



MOVIETIME 电影售票系统

开发工具学习报告

系统分析与设计 MovieLight 小组

一、Git

1. 什么是 git

git 是用于文件版本管理的分布式系统。通过 git，可以保留文件的历史版本以便随时恢复到指定时间点。

2. git 的安装

在 linux 下输入 `sudo apt-get install git` 即可完成安装.

在 Windows 下需要从 <https://git-for-windows.github.io> 下载.

3. 创建版本库

版本库是一个在 git 版本管理之下的目录。版本库中每个文件的增删查改都将被 git 跟踪.

创建方法:

在 bash 或者 windows 下的 git bash 输入以下命令

```
$ mkdir testgit  
  
$ cd testgit  
  
$ git init
```

这样就在当前目录建立了一个版本库.

4. git 的工作流程

创建新版本: 修改文件 -> 将修改后的文件添加到暂存区 -> 提交.

如上操作之后，就为版本库添加了一个可以恢复到的新状态.

合并分支: 添加分支 -> 在新分支中提交代码 -> ...(不断提交代码) -> 合并分支

合并分支可用于同时进行多个功能的开发. 使用分支功能后, 这些开发工作可以在一个稳定版本的基础上分别进行, 最后只要合并分支, 就能将对代码的修改合并到一起.

5. 具体命令列表

(1) 版本控制

将 code.java 添加到暂存区

```
$ git add code.java
```

将 code.java 从暂存区删除

```
$ git rm --cache code.java
```

提交修改

```
$ git commit -m "message"
```

查询暂存区

```
$ git status
```

文件状态对比(将当前文件状态与上次状态对比)

```
$ git diff code.java
```

删除文件

```
$ rm code.java
```

```
$ git rm code.java
```

从上一个版本恢复误删的文件

```
$ git checkout --test.txt
```

(2) 分支管理

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

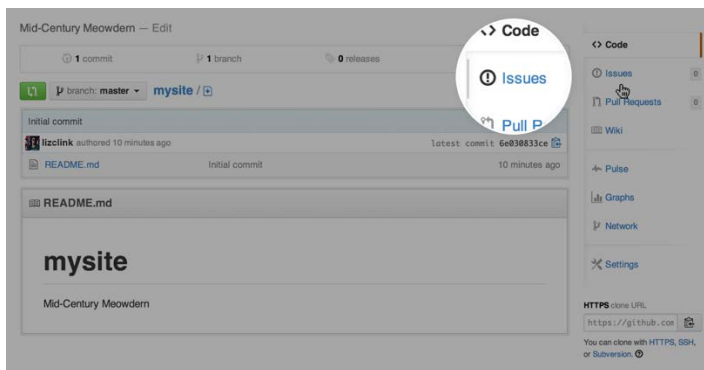
合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

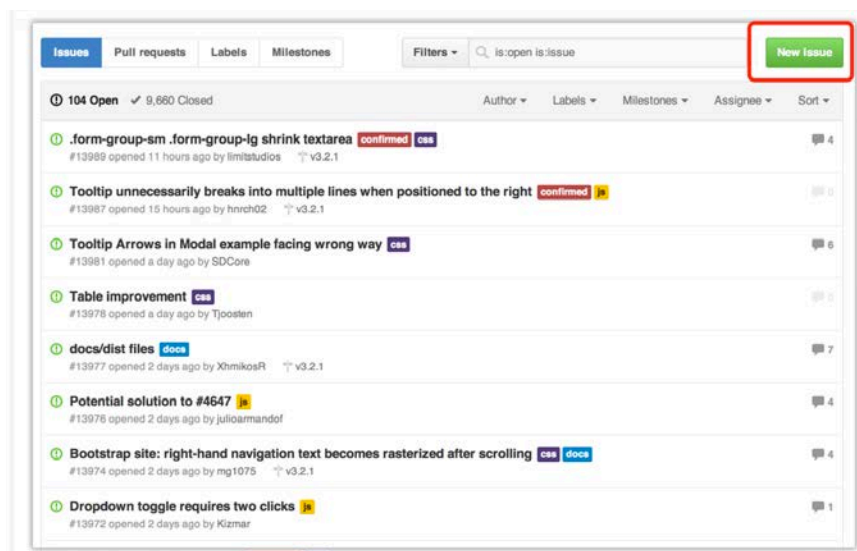
二、Github Bug Tracker

本次项目我们使用 github 的 Issues 进行 BUG 管理.操作流程如下

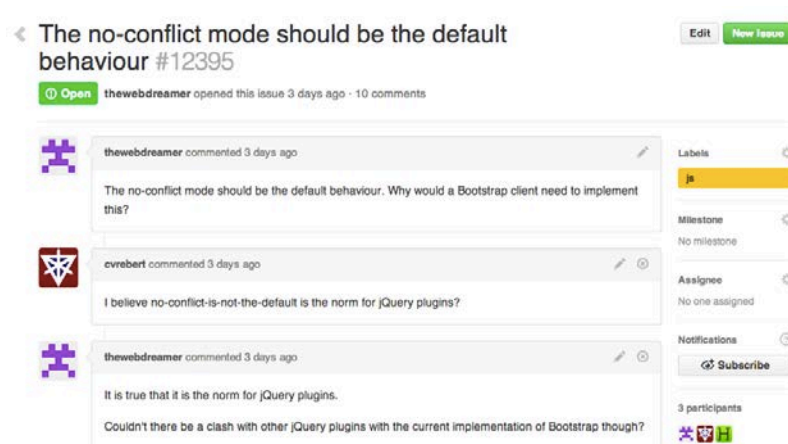
在 github 中打开 Issue 界面:



点击添加新 Issues :



填入相关信息后提交, 得到 Issues 详情如下



其中 Title 和 Description 用于描述 BUG. label 和 milestone 用于对 Bug 进行

分类. assignee 用于指派 bug 的处理者, comments 用于其他人 review 并评论.

三、Jenkins 持续集成环境

1. 简介

Jenkins 是一个持续开发环境。在现实中, 开发环境和用户使用的环境常常有所区别, 如可能会出现程序使用了一些开发环境的依赖库, 而用户环境并没有安装。Jenkins 可以在每次提交代码之后实现用户环境下程序测试的自动化。

2. 部署方法

(1) 安装 Jenkins

在官网下载 war 包, 然后解压, 把得到的 war 包直接扔到 tomcat 下, 启动 tomcat, Jenkins 就安装完毕, 访问 <http://localhost:8080>, 即可进入 Jenkins 控制目录

2. 新建一个 Job, 设置好相关信息



项目名称

JavaWeb

描述

[Plain text] [预览](#)

☒ 丢弃旧的构建

Strategy

Log Rotation

保持构建的天数

2

如果非空，构建记录将保存此天数

保持构建的最大个数

5

如果非空，最多此数目的构建记录将被保存

☒ 参数化构建过程

Choice

Name

VERSION

Choices

1.0.1

Description

产品系统 / 模块的版本号

[Plain text] [预览](#)

高级...

源码管理

☐ None

☐ CVS

☒ CVS Projectset

☒ Subversion

☐ Modules

Repository URL

svn://10.75.4.25/lin/JavaWeb

Local module directory (optional)

Repository depth

infinity

Ignore externals

☐

Add more locations...

Check-out Strategy

Use 'svn update' as much as possible

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

源码库浏览器

(自动)

ctset

1

Repository URL

svn://10.75.4.25/lin/JavaWeb

Unable to access **svn://10.75.4.25/lin/JavaWeb** : svn: E210003: connection refused by the server [\(show details\)](#)
(Maybe you need to enter credential?)

Local module directory (optional)

3. 配置 JDK 和 Maven

构建触发器

☐ Build whenever a SNAPSHOT dependency is built

☐ Build after other projects are built

☐ Build periodically

☐ Poll SCM

Pre Steps

Add pre-build step

Build

Maven Version Jenkins needs to know where your Maven is installed. Please do so from the system configuration.

Root POM pom.xml

Goals and options

JDK

JDK 安装 ☐ JDK 别名 jdk1.6 JAVA_HOME D:\Java\jdk1.6.0_45 ☐ 自动安装

新增 JDK 系统下 JDK 安装列表

Ant

Ant 安装 新增 Ant 系统下 Ant 安装列表

Maven

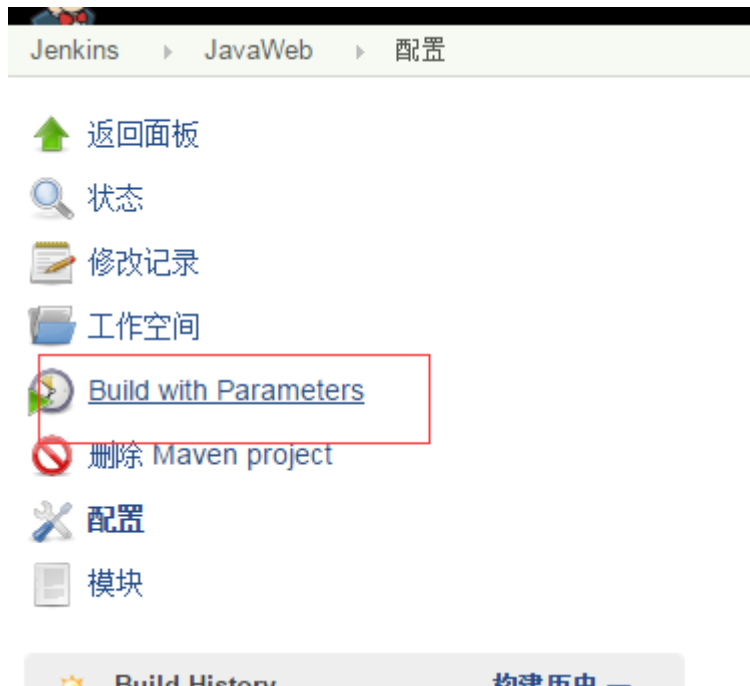
Maven 安装 ☐ Maven Name maven MAVEN_HOME D:\Java\Tool\apache-maven-3.2.1 ☐ 自动安装

新增 Maven 系统下 Maven 安装列表

Maven 项目配置

全局 MAVEN_OPTS

4. 构造项目



成功输出如下


```
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
[INFO]
[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ JavaWeb ---
[INFO] Packaging webapp [JavaWeb] in [C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\target\JavaWeb-0.0.1-SNAPSHOT]
[INFO] Assembling webapp [JavaWeb] in [C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\src\main\webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\src\main\webapp]
[INFO] Webapp assembled in [122 msecs]
[INFO] Building war: C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\target\JavaWeb-0.0.1-SNAPSHOT.war
[WARNING] Warning: selected war files include a WEB-INF/web.xml which will be ignored
(webxml attribute is missing from war task, or ignoreWebxml attribute is specified as 'true')
[INFO]
[INFO] >>> maven-source-plugin:2.4:jar (attach-sources) @ JavaWeb >>>
[WARNING] Failed to getClass for org.apache.maven.plugin.source.SourceJarMojo
[INFO]
[INFO] <<< maven-source-plugin:2.4:jar (attach-sources) @ JavaWeb <<<
[INFO]
[INFO] --- maven-source-plugin:2.4:jar (attach-sources) @ JavaWeb ---
[INFO] No sources in project. Archive not created.
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ JavaWeb ---
[INFO] Installing C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\target\JavaWeb-0.0.1-SNAPSHOT.war to D:\JavaRepository\com\lin\JavaWeb\0.0.1-SNAPSHOT\
[INFO] Installing C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\pom.xml to D:\JavaRepository\com\lin\JavaWeb\0.0.1-SNAPSHOT\JavaWeb-0.0.1-SNAPSHOT.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.875 s
[INFO] Finished at: 2016-01-14T17:54:07+08:00
[INFO] Final Memory: 15M/167M
[INFO]
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\pom.xml to com.lin\JavaWeb\0.0.1-SNAPSHOT\JavaWeb-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\Users\linbingwen\.jenkins\workspace\JavaWeb\JavaWeb\target\JavaWeb-0.0.1-SNAPSHOT.war to com.lin\JavaWeb\0.0.1-SNAPSHOT\JavaWeb-0.0.1-SNAPSHOT.war
channel stopped
Finished: SUCCESS
```

5. 可以在工作目录找到 .war 文件。 这就是输出。 可见编译是成功的。