[Git 2](#_Toc428347218)

[常用命令 2](#_Toc428347219)

[普通提交流程 2](#_Toc428347220)

[新创建一个项目的时候如下 3](#_Toc428347221)

[注册github及使用github for windows客户端托管个人代码 3](#_Toc428347222)

[Database – 数据库 9](#_Toc428347223)

[MySQL 9](#_Toc428347224)

[概念 9](#_Toc428347225)

[安装及启动（win7x64） 9](#_Toc428347226)

[方法/步骤 9](#_Toc428347227)

[第一大步：下载。 9](#_Toc428347228)

[第二大步：安装（解压） 12](#_Toc428347229)

[第三大步：配置默认文件 12](#_Toc428347230)

[第四大步：安装mysql服务 14](#_Toc428347231)

[第五大步：打开MySQL 15](#_Toc428347232)

[附加步步骤：为你的mysql瘦身 17](#_Toc428347233)

[Mysql图形化管理工具介绍 18](#_Toc428347234)

[语法 19](#_Toc428347235)

[命令行进入 19](#_Toc428347236)

[查看 19](#_Toc428347237)

[修改 19](#_Toc428347238)

[删除 20](#_Toc428347239)

[其他常用 20](#_Toc428347240)

[网上摘抄 20](#_Toc428347241)

[node上的应用 21](#_Toc428347242)

[安装 21](#_Toc428347243)

[引用 22](#_Toc428347244)

[创建链接 22](#_Toc428347245)

[查询 22](#_Toc428347246)

[连接池 22](#_Toc428347247)

[创建连接池 22](#_Toc428347248)

[获得连接池连接 22](#_Toc428347249)

[关闭连接池连接 22](#_Toc428347250)

[MongoDB 22](#_Toc428347251)

[概念 22](#_Toc428347252)

[mongodb 下载及安装 23](#_Toc428347253)

[下载路径 23](#_Toc428347254)

[网上摘抄 23](#_Toc428347255)

[语法 26](#_Toc428347256)

[启动mongodb服务 26](#_Toc428347257)

[具体命令 27](#_Toc428347258)

[查看 27](#_Toc428347259)

[查找 27](#_Toc428347260)

[增加 27](#_Toc428347261)

[删除 27](#_Toc428347262)

[node上的应用 27](#_Toc428347263)

[安装 27](#_Toc428347264)

[得到Mongo客户端 27](#_Toc428347265)

[创建链接 – 获得需要操作的集合 – 集合内查找 27](#_Toc428347266)

[查询 28](#_Toc428347267)

[Redis 28](#_Toc428347268)

[概念 28](#_Toc428347269)

[Windows下安装Redis 28](#_Toc428347270)

[启动服务 29](#_Toc428347271)

[Node上的应用 30](#_Toc428347272)

[Package.json 32](#_Toc428347273)

[安装 32](#_Toc428347274)

[删除包 32](#_Toc428347275)

[常用命令 33](#_Toc428347276)

[查看IP 33](#_Toc428347277)

[端口冲突，Kill进程 33](#_Toc428347278)

[Node\_modules 34](#_Toc428347279)

[http 34](#_Toc428347280)

[引入 34](#_Toc428347281)

[创建服务 34](#_Toc428347282)

[req --- 请求体 34](#_Toc428347283)

[res.writeHead 34](#_Toc428347284)

[res.setHeader 34](#_Toc428347285)

[res --- 响应体 35](#_Toc428347286)

[监听端口 35](#_Toc428347287)

Git

常用命令

**git init**  文件夹初始化  
**git clone /path/to/repository**创建一个本地仓库的克隆版本

**git checkout 1.txt** 检出历史区文件 替换本地文件夹的文件

**git pull** 更新你的本地仓库至最新改动 = checkout

**git rm -r --cached ignore\_file** 清除缓存

**git update-index --assume-unchanged PATH**    clone下来的仓库中**手动设置不要检查特定文件**的更改情况。

在PATH处输入要忽略的文件。

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

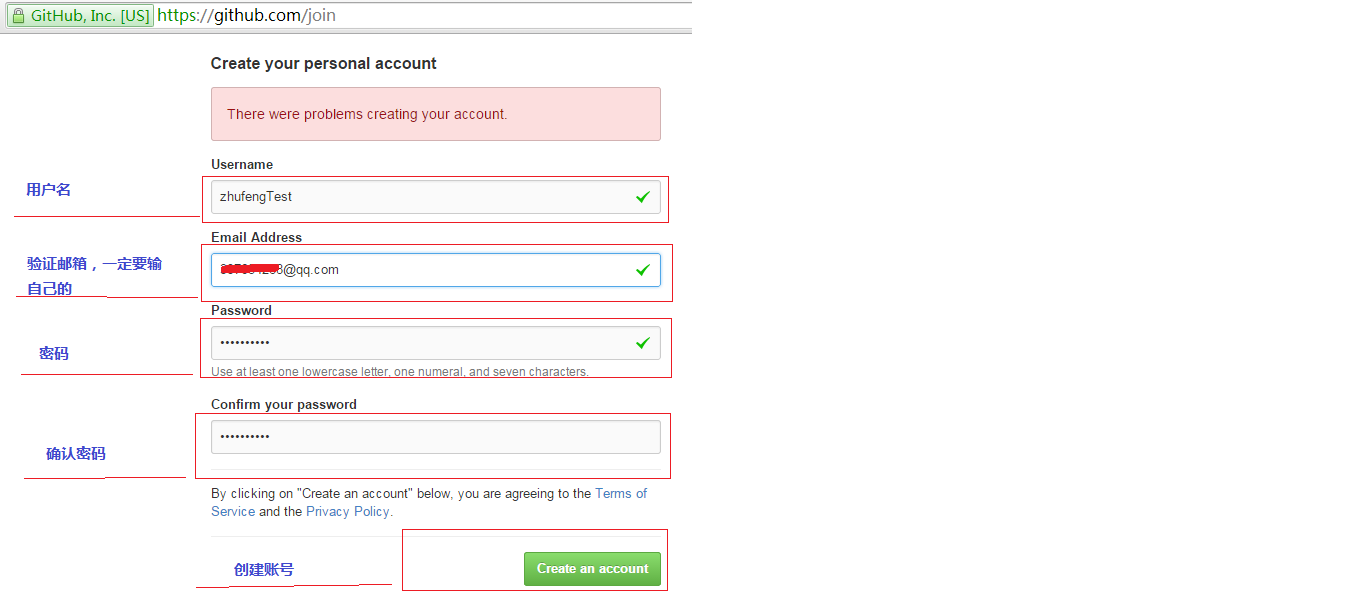
把本地文件推入远程仓库

git push -u orgin master

git push

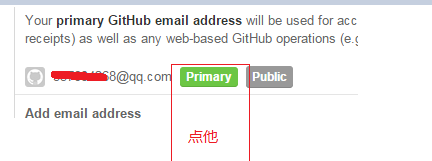
注册github及使用github for windows客户端托管个人代码

1. **登录<https://github.com/join>注册页面**

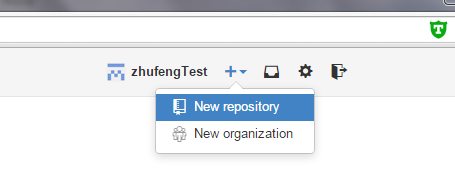
****

**验证邮箱**

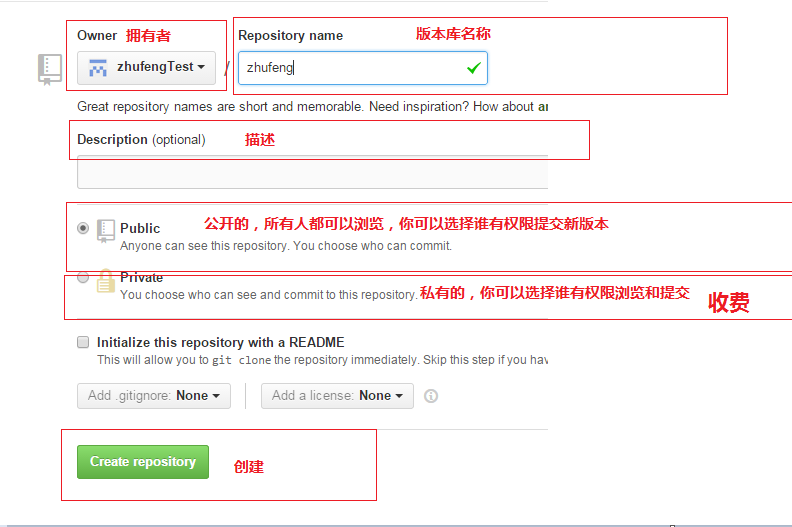
**https://github.com/settings/emails**

****

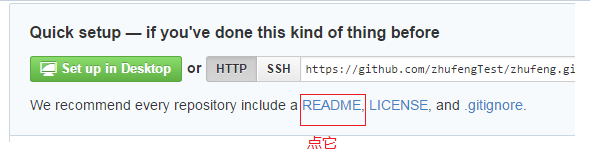
**确认注册之后会进入欢迎页面，看不懂英文没关系，不用管，下面介绍如何创建一个开源的项目**

****

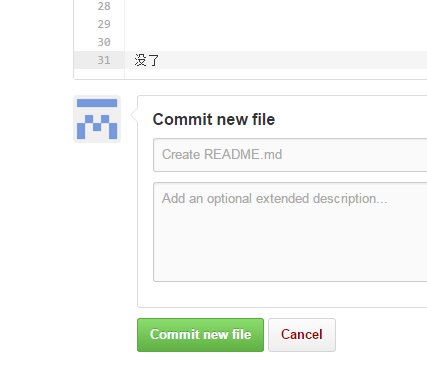
**点击New repository**

****

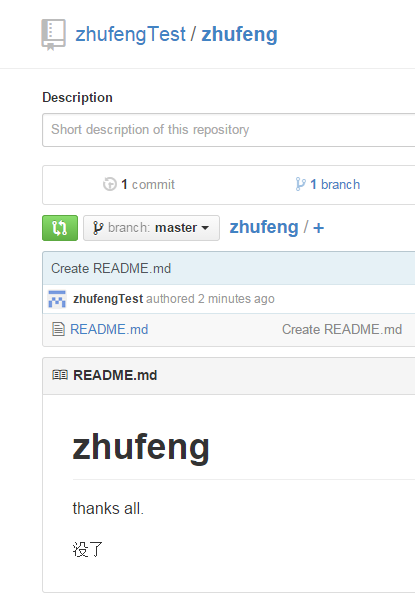
**创建完成之后**

****

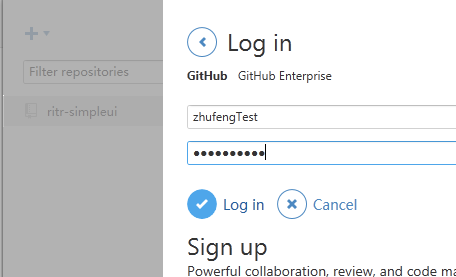
**然后**

****

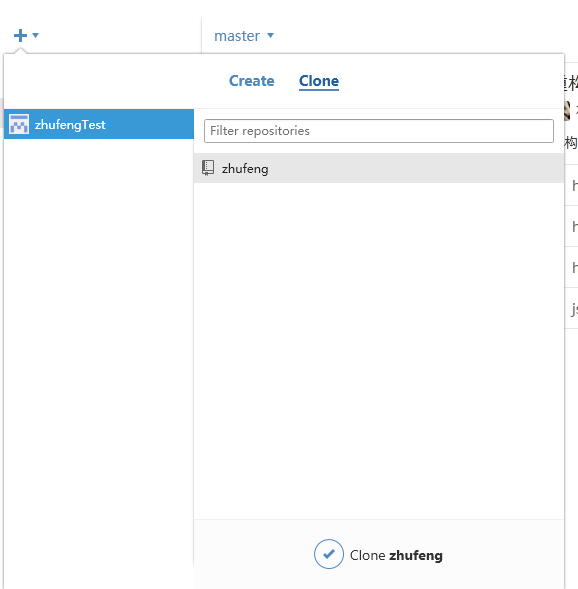
**在read me里输入一些内容，点击commit new file进入版本库管理页面**

****

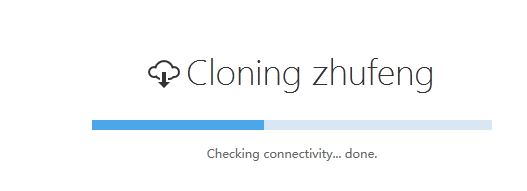
**在这里可以浏览，操作，到这会完成了一半，然后如何提交代码呢，请首先登录https://windows.github.com/下载github for windows客户端,，安装完成之后登录客户端**

****

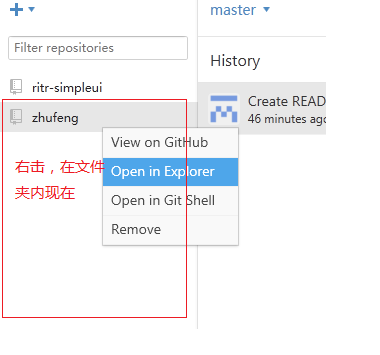
**然后克隆你的代码**

****

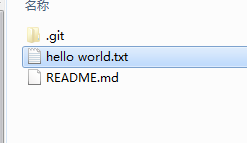
**点击Clone 选择本地的克隆地址，然后然后然我忘了截图，然后就是这个页面，正在克隆ing**

****

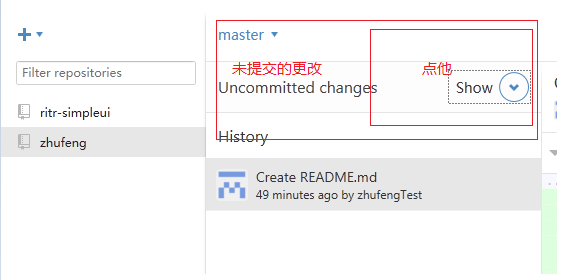
**然后**

****

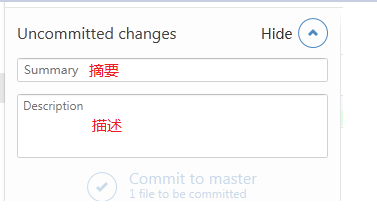
**然后我在文件内创建一个灰常牛逼的文件，写入一些内容，ok保存**

****

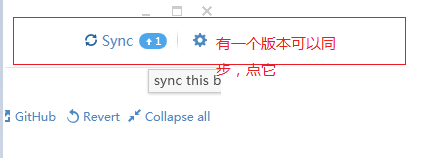
**然后在github客户端查看**

****

**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**

Database – 数据库

MySQL

概念

目前最流行的关系型数据库

1. 关系型数据库就是以行和列的形式存储数据
2. 一系列的行与列称为表
3. 一系列的表成为数据库
4. 表里一行成为一条记录，一列称为一个字段

通过SQL（结构化查询语言）操作数据库，如建库、建表、增、删、改、查、视图、索引、授权等等。

安装及启动（win7x64）

方法/步骤

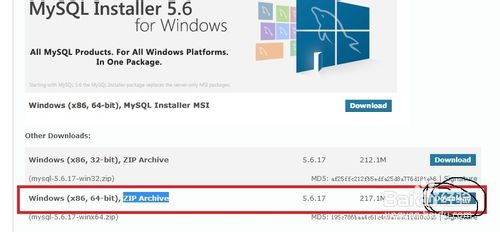
第一大步：下载。

a.俗话说：“巧妇难为无米之炊”嘛！我这里用的是 ZIP Archive 版的，win7 64位的机器支持这个，所以我建议都用这个。因为这个简单嘛，而且还干净。

地址见图

拉到最下面，看清楚了64bit ZIP Archive ，点击Download。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=1)

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=2)

1. 2

b.点开之后会进入另一个页面，但却没有开始下载，这个时候不用怕，看不懂没关系，你只要知道他是要让你们登录（login），或者 申请账号（sign up）个就行了，登录或者申请成功后就会开始下载。（记住密码要字母和数字组合，字母要有大写）

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=3)

1. 3

c.注意：密码要字母和数字组合，字母要有大写字母

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=4)

1. 4

d.当然你如果有账户的话，也可以直接登录进去

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=5)

1. 5

成功后点击DownNow。文件不大，才200多M，很快的，稍等一下（成功后按理说也是到达下面这个地方的，但假如不知这样的话，你自己找一下，或者直接退出来，重复以上步骤，选登录那个就是了。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=6)

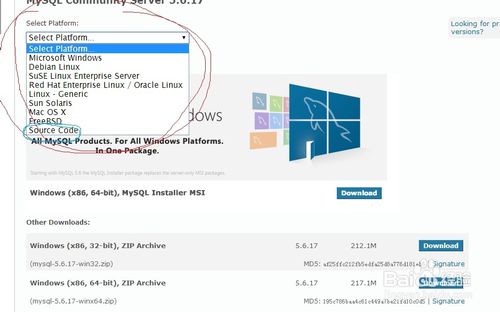
1. 6

在多说一句，MySQL 是开源（open source）数据库，所谓开源就是可以查看源代码的，我相信很多童靴都会对这个感兴趣。那么在哪下源代码呢？

在刚进-(第一个图片那地址)的时候。有一个Select Platform，选择下面的source code就行。

（同理如果要下载linux平台下的MySQL也是在这找）

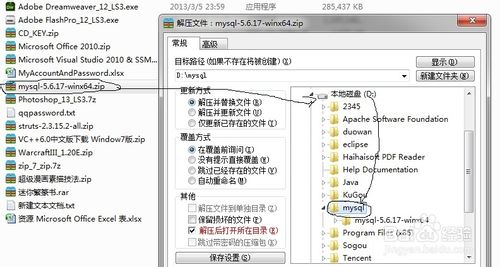
下载完毕，下面开始着手安装和配置

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=7)

1. 7

第二大步：安装（解压）

 ZIP Archive版是免安装的。只要解压就行了。和安装版的没什么不同，但就是不需要安装。找到下载好的mysql-x.x.xx-winx64.zip右键选择（解压到...），找到你想安装的地方，喜欢放C:盘的就选择C:盘，不喜欢的就自己选个就是了，小编喜欢放D：盘，毕竟c:系统盘放的东西太多不好。解压完就ok了，是不是很简单？知道 ZIP Archive版的好处了吧

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=8)

1. 8

你是不是要问了？怎么用啊？\bin点里面的mysql。exe怎么出来一个黑窗口就没反应了？

没反应是因为服务还没开！

没开，是不是打开就好了？

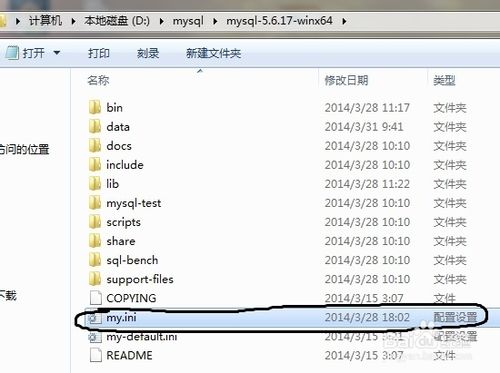
服务还没安装！

怎么安装？

下面讲！

第三大步：配置默认文件

其实理论上现在这样就可以直接安装服务了，但是因为是默认配置，我们使用的时候会出现很多问题。比如里面的汉字全是乱码之类的，所以建议你先配置一下默认文件。在mysql-x.x.xx-winx64（小编的是5.6.17根据自己的版本来定）目录下，新建个my.ini。复制下面的代码保存就ok了。my.ini会替换掉下面的my-default.ini文件。.ini文件是window里面的配置文件。保存里面各种默认的数据。安装版的是在安装的时候让你自己选然后系统给你保存进来，zip archive是自己写，都一样。里面的代码语法是什么意思童鞋们自己查相关资料吧。毕竟我一两句话说不清。这个文件在软件运行时加载。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=9)

代码：（把里面D:\mysql\mysql-5.6.17-winx64改成你自己的软件路径，保存）

[mysql]

# 设置mysql客户端默认字符集

default-character-set=utf8

[mysqld]

#设置3306端口

port = 3306

# 设置mysql的安装目录

basedir=D:\mysql\mysql-5.6.17-winx64

# 设置mysql数据库的数据的存放目录

datadir=D:\mysql\mysql-5.6.17-winx64\data

# 允许最大连接数

max\_connections=200

# 服务端使用的字符集默认为8比特编码的latin1字符集

character-set-server=utf8

# 创建新表时将使用的默认存储引擎

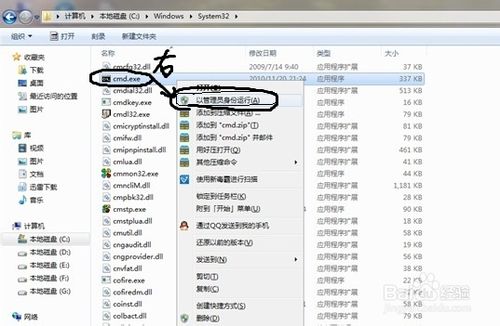
default-storage-engine=INNODB

第四大步：安装mysql服务

a：开始->在面的搜索栏里面输入cmd，上面会出来一个cmd.exe，别忙着点确定，要右键这个cmd.exe选择一管理员的身份运行这个选项。不然会出现一个身份不够的错误码，具体数忘了。这是win7的一个特点。知道就行了。若是my.ini写错的话，错误码是1067.

（你也可以在C:\Windows\System32目录下找到这个cmd.exe，同样右键它，以管理员身份确定。）

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=10)

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=11)

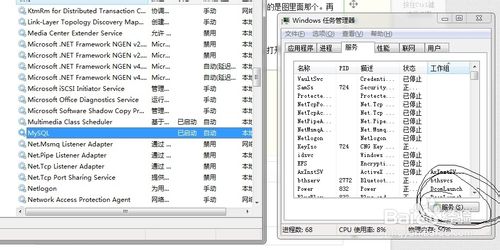
以管理员身份打开cmd窗口后，将目录切换到你解压文件的bin目录，我的目录是图里面那个，你的话，要以你自己的目录为准。再输入mysqld install回车运行就行了，注意是是mysqld，不是mysql

出现安装成功就ok了。我由于已经安装了，所以结果就是这个了。

接着就是在输入 net start mysql 启动服务。或者不嫌麻烦就手动，打开服务手动启动了mysql服务了。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=12)

告诉大家一个win7打开服务界面的简单方法。打开任务管理器（右键选择任务管理器也好，ctrl+alt+delete也罢，随你喜好）。点下面的服务就ok了，是不是很简单，比控制面板那个方法简单多了吧。当然你也可以在cmd窗口里面输入services.msc也是可以的。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=13)

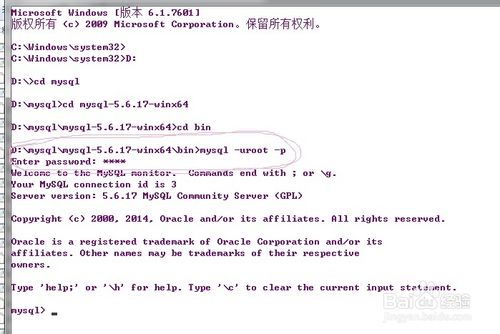
第五大步：打开MySQL

还是将目录切换到D:\mysql\mysql-x.x.xx-winx64\bin（x.xx.xxx是什么根据你自己的版本来吧）

**输入**mysql -uroot -p **,默认是没有密码的。回车进入**，当然我的已经设置密码了，所以要输入密码才能进。有密码的话，可以mysql -uroot -p密码，也是可以是mysql -uroot -p 回车输入密码，推荐第二种，原因，你动手之后就知道了。

**设置密码**的方法：mysqladmin -u root -p password 密码。

**退出**exit 就行了。记住直接关闭cmd窗口是没有退出的，要输入exit才会退出啊。

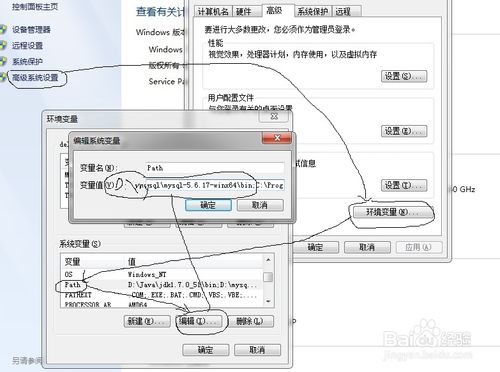
[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=14)

虽然打开mysql 了，但每次打开mysql 都要输入那么多指令切换目录是不是很讨厌？怎么弄呢？会的当然不用我说了，不会的也不用急，我来告诉你们。

右键我的电脑->属性->高级系统设置->环境变量->path->编辑，将你的mysql软件下的bin目录的全路径放里面。我建议童鞋们放在最前面，最后在那个目录的路径后面加个英文的分号（;）保存就行了。如D:\mysql\mysql-x.x.xx-winx64\bin;

为啥这样弄呢？原理是个啥？简单的说环境变量里面的path路径这东西，就是cmd系统的查找目录路径。你输入一个指令，系统怎么知道这个指令有没有呢？系统做了什么事？其实系统是在当前目录和系统环境变量path里面的路径全部查找一边，找到第一个为准，找不到就报错。所以我们要不每次都切换cmd目录，要不就设置了，以后就不需要再切换cmd路径了。

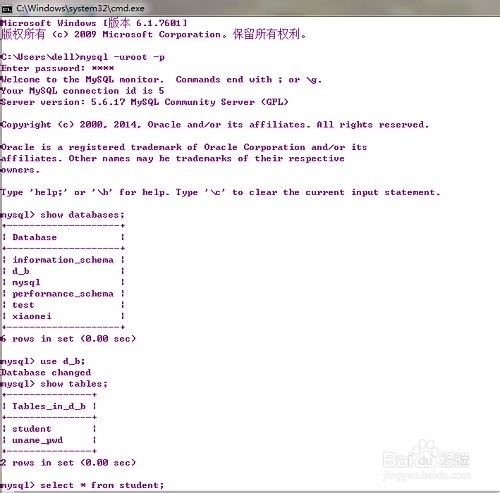
打个比方：系统就像一辆公交车。按着既定的路线走。环境变量里面的路径就是那个路线或者说是各个站。到了站（找到第一个）就下车（运行）。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=15)

现在配好了环境变量，让我们在来运行一下吧！

cmd->**mysql -uroot -p** ->回车输入密码->show databases；查找数据库->use 数据库名；切换数据库目录->show tables;查找表->sql查询工作select \* from 表名->exit退出。

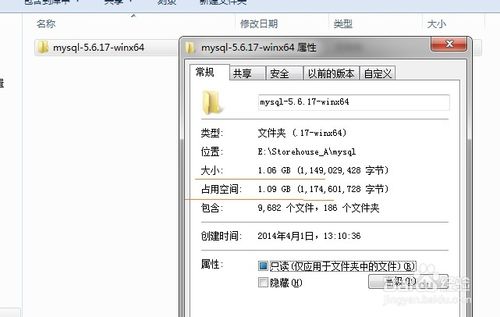
其他的指令都输数据库操作方面的事了，小编就不赘述了，你们可以找一些相关的书。

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=16)

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=17)

附加步步骤：为你的mysql瘦身

有没有没发现一个问题，你下载mysql压缩包的时候只有200M多一点，可解压只有竟然达到竟然的1G多，右键你的mysql-x.x.xx-winx64目录，选择属性来看看他的大小吧！为什么会这样呢，其实还是那个问题，mysql是开源的数据库，里面的源文件和debug等文件都没有删掉，这些文件其实没有什么用的，都删掉就是了。删掉dubug文件和.pdf文件。当然不止这一个目录里面有，自己找一下，但别删错了。删了之后是不是发现小了特别多！

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=18)

[](http://jingyan.baidu.com/album/597035521d5de28fc00740e6.html?picindex=19)

Mysql图形化管理工具介绍

小编是比较喜欢使用cmd窗口来操作mysql的，你用习惯了之后发现他是如此的方便快捷。比那什么图形化界面好用多了，而且你给mysql安个图形化界面就失去了他相比其他两个数据库的好的那些意义。小编强烈不推荐大家使用图形化界面操作mysql数据库。但你真要用我也没办法。给你们推荐几个图形化的管理工具：

1、phpMyAdmin

2、MySQLDumper

3、Navicat

4、MySQL GUI Tools

5、MySQL ODBC Connector

这些工具怎么安装我相信网上还是会有很多像我这样教程的，大家可以找下看看。

语法

命令行进入

mysql --- 进入

mysql –uroot –p  --- 默认没密码，回车进入

mysql –uroot –p123456  --- 用户名&密码进入

mysql –uroot –h127.0.0.1 --- root用户，主机名不同，进入看到的数据库也不同，是因为授权问题，权限控制严

|  |
| --- |
| 链接mysql时，指定要用的数据库  当你调用时mysql，你可在命令行上选择数据库，就在你可能需要提供的任何连接参数之后指定其名字。例如：  shell> mysql -h host -u user -p nodeBlog  Enter password: \*\*\*\*\*\*\*\* |

增加新用户

格式：grant select on 数据库.\* to 用户名@登录主机 identified by “密码”

**1、增加一个用户test1密码为abc，让他可以在任何主机上登录，并对所有数据库有查询、插入、修改、删除的权限。首先用root用户连入MYSQL，然后键入以下命令：**

    grant select,insert,update,delete on \*.\* to [email=test1@”%]test1@”%[/email]” Identified by “abc”;

但增加的用户是十分危险的，你想如某个人知道test1的密码，那么他就可以在internet上的任何一台电脑上登录你的mysql数据库并对你的数据可以为所欲为了，解决办法见2。

**2、增加一个用户test2密码为abc,让他只可以在localhost上登录，并可以对数据库mydb进行查询、插入、修改、删除的操作（localhost指本地主机，即MYSQL数据库所在的那台主机）**，这样用户即使用知道test2的密码，他也无法从internet上直接访问数据库，只能通过MYSQL主机上的web页来访问了。

    grant select,insert,update,delete on mydb.\* to [email=test2@localhost]test2@localhost[/email] identified by “abc”;

如果你不想test2有密码，可以再打一个命令将密码消掉。

    grant select,insert,update,delete on mydb.\* to [email=test2@localhost]test2@localhost[/email] identified by “”;

创建数据库

语句： mysql> CREATE DATABASE menagerie;

|  |
| --- |
| 注释： 任何人都可以访问test数据库(如果你能访问它)，但是你在该数据库创建的任何东西可以被与访问它的其他人删除，为了这个原因，你可能应该询问你的**MySQL**管理员许可你自己使用的一个数据库。假定你想要调用你的nodeBlog，管理员需要执行一个这样的命令： 这里your\_mysql\_name是分配给你的**MySQL**用户名。  mysql> GRANT ALL ON menagerie.\* TO your\_mysql\_name;  如果在设置你的权限时，管理员为你创建了数据库，你可以开始使用它。否则，你需要自己创建它：创建方法如上语句 |

删除数据库

命令：drop database <数据库名>

例如：删除名为 xhkdb的数据库

mysql> drop database xhkdb;

例子1：删除一个已经确定存在的数据库

   mysql> drop database drop\_database;

   Query OK, 0 rows affected (0.00 sec)

例子2：删除一个不确定存在的数据库

   mysql> drop database drop\_database;

   ERROR 1008 (HY000): Can't drop database 'drop\_database'; database doesn't exist

      //发生错误，不能删除'drop\_database'数据库，该数据库不存在。

   mysql> drop database if exists drop\_database;

   Query OK, 0 rows affected, 1 warning (0.00 sec)//产生一个警告说明此数据库不存在

   mysql> create database drop\_database;

   Query OK, 1 row affected (0.00 sec)

   mysql> drop database if exists drop\_database;//if exists 判断数据库是否存在，不存在也不产生错误

   Query OK, 0 rows affected (0.00 sec)

创建数据库中的 表

1.创建一个工人信息表

1. **CREATE** **TABLE** `worders` (
2. `id` **int**(11) NOT NULL AUTO\_INCREMENT,
3. `name` **varchar**(40) NOT NULL,
4. `sex` enum(F,M,S),
5. `salary` **int**(11) **DEFAULT** '0',
6. `email`  **varchar**(30),
7. `EmployedDates`  **date**,
8. `department`  **varchar**(30),
9. **PRIMARY** **KEY** (`id`)
10. ) ENGINE=MyISAM  **DEFAULT** CHARSET=utf8;

2．创建网站用户信息表

1. CREATE TABLE `user` (
2. `id` int(11) NOT NULL AUTO\_INCREMENT,
3. `username` varchar(40) NOT NULL,
4. `password` varchar(40) NOT NULL,
5. `signup\_time` datetime ,
6. `signin\_time` timestamp,
7. PRIMARY KEY (`id`)
8. ) ENGINE=MyISAM DEFAULT CHARSET=utf8;

查看

show databases; --- 看有哪些数据库

show tables; --- 查看当前库里的表

查询

select \* from ssq\_result limit 5; --- 查询 所有字段（\*） 从 ssq\_result 表中，只查5条记录

select issue,red\_1,red\_2,blue from ssq\_result limit 5;

--- 查询issue,red\_1,red\_2,blue 这4个字段 从ssq\_result 表中，只查5条记录

select issue,red\_1,red\_2,blue from ssq\_result 【where red\_1=’1’ 】【order by issue】【 desc】 【limit 5】;

--- 基本同上

--- where red\_1=’1’ 满足第一个红球是‘1’的条件

--- order by issue 根据issue字段排序

--- desc 倒序

--- limit 5 只查5条

【】里内容猜测可选，待考证，【】只为标注，代码中没有

desc course; --- 查看course表的字段设置详情

select count(\*) from course; --- **查看一共多少条记录**

select count(\*) from user where user\_name=’admin’ and password=’xx’

--- 查询user表中，用户名为admin，并且密码为xx的记录，如果有，则可以登录成功

select password from user where user\_name=’admin’

--- 查询user表中，用户名为admin的密码，然后拿到的数据和用户输入的密码字段做比较，一致的话则可以登录成功

插入 – 更新

insert into course(name,` desc`,price) values(‘javascript’,’js’,4000)

--- 插入course表一条数据 ` desc`符号是1左面的按钮

update course set price=8888 【where id-4】

--- id为4的记录的price更新成8888，如【】内没有条件，所有记录的price都会更新成8888

删除

delete from course where price=8888 --- 把所有price=8888的记录删除

其他常用

use lottery; --- 用lottery库

exit; --- 退出

网上摘抄

【1】建立员工档案表  
要求字段：员工员工编号，员工姓名，性别，工资，email，入职时间，部门。  
  
  
【2】合理选择数据类型及字段修饰符，要求有NOT NULL，auto\_increment, primary key等。

1. --
2. -- make by kakane
3. --

6. **DROP** **TABLE** IF EXISTS `workers\_info`;
7. **CREATE** **TABLE** `workers\_info` (
8. `id` **int**(11) NOT NULL AUTO\_INCREMENT,
9. `workername` **varchar**(20) NOT NULL,
10. `sex` enum(F,M,S),
11. `salary` **int**(11) **DEFAULT** '0',
12. `email`  **varchar**(30),
13. `EmployedDates`  **date**,
14. `department`  **varchar**(30),
15. **PRIMARY** **KEY** (`id`)
16. ) ENGINE=MyISAM  **DEFAULT** CHARSET=utf8;

19. mysql> **alter** **table** workers\_info **ADD** sex  enum('F','M','S');

【3】查看表的结构

1. mysql**>** desc  workers\_info;

【4】新增qq和tel字段，要求tel字段位于email前，要求入职时间是最后一个字段

 mysql> ALTER TABLE workers\_info ADD tel varchar(15) after salary;

1. mysql> **ALTER** **TABLE** workers\_info **ADD** qq **int**;
2. **ALTER** **TABLE** workers\_info **MODIFY** EmployedDates **date** **after** qq;

【5】把email字段修改成mailbox  
 mysql> ALTER TABLE workers\_info CHANGE email mailbox varchar(30);  
  
【6】向表里添加10条记录  
  mysql> INSERT INTO workers\_info values(NULL,'xing',10000,'1598232123','xing@qq.com','yanfa',736019646,20121221);

1. mysql> **INSERT** **INTO** workers\_info (workername,salary,tel,mailbox,department,qq,EmployedDates) **values**('xing',10000,'1598232123','xing@qq.com','yanfa',736019646,20121221);

【7】修改其中两条记录的tel和mailbox  
  mysql> UPDATE workers\_info SET mailbox = 'haha@qq.com' WHERE id = 14;

1. mysql> **UPDATE** workers\_info **SET** mailbox = 'haha@qq.com',tel='1246543423' **WHERE** id = 13;

【8】查看所添加记录  
mysql> select \* from workers\_info;  
  
【9】查看姓名和入职时间记录  
 mysql> select workername,EmployedDates from workers\_info;  
   
【10】查询入职时间在2003年以前的

1. mysql> **select** \* **from** workers\_info **where** year(EmployedDates) < 2003;

【11】查询工资最高和最低的员工姓名

1. mysql> **select** \* **from** workers\_info **ORDER** **BY** salary limit 1;
2. mysql> **select** \* **from** workers\_info **ORDER** **BY** salary **desc** limit 1;

【12】查询平均工资

1. mysql> **select** avg(salary) **from** workers\_info;

【13】统计男员工人数、女员工人数

1. mysql> **select** count(男) **from** workers\_info **where** sex="M";
2. mysql> **select** count(男) **from** workers\_info **where** sex="M";

【14】按照入职时间先后进行排序，并显示前5位员工姓名

1. mysql> **select** \* **from** workers\_info **ORDER** **BY** EmployedDates limit 5;

node上的应用

安装

npm install mysql

引用

var mysql = require(‘mysql’);

创建链接

普通链接

var connection = mysql.createConnection({

host: ‘localhost’,

user: ‘xxx’,

password: ‘xxx’

});

连接池

**创建连接池**

var pool = mysql.createPool({

host: 'localhost',

user: 'root'

});

**获得连接池连接**

pool.getConnection (function (err, conn) {

conn.query('select \* from test.course', function (err, rows, feilds) {

console.log(rows);

// 把借来的连接还回去

conn.release();

})

})

**关闭连接池连接**

conn.r

查询

Connection.query(‘SELECT 1 + 1 AS result’, function (err, rows, fields) {});

elease();

MongoDB

概念

非关系型数据库，KV数据

MongoDB的主要特性：

1. 面向文档的分布式数据库

可以把一堆数据分散在不同的机器上进行存储，不只可以存数据，还可以存文档

1. schema-free

关系型数据库存储的时候得建表，建表的时候得声明好字段，存储数据的时候得按字段走，多了不行，少了的字段可以为null的话可以少，不可以为null的话，必须得有值

Mongo的schema-free 存数据的时候，不管有什么字段，都可以存进去，存储的时候是key-value的结构，像JSON结构

*Mongo下没有表的概念，只有集合 – collections 的概念*

*没有集合的时候，执行插入操作，会自动创建集合*

mongodb 下载及安装

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>

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

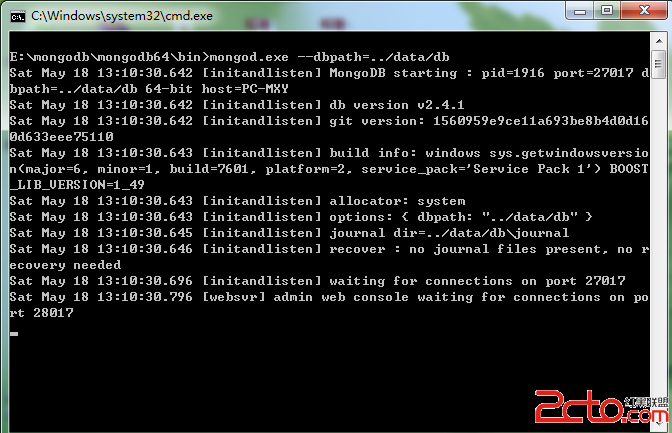
--noMoveParanoia    # 关闭偏执为moveChunk数据保存

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.

语法

启动mongodb服务

mongod -dbpath "D:\MongoDB\data\db" --- 需要在MongoDB的安装文件夹下的bin目录下运行命令行窗口

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

node上的应用

安装

npm install mongodb

得到Mongo客户端

var MongoClient = require(‘mysql’).MongoClient;

创建链接 – 获得需要操作的集合 – 集合内查找

主机 端口 库名

↓ ↓ ↓

MongoClient.connect(‘mongodb://host:port/db’, function (err, db) {

//获得操作集合对象

var collection = db.collection('aa');

// collection下查找一条记录

collection.findOne(function (err, one) {

//findOne 方法是异步的

console.log(one);

});

});

查询

collection.find()

collection.findOne()

collection.findAndModify()

collection.updata()

collection.save()

Redis

概念

key-value数据库

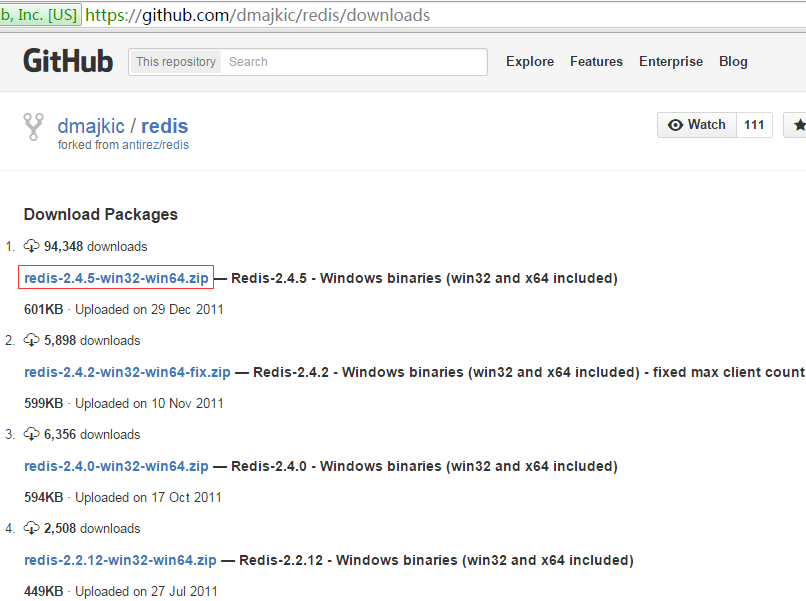
与memcached类似，但是数据可以持久化

支字符串、Hash、List、Set等多种结构

高性能，支持原子操作

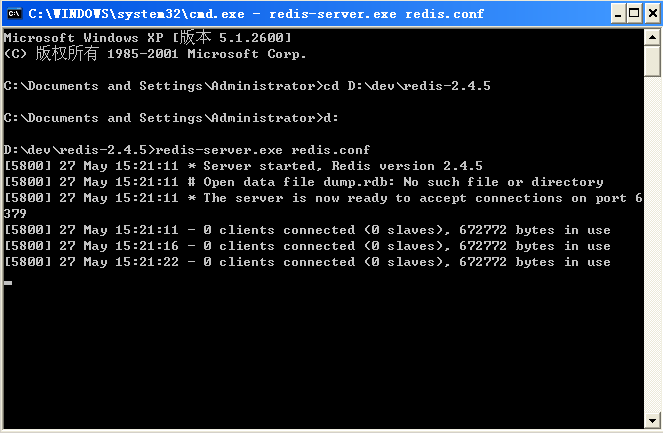
Windows下安装Redis

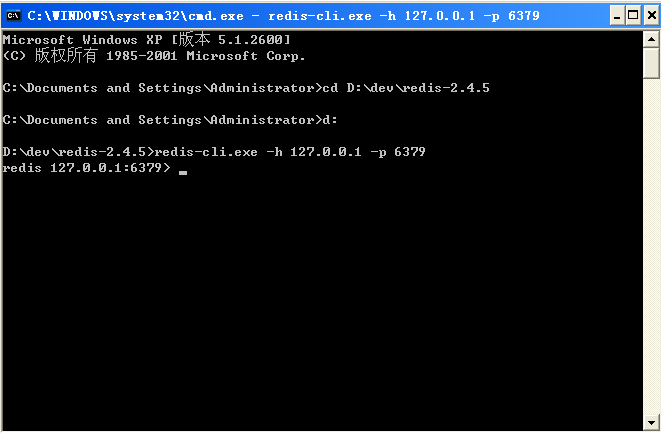
下载地址[https://github.com/dmajkic/redis/downloads](https://github.com/dmajkic/redis/downloads" \t "_blank)。

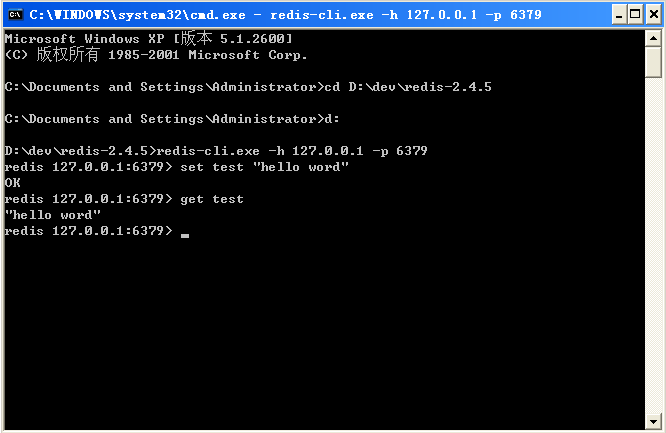


下载到的Redis支持32bit和64bit。根据自己实际情况选择，我选择32bit。把32bit文件内容拷贝到需要安装的目录下,比如：D:\dev\redis-2.4.5。

启动服务

打开一个cmd窗口，使用cd命令切换到指定目录（D:\dev\redis-2.4.5）运行 redis-server.exe redis.conf 。运行以后出现如下界面。  
  
这就说明Redis服务端已经安装成功。

重新打开一个cmd窗口，使用cd命令切换到指定目录（D:\dev\redis-2.4.5）运行 redis-cli.exe -h 127.0.0.1 -p 6379，其中 127.0.0.1是本地ip，6379是redis服务端的默认端口。运行成功如下图所示。  
这样，Redis windows环境下搭建已经完成，是不是很简单。  
  
这样，Redis windows环境下搭建已经完成，是不是很简单。

环境已经搭建好，总得测试下吧。比如：存储一个key为test，value为hello word的字符串，然后获取key值。  
  
正确输出 hell word，测试成功！

命令

启动

第一步：启动服务

redis-server.exe redis.conf ----- 启动服务

打开一个cmd窗口，使用cd命令切换到指定目录（D:\dev\redis-2.4.5）运行 redis-server.exe redis.conf

第二步：启动客户端

redis-cli.exe -h 127.0.0.1 -p 6379 ----- 启动客户端

重新打开一个cmd窗口，使用cd命令切换到指定目录（D:\dev\redis-2.4.5）运行 redis-cli.exe -h 127.0.0.1 -p 6379，其中 127.0.0.1是本地ip，6379是redis服务端的默认端口。

清空

Flushdb ---- 清空redis数据

查看数据

keys \* ---- 查看所有数据

Node上的应用

npm install redis (npm install hiredis redis)

var redis = require(‘redis’);

var client = redis.createClient(port, host[,options]);

具体方法和数据库的方法相同

client.get/set/hset/hget/hgetall…

client.expire(key,sec);

断开连接

client.quit();

Package.json

项目模块化开发的必备

安装

$ npm install

删除包

$ npm remove 包名

常用cmd命令

查看IP

ipconfig/all

端口冲突，Kill进程

未完待续

Node\_modules

http

引入

var http = require('http');

创建服务

http.createServer(function (req, res) { … });

每当有请求进来，参数的function就会被执行

req --- 请求体

{

Headers: {

host: 'localhost:9999',

connection: 'keep-alive',

accept: 'text/html,application/xhtml+xml,application…',

'user-agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleW…',

'accept-encoding': 'gzip, deflate, sdch',

'accept-language': 'zh-CN,zh;q=0.8',

cookie: 'sessionId=2b115129-1b86-4bdb-91a5-6badfe5a4f83',

…

},

url: '/',

method: 'GET',

…

}

res --- 响应体

res.writeHead(200, {"Content-Type":"text/html"});

res.setHeader('Content-Type', 'text/html;charset=utf-8');

res.setHeader('set-Cookie', cookie.serialize(this.\_sess\_key, sessId));

res.end(http.STATUS\_CODES[statusCode]);

res.setTimeout(30000);

监听端口

http .listen(port, function () {

cb.call();

});

fs

fs.createReadStream(fullpath).pipe(res);

fs.stat(fullpath, function (err, stats) {

stats.isDirectory();

stats.isFile();

}));

fs.exists(fullpath, function (isExist) {}));

url

var urlObj = url.parse(req.url);

//urlObj.pathname

// /index.html

path

var fullpath = path.join(self.\_basedir, urlObj.pathname?(urlObj.pathname.replace(/\.\./g,"")):"");

util

JSON

JSON.stringify(req.pathParams) 把对象专户成json格式字符串

Querystring – 解析form表单

Var querystring = require(‘querystring’);

Var params = querystring.parse(str);

Cookie – 解析cookie对象

var *cookie* = require('cookie');

var cookieObj = *cookie*.parse(*req*.headers.cookie);

uuid

var *uuid* = require('uuid');

// uuid用法  
// uuid.v4();

var sessId = *uuid*.v4();

Express

安装

Npm I –g express-generator --- 安装express命令行工具

Sudo npm I –g express-generator

Express –V --- 查看版本

Express –h --- 查看帮助

初始化文件夹及文件

格式：express [options] [dir]

实例：express –e demo ---- 当前目录下 创建demo文件夹，并创建express骨架

其他软件

RegexBuddy – 正则调试

调试正则神器