

# Git使用入门

## 第一部分 内容

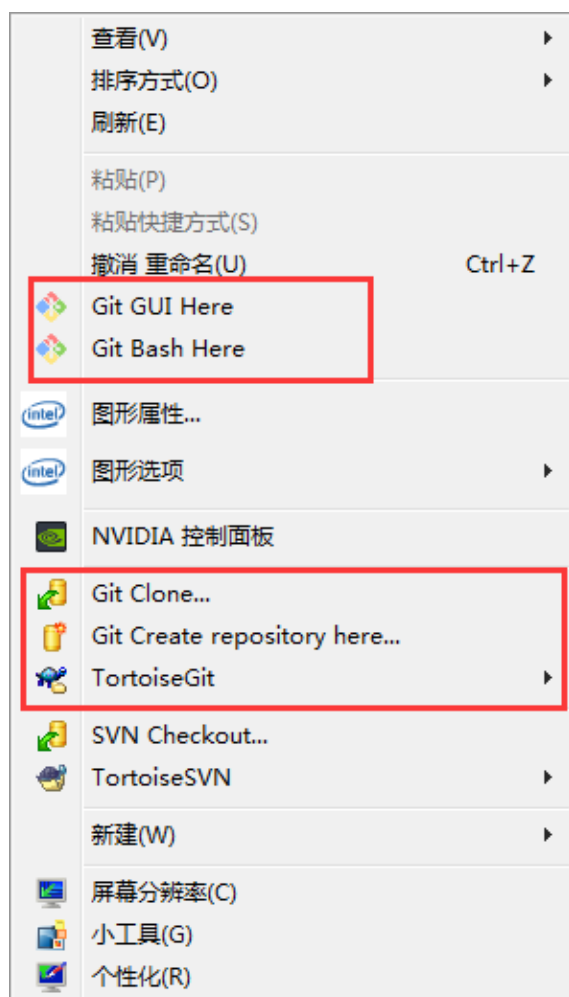
### 一、安装git

有个叫做 msysGit 的项目提供了安装包，可以到 GitHub 的页面上下载 exe 安装文件并运行，地址：<http://msysgit.github.com/>。（这里是下载好的安装包：

[file:///10.100.19.197/share/software/Git\\_2.11.0\\_64.exe](file:///10.100.19.197/share/software/Git_2.11.0_64.exe)）

完成安装之后，就可以使用命令行的 git 工具（已经自带了 ssh 客户端）了。

安装完成后，右键菜单里多了一些关于git的操作，如下图：



### 二、用户配置

配置的是你个人的用户名称和电子邮件地址。这两条配置很重要，每次 Git 提交时都会引用这两条信息，说明是谁提交了更新，所以会随更新内容一起被永久纳入历史记录。

具体操作步骤：在桌面右击——》Git Bash Here，然后在终端执行如下命令：

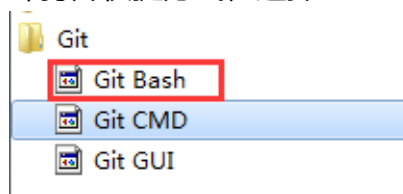
```
$ git config --global user.name "John Doe (自己的用户名) "
```

```
$ git config --global user.email johndoe@example.com (自己的邮箱地址)
```

三、 生成并在gitlab上添加ssh keys。

a) 生成密钥：在终端执行：cd ~

即打开快捷方式，选择：



\$ ssh-keygen -t rsa -C "haiyan.xu.vip@gmail.com (自己的邮箱地址)"

按3个回车，密码为空。

Your identification has been saved in /home/tekkub/.ssh/id\_rsa.

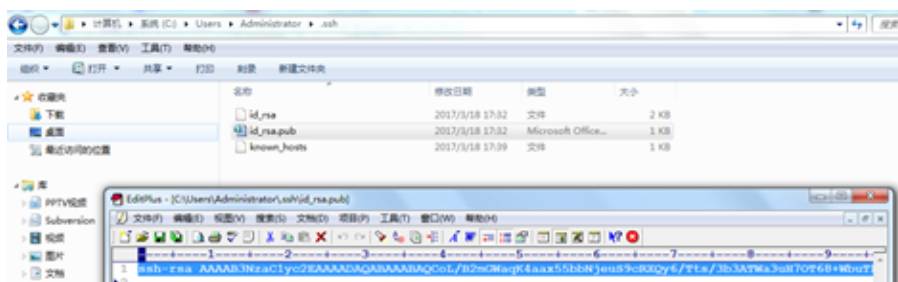
Your public key has been saved in /home/tekkub/.ssh/id\_rsa.pub.

The key fingerprint is:

.....

b)

打开目录：

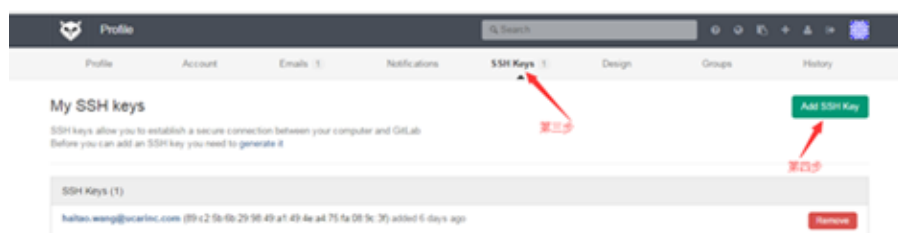


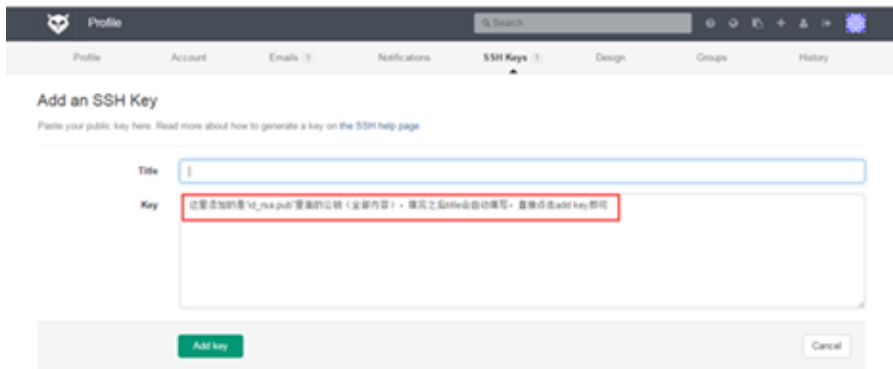
拷贝内容：

最后在~/.ssh目录中得到了两个文件：id\_rsa和id\_rsa.pub

c) 在gitlab上添加ssh密钥，这要添加的是 "id\_rsa.pub" 里面的公钥

咱们的gitlab地址：<http://gitlab.10101111.com:8888/>，登录后在SSH Keys选项卡中添加key，具体操作如下：





到此为止，git的配置就结束了，之后的使用就基本上跟SVN类似了。

## 第二部分 内容

在开始使用git之前，一定要理解了下面两张图，具体讲解大家看一下git中文文档的前两章（地址：<http://git.oschina.net/progit/index.html>），内容不是很多，强烈建议大家看一下，看完之后就会对git有一个大概的了解，理解一些内容就容易好多。（ps：这里的文档内容不错，遇到不懂得东西可以到这里查查）。

### 文件的三种状态

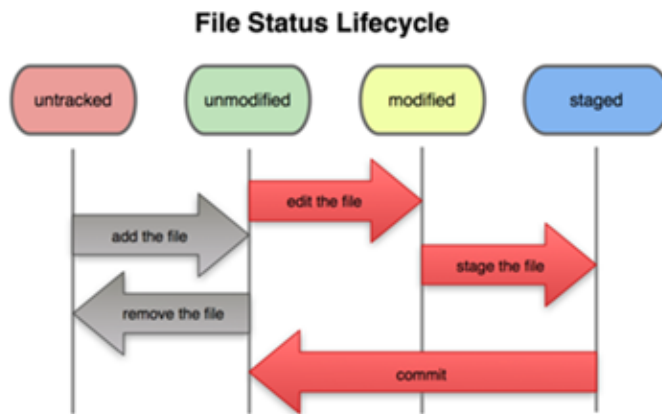
好，现在请注意，接下来要讲的概念非常重要。对于任何一个文件，在 Git 内部只有三种状态：已提交（committed）、已修改（modified）和已暂存（staged）。已提交表示该文件已经被安全地保存在本地数据库中；已修改表示修改了某个文件，但还没有提交保存；已暂存表示把已修改的文件放在下次提交时要保存的清单中。

由此我们看到 Git 管理项目时，文件流转的三个工作区域：Git 的工作目录，暂存区域，以及本地仓库。



图 1-6. 工作目录，暂存区域，以及本地仓库

使用 Git 时的文件状态变化周期如图 2-1 所示。



git常用的命令大家从网上查一下吧，网上有很多，这里就不再详细列出了。

## 1. 查看远程分支

[plain] view plain copy

```
$ git branch -a
```

```
* br-2.1.2.2
```

```
master
```

```
remotes/origin/HEAD -> origin/master
```

```
remotes/origin/br-2.1.2.1
```

```
remotes/origin/br-2.1.2.2
```

```
remotes/origin/br-2.1.3
```

```
remotes/origin/master
```

## 2. 查看本地分支[plain] view plain copy

```
$ git branch
```

\* br-2.1.2.2

master

### 3. 创建分支[plain] view plain copy

shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)

\$ git branch test

\$ git branch

\* br-2.1.2.2

master

test

### 4. 切换分支到test[plain] view plain copy

shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)

\$ git branch

\* br-2.1.2.2

master

test

shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)

```
$ git checkout test
```

```
M jingwei-
```

```
server/src/main/java/com/taobao/jingwei/server/service/cmd/GetCustomerTarCmd.java
```

```
M jingwei-server/src/main/java/com/taobao/jingwei/server/util/ServerUtil.java
```

```
Switched to branch 'test'
```

```
shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (test)
```

```
$ git branch
```

```
br-2.1.2.2
```

```
master
```

```
* test
```

5. 删除本地分支[plain] view plain copy

```
$ git checkout br-2.1.2.2
```

```
M jingwei-
```

```
server/src/main/java/com/taobao/jingwei/server/service/cmd/GetCustomerTarCmd.java
```

```
M jingwei-server/src/main/java/com/taobao/jingwei/server/util/ServerUtil.java
```

```
Switched to branch 'br-2.1.2.2'
```

```
shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)
```

```
$ git br
```

```
* br-2.1.2.2
```

```
master
```

```
test
```

```
shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)
```

```
$ git br -d test
```

```
Deleted branch test (was 17d28d9).
```

```
shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (br-2.1.2.2)
```

```
$ git br
```

```
* br-2.1.2.2
```

```
master
```

6. 查看本地和远程分支 -a。前面带\*号的代表你当前工作目录所处的分支  
[plain] view plain copy

```
shuohailhl@SHUOHAILHL-PC /f/ggg/jingwei (test)
```

```
$ git branch -a
```

```
br-2.1.2.2
```

```
master
```

\* test

remotes/origin/HEAD -> origin/master

remotes/origin/br-2.1.2.1

remotes/origin/br-2.1.2.2

remotes/origin/br-2.1.3

remotes/origin/master