# Mongodb 数据库权限管理

# 一、概念理解

## 1、用户概念

Mongodb的用户是由 用户名+所属库名组成

例如:

登录mongo  testdb1 ，创建用户testuser

登录mongo  testdb2 ，创建用户testuser

那上面创建的用户分别是：testuser@testdb1，testuser@testdb2

也就是说在哪个库下面创建用户，这个用户就是哪个库的

## 2、角色概念

Mongodb的授权采用了角色授权的方法，每个角色包括一组权限。

Mongodb已经定义好了的角色叫内建角色，我们也可以自定义角色。

这儿主要介绍内建角色，Mongodb内建角色包括下面几类：

读写权限：                read/readWrite 读写库的权限

[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库)管理角色：    dbAdmin  某数据库管理权限

     userAdmin  某数据库用户的管理权限，包括创建用户，授权的管理

     dbOwner     某数据库的所有者，拥有该库的所有权限，包括readWrite，dbAdmin和userAdmin权限

集群权限   ：

备份和恢复角色：    bakcup  restore

所有数据库角色：    readAnyDatabase

     readWriteAnyDatabase

     dbAdminAnyDatabase

     userAdminAnyDatabase

超级用户角色：        root

内部角色        ：        \_\_system  不建议使用

# 二、搭建认证环境和认证登录

1、找到mongodb配置文件,设置noauth=true

重启Mongodb后，登录admin账号，创建一个超级权限用户

use admin

db.createUser({user:'root',pwd:'root',roles:[{ "role" : "root", "db" : "admin" }]});

2、关闭mongodb

3、启用认证参数

要保证权限认证生效，需要在mongodb配置文件中加入auth=true，同时取消掉noauth=true

4、启动Mongodb

5、认证登录

> use admin  
switched to db admin  
> db.auth('root','root')

1  
> use hong  
switched to db hong  
> show collections;  
liujing  
system.users  
> db.liujing.find();  
{ "\_id" : ObjectId("56f255b77fa46acddeb4507b"), "name" : "liujing" }

# 三、用户授权详解

1、创建用户并授权

语法：db.createUser({user:"UserName",pwd:"Password",roles:[{role:"RoleName",db:"Target\_DBName"}]})

首先选择在哪个库创建用户，如test：use test;

创建用户有3项需要提供：用户名，密码，角色列表

例如我要在test下面创建用testuser，密码为testpwd，角色列表包括test库的readWrite角色和userAdmin角色：

db.createUser({user:"testuser",pwd:"testpwd",roles:[{role:"readWrite",db:"test"},{role:"userAdmin",db:"test"}]})

2、修改密码

首先进入目标库：use test

db.changeUserPassword('testuser','testPWD');

3、添加角色

首先进入目标库：use test

db.grantRolesToUser(  "testuser",  [    { role: "read",db:"admin"}  ] )

4、回收角色权限

首先进入目标库：use test

db.revokeRolesFromUser("testuser",[    { role: "read",db:"admin"}  ] )

5、删除用户

首先进入目标库：use test

db.dropUser("testuser")

6、查看用户

use admin

db.system.users.find()