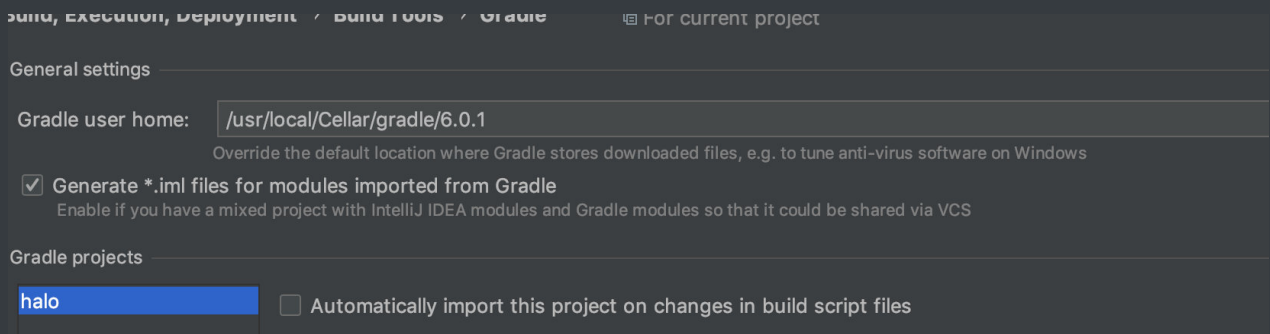
```
* [new tag]          2.2.0      -> 2.2.0
HEAD is now at adc630f1c Merge pull request #6801 from EricFromCanada/output-formatting
==> Downloading https://homebrew.bintray.com/bottles-portable-ruby/portable-ruby-2.6.3.mavericks.bottle.tar.gz
##### 100.0%
==> Pouring portable-ruby-2.6.3.mavericks.bottle.tar.gz
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
    https://github.com/Homebrew/brew#donations
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Enumerating objects: 5096, done.
remote: Counting objects: 100% (5096/5096), done.
remote: Compressing objects: 100% (4891/4891), done.
remote: Total 5096 (delta 51), reused 319 (delta 9), pack-reused 0
Receiving objects: 100% (5096/5096), 4.14 MiB | 2.46 MiB/s, done.
Resolving deltas: 100% (51/51), done.
```

这个安装好了，我们再用下面的命令就可以就把Gradle安装好了，是不是比你去下载方便很多？

```
$ brew install gradle
```

```
aobingdeMacBook-Pro:~ aobing$ brew install gradle
==> Downloading https://services.gradle.org/distributions/gradle-6.0.1-all.zip
==> Downloading from https://downloads.gradle-dn.com/distributions/gradle-6.0.1-
##### 100.0%
📦 /usr/local/Cellar/gradle/6.0.1: 14,178 files, 247.9MB, built in 1 minute 15
seconds
aobingdeMacBook-Pro:~ aobing$
```

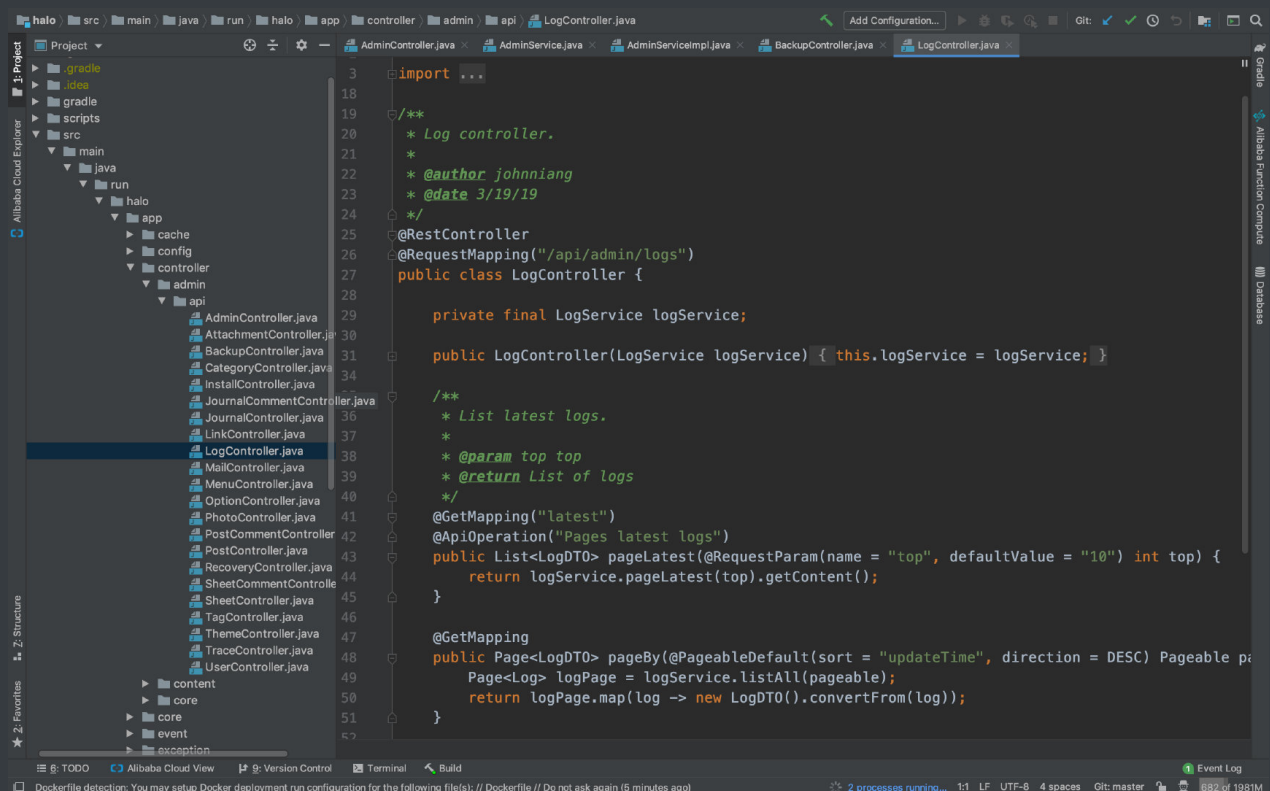
接下来去项目中设置下Gradle地址，这里介绍的是IDEA的设置

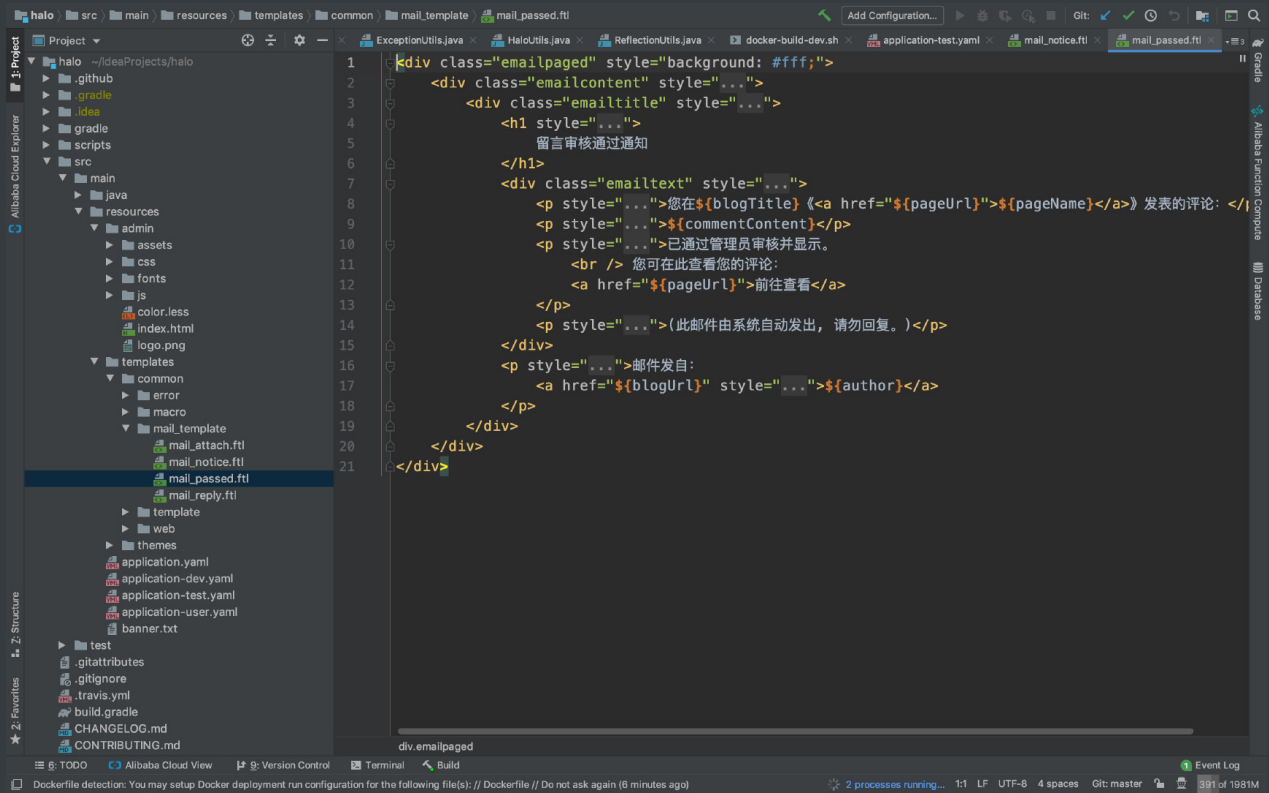


## 为啥推荐个人博客项目？

很多人问我为啥推荐这个项目，这个项目看过次代码了，没啥语法错误，项目结构完全标准的项目，注释也很齐全，代码的风格挺不错的，还是用了很多新语法和注解。

而且他也不难理解，结构很简单，前端后端都很简单，大家看看代码，看看前后的接口，很快就能适应这个项目了。





接下来我们可以用项目去打个包，或者直接下载敖丙打包好的。（jar包还有war包随意）

## 用命令行下载最新的 Halo 安装包

```
curl -L https://github.com/halo-dev/halo/releases/download/v1.1.1/halo-1.1.1.jar --output halo-latest.jar
```

或者

```
wget https://github.com/halo-dev/halo/releases/download/v1.1.1/halo-1.1.1.jar -O halo-latest.jar
```



```
root@aobing:~
Permission denied, please try again.
root@116.62.134.239's password:
Last failed login: Sun Dec 1 16:05:23 CST 2019 from 101.68.79.178 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Sun Dec 1 15:21:59 2019

Welcome to Alibaba Cloud Elastic Compute Service !

[root@aobing ~]#
[root@aobing ~]# wget https://github.com/halo-dev/halo/releases/download/v1.1.1/halo-1.1.1.jar -O halo-latest.jar
--2019-12-01 16:06:02-- https://github.com/halo-dev/halo/releases/download/v1.1.1/halo-1.1.1.jar
正在解析主机 github.com (github.com)... 13.229.188.59
正在连接 github.com (github.com)|13.229.188.59|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 302 Found
位置: https://github-production-release-asset-2e65be.s3.amazonaws.com/126178683/8f7eff80-e06e-11e9-91ee-a89a1cbbfd3c?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20191201%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191201T080602Z&X-Amz-Expires=300&X-Amz-Signature=9d7a8a99e1996514e5650eccde54b5b859868560165aad02e74fc507f6ac4d50&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dhalo-1.1.1.jar&response-content-type=application%2Foctet-stream [跟随至新的 URL]
--2019-12-01 16:06:03-- https://github-production-release-asset-2e65be.s3.amazonaws.com/126178683/8f7eff80-e06e-11e9-91ee-a89a1cbbfd3c?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20191201%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191201T080602Z&X-Amz-Expires=300&X-Amz-Signature=9d7a8a99e1996514e5650eccde54b5b859868560165aad02e74fc507f6ac4d50&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dhalo-1.1.1.jar&response-content-type=application%2Foctet-stream
正在解析主机 github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.84.64
正在连接 github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.84.64|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度: 72199537 (69M) [application/octet-stream]
正在保存至: "halo-latest.jar"

0% [ ] 442,958 14.9KB/s 剩余 69m 31s
```

文件传输呢，帅丙我习惯命令行了，FTP这样的工具还是不适应，我就发下命令行是怎么把jar包从本地传输到服务器的。

```
scp -P 22 /Java/program.war
root@123.456.789.987:/usr/server/tomcat7/webapps/
```

```
aobingdeMacBook-Pro:~ aobing$ scp -p 22 /U...ng/halo-latest.jar root@1...
2...:/
root@1...:139's password:
22: No such file or directory
halo-latest.jar 100% 69MB 3.6MB/s 00:19
aobingdeMacBook-Pro:~ aobing$
```

我们进入机器去看一下，文件已经传输进来了。

```
[root@aobing ~]# ll
总用量 3520
-rw-r--r-- 1 root root 3601974 12月 1 16:37 halo-latest.jar
[root@aobing ~]#
```

因为是一个SpringBoot项目，你只要有JDK直接就能跑起来了。

```
java -jar halo-latest.jar
```

```
aobingdeMacBook-Pro:~ aobing$ java -jar halo-latest.jar
```



```

::: Spring-Boot 2.1.7.RELEASE    ::: Halo (version: (v1.1.1))
2019-12-01 17:02:42.268 INFO 28856 --- [          main] run.halo.app.Application      : Starting Application v1.1.1 on aobingdeMacBook-Pro.local with PID 28856 (/Users/aobing/halo-latest.jar
started by aobing in /Users/aobing)
2019-12-01 17:02:42.272 INFO 28856 --- [          main] run.halo.app.Application      : No active profile set, falling back to default profiles: default
2019-12-01 17:02:43.669 INFO 28856 --- [          main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2019-12-01 17:02:43.853 INFO 28856 --- [          main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 171ms. Found 21 repository interfaces.
2019-12-01 17:02:44.863 INFO 28856 --- [          main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of type [org.
springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIJB$$152b525b] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible
for auto-proxying)
2019-12-01 17:02:45.646 WARN 28856 --- [          main] io.undertow.websockets.jsr      : UT025010: Buffer pool was not set on WebSocketDeploymentInfo, the default pool will be used
2019-12-01 17:02:45.695 INFO 28856 --- [          main] io.undertow.servlet            : Initializing Spring embedded WebApplicationContext
2019-12-01 17:02:45.695 INFO 28856 --- [          main] o.s.web.context.ContextLoader  : Root WebApplicationContext: initialization completed in 3342 ms
2019-12-01 17:02:45.982 INFO 28856 --- [          main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2019-12-01 17:02:46.347 INFO 28856 --- [          main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2019-12-01 17:02:46.459 INFO 28856 --- [          main] o.hibernate.jpa.internal.util.LogHelper : H##000204: Processing PersistenceUnitInfo [
name: default
...]
2019-12-01 17:02:46.593 INFO 28856 --- [          main] org.hibernate.Version          : H##000412: Hibernate Core {5.3.10.Final}
2019-12-01 17:02:46.597 INFO 28856 --- [          main] org.hibernate.cfg.Environment  : H##000206: hibernate.properties not found
2019-12-01 17:02:46.956 INFO 28856 --- [          main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
2019-12-01 17:02:47.486 INFO 28856 --- [          main] org.hibernate.dialect.Dialect   : H##000400: Using dialect: org.hibernate.dialect.H2Dialect
2019-12-01 17:02:49.005 INFO 28856 --- [          main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2019-12-01 17:02:50.979 INFO 28856 --- [          main] o.h.h.i.QueryTranslatorFactoryInitiator : H##000397: Using ASTQueryTranslatorFactory
2019-12-01 17:02:54.624 INFO 28856 --- [          main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 2 endpoint(s) beneath base path '/actuator'
2019-12-01 17:02:54.763 INFO 28856 --- [          main] pertySourceRequestMappingHandlerMapping : Mapped URL path [/v2/api-docs] onto method [public org.springframework.http.ResponseEntity springfox.docu
mentation.spring.web.json.Json2 Springfox.documentation.swagger2.web.Swagger2Controller.getDocumentation(java.lang.String, javax.servlet.http.HttpServletRequest)]
```

然后大家访问下自己的 ip+端口（8090）就能看到项目了

然后很多小伙伴说，渣男，我的怎么不行？

稍等稍等，敖丙啊还正准备说呢，大家的端口默认就开通了 22 、-1、3389

22是不是很熟悉，就是开头我们链接的端口号→ ssh root@123.14.123.8 -p **22** 也是因为厂商默认给我们开通了，我们才可以直接链接的。

但是我们要访问别的端口，那就去开通嘛，记得入口出口一样都要开通哟！

实例详情

本实例磁盘

本实例快照

本实例弹性网卡

本实例安全组

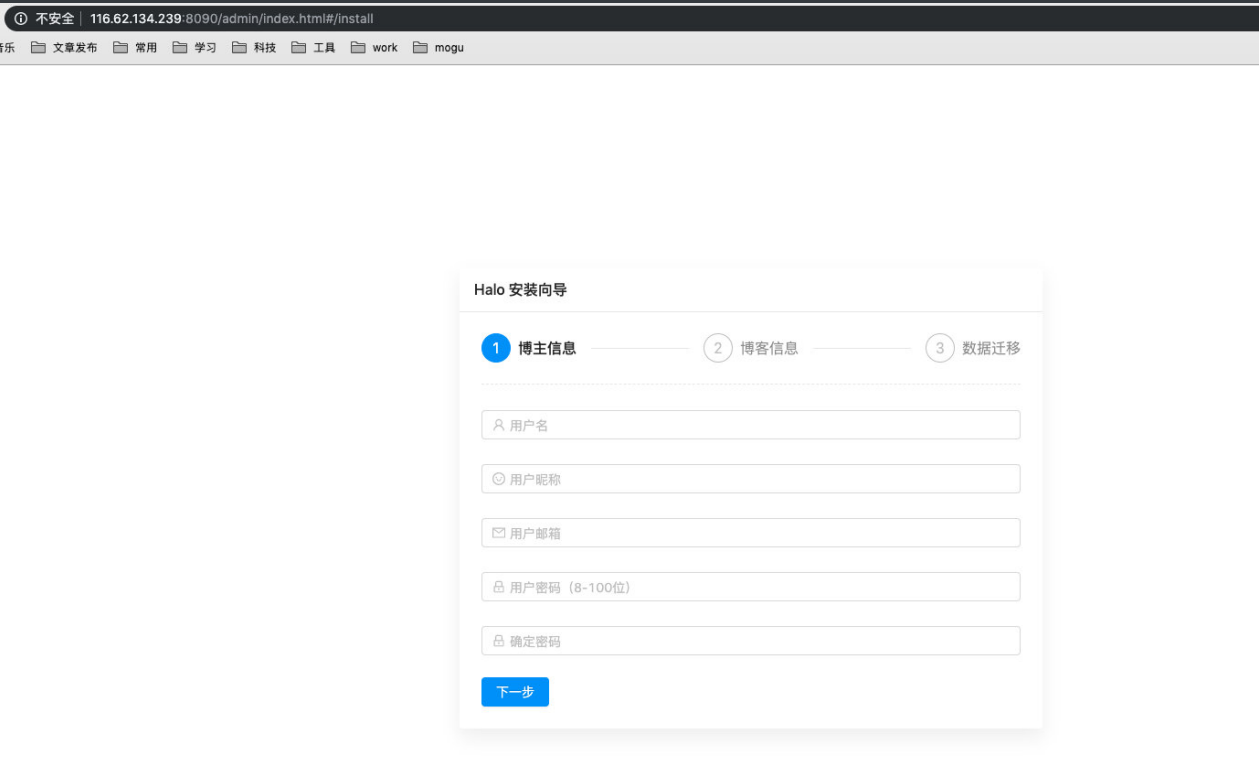
本实例安全防护



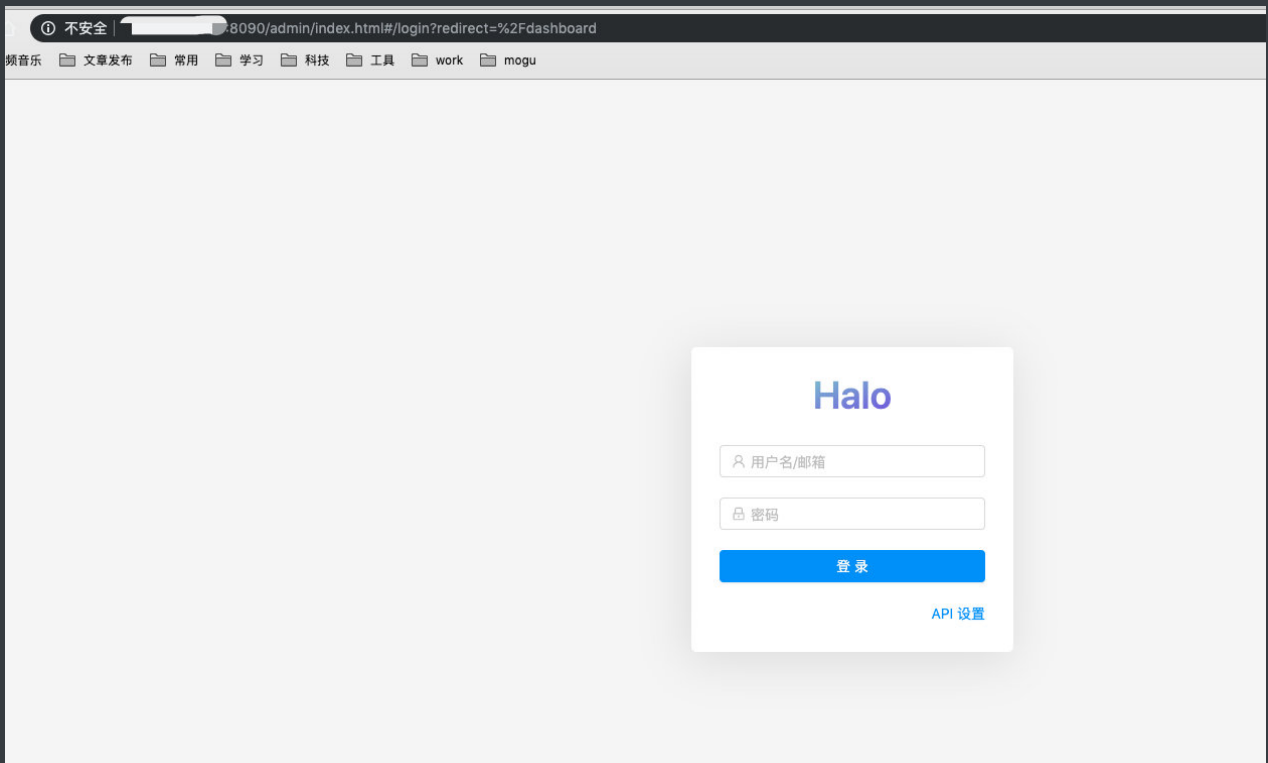
<input type="checkbox"/>	授权策略	协议类型	端口范围	授权类型(全部) ▾	授权对象	描述	优先级	创建时间	操作
<input type="checkbox"/>	允许	自定义 TCP	8090/8090	IPv4地址段访问	0.0.0.0/0	tomcat	1	2019年12月1日 17:07	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	80/80	IPv4地址段访问	0.0.0.0/0	tomcat	1	2019年12月1日 14:38	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	8009/8009	IPv4地址段访问	0.0.0.0/0	tomcat	1	2019年12月1日 13:45	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	8005/8005	IPv4地址段访问	0.0.0.0/0	tomcat	1	2019年12月1日 13:44	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	8080/8080	IPv4地址段访问	0.0.0.0/0	tomcat	1	2019年12月1日 13:30	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	全部 ICMP(IPv4)	-1/-1	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2019年11月3日 14:18	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	22/22	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2019年11月3日 14:18	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>
<input type="checkbox"/>	允许	自定义 TCP	3389/3389	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2019年11月3日 14:18	<a href="#">修改</a>   <a href="#">克隆</a>   <a href="#">删除</a>



然后重启下机器，重新启动项目，是不是！！可以访问了？？我们先注册一下



然后再登录进去。

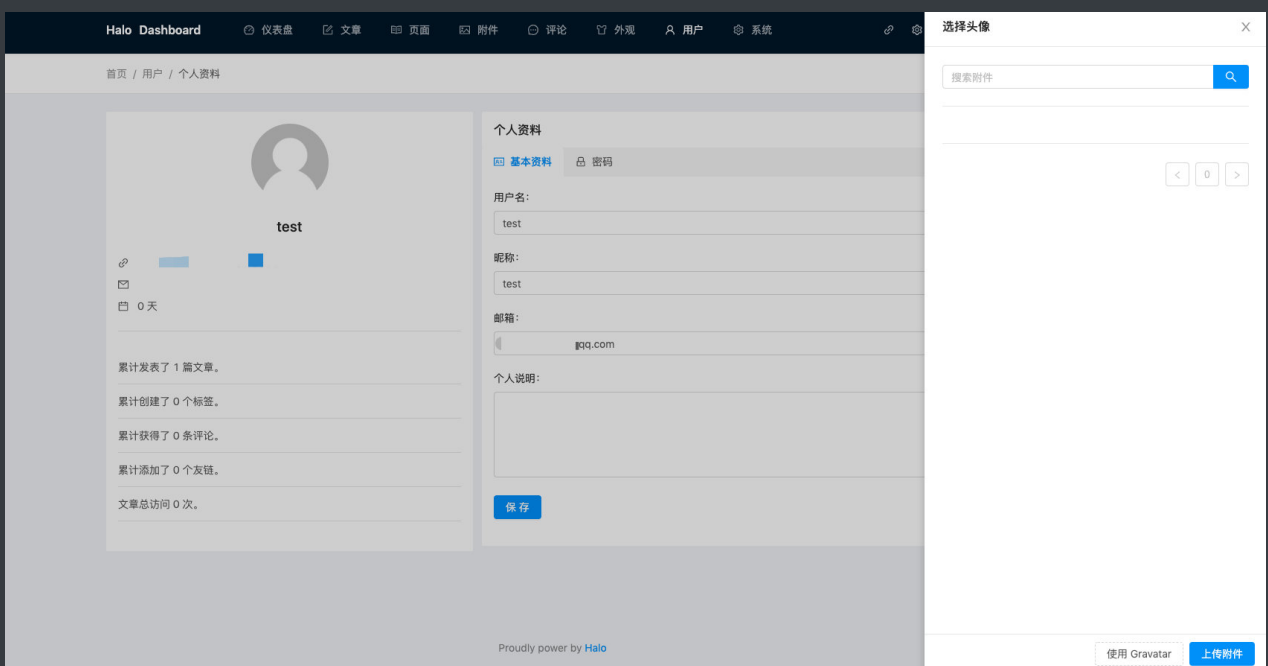
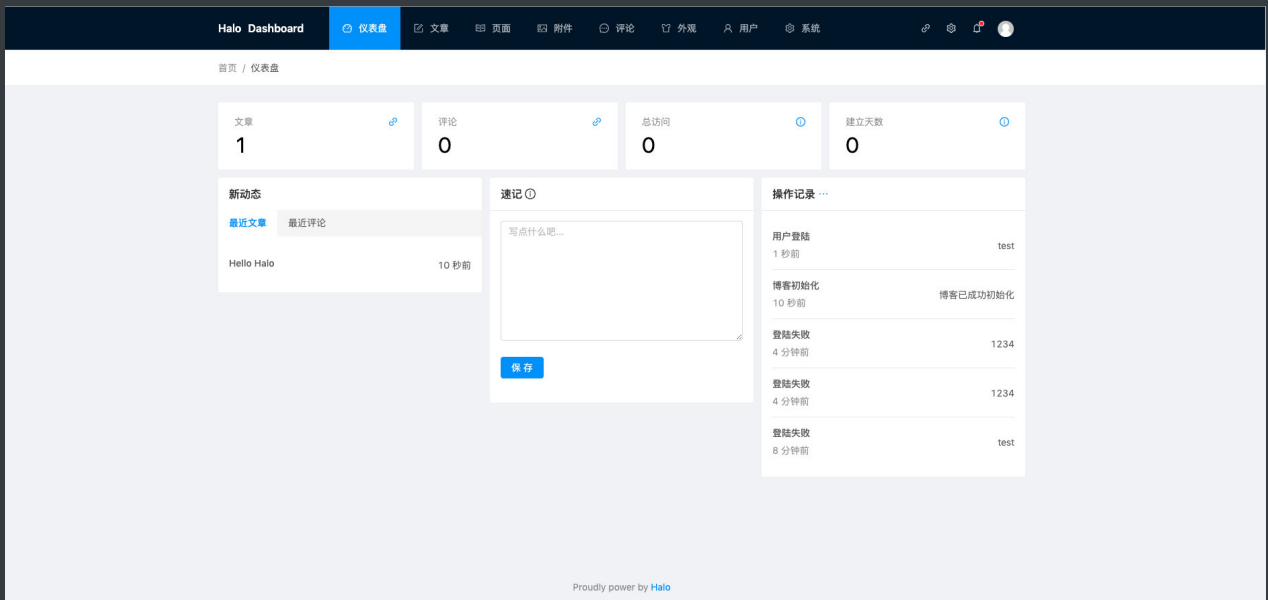


可以看到校验的异常什么的都直接日志抛出的，项目我觉得对于学生和新手来说很香，里面的代码风格大家也多适应一下。

```
run.halo.app.exception.BadRequestException: 用户名或者密码不正确
    at run.halo.app.service.impl.AdminServiceImpl.authenticate(AdminServiceImpl.java:147) ~[classes!/:1.1.1]
    at run.halo.app.controller.admin.api.AdminController.auth(AdminController.java:51) ~[classes!/:1.1.1]
    at run.halo.app.controller.admin.api.AdminController$$FastClassBySpringCGLIB$$4785b262.invoke(<generated>) ~[classes!/:1.1.1]
    at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:218) ~[spring-core-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:749) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.aspectj.MethodInvocationProceedingJoinPoint.proceed(MethodInvocationProceedingJoinPoint.java:88) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at run.halo.app.cache.lock.CacheLockInterceptor.interceptCacheLock(CacheLockInterceptor.java:69) ~[classes!/:1.1.1]
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:1.8.0_151]
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) ~[na:1.8.0_151]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:1.8.0_151]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[na:1.8.0_151]
    at org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethodWithGivenArgs(AbstractAspectJAdvice.java:644) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethod(AbstractAspectJAdvice.java:633) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.aspectj.AspectJAroundAdvice.invoke(AspectJAroundAdvice.java:70) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:175) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at org.springframework.aop.aspectj.MethodInvocationProceedingJoinPoint.proceed(MethodInvocationProceedingJoinPoint.java:88) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
    at run.halo.app.core.ControllerLogAop.controller(ControllerLogAop.java:41) ~[classes!/:1.1.1]
    at sun.reflect.GeneratedMethodAccessor163.invoke(Unknown Source) ~[na:na]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:1.8.0_151]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[na:1.8.0_151]
    at org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethodWithGivenArgs(AbstractAspectJAdvice.java:644) ~[spring-aop-5.1.9.RELEASE.jar!/:5.1.9.RELEASE]
```

登录进去我们可以发现，功能很全，博客网站基本的结构都在了。

有个不成熟的猜想，还没毕业设计的朋友。。。😂



## 絮叨

看到这里很多小伙伴可能会说你就是为了推广服务器才写的吧，我说实话有一定的成分，但是更多是因为我自己的群里，经常有很多小伙伴对**项目的整个搭建流程不是很熟悉**，所以才出这样的一篇文章的。

这里只是一个单机的项目，大家真正部署的时候要体验别的技术栈可以去Git上下载下来部署上去，体验他整个部署流程，**中间肯定很多不顺利的地方**相信我，100000%不会一次成功，但是你慢慢摸索的过程就是你学习的收获的过程。

大家经常问我那些大佬怎么成长上去的，**踩坑和付出大量实践上去的**，别无他法。

我老东家的Leader 95年的，是我们之前公司前端后端的Leader，技术深度广度，业务深度都领人发指，怎么做到的？

简单，**不断踩坑学习**咯，每天都是凌晨回去，白天开会，晚上写代码，甚至还要挤出时间学习新的业务，技术栈也是不断自己去摸索。

之前他一手带的我，我跟他太熟悉了，买了早餐刚坐下就去开会讨论业务，晚上6点回来就说：诶鸡蛋豆浆都冷了啊，然后丢垃圾桶。

学习是一条令人时而喜极若狂、时而郁郁寡欢的道路。

成长路上我们一起共勉。