**Git Skeleton User Guide**

1. git是什么？

Git是由Linus Torvalds最先创建开发的代码管理工具。它具有速度快，开源，分布式管理的特点。不同于其他的代码管理工具（如SVN，ClearCase等），它是利用快照snapshot，而不是changesets的机制来管理代码。

1. 客户端的安装和配置
2. 在windows下安装bash git 客户端Git-1.6.5.1-preview20091022.exe
3. 在git bash的console中配置git
4. 配置 username

git config --global user.name "xuhansheng"

1. 配置 email

git config --global user.email [xuhansheng@gmail.com](mailto:xuhansheng@gmail.com)

1. 生成SSH public key，用于在github服务端注册。每台客户端key只需要生成一次，然后备份起来起来即可。

cd ~/.ssh

ssh-keygen -t rsa -C "[xuhansheng@gmail.com](mailto:xuhansheng@gmail.com)"//上述step 4配置的email，一路回车

打开C:\Documents and Settings\Administrator\.ssh\id\_rsa.pub,里面的内容就是SSH public key。

1. 服务端配置
2. 申请帐号

<http://github.com/plans>，免费帐号服务端有300M的磁盘空间

1. 在服务端创建项目名，本文档以StompTest为例。
2. <https://github.com/account>的SSH public key中增加客户端生成的key。
3. 客户端初始化项目（initializing a new git repo）

Sample：

mkdir StompTest

cd StompTest

git init

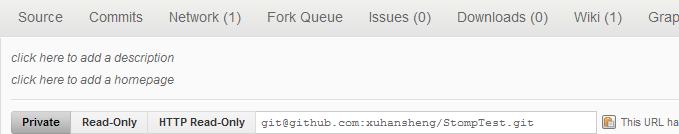
git add .

git commit -m 'initial commit'

git remote add origin git@github.com:xuhansheng/StompTest.git

git push origin master

1. 从服务端克隆一个存在的项目



git clone [git@github.com:xuhansheng/StompTest.git](mailto:git@github.com:xuhansheng/StompTest.git)

注意clone后面的参数url 需要用private里面url。Read-Only和HTTP Read-Only clone下来的代码没有修改服务端代码的权限。

1. 通常的使用流程
2. 从服务端同步代码

git pull origin master

1. 往服务端增加一个Readme的文件

git add Readme

git commit –a

git push origin master

1. 修改 后的文件discard

git checkout Readme

1. 备注

本文只是说明git的简单流程，没有涉及branch，tag等方面的应用。大家可以参考git help git 和 git help command来寻求在线帮助。

参考源：

<http://learn.github.com/>

http://help.github.com/