

一、git作为项目管理工具

1. 版本控制：是一种在开发的过程中用于管理我们对文件、目录或工程等内容的修改历史，方便查看更改历史记录，备份以便恢复以前的版本的软件工程技术（**实现多人协同开发项目的技术**）。

2. 仓库：存放项目代码

3. git有三个区域 工作区域、暂存区、本地仓库 git status查看文件的状态
git add将文件添加到暂存区 git commit将文件添加到本地仓库 -m 添加描述

4. git与svn的区别

•

5. 基本命令

- 初始化：git init
- 创建文件/目录：vi 文件名 mkdir 目录名
- 将文件添加到暂存区：git add 文件名 -A(代表所有修改) -u(修改以及删除)。(修改以及新建)
- 将文件提交到本地仓库：git commit -m "描述" 文件名(冲突时不需要加文件名)
- 本地删除：rm -rf(递归强制) 文件名/目录名
- 版本库中删除：git rm 文件名 并提交 git commit -m "描述"
- 查看日志：git log --oneline
- 查看版本变化日志：git reflog
- 恢复之前的版本(删除恢复)：git reset --hard 索引（在上一条命令中可以查看）或者 --head^

4.3.6 reset 命令的三个参数对比

➤ --soft 参数

- 仅仅在本地库移动 HEAD 指针



➤ --mixed 参数

- 在本地库移动 HEAD 指针
- 重置暂存区



➤ --hard 参数

- 在本地库移动 HEAD 指针
- 重置暂存区
- 重置工作区

- 删除恢复(未提交到本地仓库): `git reset --hard HEAD` (操作head指针)
- 比较文件差异 (缓存区与本地仓库) : `git diff 文件名`
- 比较文件差异 (工作与本地仓库) : `git diff 本地仓库版本 文件名`
- 分支管理:

4.4.1 什么是分支？

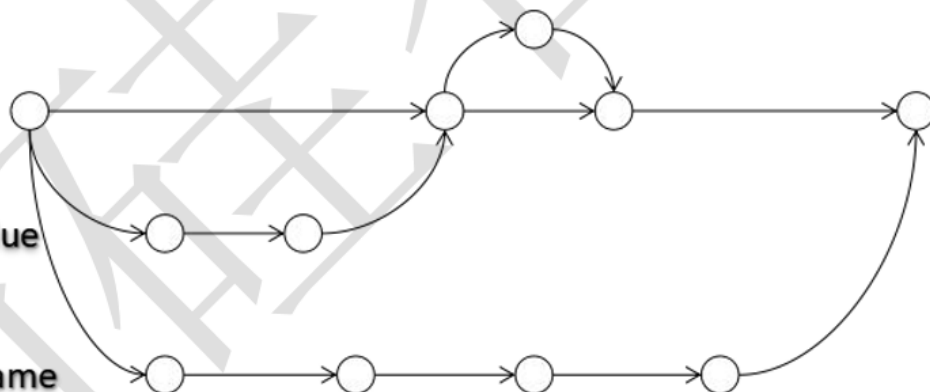
在版本控制过程中，使用多条线同时推进多个任务。

hot_fix

master

feature_blue

feature_game



4.4.2 分支的好处？

- 同时并行推进多个功能开发，提高开发效率
- 各个分支在开发过程中，如果某一个分支开发失败，不会对其他分支有任何影响。失败的分支删除重新开始即可。

4.4.3 分支操作

- 创建分支
`git branch [分支名]`
- 查看分支
`git branch -v`
- 切换分支
`git checkout [分支名]`
- 合并分支
 - 第一步：切换到接受修改的分支（被合并，增加新内容）上
`git checkout [被合并分支名]`

- 第二步：执行 merge 命令
git merge [有新内容分支名]

➤ 解决冲突

- 冲突的表现

```

8 gggggggg
9 <<<<<< HEAD
10 hhhhhhhh edit by hot_fix
11 =====
12 hhhhhhhh edit by master
13 >>>>>> master
14 iiiiiiii
15 jjjjjjjj
  
```

- 冲突的解决

- ◆ 第一步：编辑文件，删除特殊符号
- ◆ 第二步：把文件修改到满意的程度，保存退出
- ◆ 第三步：git add [文件名]
- ◆ 第四步：git commit -m "日志信息"
 - 注意：此时 commit 一定不能带具体文件名

- 创建远程库别名(github, 码云): git remote add 别名 别名地址
- 查看创建的远程库别名: git remote -v
- 克隆一个远程库到本地: git clone 地址
- 将本地文件上传到远程库: git push 别名 远程库分支
- 分支: git fetch 别名 远程库分支 (将远程库的内容获取到本地, 但是没有合并到本地可用 git checkout 别名/远程库分支名 查看)
- 合并: git merge 别名/远程库分支 (pull = fetch+merge) (将远程库合并到本地)
- 解决冲突: 冲突时不允许上传远程库, 需要git pull 别名 远程库分支 将远程库拉去到本地修改后才能提交(注意: 在提交到本地库时 后面不能带文件名)

二、手动建立maven项目（多个maven项目、父子依赖）

1. 单maven项目

- idea构建maven项目
- 新建templates文件夹存储html文件
- 新建static文件夹存储js、css、image
- 新建application.yml存放配置信息
- 在pom文件中添加

```

2  <groupId>org.springframework.boot</groupId>
3  <artifactId>spring-boot-starter-parent</artifactId>
4  <version>2.1.9.RELEASE</version>
5  <relativePath/> <!-- lookup parent from repository -->
6  </parent>
7  <dependencies>
8  <dependency>
9  <groupId>org.springframework.boot</groupId>
10 <artifactId>spring-boot-starter-web</artifactId>
11 </dependency>
12
13 <dependency>
14 <groupId>org.springframework.boot</groupId>
15 <artifactId>spring-boot-starter-test</artifactId>
16 <scope>test</scope>
17 </dependency>
18 </dependencies>
19 <build>
20 <plugins>
21 <plugin>
22 <groupId>org.springframework.boot</groupId>
23 <artifactId>spring-boot-maven-plugin</artifactId>
24 </plugin>
25 </plugins>
26 </build>

```

2.父子项目

```

1  // parent中
2  <?xml version="1.0" encoding="UTF-8"?>
3  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apac
he.org/xsd/maven-4.0.0.xsd">
5  <modelVersion>4.0.0</modelVersion>
6  <parent>
7  <groupId>org.springframework.boot</groupId>
8  <artifactId>spring-boot-starter-parent</artifactId>
9  <version>2.1.9.RELEASE</version>
10 <relativePath/> <!-- lookup parent from repository -->
11 </parent>

```

```
12 <groupId>com.pay</groupId>
13 <artifactId>parent</artifactId>
14 <version>0.0.1-SNAPSHOT</version>
15 <name>parent</name>
16
17 <packaging>pom</packaging>
18
19 <description>parent for Spring Boot</description>
20
21
22 <modules>
23 <module>wechat</module>
24 </modules>
25
26 <properties>
27 <java.version>1.8</java.version>
28 </properties>
29
30 <dependencies>
31
32 </dependencies>
33
34 <build>
35 <plugins>
36 <plugin>
37 <groupId>org.springframework.boot</groupId>
38 <artifactId>spring-boot-maven-plugin</artifactId>
39 </plugin>
40 </plugins>
41 </build>
42
43 </project>
44
45
46 // 子项目
47 <?xml version="1.0" encoding="UTF-8"?>
48 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance"
49 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apa
che.org/xsd/maven-4.0.0.xsd">
50 <modelVersion>4.0.0</modelVersion>
51
```

```
52 <parent>
53 <groupId>com.pay</groupId>
54 <artifactId>parent</artifactId>
55 <version>0.0.1-SNAPSHOT</version>
56 </parent>
57
58 <groupId>com.pay</groupId>
59 <artifactId>admin</artifactId>
60 <version>0.0.1-SNAPSHOT</version>
61 <name>admin</name>
62 <description>admin for Spring Boot</description>
63
64 <properties>
65 <java.version>1.8</java.version>
66 </properties>
67
68 <dependencies>
69 <dependency>
70 <groupId>org.springframework.boot</groupId>
71 <artifactId>spring-boot-starter-web</artifactId>
72 </dependency>
73 </dependencies>
74
75
76 </project>
77
```

三、微信登录

- 网站微信登录流程
- 扫一扫微信登录
- 微信登录的api使用（知道微信服务器的回调流程以及各个参数的含义）

四、微信支付

1. 生成微信支付二维码
2. 扫一扫付款

