MongoDB指导文档



mongo.conf(可点击下载)

**一、几个网站：**

1、官网：www.mongodb.org

安装包下载 使用文档

2、国内网站：www.mongoing.com

中文使用文档

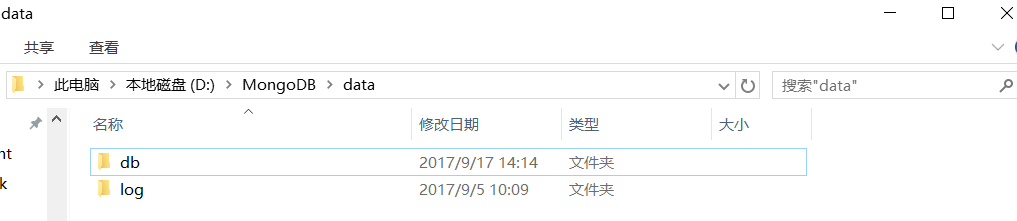
3、MongoDB GitHub：https://github.com/mongodb

4、MongoDB下载地址：http://dl.mongodb.org/dl/win32/x86\_64

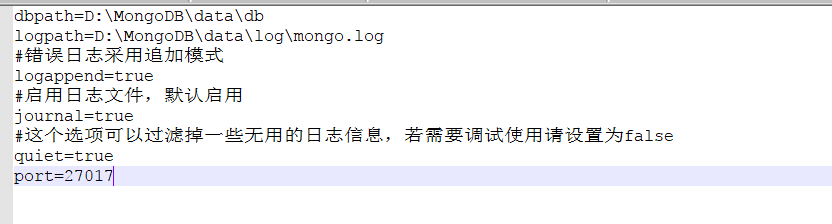
**二、mongodb安装与配置**（-- 前面都是有空格的，这个间距太小看不出来）

1、执行安装包安装mongodb

2、创建数据库路径（db目录）、日志路径（log目录）和日志文件（mongo.log文件），最好放在mongodb安装目录下



3、创建配置文件mongo.conf



4、创建并启动MongoDB服务

mongod --config "D:\MongoDB\mongo.conf" --install --serviceName "MongoDB"

net start MongoDB

5、删除mongo服务

mongod.exe --remove --serviceName "MongoDB"

提示：第4步和第5步在D:\MongoDB\bin目录下并以管理员身份运行cmd

**三、常用指令**

1、 mongodb数据库权限方面操作

mongodb 添加超级管理员

use admin

db.createUser(

{

user: "disanmeng",

pwd: "123",

roles: [ "root" ]

}

)

在admin数据库中创建了超级用户之后，登陆的时候发现用户名密码不起作用，即不需要用户名密码也可以操作数据库，这是需要为服务添加 --auth 字段。

需要删除MongoDB服务，在执行mongod --config "D:\MongoDB\mongo.conf" --install --auth --serviceName "MongoDB" 添加服务，启动服务之后就OK了。

加权之后登陆方式，进入D:\MongoDB\bin目录，

1. 可以使用mongo -u disanmeng -p 123 localhost:27017/admin 命令
2. 可以使用mongo admin登陆，

然后执行db.auth('disanmeng', '123')赋权限，返回1，即db.auth()执行成功；失败则表示admin数据库下不存在disanmeng用户或密码错误

1. 可以mongo 其他数据库（非超级用户所在数据库），

其他数据库也可以创建自己的用户，然后db.auth(用户名，密码)赋权限

**注意：--auth 启动服务之后，任何数据库都必须执行db.auth() 赋权限之后才可以对其操作。如果该数据库没有用户，则无法对其操作，必须进入admin数据库，然后执行use database\_name跳转到该数据库下为其添加用户**

2、创建、删除、显示数据库

**创建**（如果名字存在则切换到改数据库下，否则创建）：use database\_name

**显示所有**数据库：show dbs

**显示当前**所在数据库：db

**删除**：use database\_name切换到要删除的库下，db.dropDatabase()删除该库

3、数据库增删改查**（**存储的数据结构和JSON基本一样。所有存储在集合中的数据都是BSON格式。BSON是一种类json的一种二进制形式的存储格式,简称Binary JSON。**）**

**增**：

db.COLLECTION\_NAME.insert(document)

例子：

db.yang.insert({

title: 'MongoDB教程',

description: 'MongoDB是一个Nosql数据库',

by: '菜鸟教程',

url: 'http: //www.runoob.com',

tags: ['mongodb','database','NoSQL'],

likes: 100

})

**删**：

db.collection.remove(

<query>,

{

justOne: <boolean>,

writeConcern: <document>

}

)

参数说明：

query :（可选）删除的文档的条件，相当于where。

justOne : （可选）如果设为 true 或 1，则只删除一个数据。

writeConcern :（可选）抛出异常的级别。

**改**：

1. 、db.collection.update(

<query>,

<update>,

{

upsert: <boolean>,

multi: <boolean>,

writeConcern: <document>

}

)

参数说明：

query :类似sql update查询内where后面的。

update :理解为sql update查询内set后面的

upsert : 可选，默认false，如果不存在update的记录，是否插入该数据：true为插入，false不插入。

multi : 可选，默认false，只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新。

writeConcern :可选，抛出异常的级别。

例子：

db.yang.update(

{ title: 'MongoDB教程'},

{$set:{'title':'MongoDB'}},

)

1. 、save()根据\_id来修改对应\_id的数据，如果没有该\_id，则插入该数据

db.collection.save(

<document>,

{

writeConcern: <document>

}

)

参数说明：

document : 文档数据。

writeConcern :可选，抛出异常的级别

例子：

db.yang.save({

"\_id" : ObjectId("59c6919666661c423f6e104d"),

"title" : "MongoDB",

"description" : "MongoDB 是一个 Nosql 数据库",

"by" : "Runoob",

"url" : "http://www.runoob.com",

"tags" : [

"mongodb",

"NoSQL"

],

"likes" : 110

})

**查**：

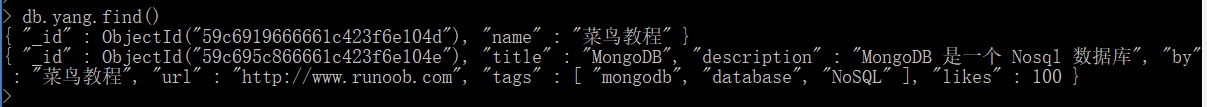
db.collection.find(query, projection)

参数说明

query：可选，使用查询操作符指定查询条件，相当于where

projection：可选，默认相当于 select \* 即显示全部字段，显示特定字段如下格式：db.yang.find( {}, {"name" ：1, "age" ：1} )。字段1表示显示该字段，0不显示

db.yang.find()



db.yang.find().pretty() 以json形式展开



**四、操作符**

常用操作符：

$gt >

$gte >=

$lt <

$lte <=

$ne !=

$eq =

$in 相当于sql语句中in

$nin $in反义词

操作符大全：http://blog.csdn.net/qq\_16313365/article/details/58599253