

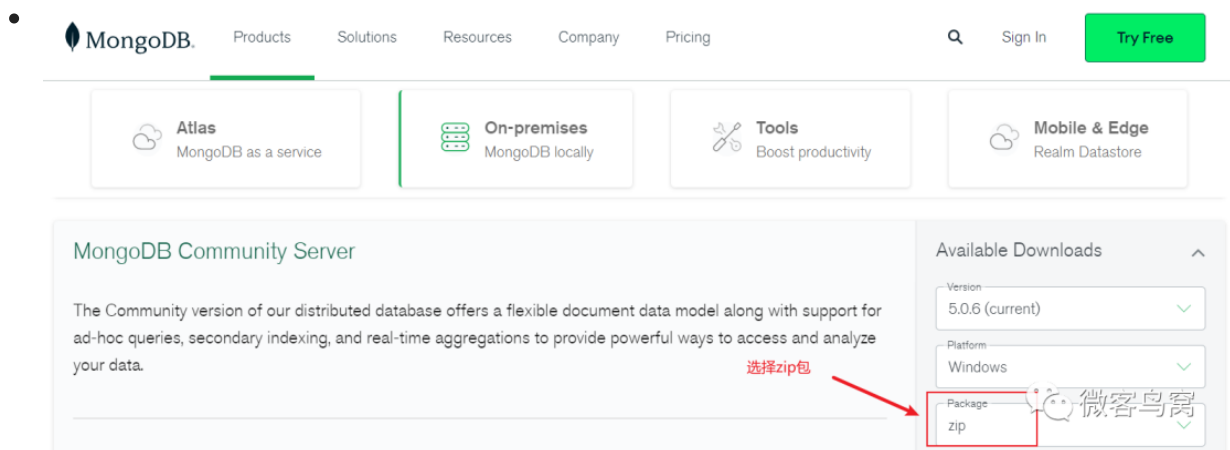
一文搞定 mongodb (学习笔记)

mp.weixin.qq.com/s/eIXW aw5umSBBTzQgZUuag

相见即是缘，本号致力于后端技术分享，欢迎大家加我wucs_dd进技术交流群,更多学习资料获取请见[王货分享](#)。

一、安装MongoDB

- 下载地址：<https://www.mongodb.com/try/download/community>



- 此处我下载的是zip版本，在 E 盘 software 目录创建了一个 **mongoDB** 的文件夹，并将压缩包解压到了这个文件夹里。

创建目录及配置文件

1. 在 bin 目录同级的目录创建 data 目录，继续在 data 目录下创建 db 及 log 文件夹。
log 目录中还需要创建 mongod.log 文件。这个文件一定要创建否则找不到会报错。
- db：表示数据存储的文件夹
• log：表示日志打印的文件夹
2. 在 bin 目录的同级目录创建 mongod.cfg 文件，并写入如下内容（注意：配置件缩进需要使用tab键，路径根据自己的实际情况填写）。

```
systemLog:
  destination: file
  path: E:\software\mongoDB\data\log\mongod.log
storage:
  dbPath: E:\software\mongoDB\data\db
net:
  port: 27017
```

- path：配置打印日志的目录
- dbpath：配置数据的存储位置
- port：配置的端口号

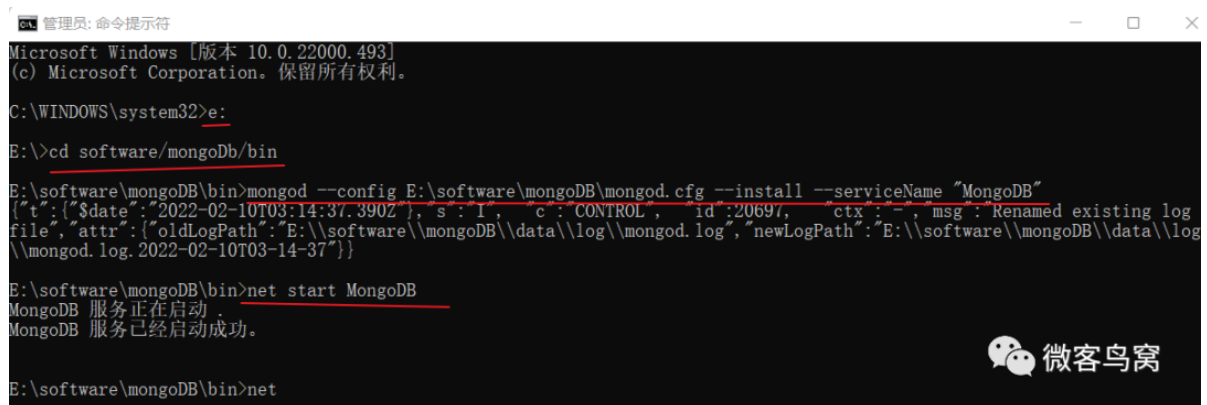
配置环境变量

打开环境变量中的path，选择path双击，打开后点击“新建”，把bin目录的路径粘贴进去（我的是：E:\mongodb\bin）

添加服务（开机自动启动）

注意：以管理员身份运行CMD

1. 方法一：从配置文件启动



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.22000.493]
(c) Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>e:

E:\>cd software/mongodb/bin

E:\software\mongodb\bin>mongod --config E:\software\mongodb\mongod.cfg --install --serviceName "MongoDB"
{"t":{"$date":"2022-02-10T03:14:37.390Z"},"s":"I","c":"CONTROL","id":20697,"ctx":"-","msg":"Renamed existing log file","attr":{"oldLogPath":"E:\\software\\mongodb\\data\\log\\mongod.log","newLogPath":"E:\\software\\mongodb\\data\\log\\mongod.log.2022-02-10T03-14-37"}}

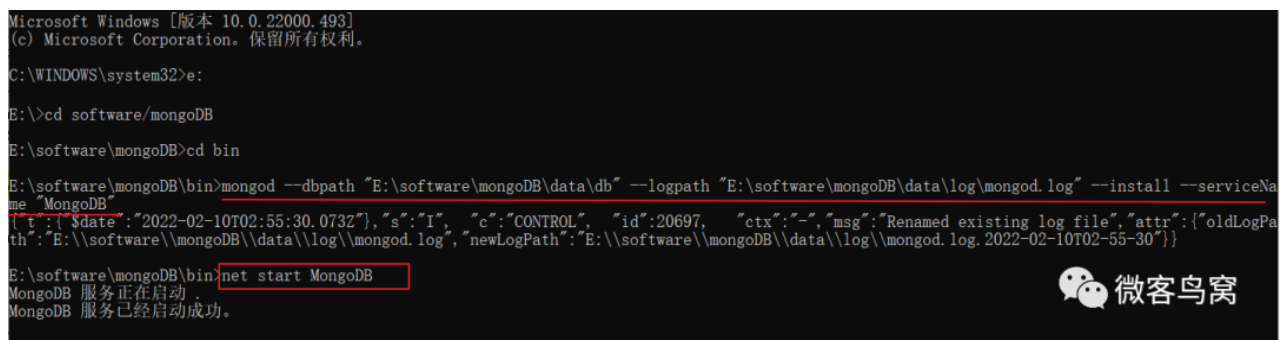
E:\software\mongodb\bin>net start MongoDB
MongoDB 服务正在启动。
MongoDB 服务已经启动成功。

E:\software\mongodb\bin>net
```

切换到bin目录中，执行此命令，表示执行配置文件(注意修改自己的路径)：`mongod --config E:\software\mongodb\mongod.cfg --install --serviceName "MongoDB"`

2. 方法二：直接命令行中配置

如果配置文件执行报错，无法启动，那么就自己在命令行配置mongodb，切换到bin目录中，执行此命令(注意修改自己的路径)：



```
Microsoft Windows [版本 10.0.22000.493]
(c) Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>e:

E:\>cd software/mongodb

E:\software\mongodb>cd bin

E:\software\mongodb\bin>mongod --dbpath "E:\software\mongodb\data\db" --logpath "E:\software\mongodb\data\log\mongod.log" --install --serviceName "MongoDB"
{"t":{"$date":"2022-02-10T02:55:30.073Z"},"s":"I","c":"CONTROL","id":20697,"ctx":"-","msg":"Renamed existing log file","attr":{"oldLogPath":"E:\\software\\mongodb\\data\\log\\mongod.log","newLogPath":"E:\\software\\mongodb\\data\\log\\mongod.log.2022-02-10T02-55-30"}}

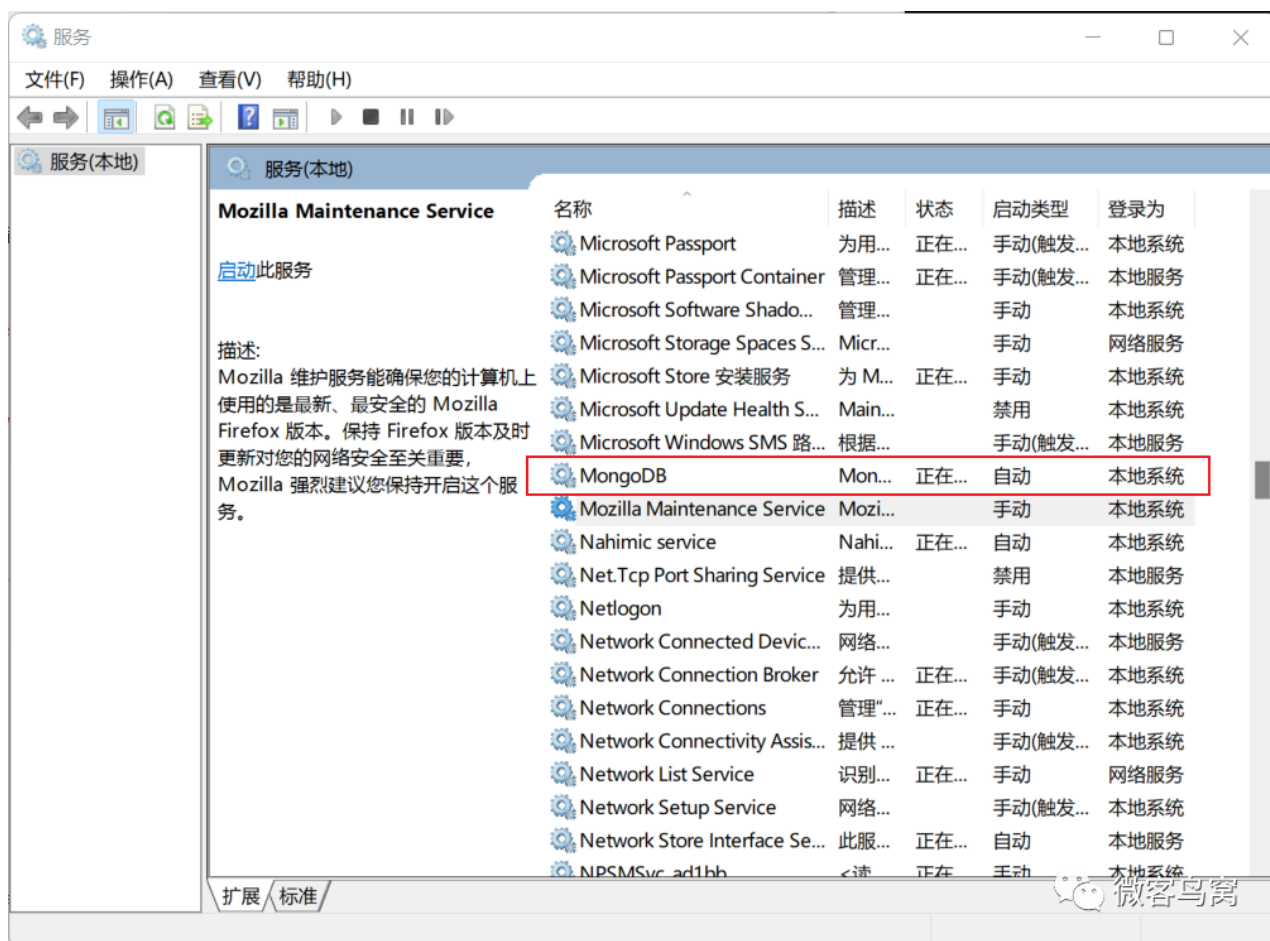
E:\software\mongodb\bin>net start MongoDB
MongoDB 服务正在启动。
MongoDB 服务已经启动成功。
```

`mongod --dbpath "E:\software\mongodb\data\db" --logpath "E:\software\mongodb\data\log\mongod.log" --install --serviceName "MongoDB"`

- 开启服务器: net start MongoDB
- 关闭服务器: net stop MongoDB
- 删除服务: sc delete MongoDB
- 打开客户端: mongo，在cmd窗口输入便可以连接上数据库

查看服务

在键盘上按“WIN+R”，输入“services.msc”指令确定，打开服务后，找到MongoDB，如下图所示表示已经成功！



后续如果修改 mongoDB 的端口就直接改配置文件，删除服务，再重新执行“添加服务”。

二、基础介绍

三个概念

1. 数据库 (database) 数据库是一个仓库，在仓库中可以存放集合。
2. 集合 (collection) 集合类似于数组，在集合中可以存放文档。
3. 文档 (document) 文档数据库中的最小单位，我们存储和操作的内容都是文档。

常用命令

- show dbs : 显示当前所有数据库 show database : 显示当前所有数据库
- use 数据库名 : 进入到指定的数据库中 (不存在则创建)
- db : 表示我们当前所处的数据库
- show collections : 显示我们数据库中所有的集合

三、安装图形化工具

- mongodbmanger :
- 下载地址 : <https://www.mongodbmanager.com/download>

三、CRUD 操作

插入文档

```
db.<collection>.insert()
```

- 向集合中插入一个或多个文档
- 当我们向集合中插入文档时，如果没有给文档指定_id属性，则数据库会自动给文档添加_id
- 该属性用来作为文档的唯一标识
- _id可以自己指定，如果我们指定了，数据库就不会再添加了,如果自己指定_id也必须确保唯一性

```
db.<collection>.insertOne() //插入一个文档对象
```

```
db.<collection>.insertMany() //插入多个文档对象
```

```
//插入一个文档对象
```

```
db.stus.insert({name:"微客鸟窝",age:18,gender:"男"})
```

```
//插入多个文档对象
```

```
db.stus.insert([
  {name:"沙和尚",age:36,gender:"男"},
  {name:"白骨精",age:16,gender:"女"},
  {name:"蜘蛛精",age:14,gender:"女"}
])
```

查询文档

```
db.<collection>.find()
```

- find () 用来查询集合中所有符合条件的文档
- find () 可以接收一个对象作为条件参数
 - {} : 表示查询集合中所有文档
 - { 属性 : 值 } : 查询属性是指定值的文档
 - 返回值是一个数组

```
db.<collection>.findOne()
```

- 用来查询集合中符合条件的第一个文档
- 返回的是一个文档

```
db.<collection>.find({}).count()
```

查询所有结果的数量

```
db.stus.find({name:"沙和尚"})
db.stus.find({name:"白骨精",age:16})
db.stus.findOne({gender:"女"})
db.stus.find({}).count()
db.stus.find({}).length()
```

修改文档

- `db.<collection>.update(查询条件, 新对象)`
 - `update()` 默认情况下会使用新对象来替换旧对象
 - `update()` 默认只会修改一个对象 如果需要修改指定的属性，而不是替换，需要使用“修改操作符”来完成修改
- `$set`：可以用来修改文档中的指定属性
- `$unset`：可以用来删除文档的指定属性
- `db.<collection>.updateMany()`：同时修改多个符合条件的文档
- `db.<collection>.updateOne()`：修改一个符合条件的文档
- `db.<collection>.replaceOne()`：替换一个符合条件的文档
 - MongoDB的文档的属性值也可以是一个文档，当一个文档的属性值是文档时，我们称这个文档为内嵌文档
 - MongoDB支持直接通过内嵌文档的属性进行查询，如果要查询内嵌文档可以则可以通过`==`的形式来匹配，且属性名必须使用引号`==`,双引号单引号都可以

```
db.stus.find()
```

```
db.stus.update(  
  {name:"白骨精"}, {age:28}  
)
```

```
db.stus.update(  
  {"name":"沙和尚"},  
  {$set:{  
    gender:"男",  
    address:"流沙河"  
  }}  
)
```

```
//update()默认修改一条 = updateOne()  
db.stus.update(  
  {"name":"沙和尚"},  
  {$unset:{address:"流沙河"}}  
)
```

```
db.stus.updateMany(  
  {"name" : "猪八戒"},  
  {$set:{address:"高老庄"}}  
)  
//multi:true 则可修改所有满足条件的文档  
db.stus.update(  
  {"name" : "猪八戒"},  
  {$set:{address:"猪老庄"}},  
  {multi:true}  
)
```

删除文档

`db.<collection>.remove()`

- 可以根据条件来删除文档，传递条件的方式和find()一样
- 能删除符合条件的所有文档，默认删除多个
- 如果第二个参数传递一个true，则只会删除一个
- 如果只传递一个{}作为参数，则会删除集合中的所有文档(效果相当于drop()，但是性能不如drop())

- `db.<collection>.deleteOne()`

- `db.<collection>.deleteMany()`

- `db.<collection>.drop()`

删除集合（如果最后一个集合没了，数据库也没了）

- `db.dropDatabase()` 删除数据库

练习

```
//添加两万条数据的性能高，尽量少调用系统的方法
var arr=[];
for(var i=1;i<=20000;i++){
    arr.push({num:i});
}
db.user.insert(arr);
```

```
//查询numbers中num大于5000的文档
db.unmbers.find({num:{$gt:500}})
```

```
//查询numbers中num小于30的文档
db.unmbers.find({num:{$lt:500}})
```

```
//查询numbers中num大于40小于50的文档
db.numbers.find({num:{$gt:40,$lt:50}})
```

```
//查询numbers前10条的数据
db.numbers.find({num:{$lte:10}})
```

```
//limit()设置显示数据的上限
db.numbers.find().limit(10)
```

```
//查询numbers中第11条到20条的数据
//skip()用于跳过指定数量的数据 skip( (页码-1)*每页显示的条数 ).limit(每页显示的条数)
//MongoDB会自动调整limit()和skip()的位置，下面两条语句效果一样
db.numbers.find().skip(10).limit(10)
db.numbers.find().limit(10).skip(10)
```

文档间的关系

一对一 (one to one)

- 夫妻
- 在MongoDB中，可以通过内嵌文档的形式来体现出一对一的关系

```
db.WifeAndHusband.insert([
  {
    wife:"黄蓉",
    husband:{
      name:"郭靖"
    }
  },

  {
    wife:"潘金莲",
    husband:{
      name:"武大郎"
    }
  }
])
```

一对多 (one to many)

一对多：父母和孩子、用户和订单、文章和评论，也可以通过内嵌文档的方式来映射一对多的关系（将1的那个属性设置为多的里面的字段）

```
// 第一句是创建了两个用户，分别是孙悟空和猪八戒
db.users.insert([{name:"孙悟空"}, {name:"猪八戒"}]);
// 这一句是查询用户内容，这是需要知道用户的_id，从而在第三句加入
db.users.find();
// 这是生成了一个订单，并且user_id存储的是孙悟空的id
db.orders.insert({list_name:["香蕉", "桃子"], user_id: ObjectId("812419dc00852cj76m2b4d10")})
// 再生成了一个订单，并且user_id存储的是孙悟空的id
db.orders.insert({list_name:["西瓜", "人参果"], user_id: ObjectId("812419dc00852cj76m2b4d10")})
// 在users里查找到孙悟空这个用户的id，再通过此id查找孙悟空的订单
let userId = db.users.findOne({name:"孙悟空"})._id;
// 使用id来查找订单
db.orders.find({user_id: userId});
```

多对多 (many to many)

分类和商品，通过内嵌文档的方式

```

db.teacher.insert([
  {name:"洪七公"},
  {name:"黄药师"},
  {name:"龟仙人"}
])

db.stus.insert([
  {
    name:"郭靖",
    tech_ids:[
      ObjectId("62077e5c10b3aaaec77e55b9"),
      ObjectId("62077e5c10b3aaaec77e55ba")
    ]
  },
  {
    name:"孙悟空",
    tech_ids:[
      ObjectId("62077e5c10b3aaaec77e55bb"),
      ObjectId("62077e5c10b3aaaec77e55bc"),
      ObjectId("62077eb210b3aaaec77e55bd")
    ]
  }
])

```

sort和投影

1. sort

- find () 查询文档时，默认情况是按照_id的值进行升序排列
- sort () 可以用来指定文档的排序的规则，需要传递一个属性来指定排序规则，1表示升序，-1表示降序

```

db.users.find({}).sort({age:1})
db.users.find({}).sort({age:1, salary:-1}) //先指定age的升序 再salary的降序

```

limit、skip、sort可以任意顺序的调用

2. 投影

- 查询时，我们可以在第二个参数的位置来设置查询结果的投影(指定要显示的字段)
 {name: 1, _id: 0}：表示显示name属性，不显示_id属性。

```

db.users.find({}, {name: 1, _id: 0, salary: 1});
/*
  这句的含义是只显示 name和salary属性，不显示 _id属性
*/

```


图片及部分相关技术知识点来源于网络搜索，侵权删！

参考资料：

<https://www.bilibili.com/video/BV18s411E78K>

NEW////ARRIVAL

微信公众号

gophpython

我的微信

wucs_dd

