Node.js核心模块

第三方扩展：Express/Koa/Hopi、Socket.IO、NPM(前端/后台)

Git => Github => 新浪云上传Node项目

1.使用Git

项目中代码文件的共享、存储、合并、历史对比、回退等 —— Source Code Management

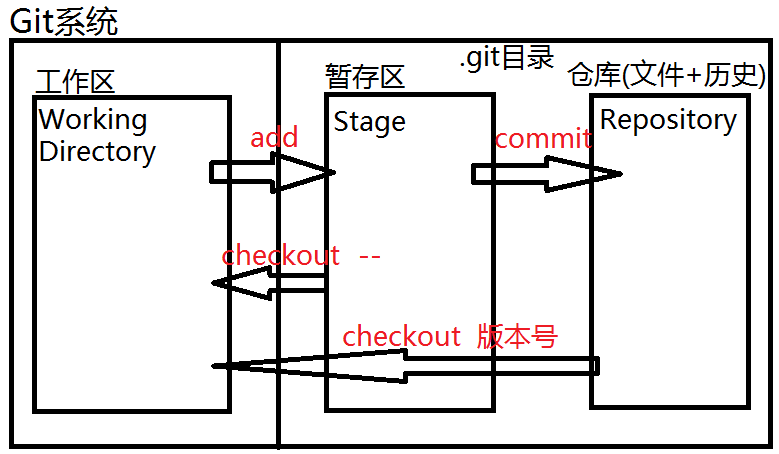
解决方法：每次保存项目任何文件都自动给项目一个“快照”，并分配一个唯一的版本号(Revesion)，供以后查找、对比、回退 —— Version Control System（VCS）——所有的项目文件+“快照”(历史记录)称为“代码仓库(Repository)”

常见的VCS软件有：

**(1)集中式VCS：** SVN等，需要安装服务器(仓库)和客户端，版本号可以使用1/2/3...

**(2)分布式VCS：** Git等，每个客户端同时也是服务器(仓库)，本地仓库可以和远程仓库合并，版本号只能是随机数

Linux操作系统 => Linus开发了一套最快的SCM => Git



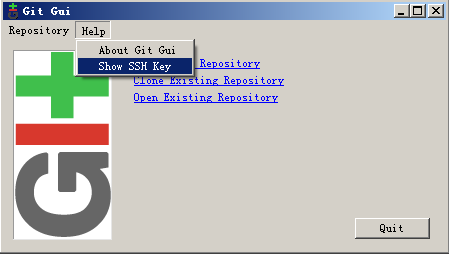
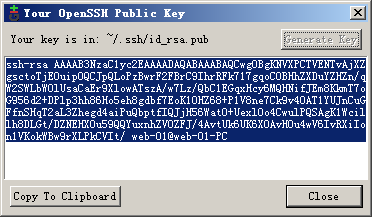
使用步骤：

(1)下载地址：https://git-scm.com

(2)安装Git：一路Next即可

-----------------------使用前设置------------------------

(3)生成自己的私钥和公钥，用于今后的远程仓库文件传输

(4)为Git指定当前操作用户的用户名和邮箱

git config --global user.name "LiWenhua"

git config --global user.email "lwh@tedu.cn"

git config --list 查看是否已经修改成功

-------------------------初始化工作区----------------------

(5)创建一个项目目录，将其初始化为“工作区”，生成仓库必需的文件

git init

--------------------------基础使用--------------------------

(6)创建/拷贝项目文件到工作区，添加到暂存区

git add 文件名 #添加工作区中指定文件到暂存区

git add . #添加工作区中所有文件到暂存区

(7)把暂存区内容提交到仓库中，生成版本号，永久存储

git commit -m "说明文字" #将暂存区内容提交到仓库

git commit -am "说明文字" #将工作区中的修改添加到暂存区再立即提交到仓库

|  |
| --- |
| 扩展命令：  git status：查看当前的工作区/暂存区的状态  git log：查看仓库的保存日志 |
| 在当前工作区中创建一个.gitignore文本文件，其中可以添加想被Git仓库忽略的所有文件和目录，提示：目录必须以/结尾 |
| 在命令行中创建一个目录： mkdir 目录名  在命令行中创建一个空文件： touch 文件名 |

------------------------撤销和回退---------------------------

(8)把暂存区中的内容撤销到工作区(最新版本号无变化)

git checkout -- 文件名

(9)把仓库中指定版本的内容撤销到暂存区和工作区(最新版本号无变化)

git checkout 版本号 文件名

(10)把整个仓库中的最新版本回退到历史上的某个版本(最新版本号变化)

git reset --hard 版本号

|  |
| --- |
| git log #查看最新版本(HEAD)以及之前的版本号  git reflog #查看最新版本号的变化历史 |
| 面试题： git checkout 和 git reset 的异同？ |

------------------------分支管理-----------------------------

项目已经到达某个“里程碑”，可以单独发布运行；后续仍然要继续开发，新增的内容不希望影响到已经稳定运行的内容 —— 再开一个新的分支(branch)即可。

(11)创建一个新的分支

git branch 分支名

(12)切换到另一个分支，就是将HEAD移动到指定分支的最新版

git checkout 分支名

(13)把其它分支合并到当前分支

git checkout master #切换到主分支

git merge mobile #把指定的mobile分支内容当前分支

|  |
| --- |
| 扩展：  git branch #查看当前仓库中已有的所有分支 |

2.使用远程仓库

Git中的每个子系统都有完整的仓库，即使没有联网。项目中，一般需要有一个公共的远程仓库（7x24小时服务），每个Git都可以随时把自己的仓库与此仓库做同步！如何搭建远程Git仓库：

方法1：在公司创建一个Git服务器，搭建远程Git仓库。

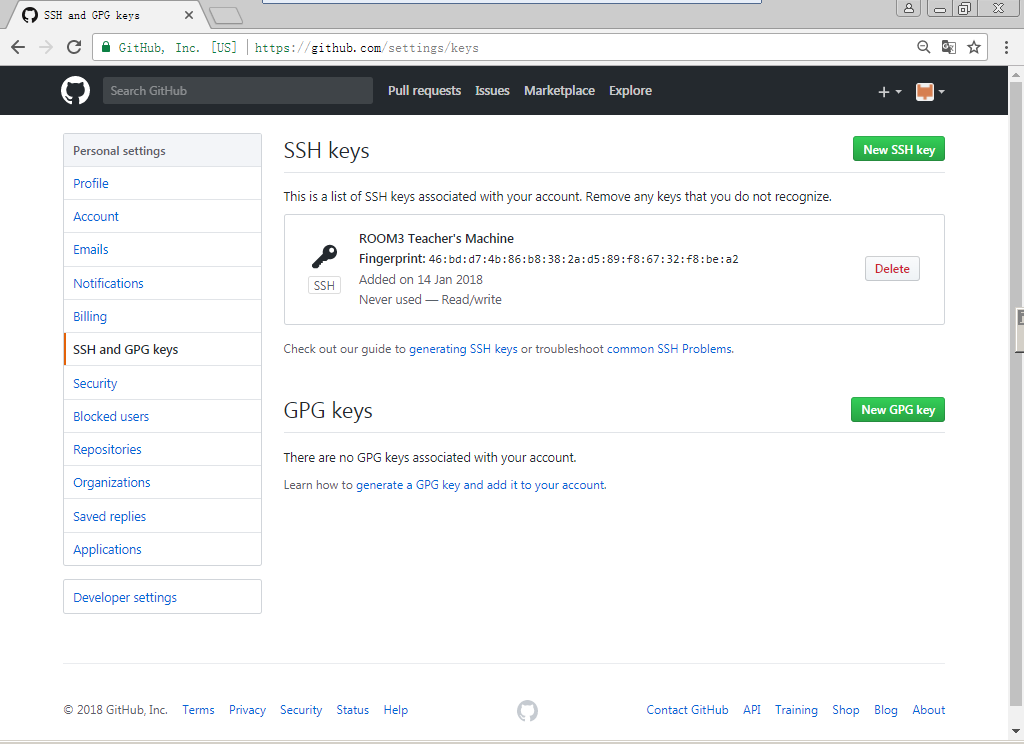
方法2：把本公司的项目托管在某个公共的Git服务器上，如Github.com。

(1)在github.com注册账户 Sign Up

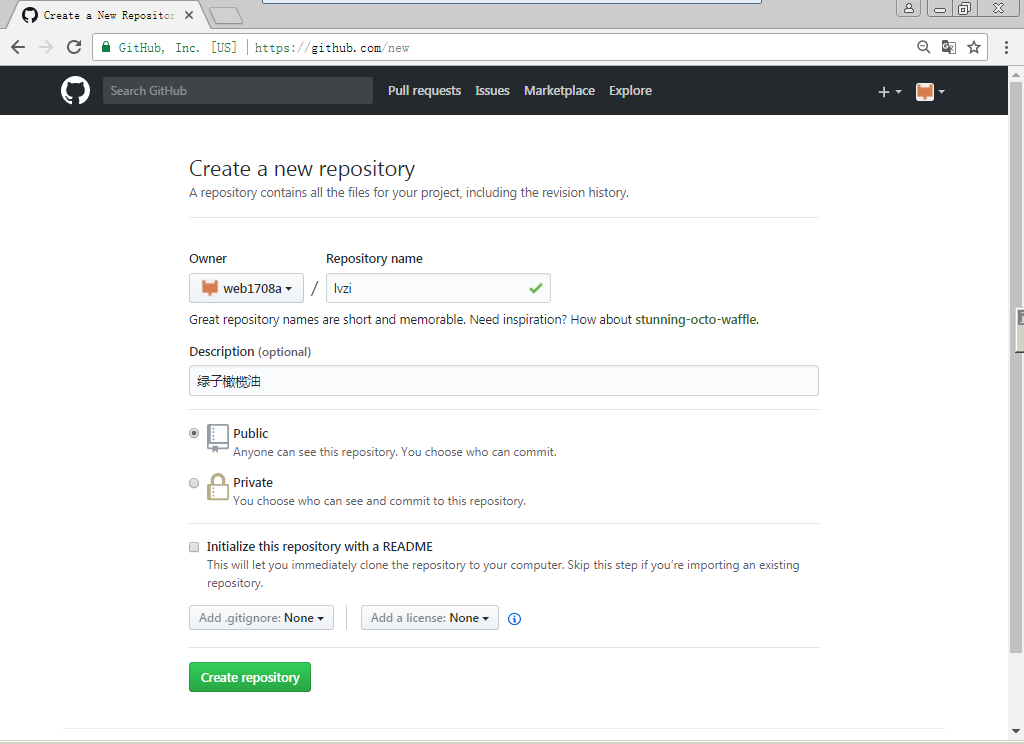
记得点击邮箱中的确认邮件，以完成注册

(2)若希望向github上传代码，必须告诉它用户自己的SSH公钥

Settings => SSH and GPG Keys => New ssh key



(3)创建自己的项目仓库(远程仓库)



每个项目的仓库都有两种地址：HTTPS协议和SSH协议，后者的访问效率更高些。

---------------------------------------------------------

(4)在自己的电脑上克隆一份远程的仓库

git clone 远程仓库的地址

(5)在本地仓库中添加项目文件、添加到暂存区、提交到仓库

(6)把本地仓库中的内容“推送(push)”到远程仓库

git push 远程仓库的地址

(7)如果远程仓库中的内容被其它的客户端修改了，本地仓库需要重新“拉取(pull)”最新内容

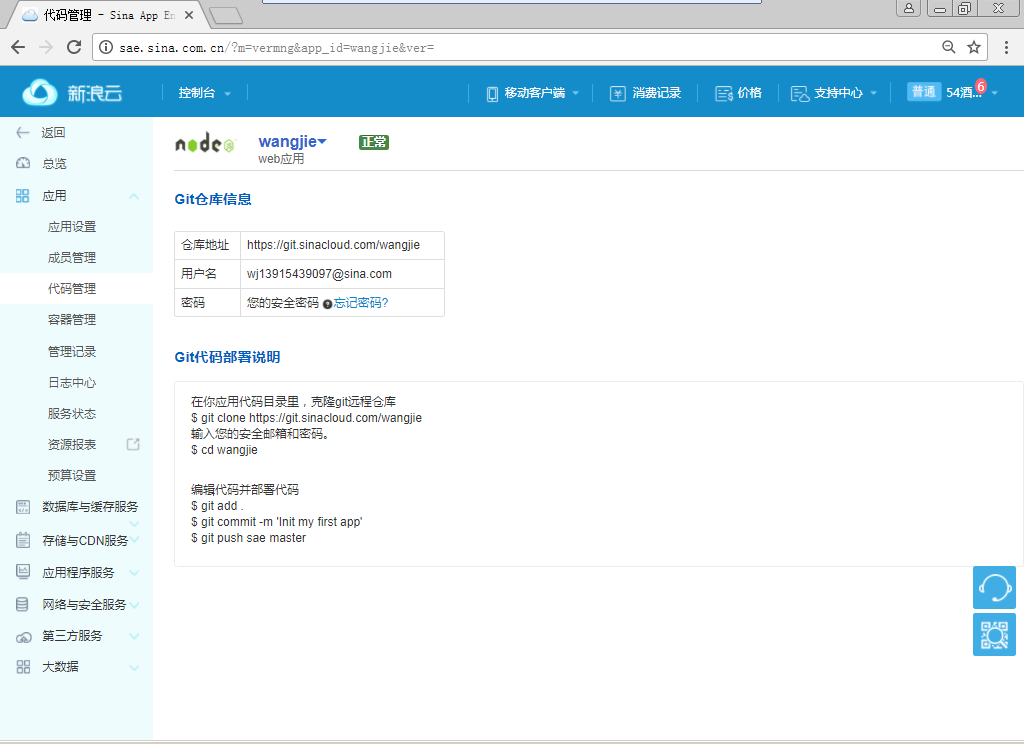
git pull 远程仓库的地址 本地分支名

|  |
| --- |
| 扩展知识：可以把一个远程仓库的地址在本地取一个别名，以后的push和pull就可以直接使用该别名  git remote add 别名 远程仓库的地址 |
| 所有人都可以clone或pull一个公共的仓库内容；而只有项目创建者和合作开发者可以push仓库内容。若想让其他人共同来修改一个项目的仓库内容，可以由项目创建者邀请其他人称为项目仓库的合作开发者（Collaborar）。  项目的Settings => Collaborators |

**绿子橄榄油项目仓库地址：git@github.com:web1708a/lvzi.git**

3.新浪云部署Node云应用

(1)创建一个Node.js云应用（Docker环境已经配置好Node.js解释器），会自动在新浪云中创建一个远程Git仓库



(2)在本地克隆一份远程仓库

git clone 远程仓库地址

(3)在本地编写Node.js应用，要求Node.js监听5050端口；在项目根目录下创建package.json文件，要求声明"scripts"{ "start": "node x.js" }，因为新浪云只会使用 npm start启动咱们的node应用

(4)添加到本地暂存区，提交给本地仓库

(5)把本地仓库中的内容推送到远程仓库

git push 远程仓库地址

更多细节参考使用手册：http://www.sinacloud.com/doc/sae/nodejs.html

4.WebStorm整合Git客户端

克隆远程仓库 => 本地编辑 => 本地仓库添加/提交 => PUSH到远程仓库。

