Git

常用命令

文件夹初始化

git init

执行如下命令以创建一个本地仓库的克隆版本：  
git clone /path/to/repository

检出历史区文件 替换本地文件夹的文件

git checkout 1.txt

更新你的本地仓库至最新改动 = checkout

git pull

普通提交流程

Add → commit → push 具体如下

新创建一个项目的时候如下

把文件夹中修改文件 推入暂存区

git add (文件名 or \* - 所有)

暂存区内容 推入历史区

git commit -m "注释"

链接远程仓库 - 只有第一次关联本地和git仓库的时候需要 ，之后不需要

git remote add orgin https:github.com/happychong/20150401.git

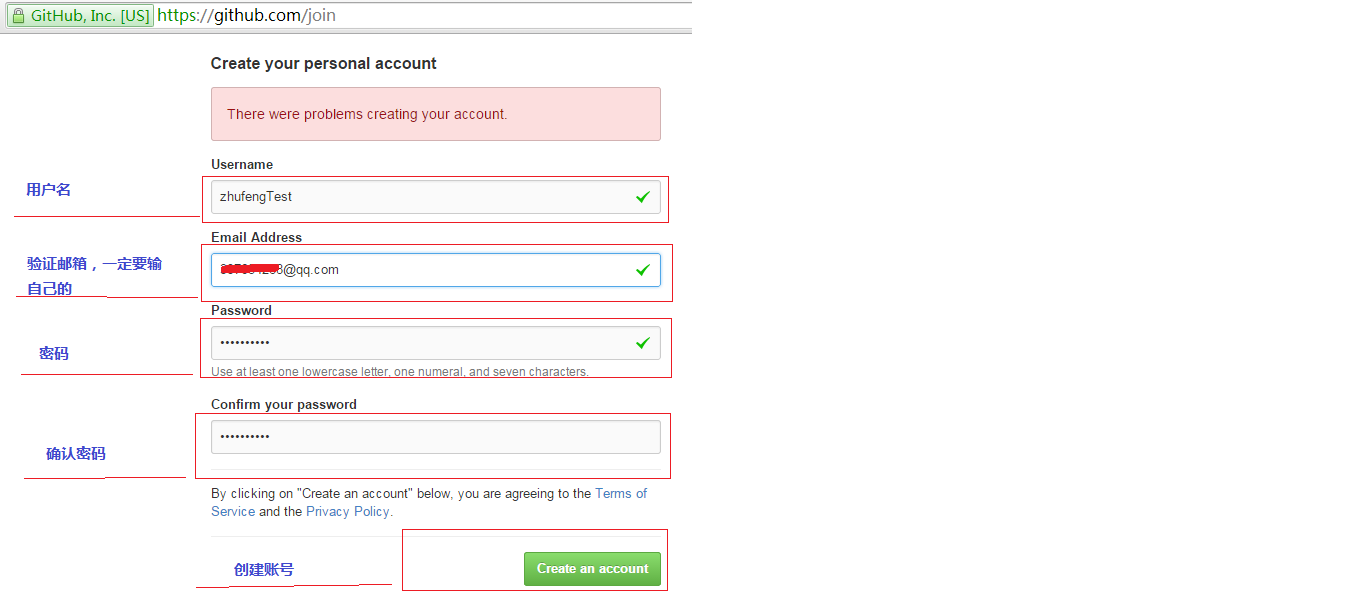
把本地文件推入远程仓库

git push -u orgin master

git push

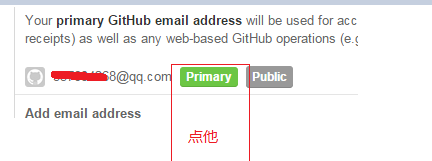
注册github及使用github for windows客户端托管个人代码

1. **登录**[**https://github.com/join**](https://github.com/join)**注册页面**

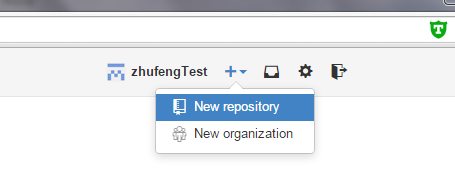
****

**验证邮箱**

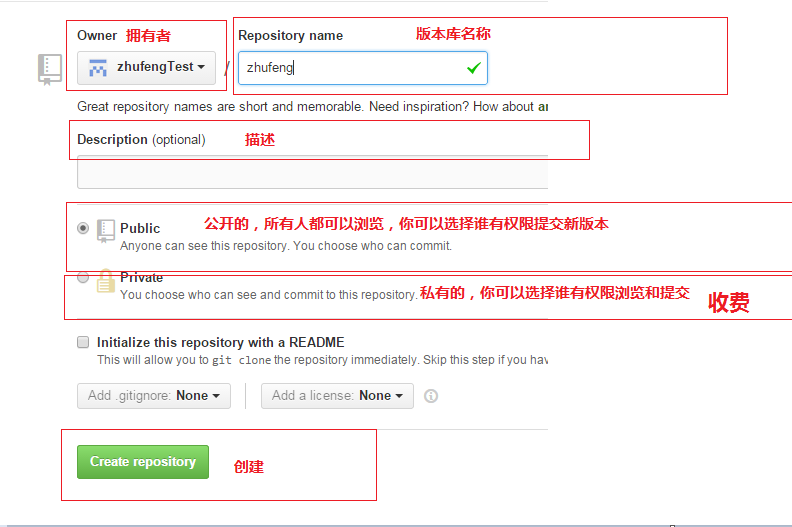
**https://github.com/settings/emails**

****

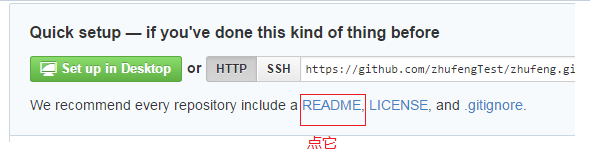
**确认注册之后会进入欢迎页面，看不懂英文没关系，不用管，下面介绍如何创建一个开源的项目**

****

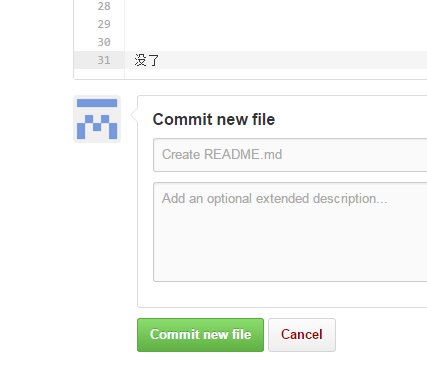
**点击New repository**

****

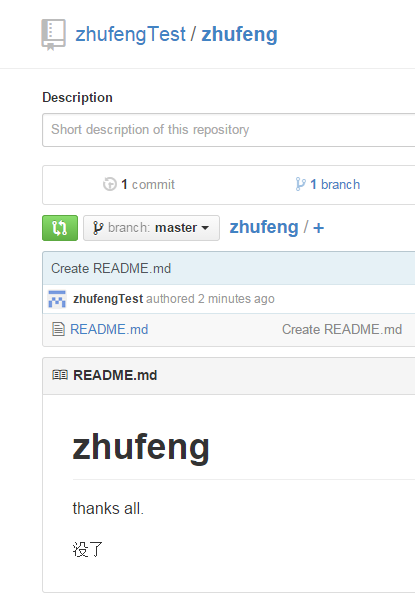
**创建完成之后**

****

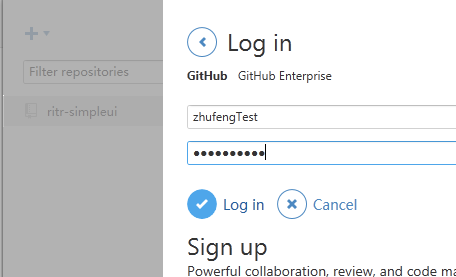
**然后**

****

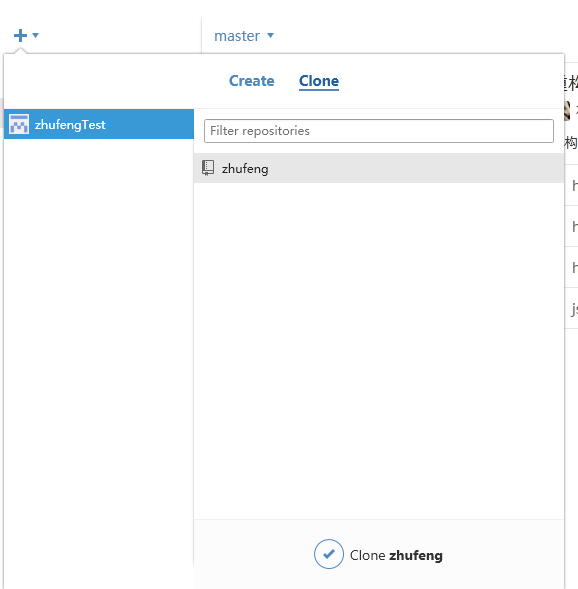
**在read me里输入一些内容，点击commit new file进入版本库管理页面**

****

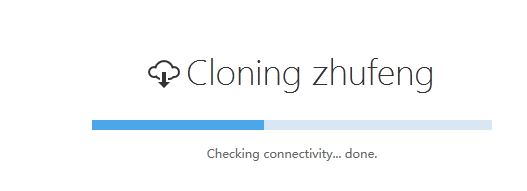
**在这里可以浏览，操作，到这会完成了一半，然后如何提交代码呢，请首先登录https://windows.github.com/下载github for windows客户端,，安装完成之后登录客户端**

****

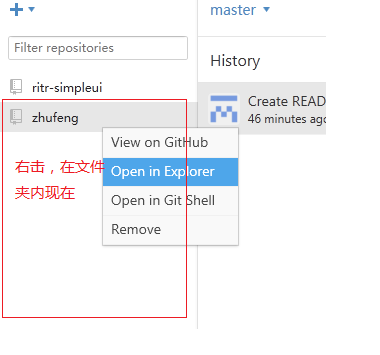
**然后克隆你的代码**

****

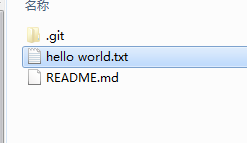
**点击Clone 选择本地的克隆地址，然后然后然我忘了截图，然后就是这个页面，正在克隆ing**

****

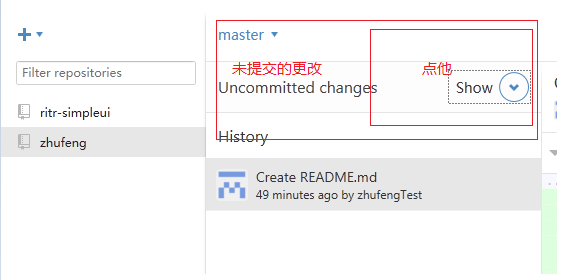
**然后**

****

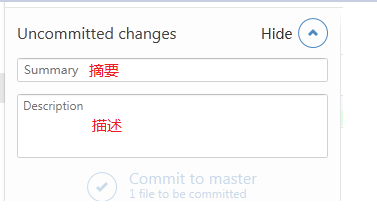
**然后我在文件内创建一个灰常牛逼的文件，写入一些内容，ok保存**

****

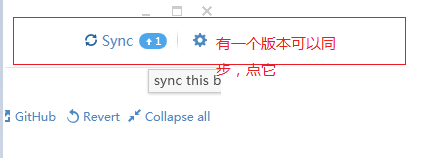
**然后在github客户端查看**

****

**输入摘要和描述点击Commit**

****

**然后请看右上角，是右上角不是左上角**

****

**Ok，过一会即可同步完成。**

**至此，你的个人代码在github托管的步骤已经完毕。**

**业余编写，定有很多疏漏，还望各位多多补正。**

**节日快乐。**

**Thanks!**

**2014.11.10**

**石博文**

mongoose

mongodb 下载路径

http://downloads.mongodb.com/win32/mongodb-win32-x86\_64-enterprise-windows-64-2.6.4-signed.msi?\_ga=1.238525191.607472782.1411452026

Mongodb相关

启动mongodb

mongod -dbpath "D:\MongoDB\data\db"

另一个窗口中 - **进入命令行窗口**

D:\MongoDB\bin>mongo.exe

具体命令

查看

help - 查看命令帮助

show dbs show database names - 显示数据库名称

show collections show collections in current database - 显示当前数据库集合

use <db\_name> set current database - 设置当前的数据库

查找

db.foo.find() list objects in collection foo - 列表中的对象集合foo

db.foo.find( { a : 1 } ) list objects in foo where a == 1 - 列表中的对象foo = = 1

db.foo.find( { $or:[{id:3} , {id:2} , {name:’sxd’}] } ) - 查找foo表中，id=3 或者 id = 2 或者 name = ‘sxd’

增加

db.user.insert({a:1}) - 插入一条数据到user的表collections

db.user.insert([{id:1}, {id:2}]) - 插入多条数据到user的表collections

删除

Db.user.remove({id:4}) - 删除id=4的数据

网上摘抄

[[html](http://www.2cto.com/kf/qianduan/css/)]

--quiet # 安静输出

--port arg  # 指定服务端口号，默认端口27017

--bind\_ip arg   # 绑定服务IP，若绑定127.0.0.1，则只能本机访问，不指定默认本地所有IP

--logpath arg   # 指定MongoDB日志文件，注意是指定文件不是目录

--logappend # 使用追加的方式写日志

--pidfilepath arg   # PID File 的完整路径，如果没有设置，则没有PID文件

--keyFile arg   # 集群的私钥的完整路径，只对于Replica Set 架构有效

--unixSocketPrefix arg  # UNIX域套接字替代目录,(默认为 /tmp)

--fork  # 以守护进程的方式运行MongoDB，创建服务器进程

--auth  # 启用验证

--cpu   # 定期显示CPU的CPU利用率和iowait

--dbpath arg    # 指定数据库路径

--diaglog arg   # diaglog选项 0=off 1=W 2=R 3=both 7=W+some reads

--directoryperdb    # 设置每个数据库将被保存在一个单独的目录

--journal   # 启用日志选项，MongoDB的数据操作将会写入到journal文件夹的文件里

--journalOptions arg    # 启用日志诊断选项

--ipv6  # 启用IPv6选项

--jsonp # 允许JSONP形式通过HTTP访问（有安全影响）

--maxConns arg  # 最大同时连接数 默认2000

--noauth    # 不启用验证

--nohttpinterface   # 关闭http接口，默认关闭27018端口访问

--noprealloc    # 禁用数据文件预分配(往往影响性能)

--noscripting   # 禁用脚本引擎

--notablescan   # 不允许表扫描

--nounixsocket  # 禁用Unix套接字监听

--nssize arg (=16)  # 设置信数据库.ns文件大小(MB)

--objcheck  # 在收到客户数据,检查的有效性，

--profile arg   # 档案参数 0=off 1=slow, 2=all

--quota # 限制每个数据库的文件数，设置默认为8

--quotaFiles arg    # number of files allower per db, requires --quota

--rest  # 开启简单的rest API

--repair    # 修复所有数据库run repair on all dbs

--repairpath arg    # 修复库生成的文件的目录,默认为目录名称dbpath

--slowms arg (=100) # value of slow for profile and console log

--smallfiles    # 使用较小的默认文件

--syncdelay arg (=60)   # 数据写入磁盘的时间秒数(0=never,不推荐)

--sysinfo   # 打印一些诊断系统信息

--upgrade   # 如果需要升级数据库  \* Replicaton 参数

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

--fastsync  # 从一个dbpath里启用从库复制服务，该dbpath的数据库是主库的快照，可用于快速启用同步

--autoresync    # 如果从库与主库同步数据差得多，自动重新同步，

--oplogSize arg # 设置oplog的大小(MB)  \* 主/从参数

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

--master    # 主库模式

--slave # 从库模式

--source arg    # 从库 端口号

--only arg  # 指定单一的数据库复制

--slavedelay arg    # 设置从库同步主库的延迟时间  \* Replica set(副本集)选项：

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

--replSet arg   # 设置副本集名称  \* Sharding(分片)选项

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

--configsvr # 声明这是一个集群的config服务,默认端口27019，默认目录/data/configdb

--shardsvr  # 声明这是一个集群的分片,默认端口27018

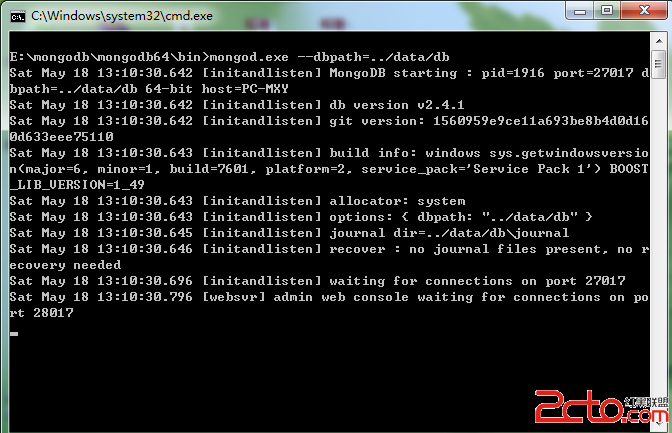
--noMoveParanoia    # 关闭偏执为moveChunk数据保存

在使用mongod这个命令的时候，必须指定一下数据库文件存放的位置，也就是--path参数必须得配置。这里我们建立一个data目录，用于存放数据库文件。建立好了data目录，我们可以使用如下的命令启动（我的data目录与bin同级）：

[html]

mongod.exe --dbpath=../data/db

这时候我们可以看到如下的画面：



这里我们看到一个27017，这个代表mongodb监听的端口，我们连接的时候都会使用到这个端口，就跟[mysql](http://www.2cto.com/database/MySQL/" \t "_blank)的3306端口一下。同样我们可以--port修改这个端口。

另外还有一个28017，这个是mongodb在启动的时候为我们启动的一个web服务的端口。我们可以使用http://localhost:28017/这个地址查看mongodb的一些参数。

这样我们就安装好了并且启动了mongodb。但是这样每次在cmd中输入很麻烦，我建议将这个命令存入一个bat文件，这样我们就可以直接点击bat启动mongodb了。

下面介绍一下mongo这个命令，这个是mongodb自带的客户端shell。我们直接在cmd中输入mongo命令，出现如下的画面，表示我们连接成功。



默认情况下我们连接的是本机（127.0.0.1）的test数据库。同样mongo的启动参数也可以设置，这里我简单介绍几个：

[html]

--port 设置连接的mongodb的端口

--host 连接的mongodb的ip地址

-u 登陆用户

-p 登陆密码

[windows](http://www.2cto.com/os/windows/)下安装mongodb就这样了,在linux中安装没有试过,有兴趣的可以试试.

Package.json

项目模块化开发的必备

安装

$ npm install

删除包

$ npm remove 包名