 作为系列的最后一篇，得要说说C#驱动对mongodb的操作，目前驱动有两种：官方驱动和samus驱动，不过我个人还是喜欢后者，

因为提供了丰富的linq操作，相当方便。

官方驱动：[https://github.com/mongodb/mongo-csharp-driver/downloads](https://github.com/mongodb/mongo-csharp-driver/downloads" \t "_blank)。下载后，还提供了一个酷似msdn的帮助文档。

samus驱动：[https://github.com/samus/mongodb-csharp/downloads](https://github.com/samus/mongodb-csharp/downloads" \t "_blank)。

下面就具体看看samus驱动，[https://github.com/samus/mongodb-csharp/blob/master/examples/Simple/Main.cs](https://github.com/samus/mongodb-csharp/blob/master/examples/Simple/Main.cs" \t "_blank)上面提供了

一个简单的demo，大体上看看我们就知道怎么玩了。

一： 实践

1：我们建立一个Person实体，MongoAlias特性表示取别名，这里的ID值将会覆盖掉数据库自动生成的\_id。

复制代码

1 #region 数据实体  
2 /// <summary>  
3 /// 数据实体  
4 /// </summary>  
5 public class Person  
6     {  
7         [MongoAlias("\_id")]  
8 public string ID { get; set; }  
9   
10 public string Name { get; set; }  
11   
12 public int Age { get; set; }  
13   
14 public DateTime CreateTime { get; set; }  
15     }  
16 #endregion

复制代码

2：初始化一些变量

复制代码

1 string connectionString = string.Empty;  
2   
3 string databaseName = string.Empty;  
4   
5 string collectionName = string.Empty;  
6   
7 static MongodbHelper<T> mongodb;  
8   
9 #region 初始化操作  
10 /// <summary>  
11 /// 初始化操作  
12 /// </summary>  
13 public MongodbHelper()  
14         {  
15             connectionString = "Server=127.0.0.1:2222";  
16             databaseName = "shopex";  
17             collectionName = "person";  
18         }  
19 #endregion

复制代码

3：为了方便T的继承类使用linq功能，我们还需要映射一下。

复制代码

1 #region 实现linq查询的映射配置  
2 /// <summary>  
3 /// 实现linq查询的映射配置  
4 /// </summary>  
5 public MongoConfiguration configuration  
6         {  
7 get  
8             {  
9 var config = new MongoConfigurationBuilder();  
10   
11                 config.Mapping(mapping =>  
12                 {  
13                     mapping.DefaultProfile(profile =>  
14                     {  
15                         profile.SubClassesAre(t => t.IsSubclassOf(typeof(T)));  
16                     });  
17                     mapping.Map<T>();  
18                     mapping.Map<T>();  
19                 });  
20   
21                 config.ConnectionString(connectionString);  
22   
23 return config.BuildConfiguration();  
24             }  
25         }  
26 #endregion

复制代码

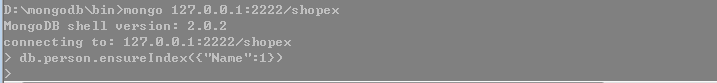
4：下面是一些基本的CURD的代码，跟写EF代码很类似，写起来好舒服。

复制代码

  1 #region 插入操作  
  2 /// <summary>  
  3 /// 插入操作  
  4 /// </summary>  
  5 /// <param name="person"></param>  
  6 /// <returns></returns>  
  7 public void Insert(T t)  
  8         {  
  9 using (Mongo mongo = new Mongo(configuration))  
10             {  
11 try  
12                 {  
13                     mongo.Connect();  
14   
15 var db = mongo.GetDatabase(databaseName);  
16   
17 var collection = db.GetCollection<T>(collectionName);  
18   
19                     collection.Insert(t, true);  
20   
21                     mongo.Disconnect();  
22   
23                 }  
24 catch (Exception)  
25                 {  
26                     mongo.Disconnect();  
27 throw;  
28                 }  
29             }  
30         }  
31 #endregion  
32   
33 #region 更新操作  
34 /// <summary>  
35 /// 更新操作  
36 /// </summary>  
37 /// <param name="person"></param>  
38 /// <returns></returns>  
39 public void Update(T t, Expression<Func<T, bool>> func)  
40         {  
41 using (Mongo mongo = new Mongo(configuration))  
42             {  
43 try  
44                 {  
45                     mongo.Connect();  
46   
47 var db = mongo.GetDatabase(databaseName);  
48   
49 var collection = db.GetCollection<T>(collectionName);  
50   
51                     collection.Update<T>(t, func, true);  
52   
53                     mongo.Disconnect();  
54   
55                 }  
56 catch (Exception)  
57                 {  
58                     mongo.Disconnect();  
59 throw;  
60                 }  
61             }  
62         }  
63 #endregion  
64   
65 #region 获取集合  
66 /// <summary>  
67 ///获取集合  
68 /// </summary>  
69 /// <param name="person"></param>  
70 /// <returns></returns>  
71 public List<T> List(int pageIndex, int pageSize, Expression<Func<T,bool>> func, out int pageCount)  
72         {  
73             pageCount = 0;  
74   
75 using (Mongo mongo = new Mongo(configuration))  
76             {  
77 try  
78                 {  
79                     mongo.Connect();  
80   
81 var db = mongo.GetDatabase(databaseName);  
82   
83 var collection = db.GetCollection<T>(collectionName);  
84   
85                     pageCount = Convert.ToInt32(collection.Count());  
86   
87 var personList = collection.Linq().Where(func).Skip(pageSize \* (pageIndex - 1))  
88                                                    .Take(pageSize).Select(i => i).ToList();  
89   
90                     mongo.Disconnect();  
91   
92 return personList;  
93   
94                 }  
95 catch (Exception)  
96                 {  
97                     mongo.Disconnect();  
98 throw;  
99                 }  
100             }  
101         }  
102 #endregion  
103   
104 #region 读取单条记录  
105 /// <summary>  
106 ///读取单条记录  
107 /// </summary>  
108 /// <param name="person"></param>  
109 /// <returns></returns>  
110 public T Single(Expression<Func<T, bool>> func)  
111         {  
112 using (Mongo mongo = new Mongo(configuration))  
113             {  
114 try  
115                 {  
116                     mongo.Connect();  
117   
118 var db = mongo.GetDatabase(databaseName);  
119   
120 var collection = db.GetCollection<T>(collectionName);  
121   
122 var single = collection.Linq().FirstOrDefault(func);  
123   
124                     mongo.Disconnect();  
125   
126 return single;  
127   
128                 }  
129 catch (Exception)  
130                 {  
131                     mongo.Disconnect();  
132 throw;  
133                 }  
134             }  
135         }  
136 #endregion  
137   
138 #region 删除操作  
139 /// <summary>  
140 /// 删除操作  
141 /// </summary>  
142 /// <param name="person"></param>  
143 /// <returns></returns>  
144 public void Delete(Expression<Func<T, bool>> func)  
145         {  
146 using (Mongo mongo = new Mongo(configuration))  
147             {  
148 try  
149                 {  
150                     mongo.Connect();  
151   
152 var db = mongo.GetDatabase(databaseName);  
153   
154 var collection = db.GetCollection<T>(collectionName);  
155   
156 //这个地方要注意，一定要加上T参数，否则会当作object类型处理  
157 //导致删除失败  
158                     collection.Remove<T>(func);  
159   
160                     mongo.Disconnect();  
161   
162                 }  
163 catch (Exception)  
164                 {  
165                     mongo.Disconnect();  
166 throw;  
167                 }  
168             }  
169         }  
170 #endregion

复制代码

5.   好，我们开一下2222端口，由于前前篇我已经把这个mongodb做成了服务，现在就直接连过去了，并做一下对Name的索引。



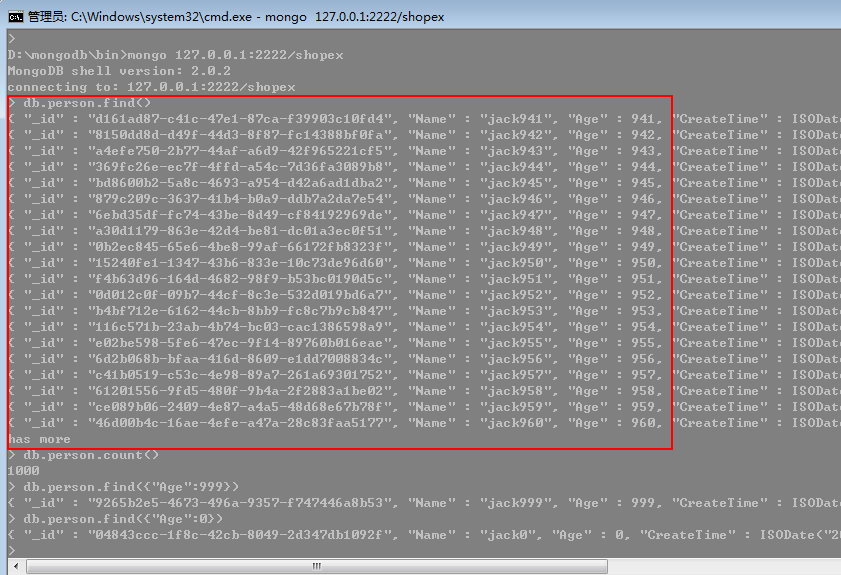
6. 一切准备妥当，我们做下基本的操作，比如这里我添加一千条数据，注意我开启的是安全模式，如果插入不成功，将会抛出异常。

 <1> Add:

复制代码

1 static void Main(string[] args)  
2         {  
3             MongodbHelper<Person> helper = new MongodbHelper<Person>();  
4   
5 //插入1000条数据  
6 for (int i = 0; i < 1000; i++)  
7             {  
8                 helper.Insert(new Person()  
9                 {  
10                     ID = Guid.NewGuid().ToString(),  
11                     Name = "jack" + i,  
12                     Age = i,  
13                     CreateTime = DateTime.Now  
14                 });  
15             }  
16   
17             Console.WriteLine("插入成功");  
18   
19             Console.Read();  
20         }

复制代码



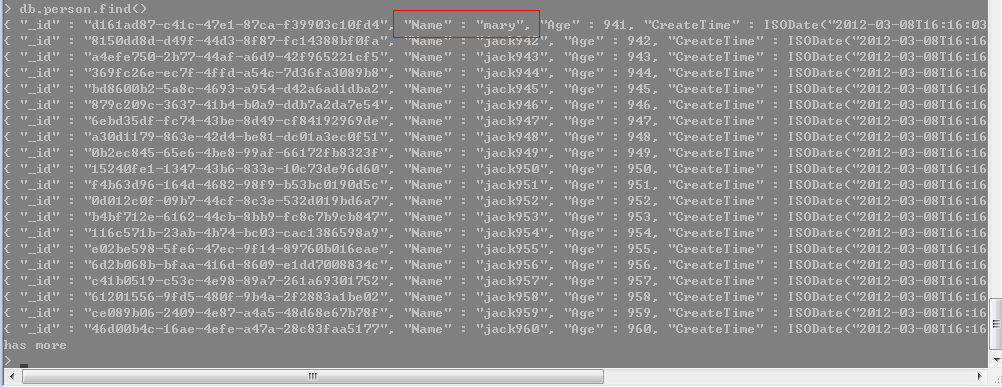
乍一看显示的数据以为有问题，为什么没有出现jack0或者jack999，不过find的一下后心情舒坦了。

<2> update:   这里就把jack941的名字改掉“mary”

复制代码

1 static void Main(string[] args)  
2         {  
3             MongodbHelper<Person> helper = new MongodbHelper<Person>();  
4   
5 //修改jack941改成mary  
6 var single = helper.Single(i => i.Name == "jack941");  
7             single.Name = "mary";  
8             helper.Update(single, i => i.ID == single.ID);  
9   
10             Console.WriteLine("修改成功");  
11             Console.Read();  
12         }

复制代码

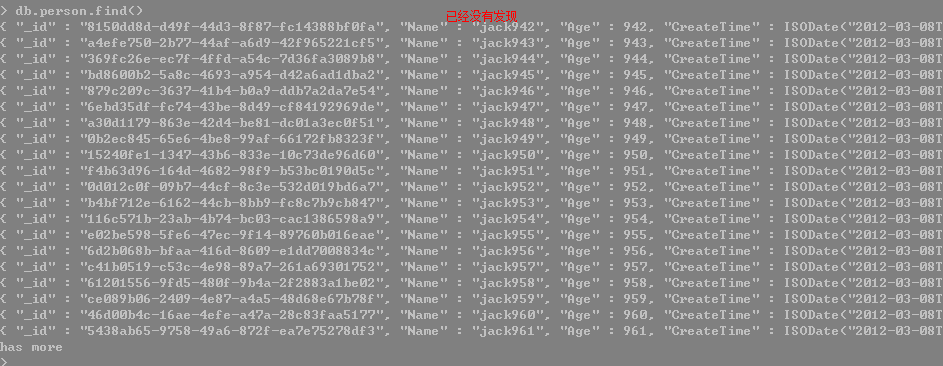


<3>Delete：  删除mary这条记录

复制代码

1 static void Main(string[] args)  
2         {  
3             MongodbHelper<Person> helper = new MongodbHelper<Person>();  
4   
5 //删除mary这个记录  
6             helper.Delete(i => i.Name == "mary");  
7   
8             Console.WriteLine("删除成功");  
9             Console.Read();  
10         }

复制代码

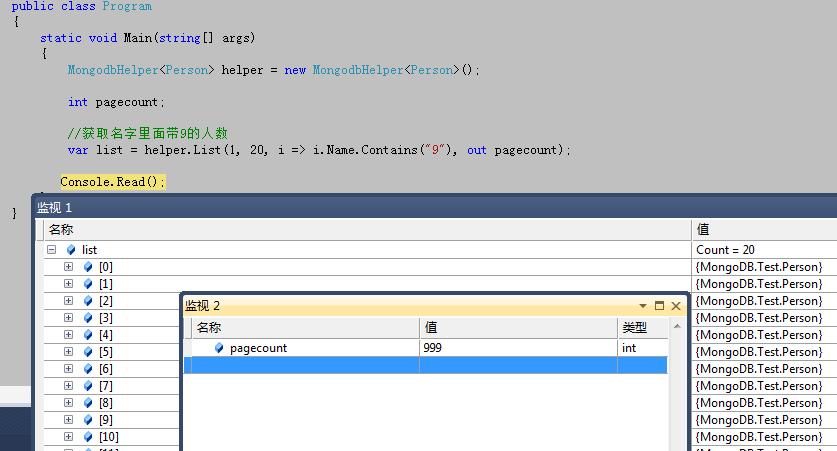


<4> list操作： 这里我获取一下名字里面带9的人数列表

复制代码

1 static void Main(string[] args)  
2         {  
3             MongodbHelper<Person> helper = new MongodbHelper<Person>();  
4   
5 int pagecount;  
6   
7 //获取名字里面带9的人数  
8 var list = helper.List(1, 20, i => i.Name.Contains("9"), outpagecount);  
9   
10            Console.Read();  
11         }

复制代码



总的运行代码

C:\Users\wyz\Documents\My Knowledge\temp\8e938dba-b45a-4d4c-8299-17c6cc218a0a_4_files\dc4b210a-d516-484b-ae5f-274466304561.pngView Code

wow，趁着3天的休假，不断的努力终于把这个系列写完了，很感谢一直关注此系列的朋友。C:\Users\wyz\Documents\My Knowledge\temp\8e938dba-b45a-4d4c-8299-17c6cc218a0a_4_files\741ca3d4-f66d-4e68-97d9-c7bd37d43994.jpg

来源： <<http://blog.csdn.net/tengbaichuan/article/details/11120767>>