

MongoDB数据库

介绍

- 1. MongoDB 是一个基于分布式文件存储的数据库，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案
- 2. MongoDB 是一个NoSQL类型的非关系型数据库
- 3. MongoDB将数据存储为文档格式，数据结构由键值对组成
- 4. 数据存储使用类似于JSON格式的BSON数据格式，查询方便
- 5. MongoDB允许数据存储使用非固定模式，易扩展
- 6. MongoDB数据库是开源免费的

下载及安装

[MongoDB官网](#)

选择版本

目前官网最新版本为：3.4.4

傻瓜式安装

考虑系统安装就OK！

非关系型MongoDB数据库类比关系型数据库MySQL

数据库名	MongoDB	MySQL
数据库	Database	Database
多条数据	collection(集合)	table(表)
单条数据	document(文档)	row(记录)
数据属性	field(字段)	field(字段)
数据库 school 学校		
集合：teachers 老师数据 stus 学生数据		
文档：在stus存在条文档信息 --- 一条用户信息		
_id	编号	字段名
name	名称	字段名
age	年龄	字段名

添加mongo路径到环境变量中

1. 找到mongodb的安装目录下的bin目录（mongod.exe所在的目录）

查询路径：D:\Program Files\MongoDB\Server\3.2\bin

2. 右键我的电脑-->属性-->高级系统设置-->环境变量

将路径添加到path变量中

创建MongoDB数据库存储路径

默认存储为 c:/data

```
// 创建存储数据的目录 D:/data/db
```

MonggoDB数据库基本操作命令

```
// 启动mongodb服务器，设置数据的存储位置和数据存储引擎
mongod --dbpath=d:/data/db --storageEngine=mmapv1

// 再开启一个新的命令行
mongo

// 选择、创建数据库，当数据库不存在时自动创建
user dbName

// 显示当前数据库系统下的所有数据库
show dbs

// 显示当前所在的数据库
db

// 创建集合
1. db.createCollection(collectionName);
2. 直接插入数据时自动创建该集合(当集合不存在时)
   db.collectionName.insert(data)

// 插入数据
db.collectionName.insert(data)

// 查询数据
db.collectionName.find()

// 格式化、美化查询
db.collectionName.find().pretty()

// 更新数据 -- $set操作符
db.collectionName.update(con,newData)

// 删除数据
db.collectionName.remove(con)
```

MongoDB数据存储允许的值类型

数据类型

1. null
表示空值
2. 布尔值
true 或 false
3. 数值
4. 字符串
5. 日期
存储从新纪元以来经过的毫秒数
`new Date()`
6. 正则表达式
7. 数组
8. 内嵌文档
9. 对象id
`ObjectId()`
对象id是一个12字节的ID，是文档的唯一标识
10. 二进制数据
11. 代码

MongoDB数据库的增删改查命令

增加

```
db.collectionName.insert() 单条插入
db.collectionName.insert(数组)
```

删除

```
// 切换到对应的数据库
db.dropDatabase()

// 删除集合
db.collectionName.drop()

// 清空集合数据
db.collection.remove({})
```

修改

```

// 默认在更新时，选择一条匹配的数据进行更新
db.collectionName.update(con,newData,{upsert:false},{multi:false})

// $set操作符，修改指定的字段
db.collectionName.update(con,{ $set:{field:value}})

// $inc操作符，数据的增加与减少
db.collectionName.update(con,{ $inc:{field:value}})

// 数组修改器
    $push操作符 向数组中压入一个单元
    db.collectionName.update(con,{ $push:{field:value}})

    // $push和$each结合操作
    db.users.update({uname:'小白龙'}, { $push:{book:{ $each:['西游记','水浒传']}}}),

    // $addToSet 数据已经存在不再进行数据添加

    // 删除元素
    $pop 在数组开头或者末尾删除一个
    $pull 根据条件删除

// $rename字段重命名

```

查询

```

// 查询所有的数据
db.collection.find()

// 在查询时返回指定的键
find({}, {uname:1})          // 1 表示保留
find({}, {uname:0})          // 0 表示删除

// 查询一条
db.collection.findOne()

// 使查询的数据显示更加美观
.pretty()

```

查询条件

```

// $and 与
{age:1,sex:2}
{$and:[{age:20},{sex:2}]}

// $or 或
{$or:[{age:20},{sex:1}]}

// $in 判断值是否在某个范围内
find({uname:{$in:['玉帝','王母']}})

// $nin 值不再某个范围

// $nor 排除，满足一个条件就会被剔除

// $mod 求余
{"键": {"$mod": [num1, num2]}}

// $not 取反

// 范围查询
$lt      <
$lte     <=
$gt      >
$gte     >=
$eq      ==
$ne      !=

// 特殊查询
判断值是否为: null
$exists 操作符表示存在
db.users.find({x:null,x:{$exists:true}}).pretty()
db.stu.find({$and:[{age:null},{age:{$exists:true}}]})

// 使用正则表达式进行查询
db.users.find({class:/王/})

// 查询数组的值

// 查询该值中包含某个值
db.users.find({book:'论持久战'}).pretty()
$all
db.users.find({book:{$all:['论持久战','雅典娜']}})
精确匹配
db.users.find({'book.1':'论持久战'}).pretty()
查询指定长度
db.users.find({book:{$size:3}}).pretty()

```

排序查询

| `sort()` 排序 1 正序 -1 倒序

限制查询

| `limit()` 限制条数

跳过查询

| `skip()` 跳过条数