

# **新一代电影购票与影视服务网站**

## **—— 皮皮虾电影购票网**

**Git 和 Bug 管理系统、持续集成环境的安装和学习报告**

**PiPiShrimp Team**  
**2017.04.08**

## 修订历史

版本	日期	描述	作者
正式版	2017 年 4 月 8 日	Git 和 Bug 管理系统、持续集成环境的安装和学习	吴思

## Github :

### 1、什么是 Github ?

Git 是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。

相对于集中式版本控制系统，分布式版本控制系统有以下优势：分布式版本控制系统没有“中央服务器”，每个人的电脑上都是一个完整的版本库，多人协作只需把各自的修改推送给对方，就可以互相看到对方的修改了；分布式版本控制系统的安全性要高很多，因为每个人电脑里都有完整的版本库，某一个人的电脑坏掉了不要紧，随便从其他人那里复制一个就可以了。

### 2、Github 的安装

以在 Mac OS X 上安装 Git 为例，方法如下：

安装 homebrew，然后通过 homebrew 安装 Git，指令如下：

**\$ brew install git**

（由于电脑早已安装过 git，故不能截图安装过程）

安装完成，查看版本信息如下：



```
wusideMacBook-Pro:~ Sophia$ git --version
git version 2.11.0 (Apple Git-81)
wusideMacBook-Pro:~ Sophia$
```

### 3、如何使用 Github ?

以下是简单的 git 使用：

（1）创建版本库：创建一个空目录，通过 git init 命令把这个目录变成 Git 可以管理的仓库：

```
wusideMacBook-Pro:Desktop Sophia$ cd demo/  
wusideMacBook-Pro:demo Sophia$ git init  
Initialized empty Git repository in /Users/Sophia/Desktop/demo/.git/  
wusideMacBook-Pro:demo Sophia$
```

(2) 把文件添加到版本库。

例 :把 readme 文件放到该文件夹。用命令 `git add` 告诉 Git, 把文件添加到仓库 ;  
用命令 `git commit` 告诉 Git, 把文件提交到仓库 :

```
wusideMacBook-Pro:demo Sophia$ git add .  
wusideMacBook-Pro:demo Sophia$ git commit -m "test"  
[master (root-commit) 91dd39e] test  
Committer: 吴思 <Sophia@172-1-1-33.lightspeed.hstntx.sbcglobal.net>  
Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.  
You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:  
  
    git config --global --edit  
  
After doing this, you may fix the identity used for this commit with:  
  
    git commit --amend --reset-author  
  
2 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 .DS_Store  
create mode 100644 readme.md  
wusideMacBook-Pro:demo Sophia$
```

(3) 远程仓库

(a) 添加远程库 :

登陆 GitHub, 然后, 在右上角找到“Create a new repo”按钮, 创建一个新的仓库, 把一个已有的本地仓库与之关联, 然后, 把本地仓库的内容推送到 GitHub 仓库, 远程库的名字就是 origin。指令形如 :

```
$ git remote add origin git@github.com:xxxx/xxx.git
```

```
$ git push origin master
```

(b) 从远程库克隆 :

指令形如 : **\$ git clone https://github.com/xxxx**

(4) 分支管理

(a) 创建与合并分支

```
$ git checkout -b dev
```

`git checkout` 命令加上 `-b` 参数表示创建并切换到 dev 开发分支。

```
$ git branch
```

用 `git branch` 命令查看当前分支。

**\$ git checkout master**

**\$ git merge dev**

切换回 master 分支；把 dev 分支的工作成果合并到 master 分支上。

**\$ git branch -d dev**

删除 dev 分支

#### (5) 多人协作

多人协作的工作模式通常是这样：

- 首先，可以试图用 **git push origin branch-name** 推送自己的修改；
- 如果推送失败，则因为远程分支比你的本地更新，需要先用 **git pull** 试图合并；
- 如果合并有冲突，则解决冲突，并在本地提交；
- 没有冲突或者解决掉冲突后，再用 **git push origin branch-name** 推送就能成功！
- 如果 git pull 提示“no tracking information”，则说明本地分支和远程分支的链接关系没有创建，用命令 **git branch --set-upstream branch-name origin/branch-name**。

## Bug 管理系统：

### 1、什么是 Bug 管理系统？

顾名思义，bug 管理系统就是一款用于管理软件开发过程中出现的 bug，目前，市面上有很多 bug 管理工具，比如：EasyBUG、Bugzilla、QC 等等。

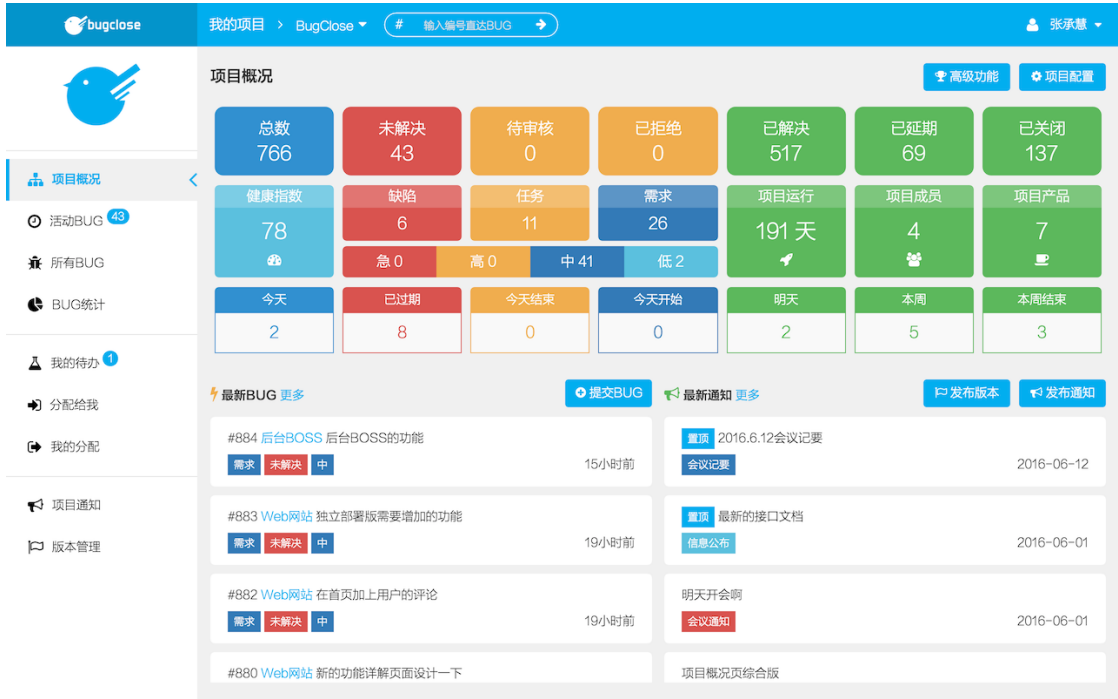
### 2、Bugclose

以 Bugclose 为例，其优势有：

- 简单易用：注册即用 无需部署；灵活组织团队和项目；只做必要的记录和跟踪
- 创新功能：统一管理缺陷任务和需求；独创工作台模式 高效捉虫；手机传图 免去中间环节
- 安全稳定：基于阿里云构建；良好的稳定性和伸缩性；HTTPS 协议通讯加密
- 专业素质：ALM 行业多年经验；为敏捷开发量身定做；专注服务中小团队

这款 Bug 管理系统的功能以及使用方法如下：

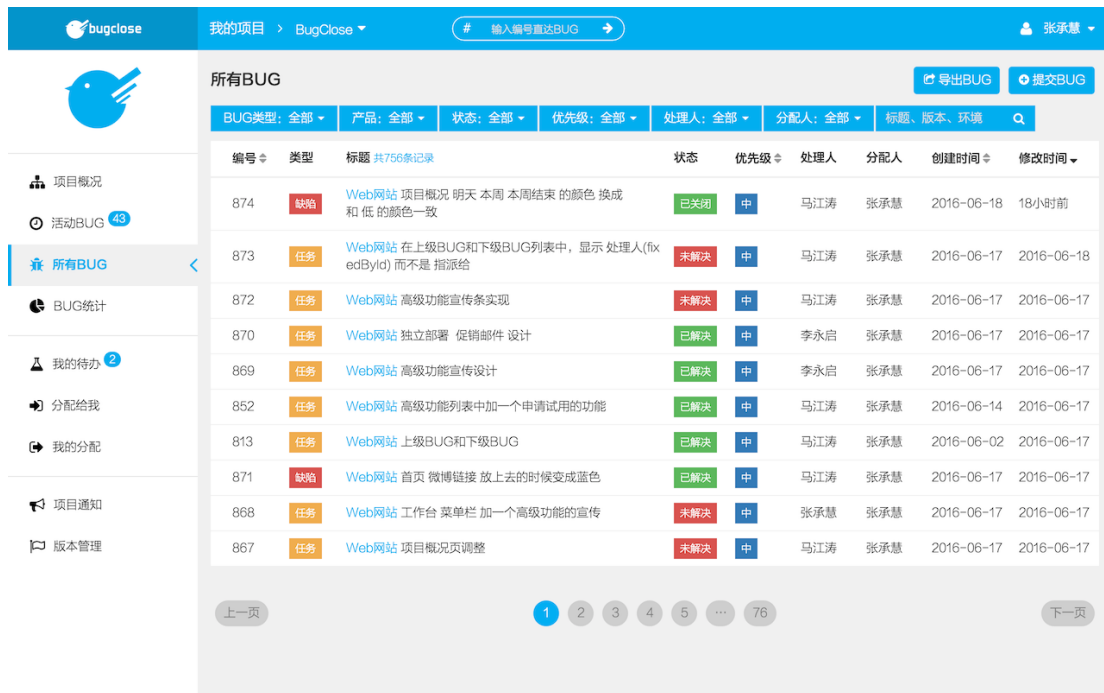
进入工作台，默认选择的是<项目概况>，<项目概况>主要是列出项目的一些现状数字，包括按状态，按类型，按优先级统计的 BUG 数量。最新通知和最新 BUG 让你了解项目的最新动态。用户可以在该页直接提交 BUG，发布版本及发布通知。另外项目管理员还可以在这里进行项目配置。



直接在网页顶端菜单栏上可以切换到不同的项目。点击我的项目可以进入项目列表；

<活动 BUG>列出了所有未完成的 BUG，重点突出了计划日期和截止日期，活动 BUG 可以当作团队的任务看板；

<所有 BUG>则提供了项目的所有 BUG 的筛选排序（包括标题，版本，环境等的模糊搜索）；



<活动 BUG>和<所有 BUG>都提供了导出功能；

当点击BUG列表或搜索BUG编号时,会从右边滑出BUG详情,随时可切换BUG,非常方便。

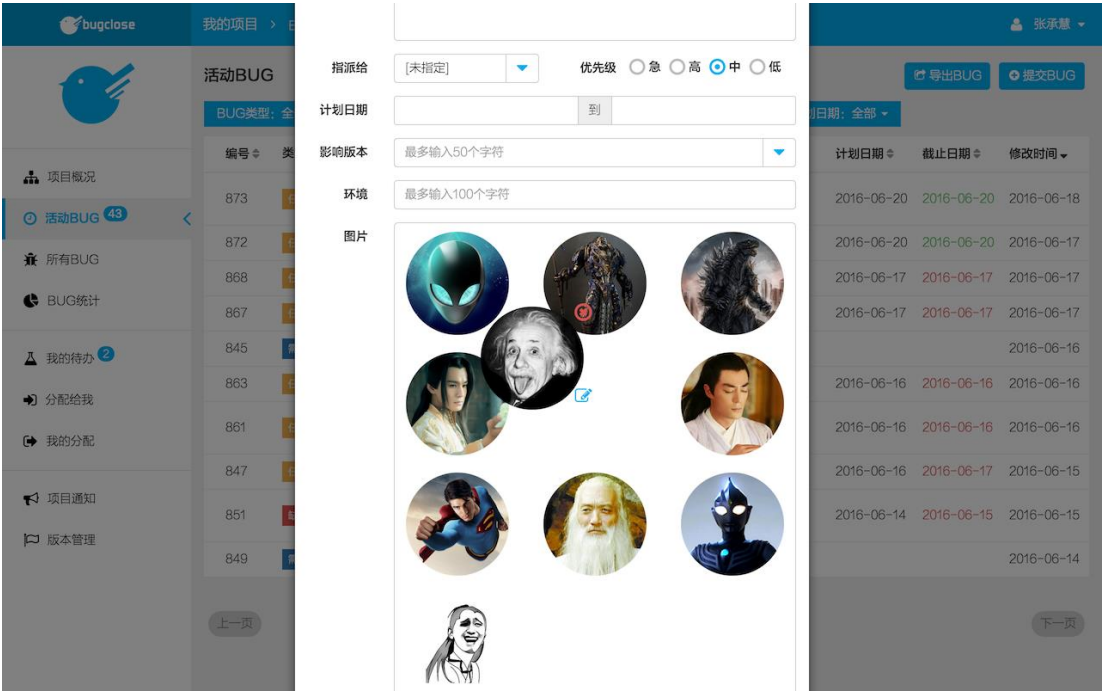
几乎每个板块都有<提交 BUG>, 点击按钮后直接弹出对话框：



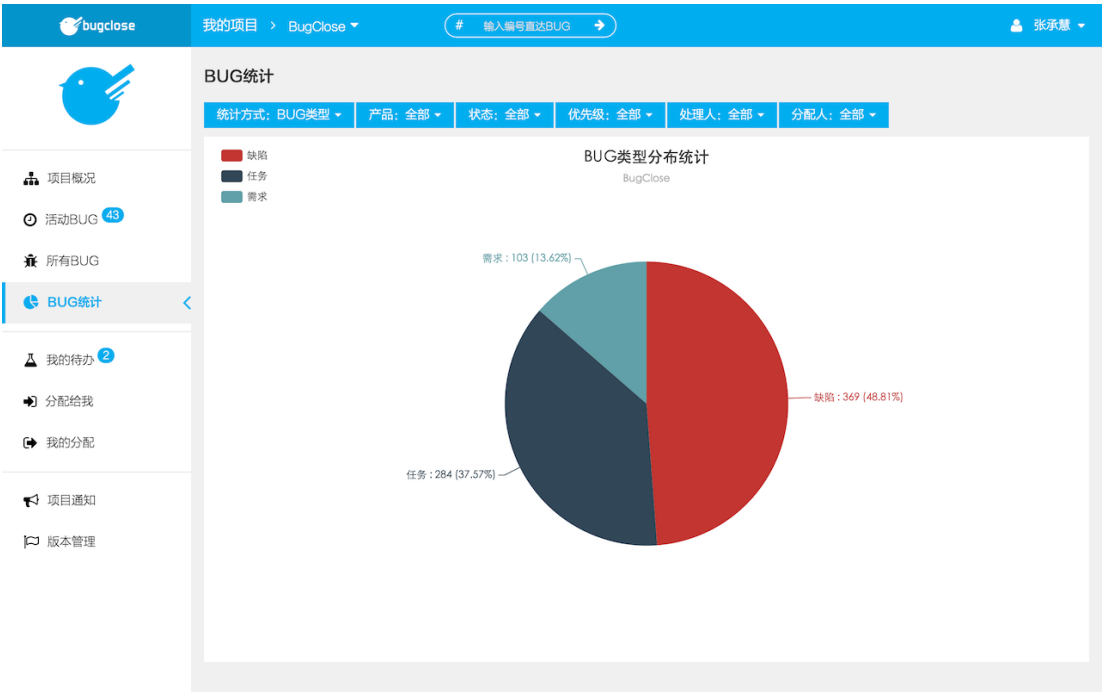
<提交 BUG>上传图片有一个重要的功能,可以直接用手机扫描二维码(微信,QQ,浏览器都可以)来上传图片,在手机上上传完图片会马上在PC的图片框内

显示出来。

<提交 BUG>时可以传多张图片，支持从其他程序拖拉图片上传，支持从剪贴板获取图片上传（仅 Chrome 内核的浏览器支持），支持上传好的图片在图片框进行拖拉排序。



<BUG 统计>主要是对 BUG 的分布统计，可以按 BUG 类型，状态，处理人，分配人，产品等进行分布统计：

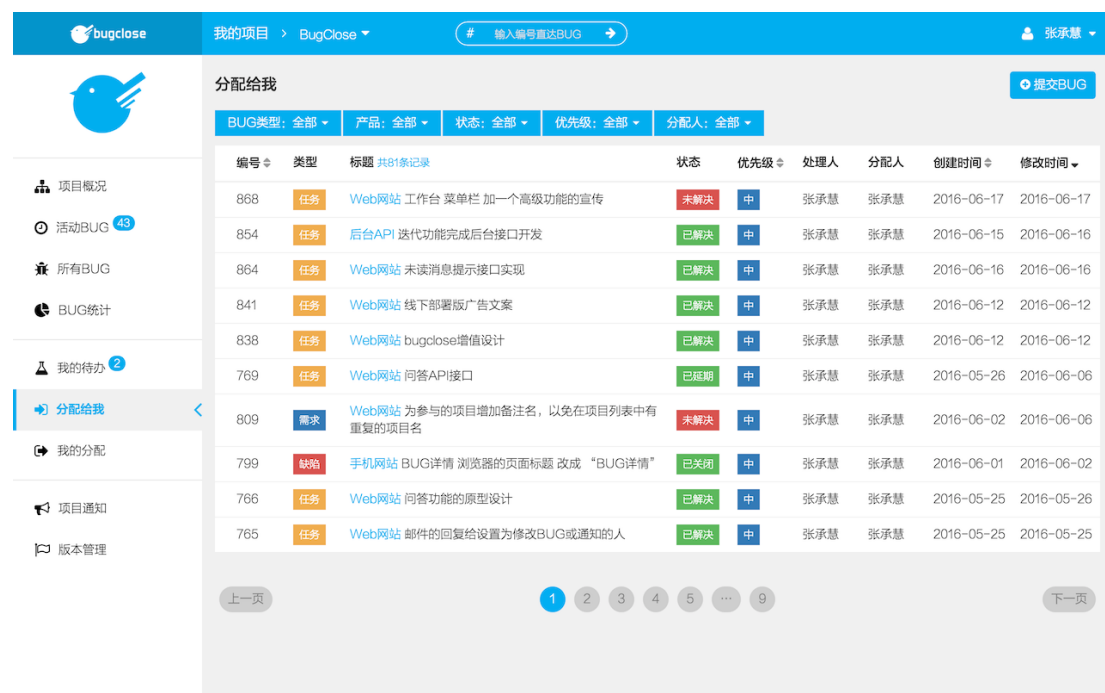


<我的待办>相当于一个待办清单，列出当前指派给你的 BUG，你可以根据优先



级或者计划日期（如今天，明天等）进行筛选。

<分配给我>主要是给开发人员使用的一个视图，该视图列出所有分配给你的BUG（包括各种状态）：



编号	类型	标题	状态	优先级	处理人	分配人	创建时间	修改时间
868	任务	Web网站 工作台 菜单栏 加一个高级功能的宣传	未解决	中	张承慧	张承慧	2016-06-17	2016-06-17
854	任务	后台API 迭代功能完成后台接口开发	已解决	中	张承慧	张承慧	2016-06-15	2016-06-16
864	任务	Web网站 未读消息提示接口实现	已解决	中	张承慧	张承慧	2016-06-16	2016-06-16
841	任务	Web网站 线下部署版广告文案	已解决	中	张承慧	张承慧	2016-06-12	2016-06-12
838	任务	Web网站 bugclose增值设计	已解决	中	张承慧	张承慧	2016-06-12	2016-06-12
769	任务	Web网站 问答API接口	已延期	中	张承慧	张承慧	2016-05-26	2016-06-06
809	需求	Web网站 为参与的项目增加备注名，以免在项目中列表中有重复的项目名	未解决	中	张承慧	张承慧	2016-06-02	2016-06-06
799	缺陷	手机网站 BUG详情 浏览器的页面标题 改成“BUG详情”	已关闭	中	张承慧	张承慧	2016-06-01	2016-06-02
766	任务	Web网站 问答功能的原型设计	已解决	中	张承慧	张承慧	2016-05-25	2016-05-26
765	任务	Web网站 邮件的回复给设置为修改BUG或通知的人	已解决	中	张承慧	张承慧	2016-05-25	2016-05-25

<我的分配>主要是给测试人员使用的一个视图，该视图列出所有你分配给别人的BUG（包括各种状态）；

<项目通知>则列出了项目的所有通知，通知分成了常用的几种类型，可以通过筛选类型和搜索标题尽快的找到以前发的通知；

<版本管理>列出了所有发布的版本，可以根据产品，发布人及产品名称进行筛选。

## 持续集成环境：

### 1、什么是持续集成环境？

持续集成（Continuous integration），简称 CI，是一种软件开发实践，即团队开发成员经常集成他们的工作，通过每个成员每天至少集成一次，每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽早地发现集成错误。

### 2、持续集成的两大优点

（1）快速发现错误。每完成一点更新，就集成到主干，可以快速发现错误，



定位错误也比较容易。

(2) 防止分支大幅偏离主干。如果不是经常集成，主干又在不断更新，会导致以后集成的难度变大，甚至难以集成。

### 3、持续集成的设计及流程

根据持续集成的设计，代码从提交到生产，整个过程有以下几步：

(1) 提交：流程的第一步，是开发者向代码仓库提交代码。所有后面的步骤都始于本地代码的一次提交（commit）。

(2) 测试（第一轮）：代码仓库对 commit 操作配置了钩子（hook），只要提交代码或者合并进主干，就会跑自动化测试。第一轮至少要跑单元测试。

(3) 通过第一轮测试，代码就可以合并进主干，就算可以交付了。交付后，就先进行构建（build），再进入第二轮测试。所谓构建，指的是将源码转换为可以运行的实际代码，比如安装依赖，配置各种资源（样式表、JS 脚本、图片）等等。常用的构建工具如下：

- Jenkins

- Travis

- Codship

- Strider

(4) 测试（第二轮）：第二轮是全面测试，单元测试和集成测试都会跑，有条件的话，也要做端对端测试。所有测试以自动化为主，少数无法自动化的测试用例，就要人工跑。

(5) 部署：通过了第二轮测试，当前代码就是一个可以直接部署的版本（artifact）。将这个版本的所有文件打包（tar filename.tar \*）存档，发到生产服务器。生产服务器将打包文件，解包成本地的一个目录，再将运行路径的符号链接（symlink）指向这个目录，然后重新启动应用。这方面的部署工具有 Ansible，Chef，Puppet 等。

(6) 回滚：一旦当前版本发生问题，就要回滚到上一个版本的构建结果。最简单的做法就是修改一下符号链接，指向上一个版本的目录。

### 4、如何快速搭建持续集成环境？

Jenkins 是一个配置简单和使用方便的持续集成服务器，以下以 Mac 环境安装为例：

## 1. homebrew 安装 Jenkins

### brew install Jenkins

```
wusideMacBook-Pro:Desktop Sophia$ brew install jenkins
Updating Homebrew...
```

```
> Using the sandbox
> Downloading http://mirrors.jenkins-ci.org/war/2.53/jenkins.war
> Downloading from http://mirrors.tuna.tsinghua.edu.cn/jenkins/war/2.53/jenkins.war
##### 100.0%
> jar xvf jenkins.war
> Caveats
Note: When using launchctl the port will be 8080.

To have launchd start jenkins now and restart at login:
  brew services start jenkins
Or, if you don't want/need a background service you can just run:
  jenkins
> Summary
📦 /usr/local/Cellar/jenkins/2.53: 7 files, 66.6MB, built in 27 seconds
wusideMacBook-Pro:Desktop Sophia$
```

## 2. 链接 launchd 配置文件

### ln -sfv /usr/local/opt/jenkins/\*.plist ~/Library/LaunchAgents

如果要其他机器也可以访问，把 ip 地址改为广播地址:<string>--  
httpListenAddress=0.0.0.0</string>

## 3. 修改完成,执行命令行:

### launchctl load ~/Library/LaunchAgents/homebrew.mxcl.jenkins.plist

## 4. 命令行启动 Jenkins

### Jenkins

```
wusideMacBook-Pro:Desktop Sophia$ ln -sfv /usr/local/opt/jenkins/*.plist ~/Library/LaunchAgents
/Users/Sophia/Library/LaunchAgents/homebrew.mxcl.jenkins.plist -> /usr/local/opt/jenkins/homebrew.mxcl.jenkins.plist
wusideMacBook-Pro:Desktop Sophia$ launchctl load ~/Library/LaunchAgents/homebrew.mxcl.jenkins.plist
wusideMacBook-Pro:Desktop Sophia$ jenkins
Running from: /usr/local/Cellar/jenkins/2.53/libexec/jenkins.war
webroot: $user.home/.jenkins
四月 09, 2017 8:35:01 下午 Main deleteWinstoneTempContents
警告: Failed to delete the temporary Winstone file /var/folders/yk/0yczvk994k55fx0y82s9d26c0000gn/T/winstone/jenkins.war
四月 09, 2017 8:35:02 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: Logging initialized @739ms
四月 09, 2017 8:35:02 下午 winstone.Logger logInternal
信息: Beginning extraction from war file
四月 09, 2017 8:35:02 下午 org.eclipse.jetty.util.log.JavaUtilLog warn
警告: Empty contextPath
四月 09, 2017 8:35:02 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: jetty-9.2.z-SNAPSHOT
四月 09, 2017 8:35:03 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
Jenkins home directory: /Users/Sophia/.jenkins found at: $user.home/.jenkins
四月 09, 2017 8:35:06 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: Started w.@202b0582{/file:/Users/Sophia/.jenkins/war/,AVAILABLE}{/Users/Sophia/.jenkins/war}
四月 09, 2017 8:35:06 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: Started ServerConnector@2152b527{HTTP/1.1}{0.0.0.0:8080}
四月 09, 2017 8:35:06 下午 org.eclipse.jetty.util.log.JavaUtilLog info
信息: Started @4624ms
```

## 5. 打开浏览器,输入 localhost:8080 就可看到 Jenkins 的 web 界面

## 6. Jenkins 的一些操作

http://localhost:8080/exit //退出 Jenkins

`http://localhost:8080/restart` //重启

`http://localhost:8080/reload` //重新加载

7. 您可以有选择地安装一些插件，常见插件

**Git plugin git 集成**

**Workspace Cleanup Plugin 构建前工作空间清理**

**build timeout plugin 构建超时设置**

**user build vars plugin build variables**

**Post-Build Script Plug-in 执行脚本**

**Log Parser Plugin log**

**Gradle plugin gradle 插件**