typora-copy-images-to: mongodb_img

# mongodb安装

# 1 mongodb介绍

## 简介

编辑

MongoDB [1] 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。

MongoDB [2] 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。它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最大的特点是它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。 [3]


mongoDB [1]

# 2下载mongodb

MongoDB 提供了可用于 32 位和 64 位系统的预编译二进制包，你可以从MongoDB官网下载安装。

官方地址：https://www.mongodb.com/

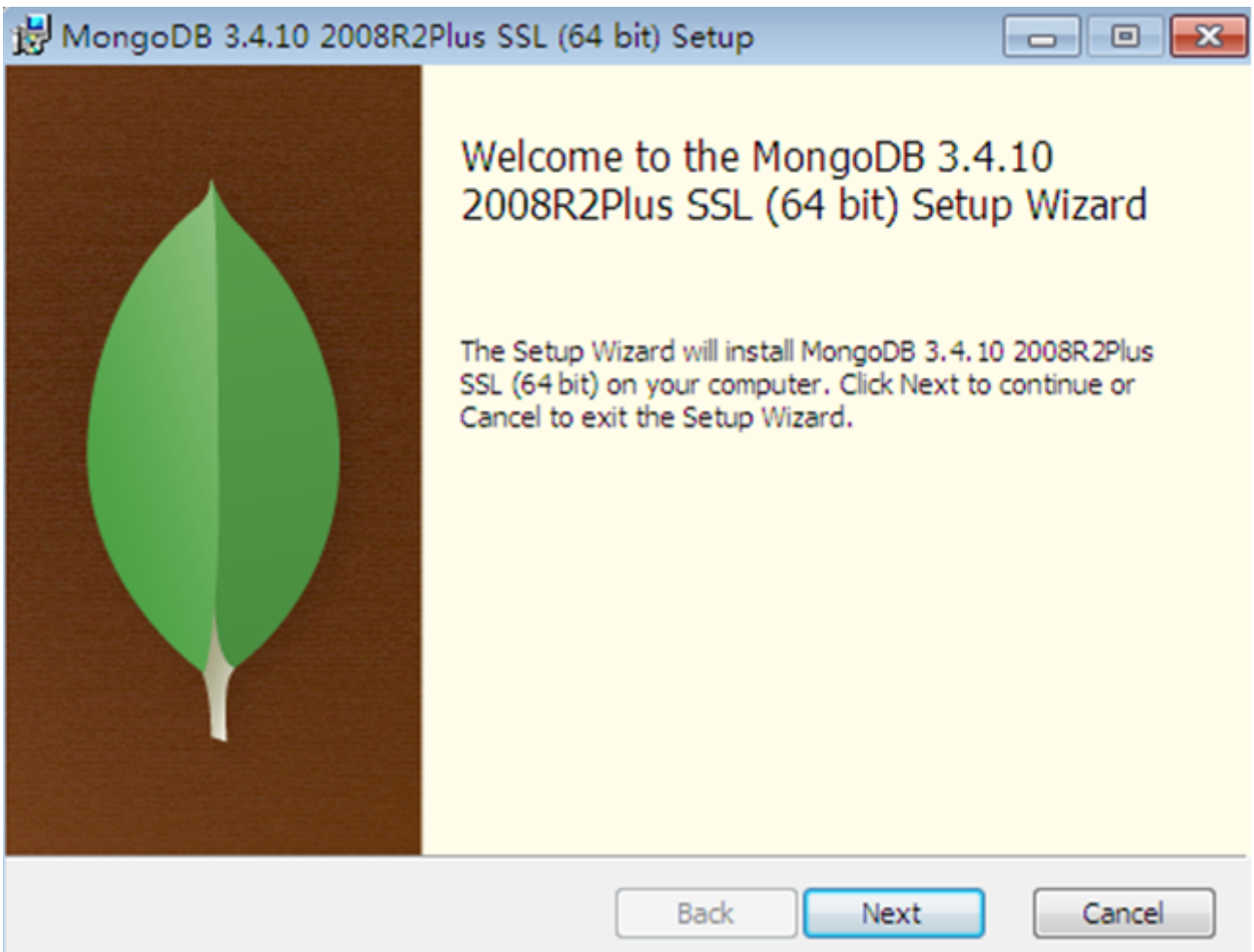
本教程下载3.4版本：http://downloads.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-v3.4-latest-signed.msi

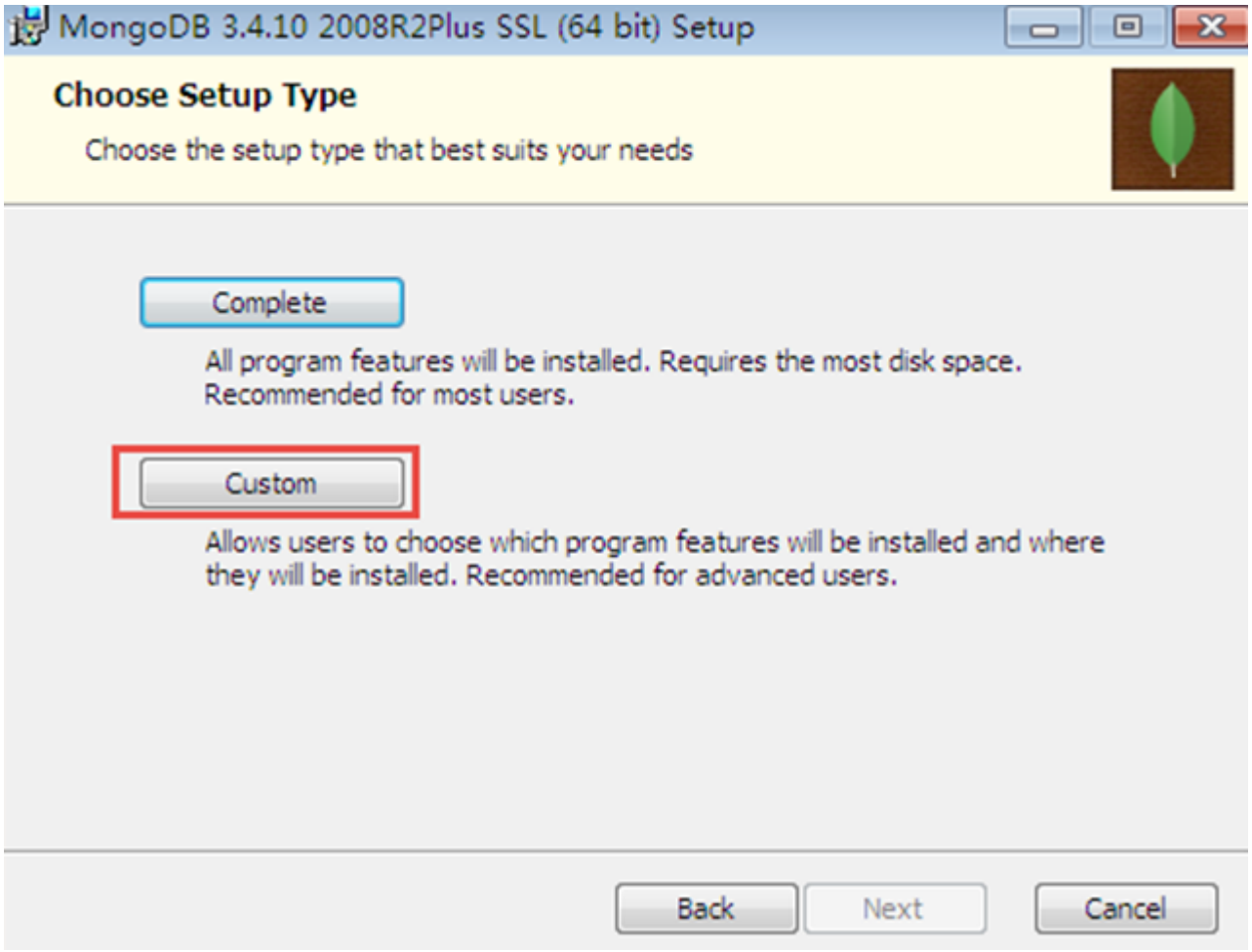
# 3安装mongodb

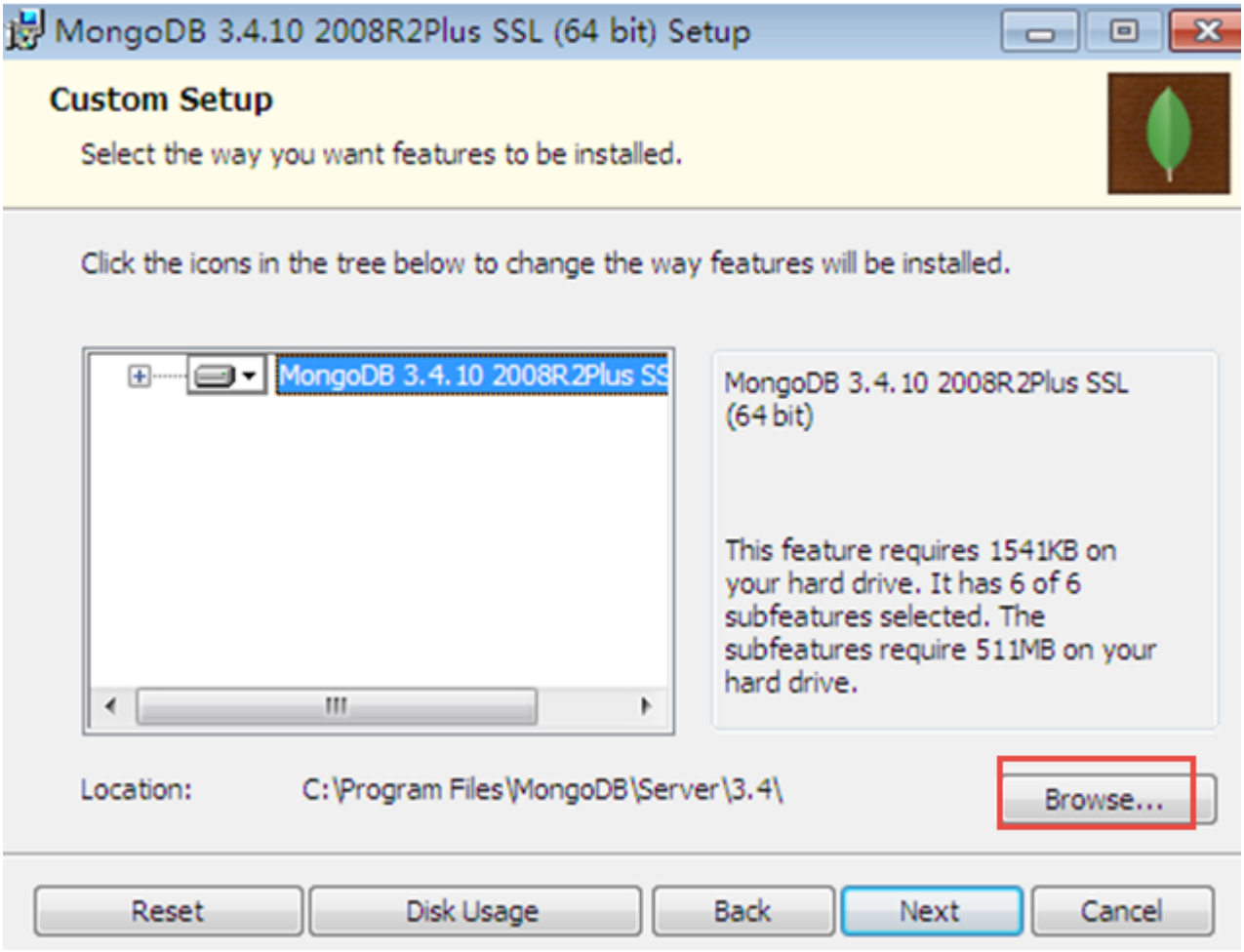
在win7系统安装mongodb需要vc++运行库，如果没有则会提示"无法启动此程序，因为计算机中丢失VCRUNTIME140.dll"。

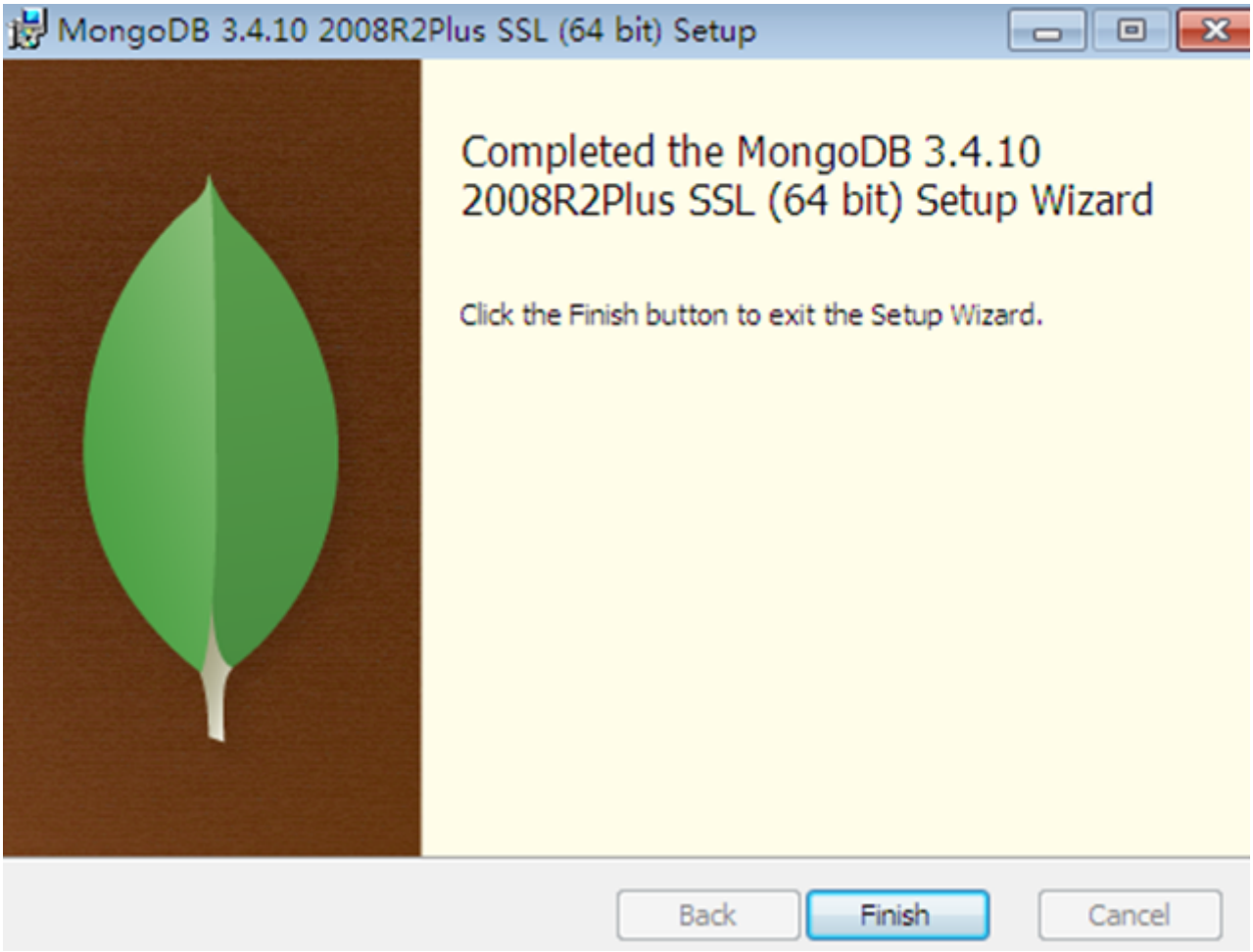
去网上下载或使用360安装：


运行 mongodb-win32-x86_64-2008plus-ssl-v3.4-latest-signed.msi



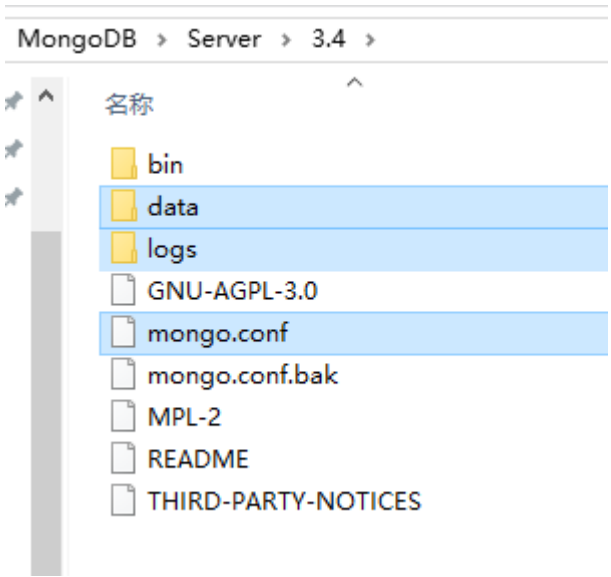

MongoDB 3.4.10 2008R2Plus SSL (64 bit) Setup
Choose Setup Type
Choose the setup type that best suits your needs
Complete
All program features will be installed. Requires the most disk space.
Recommended for most users.
Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.
Back
Next
Cancel


MongoDB 3.4.10 2008R2Plus SSL (64 bit) Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
MongoDB 3.4.10 2008R2Plus SS
MongoDB 3.4.10 2008R2Plus SSL (64 bit)
This feature requires 1541KB on your hard drive. It has 6 of 6 subfeatures selected. The subfeatures require 511MB on your hard drive.
Location:
C:\Program Files\MongoDB\Server\3.4\
Browse...
Reset
Disk Usage
Back
Next
Cancel




# 4 启动mongodb

创建几个文件夹具体如下：数据库路径（data目录）、日志路径（logs目录）和日志文件（mongo.log文件）



创建配置文件mongo.conf，文件内容如下：

```
#数据库路径
dbpath=d:\MongoDB\Server\3.4\data
#日志输出文件路径
logpath=d:\MongoDB\Server\3.4\logs\mongo.log
#错误日志采用追加模式
logappend=true
#启用日志文件，默认启用
journal=true
#这个选项可以过滤掉一些无用的日志信息，若需要调试使用请设置为false
quiet=true
#端口号 默认为27017
port=27017
```

安装 MongoDB服务

通过执行bin/mongod.exe，使用--install选项来安装服务，使用--config选项来指定之前创建的配置文件。 cmd进入d:\MongoDB\Server\3.4\bin

```
mongod.exe --config "d:\MongoDB\Server\3.4\mongo.conf" --install
```

启动MongoDB服务

```
net start MongoDB
```

关闭MongoDB服务

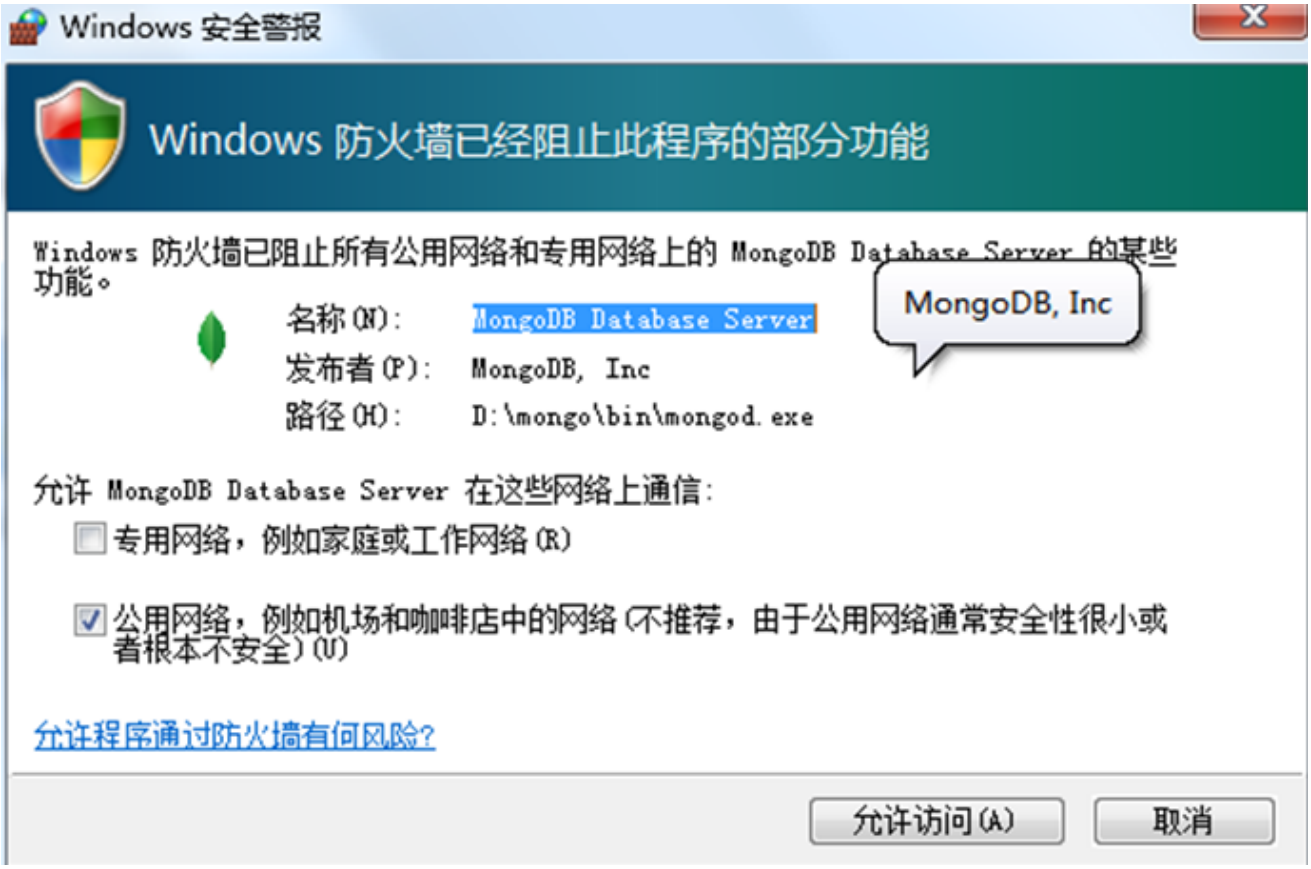
```
net stop MongoDB
```

移除MongoDB服务

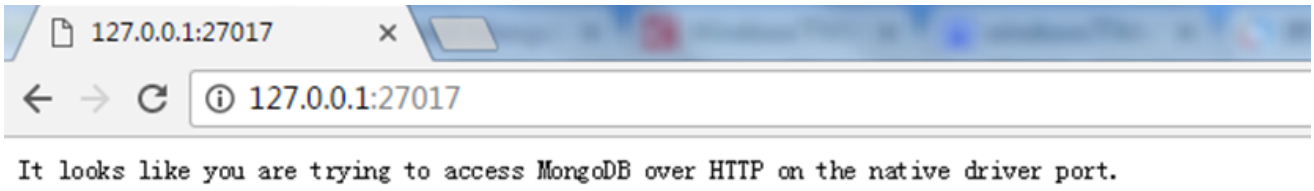
```
" d:\MongoDB\Server\3.4\bin\mongod.exe" --remove
```

启动mongodb服务，

命令执行后，浏览器中输入http://127.0.0.1:27017看到如下界面即说明启动成功




出现下图说明安装成功



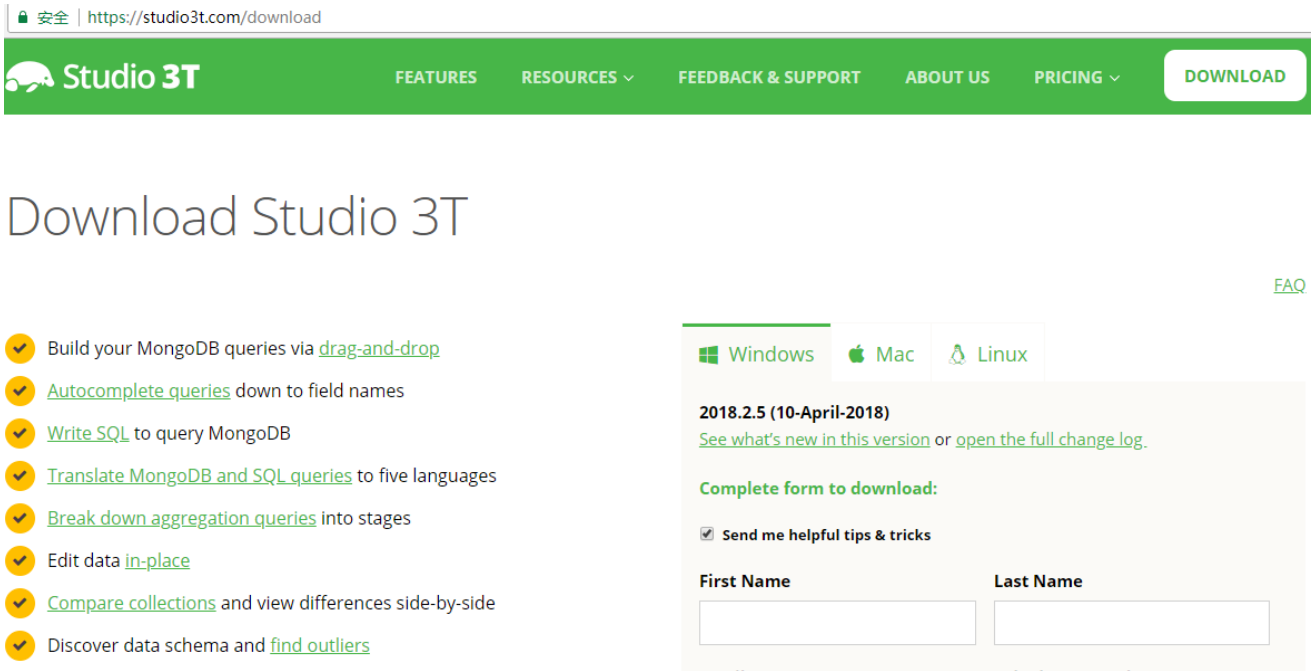

也可以通过bin目录下的mongo.exe连接mongodb

```
PS D:\MongoDB\Server\3.4\bin> .\mongo.exe
MongoDB shell version v3.4.10-72-gfbb20bc
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.10-72-gfbb20bc
Server has startup warnings:
2018-04-19T14:39:34.358+0800 I CONTROL  [initandlisten]
2018-04-19T14:39:34.358+0800 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the
2018-04-19T14:39:34.358+0800 I CONTROL  [initandlisten] **          Read and write access to data and conf
nrestricted.
2018-04-19T14:39:34.358+0800 I CONTROL  [initandlisten]
>
```

# 5 安装studio3t

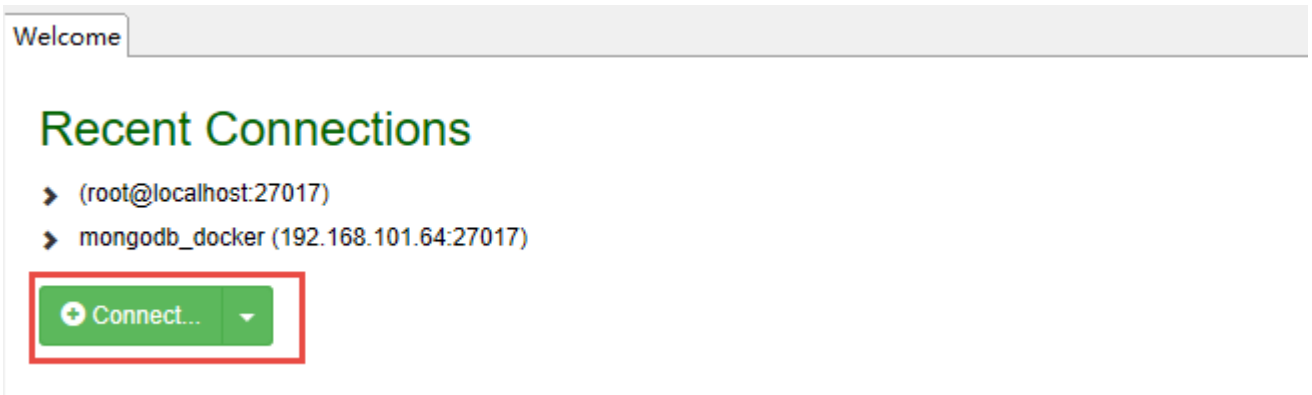
studio3t是mongodb优秀的客户端工具。官方地址在https://studio3t.com/

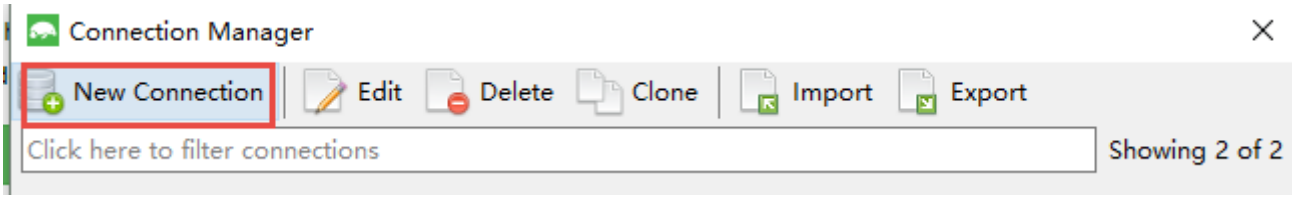
下载studio3t



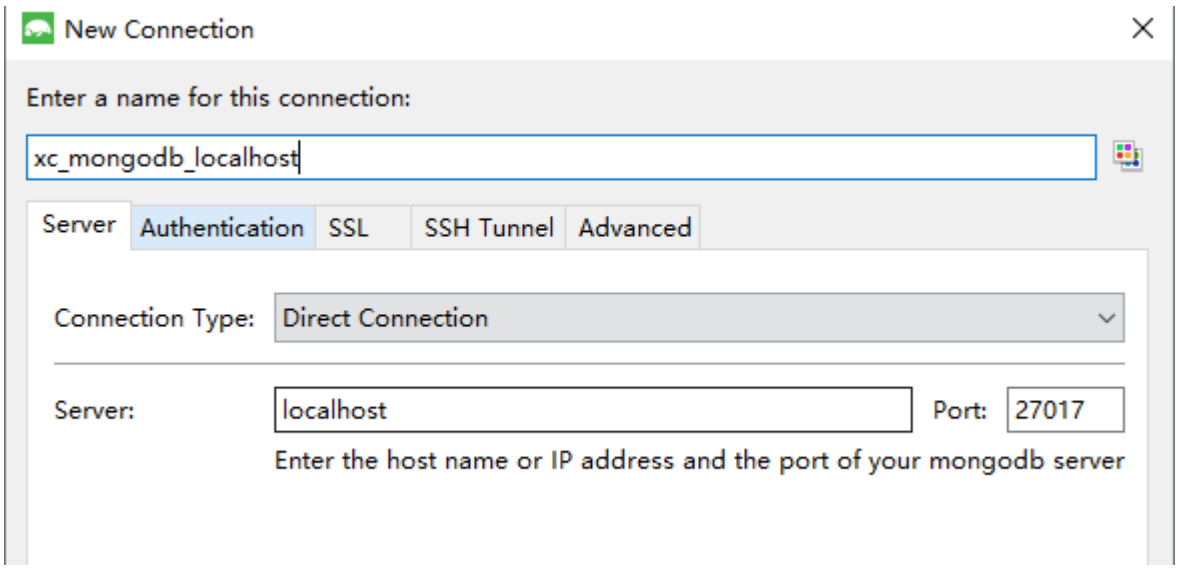

安装并启动：


创建一个新连接：

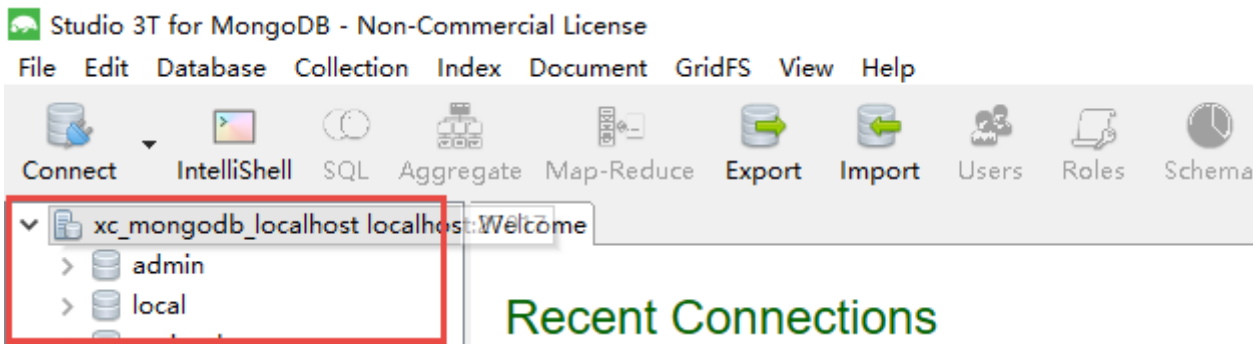





填写连接信息：



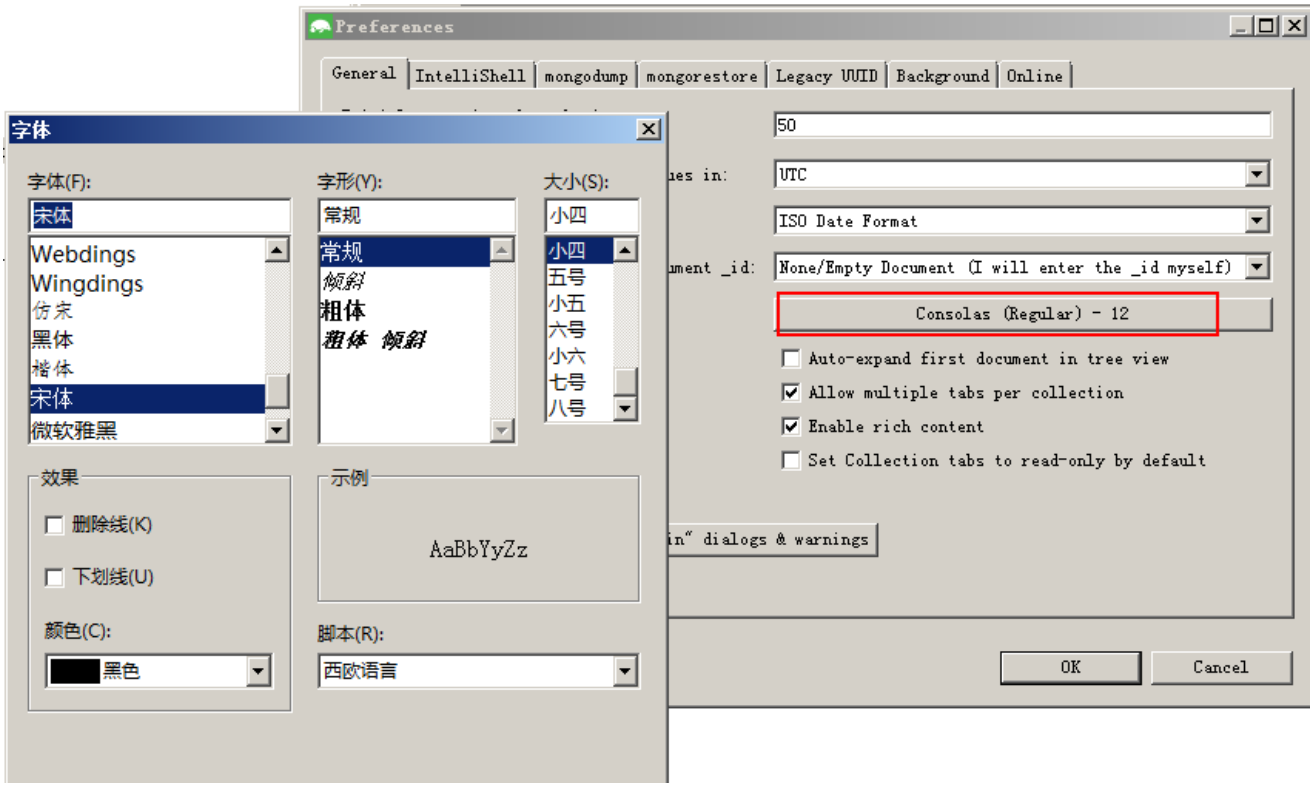

连接成功：




修改字体：

默认Studio3t的字体太小，需要修改字体：

点击菜单：Edit--->Preferences

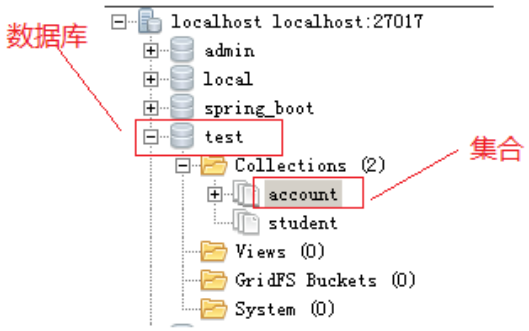



# 6 mongodb入门

## 6.1 基础概念

在mongodb中是通过数据库、集合、文档的方式来管理数据，下边是mongodb与关系数据库的一些概念对比：

| SQL术语/概念 | MongoDB术语/概念 | 解释/说明 |
| --- | --- | --- |
| database | database | 数据库 |
| table | collection | 数据库表/集合 |
| row | document | 数据记录行/文档 |
| column | field | 数据字段/域 |
| index | index | 索引 |
| table joins | | 表连接（MongoDB不支持） |
| primary key | primary key | 主键,MongoDB自动在每个集合中添加_id的主键 |




一个文档

```
{
    "_id" : ObjectId("5b2cc4bfa6a44812707739b5"),
    "name" : "黑马程序员"
}
```

1、一个mongodb实例可以创建多个数据库

2、一个数据库可以创建多个集合

3、一个集合可以包括多个文档。

## 6.2 连接mongodb

mongodb的使用方式是客户服务器模式，即使用一个客户端连接mongodb数据库（服务端）。

1、 命令格式

```
mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?
options]]
```

mongodb:// 固定前缀

username：账号，可不填

password：密码，可不填

host：主机名或ip地址，只有host主机名为必填项。

port：端口，可不填，默认27017

/database：连接某一个数据库

?options：连接参数，key/value对

例子：

```
mongodb://localhost 连接本地数据库27017端口
mongodb://root:itcast@localhost 使用用户名root密码为itcast连接本地数据库27017端口
mongodb://localhost,localhost:27018,localhost:27019，连接三台主从服务器，端口为27017、27018、27019
```

2、使用mongodb自带的javascript shell（mongo.exe）连接

windows版本的mongodb安装成功，在安装目录下的bin目录有mongo.exe客户端程序


cmd状态执行mongo.exe：

```
D:\MongoDB\Server\3.4\bin>mongo.exe
MongoDB shell version v3.4.10-72-gfbb20bc
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.10-72-gfbb20bc
Server has startup warnings:
2018-06-22T12:34:57.411+0800 I CONTROL  [initandlisten]
2018-06-22T12:34:57.411+0800 I CONTROL  [initandlisten] ** WARNING: Access
2018-06-22T12:34:57.411+0800 I CONTROL  [initandlisten] **          Read a
2018-06-22T12:34:57.411+0800 I CONTROL  [initandlisten]
>
```

此时就可以输入命令来操作mongodb数据库了，javascript shell可以运行javascript程序。

3、使用studio3T连接

4、使用java程序连接

详细参数：http://mongodb.github.io/mongo-java-driver/3.4/driver/tutorials/connect-to-mongodb/

添加依赖：

```
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.4.3</version>
</dependency>
```

测试程序：

```
@Test
public void testConnection(){
    //创建mongodb 客户端
    MongoClient mongoClient = new MongoClient( "localhost" , 27017 );
    //或者采用连接字符串
    //MongoClientURI connectionString = new
MongoClientURI("mongodb://root:root@localhost:27017");
     //MongoClient mongoClient = new MongoClient(connectionString);
    //连接数据库
```

```
    MongoDatabase database = mongoClient.getDatabase("test");
    // 连接collection
    MongoCollection<Document> collection = database.getCollection("student");
    //查询第一个文档
    Document myDoc = collection.find().first();
    //得到文件内容 json串
    String json = myDoc.toJson();
    System.out.println(json);
}
```

## 6.3 数据库

1、查询数据库

show dbs 查询全部数据库

db 显示当前数据库

2、创建数据库

命令格式：

```
use DATABASE_NAME
```

例子：

use test02

有test02数据库则切换到此数据库，没有则创建。

注意：

新创建的数据库不显示，需要至少包括一个集合。

3、删除数据库（慎用！！！）

命令格式：

```
db.dropDatabase()
```

例子：

删除test02数据库

先切换数据库：use test02

再执行删除：db.dropDatabase()

## 6.4 集合

集合相当于关系数据库中的表，一个数据库可以创建多个集合，一个集合是将相同类型的文档管理起来。

1、创建集合

```
db.createCollection(name, options)
name: 新创建的集合名称
options: 创建参数
```

2、删除集合

```
db.collection.drop()
例子：
db.student.drop() 删除student集合
```

# 6.5 文档

## 6.5.1 插入文档

mongodb中文档的格式是json格式，下边就是一个文档，包括两个key：_id主键和name

```
{
    "_id" : ObjectId("5b2cc4bfa6a44812707739b5"),
    "name" : "黑马程序员"
}
```

插入命令：

```
db.COLLECTION_NAME.insert(document)
```

每个文档默认以_id作为主键，主键默认类型为ObjectId（对象类型），mongodb会自动生成主键值。

例子：

```
db.student.insert({"name":"黑马程序员","age":10})
```

注意：同一个集合中的文档的key可以不相同！但是建议设置为相同的。

## 6.5.2 更新文档

命令格式：

```
db.collection.update(
   <query>,
   <update>,
   <options>
)
query:查询条件，相当于sql语句的where
update：更新文档内容
options：选项
```

1、替换文档

将符合条件 "name":"北京黑马程序"的第一个文档替换为{"name":"北京黑马程序员","age":10}。

```
db.student.update({"name":"黑马程序员"},{"name":"北京黑马程序员","age":10})
```

2、$set修改器

使用$set修改器指定要更新的key，key不存在则创建，存在则更新。

将符合条件 "name":"北京黑马程序"的所有文档更新name和age的值。

```
db.student.update({"name":"黑马程序员"},{$set:{"name":"北京黑马程序员","age":10}},{multi:true})
```

multi：false表示更新第一个匹配的文档，true表示更新所有匹配的文档。

## 6.5.3 删除文档

命令格式：

```
db.student.remove(<query>)
query：删除条件，相当于sql语句中的where
```

1、删除所有文档

db.student.remove({})

2、删除符合条件的文档

db.student.remove({"name":"黑马程序员"})

## 6.5.4 查询文档

命令格式：

```
db.collection.find(query, projection)
query：查询条件，可不填
projection：投影查询key，可不填
```

1、 查询全部

```
db.student.find()
```

2、查询符合条件的记录

查询name等为"黑马程序员"的文档。

```
db.student.find({"name":"黑马程序员"})
```

3、投影查询

只显示name和age两个key，_id主键不显示。

```
db.student.find({"name":"黑马程序员"},{name:1,age:1,_id:0})
```

# 6.6 用户

## 6.6.1 创建用户

语法格式：

```
mongo>db.createUser(
{ user: "<name>",
  pwd: "<cleartext password>",
  customData: { <any information> },
  roles: [
    { role: "<role>", db: "<database>" } | "<role>",
    ...
  ]}
)
```

例子：

创建root用户，角色为root

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

内置角色如下：

1. 数据库用户角色：read、readWrite;
2. 数据库管理角色：dbAdmin、dbOwner、userAdmin；

3. 集群管理角色：clusterAdmin、clusterManager、clusterMonitor、hostManager；
4. 备份恢复角色：backup、restore；
5. 所有数据库角色：readAnyDatabase、readWriteAnyDatabase、userAdminAnyDatabase、dbAdminAnyDatabase
6. 超级用户角色：root

## 6.6.2 查询用户

查询当前库下的所有用户：

show users

## 6.6.3 删除用户

语法格式：

db.dropUser("用户名")

例子：

删除root1用户

db.dropUser("root1")

## 6.6.4 修改用户

语法格式：

```
db.updateUser(
  "<username>",
  {
    customData : { <any information> },
    roles : [
              { role: "<role>", db: "<database>" } | "<role>",
              ...
            ],
    pwd: "<cleartext password>"
    },
    writeConcern: { <write concern> })
```

例子：

修改root用户的角色为readWriteAnyDatabase

```
use admin
db.updateUser("root",{roles:[{role:"readWriteAnyDatabase",db:"admin"}]})
```

## 6.6.5 修改密码

语法格式：

```
db.changeUserPassword("username","newPasswd")
```

例子：

修改root用户的密码为123

```
use admin
db.changeUserPassword("root","123")
```