

赫赫小虾的专栏

转 mgo使用指南

2016年12月29日 14:32:43

mgo简介

mgo（音mango）是MongoDB的Go语言驱动，它用基于Go语法的简单API实现了丰富的特性，并经过良好测试。

官方网站：<http://labix.org/mgo>

[golang.tc \(golangtc.com\)](http://golangtc.com) 网站存储就是用的MongoDB+mgo。近一年使用下来表现良好。

API文档

下面是mgo、mgo/bson、mgo/txn的在线文档。

- [mgo GoDoc GoWalker](#)
- [mgo/bson GoDoc GoWalker](#)
- [mgo/txn GoDoc GoWalker](#)

安装

安装mgo之前请先安装Golang和MongoDB，安装过程不再赘述。

安装bzd版本工具（mgo使用Bazaar作为版本控制系统，因安装的时候需要去Bazaar拉取代码）。

安装命令

```
go get gopkg.in/mgo.v2
```

示例

下面的代码是个示例。

```
package main
import (
    "fmt"
    "labix.org/v2/mgo"
    "labix.org/v2/mgo/bson"
)
type Person struct {
    Name string
    Phone string
}

func main()
{
    session, err := mgo.Dial("")
    if err != nil { panic(err) }
    defer session.Close()

    session.SetMode(mgo.Monotonic, true)
    c := session.DB("test").C("people")
    err = c.Insert(&Person{"Ale", "111111"}, &Person{"Cla", "22222222"})
    if err != nil { panic(err) }
    result := Person{}
    err = c.Find(bson.M{"name": "Ale"}).One(&result) if err != nil { panic(err) }
    fmt.Println("Phone:", result.Phone)
}
```

如何使用

下面介绍如何使用mgo，主要介绍连接数据库，用户等操作，请自行查看文档。

第一步当然是先导入mgo包

```
import (
    "gopkg.in/mgo.v2"
    "gopkg.in/mgo.v2/bson"
)
```

连接服务器

通过方法Dial()来和MongoDB服务器建立连接。Dial()定义如下：

```
func Dial(url string) (*Session, error)
```

具体使用：

```
1 session, err := mgo.Dial(url)
2
```

如果是本机，并且MongoDB是默认端口27017启动的话，下面几种方式都可以。

```
1 session, err := mgo.Dial("")
2 session, err := mgo.Dial("localhost")
3 session, err := mgo.Dial("127.0.0.1")
4 session, err := mgo.Dial("localhost:27017")
5 session, err := mgo.Dial("127.0.0.1:27017")
6
```

如果不在本机或端口不同，传入相应的地址即可。如：

```
mongodb://myuser:mypass@localhost:40001,otherhost:40001/mydb
```

切换数据库

通过Session.DB()来切换相应的数据库。

```
func (s *Session) DB(name string) *Database
```

如切换到test数据库。

```
1 db := session.DB("test")
2
```

切换集合

通过Database.C()方法切换集合（Collection），这样我们就可以通过对集合进行增删查改操作了。

```
func (db *Database) C(name string) *Collection
```

如切换到`users`集合。

```
1 c := db.C("users")
2
```

对集合进行操作

这里我们定义一个struct，用来和集合对应。

```
type User struct {
    Id_ bson.ObjectId `bson:"_id"`
    Name string `bson:"name"`
    Age int `bson:"age"`
    JonedAt time.Time `bson:"joined_at"`
    Interests []string `bson:"interests"`
}
```

注解

注意User的字段首字母大写，不然会报错。通过bson:" name" 这种方式可以定义MongoDB中集合的字段名，如果不定义，mgo自动把struct的字段名首字母小写作为字段名。获得id_，Id_可以不用定义，在插入的时候会自动生成。

插入

插入方法定义如下：

```
func (c *Collection) Insert(inserts ...interface{}) error
```

下面代码插入两条集合数据。

```
1 err = c.Insert(&User{ Id_: bson.NewObjectId(),
2     Name: "Jimmy Kuu",
3     Age: 33,
4     JoinedAt: time.Now(),
5     Interests: []string{"Develop", "Movie"}, }
6 )
7 if err != nil {
8     panic(err)
9 }
10 err = c.Insert(&User{ Id_: bson.NewObjectId(),
11     Name: "Tracy Yu",
12     Age: 31,
13     JoinedAt: time.Now(),
14     Interests: []string{"Shoping", "TV"}, }
15 )
16 if err != nil {
17     panic(err)
18 }
```

这里通过bson.NewObjectId()来创建新的ObjectId，如果创建完需要用到他的话，放在一个变量中即可，一般在Web开发中可以作为参数跳转到其他页面。

通过MongoDB客户端可以发现，两条即可已经插入。

```
{ "_id" : ObjectId( "5204af979955496907000001" ),
  "name" : "Jimmy Kuu",
  "age" : 33,
  "joined_at" : Date( 1376038807950 ),
  "interests" : [ "Develop", "Movie" ]
}
{ "_id" : ObjectId( "5204af979955496907000002" ),
  "name" : "Tracy Yu",
  "age" : 31,
  "joined_at" : Date( 1376038807971 ),
  "interests" : [ "Shoping", "TV" ]
}
```

查询

通过func (c *Collection) Find(query interface{}) *Query来进行查询，返回的Query struct可以有附加各种条件来进行过滤。

通过Query.All()可以获得所有结果，通过Query.One()可以获得一个结果，注意如果没有数据或者数量超过一个，One()会报错。

条件用bson.M{key: value}，注意key必须用MongoDB中的字段名，而不是struct的字段名。

查询所有

```
var users []User c.Find(nil).All(&users) fmt.Println(users)
```

上面代码可以把所有Users都查出:

```
[{ObjectIdHex("5204af9799554969697000001") Jimmy Kuu 33 2013-08-09 17:00:07.95 +0800 CST [Develop Movie]} {ObjectIdHex("5204af9799554969697000001") Jimmy Kuu 33 2013-08-09 17:00:07.971 +0800 CST [Shoping TV]}]
```

根据ObjectId查询

```
1 id := "5204af9799554969697000001" objectId := bson.ObjectIdHex(id) user := new(User) c.Find(bson.M{"_id": objectId}).One(&user) fmt.Println(*user)
2
```

结果如下:

```
&{ObjectIdHex("5204af9799554969697000001") Jimmy Kuu 33 2013-08-09 17:00:07.95 +0800 CST [Develop Movie]}
```

更简单的方式是直接FindId()方法:

```
c.FindId(objectId).One(&user)
```

注解

注意这里没有处理err。当找不到的时候用One()方法会出错。

单条件查询

=(\$eq)

```
c.Find(bson.M{"name": "Jimmy Kuu"}).All(&users)
```

!= (\$ne)

```
c.Find(bson.M{"name": bson.M{"$ne": "Jimmy Kuu"}}).All(&users)
```

> (\$gt)

```
c.Find(bson.M{"age": bson.M{"$gt": 32}}).All(&users)
```

< (\$lt)

```
c.Find(bson.M{"age": bson.M{"$lt": 32}}).All(&users)
```

>= (\$gte)

```
c.Find(bson.M{"age": bson.M{"$gte": 33}}).All(&users)
```

<= (\$lte)

```
c.Find(bson.M{"age": bson.M{"$lte": 31}}).All(&users)
```

in (\$in)

```
c.Find(bson.M{"name": bson.M{"$in": []string{"Jimmy Kuu", "Tracy Yu"}}).All(&users)
```

多条件查询

and (\$and)

```
c.Find(bson.M{"name": "Jimmy Kuu", "age": 33}).All(&users)
```

```
c.Find(bson.M{"$or": []bson.M{bson.M{"name": "Jimmy Kuu"}, bson.M{"age": 31}}}).All(&users)
```

修改

通过func (*Collection) Update来修改操作。

```
func (c *Collection) Update(selector interface{}, change interface{}) error
```

注意修改单个或多个字段需要通过update操作符号，否则集合会被替换。

修改字段的值(\$set)

```
c.Update(bson.M{"_id": bson.ObjectIdHex("5204af979955496907000001")}, bson.M{"$set": bson.M{ "name": "Jimmy Gu", "age": 34, }})
```

inc(\$inc)

字段增加值

```
c.Update(bson.M{"_id": bson.ObjectIdHex("5204af979955496907000001")}, bson.M{"$inc": bson.M{ "age": -1, }})
```

push(\$push)

从数组中增加一个元素

```
c.Update(bson.M{"_id": bson.ObjectIdHex("5204af979955496907000001")}, bson.M{"$push": bson.M{ "interests": "Golang", }})
```

pull(\$pull)

从数组中删除一个元素

```
c.Update(bson.M{"_id": bson.ObjectIdHex("5204af979955496907000001")}, bson.M{"$pull": bson.M{ "interests": "Golang", }})
```

删除

```
c.Remove(bson.M{"name": "Jimmy Kuu"})
```

注解

这里也支持多条件，参考多条件查询。

文章标签: [ubuntu](#) [mgo](#)

个人分类: [Golang](#)

相关热词: [mgo](#) [mgo正则表达式](#) [mgo命令](#) [mgo排序](#) [mgo文档](#)

上一篇 [Golang开发环境搭建-Vim篇](#)

下一篇 [通过SublimeCodeIntel设置JavaScript自动补全](#)

golang mgo 查询和更新内嵌文档

在mongoDB的shell中查找并显示特定的键值对的语法是: > db.logs.find({},{"log":1}) {"_id": "3a06384a40a8e587806f194c6...

想对作者说点什么? [我来说一句](#)

Golang MongoDB 驱动 mgo 的若干问题

mgo 是 MongoDB 的 Golang 驱动。连接池 我们通过 Dial 函数创建一个新的 session: session, err := mgo.Dial(url) ...

Go语言mgo使用情况

本文重点介绍mgo使用,仅简单介绍mongodb。mongodb特性mongodb简单介绍注意: 上图已经告知我们mongo不支持事务，在开发项...

mgo(音mango)是MongoDB的Go语言驱动,它用基于Go语法的简单API实现了丰富的特性...func (c *Collection) Update(selector interface...

MongoDB的Go语言驱动-mgo的使用指南 - CSDN博客

mgo(音mango)是MongoDB的Go语言驱动,它用基于Go语法的简单API实现了丰富的特性...func (c *Collection) Update(selector interface...

写评论

Golang+Mgo 实现对于MongoDB的正则表达式查找

1.xxdb.go程序内容如下 package main import ("fmt" "gopkg.in/mgo.v2" "gopkg.in/mgo.v2/bson")

收藏

三、go语言操作 mongodb mgo --go语言学习笔记 - CSDN博客

mgo(音mango)是MongoDB的Go语言驱动,它用基于Go语法的简单API实现了丰富的特性...func (c *Collection) Update(selector interface...

微信

三、go语言操作 mongodb mgo --go语言学习笔记 - alvine008的专栏...

mgo(音mango)是MongoDB的Go语言驱动,它用基于Go语法的简单API实现了丰富的特性...func (c *Collection) Update(selector interface...

微博

QQ

go 使用mgo驱动示例

package model import (// "fmt" "errors" "infrastructure/collnamegenerator" "infrastructure/log..."

golang mgo 使用

golang使用mongodb，目前比较多人用的是mgo（pronounced as mango）github地址 官网 文档直接上官方example的代码Examp...

golang mgo的mongo连接池设置：必须手动加上maxPoolSize

golang的mgo库说明里是说明了开启连接复用的，但观察实验发现，这并没有根本实现连接的控制，连接复用仅在有空闲连接时生效...

mgo中DBRef的使用方法

type User struct { Name string Phone string Fkey mgo.DBRef } user:={Name:"abc",Phone:"1234"} us...

golang 使用mgo 连接MongoDB

golang访问MongoDB数据库。

Go实战--golang中使用MongoDB(mgo)

生命不止，继续 go go go !!! 昨天分享了golang如何操作Redis数据库，那今天就介绍一下golang中如何使用mongodb数据库。何为...

Golang mgo 剖析之 Session

golang 操作 mongodb 使用的包是“gopkg.in/mgo.v2”，coding 过程中需要并发读写 mongodb 数据库，简单观摩了下源码，记录下...

golang mongodb驱动mgo异常崩溃问题

##背景在我的后端游戏架构中，有一个golang开发的数据服务器，用来存储落地玩家的历史数据，并提供给客户端http接口查询。由...

三、go语言操作 mongodb mgo --go语言学习笔记

mgo使用指南mgo简介 mgo（音mango）是MongoDB的Go语言驱动，它用基于Go语法的简单API实现了丰富的特性，并经过良好测...

MongoDB的Go语言驱动----mgo的使用指南

原文地址：https://my.oschina.net/ffs/blog/300148 mgo简介 mgo（音mango）是MongoDB的Go语言驱动，它用基于Go语法的简单...

mgo 的 session 与连接池

简介 mgo是由Golang编写的开源mongodb驱动，由于mongodb官方并没有开发Golang驱动，因此该驱动被广泛使用。mgo...

Go语言下与MongoDB通信-使用mgo

最近写游戏服.用的全是新货.go语言下与mongodb.嘿嘿.关于mgo如何使用就不详细说了.官网上写的很清楚.这块就说说怎么用.首先导...

mgo EnsureIndex注意事项

问题描述：开发过程中，手动drop一个collection，由于程序在写入该collection前做了EnsureIndex，等写入了文档之后，在console...

MonogDB - 索引 (三) G...

MongoDB 2d Index 和 2sphere Index 索引的使用。

Go 在 MongoDB 中对数组进行查询

MongoDB中根据数组子元素进行查询的两种方式。使用 "[数组名].[子元素字段名]" 的方式进行匹配。使用 "[数组名]" \$elemMatch {...

mgo 实现mongodb中的\$addToSet和\$push/Count方法

mgo使用pipe来实现mongodb中的aggregation,所以在实现mongodb中的一些方法的时候就需要我们自己去拼接，现在使用go给大家...

在golang中使用mgo多条件查询

今天被mgo凶残的语法折腾跪了 磨了1个多小时才搞出来 特此纪念: 一般做简单查询，是这样写的: collection := mgodbcontroller.Get...

mgo简介以及使用说明

简介：mgo是Go语言操作monggodb数据库的一个驱动，封装了基于Go语法的API官网地址：http://labix.org/mgo文档地址：mgo G...

mgo

选择列 https://stackoverflow.com/questions/31116528/select-column-from-mongodb-in-golang-using-mgo 2....

关于谷歌地图获取定位然后mongodb实现矩形框、距离远近排序、mgo驱动。

一、谷歌地图不能正常显示： 情形①、谷歌地图的div原本设置成隐藏的，在点击的时候才出现。隐藏使用的方案：display：none...

mgo使用指南-高级篇(长连接/重连)

Keeping an mgo (any other db engine) connection alive I have yet to write a service that would k...

golang-gin-mgo高并发服务器搭建

gin-mgo服务器搭建 该服务器实现简单接收请求并将请求参数封装存储在mongodb数据库中,本文将讲述gin - mgo的使用方法。项目...

go mongodb 模糊查询

根据某一个字段的关键字，查询mgo中的信息 func GetOrdersByCity(city string)([]ParkingOrderModel,error){ query:=bson...

golang mongodb upsert

//UpsertPhoneInterface func UpsertPhoneInterface(_query, update bson.M) (err error) { _query :...

mgo 连接需要授权登录的mongodb

golang mgo authentication

什么场景应该用 MongoDB ?

很多人比较关心 MongoDB 的适用场景，也有用户在话题里分享了自己的业务场景，比如 案例1 用在应用服务器的日志记录，查找...

先说一句题外话，自己的亲身感受，看别人再多的博客，也不如自己去查看官方文档，自己就是看了网上的博客，感觉似懂非懂，就...

基于mgo写的mgo小工具

2018年02月11日 5KB 下载 写评论

mgo 聚类统计

// GroupCountByIssueType return 目录 ount list func (mgr *issueMgr) GroupCountByIssueType(field st...

JSON和BSON的区别

BSON是由10gen开发的一个数据 目前主要用于MongoDB中，是MongoDB的数据存储格式。BSON基于JSON格式，选择JSO... 微信

MongoDB - 索引 （二）使

MongoDB数据库中的索引相关操 本使用命令。 QQ

【寒江雪】golang-mgo封装的数据映射器

github:https://github.com/HJXTSTU/mgorm 写完后觉得有点多余，但是用着还是蛮舒服的，在启动的时候初始化好。获得一个全局s...

linux(ubuntu)下go(golang)操作mongodb数据库

一、安装git\$ sudo add-apt-repository ppa:git-core/ppa\$ sudo apt-get update\$ sudo apt-get install git 二、安...

Golang 中操作 Mongo Update 的方法

这篇博客主要记录下 Golang 中引入 Mongo 数据库对 Mongo 的修改操作的方法。使用gopkg.in/mgo.v2库操作，修改操作主要使用...

mgo 遇到的问题以及解决方案集合

1. main redeclared in this block previous declaration at ./test03.go:13 解决方案：在分别建立新文件夹，把文件放进去，重...

第9章 完善之个人中心、密码找回功能

小站：http://oideas.herokuapp.com 教程演示地址：http://omind.herokuapp.com 教程代码地址：https://github.com/jov...

mgo的小技巧

使用mgo，下面是mgo官网的一段代码。细看之下发现一个问题，Person中的变量定义是大写的"Name"和"Phone"，但是后边Find...

MongoDB 自动分片 (二) (Auto-Sharding)

深入浅出得详细介绍了MongoDB分片得原理，深入介绍了MongoDB片键得选取原则

golang mongoDB GridFS查询 存储 删除文件

先讲一下GridFS的作用 GridFS 用于存储和恢复那些超过16M（BSON文件限制）的文件(如：图片、音频、视频等)，如果没有超过...

CDC使用指南，CDC使用指南

2011年07月12日 38KB 下载

mog使用指南

mog使用指南 mgo简介 mgo（音mango）是MongoDB的Go语言驱动，它用基于Go语法的简单API实现了丰富的特性，并经过良好...

iTunes使用指南

jboss学习资料/JBOSS使用指南

2011年06月27日 706KB 1

Beego 连接 MongoDB主从

MongoDB一主两从，Beego框架里写评论方法。

ANT使用指南ANT使用指南

2011年05月08日 407KB 收藏

运算放大器使用指南，包含运放单电源使用

2010年03月09日 3.11MB 微博

QQ

个人资料



hehexiaoxia

原创 28 粉丝 13

等级： 博客 4 访问
积分： 1460 排名
勋章： 恒

最新文章

MAC 常用软件汇总
mysql基础命令汇总
Sublime Text3通过HTML-
自动格式化代码
简单的Http服务器监听和管
vs2010 编译mosquito wir
bsockets，实现消息推送

个人分类

Golang
环境搭建
C++
C#
mongodb

展开

归档

2017年10月
2017年8月
2017年7月
2017年6月
2017年5月

展开

1

写评论

目录

收藏

微信

微博

QQ

- 利用Access-Control-Allow
决跨域请求原理
阅读量：63214
- 通过SublimeCodeIntel设置
补全
阅读量：24320
- Mysql引起索引失效的原因
阅读量：12784
- WPF中WrapPanel、Stack
动条ScrollViewer
阅读量：9685
- MySQL数据库order by 主
解决方案
阅读量：7457

最新评论

- ubuntu14.04中离线安装d...
qq_25093037：你好，为什么
安装不成功呢
- 通过SublimeCodeInte...
xiaofeifeifeihaha：为什么我每
载别的插件可以
- 利用Access-Control-...
Aloor：感谢看明白了，三克油
- 在Docker上部署mongodb
caofeiliju：您确定，\$ sudo do
mongo:latest ~...
- 通过SublimeCodeInte...
Huimin_520：没提示的可能要