

mongoDB

一、MongoDB 的引言

MongoDB 是一个基于分布式文件存储的数据库。由 C++ 语言编写。旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。 Nosql 技术门类 redis 内存型 mongodb 文档型

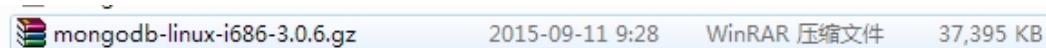
MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似 json 的 bson 格式，因此可以存储比较复杂的数据类型。Mongo 最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

二、MongoDB 的特点

- 面向集合存储，易存储对象类型的数据
- 支持查询,以及动态查询
- 支持 RUBY, PYTHON, JAVA, C++, PHP, C#等多种语言
- 文件存储格式为 BSON（一种 JSON 的扩展）
- 支持复制和故障恢复和分片

三、MongoDB 的安装和使用

1. 下载 mongoDB 的安装包(最新版本 3.6,只能在 64 位系统安装)这里使用的 3.0.6 版本



2. 上传至 linux 系统中解压当前的 linux 系统



```
mongodb-linux-i686-3.0.6.gz
```

```
[root@localhost ~]# tar -zxvf mongodb-linux-i686-3.0.6.gz
mongodb-linux-i686-3.0.6/README
mongodb-linux-i686-3.0.6/THIRD-PARTY-NOTICES
mongodb-linux-i686-3.0.6/GNU-AGPL-3.0
mongodb-linux-i686-3.0.6/bin/mongodump
mongodb-linux-i686-3.0.6/bin/mongorestore
```

3. 将解压的文件为了方便目录进行重命名(这步可以跳过)

```
[root@localhost ~]# mv mongodb-linux-i686-3.0.6 mongodb
```

4. 进入 mongodb 的文件夹中查看目录

```
[root@localhost mongodb]# ls
bin  GNU-AGPL-3.0  README  THIRD-PARTY-NOTICES
```

5. 在 bin 目录中存在大量 mongodb 使用的命令

```
[root@localhost bin]# ls
bsondump  mongod  mongoexport  mongoimport  mongoperf  mongos  mongotop
mongo     mongodump  mongofiles  mongooplog  mongorestore  mongostat
```

6. 启动 MongoDB 数据库服务

```
./mongod --port 27017 --dbpath /root/data
```

```
[root@localhost bin]# ./mongod --port 27017 --dbpath /root/data
2017-12-12T22:59:36.944+0800 I CONTROL
2017-12-12T22:59:36.944+0800 W CONTROL 32-bit servers don't have journaling enabled, you want durability.
```

注意:

启动时要求存放数据的目录必须存在

默认的端口号是 27017 可以通过--port 指定端口启动

7. 出现如下结果代表启动成功

```
[initandlisten] options: { net: { port: 27017 }, storage
[initandlisten] allocating new ns file /root/data/local.
[FileAllocator] allocating new datafile /root/data/local
[FileAllocator] creating directory /root/data/_tmp
[FileAllocator] done allocating datafile /root/data/local
[initandlisten] waiting for connections on port 27017
```

四、MongoDB 的 shell(客户端)操作

1. 进入 mongo 的 bin 目录中找到如下指令



```
[root@localhost ~]# cd mongodb/bin/
[root@localhost bin]# ls
bsondump  mongod  mongoexport  mongoimport  mongoperf  mongos  mongotop
mongo     mongodump  mongofiles  mongooplog  mongorestore  mongostat
```

2. 使用如下命令连接到 mongodb 的服务中

```
[root@localhost bin]# ./mongo --port 27017
MongoDB shell version: 3.0.6
connecting to: 127.0.0.1:27017/test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2017-12-12T23:01:37.024+0800 I CONTROL [initandlisten] ** WARNING
is not recommended.
```

注意:

- a) 连接到 mongodb 后, mongo 和 mysql 数据库有点像, 先是一个一个库的概念, 操作之前需要先选中库

3. 查看系统中默认的所有库

show dbs;

```
> show dbs;
local 0.078GB
```

注意:

默认只有一个 local 库

还有一个默认隐藏不可见的数据库 admin

4. 选中一个库

- a) use 数据库名称

```
> use test
switched to db test
```

注意:

use 命名 存在库使用当前库 不存在则创建当前库

5. 删除一个库

db.dropDatabase();



```
> show dbs;
local  0.078GB
test   0.078GB
> use test
switched to db test
> db.dropDatabase();
{ "dropped" : "test", "ok" : 1 }
> show dbs;
local  0.078GB
```

注意: 选中那个库,删除的就是当前选中的库

6. Mongodb 的数据库中,库中是一个一个集合的概念,选中库后要创建一个一个的集合,集合类似于传统的关系型数据库中的表

- a) 显示创建集合 `db.createCollection("t_user");`

```
> use test
switched to db test
> db.createCollection("t_user");
{ "ok" : 1 }
> show collections;
system.indexes
t_user
```

- b) 隐式创建集合 创建集合同时添加元素

```
> db.t_product.insert({name:"iphone",color:["金色","红色","白色"],bir:new Date()});
WriteResult({ "nInserted" : 1 })
> show collections
system.indexes
t_product
t_user
```

7. 查看 mongo 中当前库

- a) db 命令显示当前库

```
> db
test
```

8. 显示当前库中的所有集合

- a) show collections;

```
> show collections;
system.indexes
t_product
t_user
```

9. Mongo 中插入数据



- a) 向集合中插入数据
`db.集合名称.insert({name:'xiaohei',age:23,sex:true});`
- b) 向集合中插入多条数据
`db.集合名称.insert([{name:'xiaohei',age:23,sex:true},...]);`
- 10. Mongo 中的删除数据
 - a) `db.集合名称.remove({条件})` //删除满足条件的数据
 - b) `db.集合名称.remove({不加任何条件})`//删除所有文档 保留空的集合
- 11. Mongo 中的修改数据
 - a) `db.集合名称.update({条件},{更新内容});`
 - b) `db.集合名称.update({"name":"zhangsan"},{name:"11",bir:new Date()})` --这个更新是将符合条件的全部更新成后面的文档,相当于先删除在更新
 - c) `db.集合名称.update({"name":"xiaohei"},{$set:{name:"mingming"}})` --保留原来的值修改,但是只更新符合条件的第一条数据
 - d) `db.集合名称.update({name:"小黑"},{$set:{name:"小明"}},{multi:true})` ---保留原来数据更新,更新符合条件的所有数据
 - e) `db.集合名称.update({name:"小黑"},{$set:{name:"小明"}},{multi:true,upsert:true})` ---保留原来数据更新,更新符合条件的所有数据 没有条件符合时插入数据
 - f) `db.t_user.update({name:"zhangsan"},{$inc:{age:1}},{upsert:true,multi:true})` -- 在保留原始数据同时给符合条件的所有 age 这列的值自增指定的大小
 - g)
- 12. 删除集合
 - a) `db.集合名称.drop();`
- 13. 查询集合
 - a) `db.集合名称.find();`
 - b) `db.集合名称.find({条件})`
 - c) `db.集合名称.find({条件},{显示字段,name:1,age:1})` 1 显示 0 不显示 1 和 0 不能混合出现
 - d) 查询结果排序:`db.集合名称.find().sort({条件 name:1,age:1}), 1 升序 -1 降序`
 - e) 分页查询:`db.集合名称.find().sort({条件}).skip(起始条数).limit(显示总记录数);`
 - f) 总条数:`db.集合名称.count();|db.t_user.find({"name":"aa"}).count();`
 - g) 模糊查询:使用正则表达式 `db.集合名称.find({"name":"/go/})`
 - h) 等值(==)查询
`db.user.find({name:"张三"});`
`db.user.find({name:{$eq:"张三"}});`
 - i) 且 (\$and) 查询
`db.t_user.find({name:"zhangsan",age:12});`
`db.t_user.find({$and:[{name:"zhangsan"},{id:10}]})`
 - j) \$or 使用:
 - i.
`db.集合名称.find({`
 `$or:[`
 `{key:value},{age:{$gte:20}}`
 `]`
 `});`



- k) \$gt 大于 \$gte 大于等于 \$lt 小于 \$lte 小于等于 \$eq 等于:
 - i. db.集合名称.find({"age":{"\$lte":18,"\$gte":30}})
 - l) \$nor 查询使用
 - i. db.t_user.find({\$nor:[{name:"chenyn"},{age:26}]});
14. shell 非正常关闭时,下次无法连接问题解决方案:
- i. 删除数据目录中的 mongo.lock 文件即可

五、Java 操作 MongoDB

1. 项目中引入 mongo 的坐标

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongo-java-driver</artifactId>
  <version>3.0.0</version>
</dependency>
```

2. 使用 java 操作 mongo
- a) 参见代码 mongo-java

六、Spring Boot 整合 MongoDB

1. pom 配置

pom 包里面添加 spring-boot-starter-data-mongodb 包引用

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-mongodb</artifactId>
  </dependency>
</dependencies>
```

2. 在 application.properties 中添加配置
- ```
spring.data.mongodb.uri=mongodb://localhost:27017/test
```
3. 详细操作参见 springboot-mongo