首先要确保mongodb的正确安装，安装参照：[http://docs.mongodb.org/manual/tutorial/install-mongodb-on-debian-or-ubuntu-linux](http://docs.mongodb.org/manual/tutorial/install-mongodb-on-debian-or-ubuntu-linux/)

然后下载nodejs的mongodb的driver

npm install mongodb

编写一个测试的程序：

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 1 var mongodb = require('mongodb');  
 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7         console.log('connect');  
 8     }else{  
 9         console.log(err);  
10     }     
11   
12 });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

如果最终显示connect则说明成功。

对mongodb的collection的操作

有两种方法链接collection，分别为：

db.collection('mycoll',function(err,coll){});

db.createCollection('mycoll',function(err,coll){});

这两种方法还有第二个可选参数{safe:true},这个参数的作用对于第一种方法，如果加上了这个参数，那么当collection不存在的时候则报错，对于第二种方法，则当collection存在的时候报错

示例：

  1 var mongodb = require('mongodb');

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7       console.log('connect');  
 8       db.collection('mycoll',{safe:true},function(err,collection){  
 9           if(err){  
10               console.log(err);  
11           }     
12       });   
13   
14     }else{  
15         console.log(err);  
16     }     
17   
18 });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 结果如图所示：

http://images.cnblogs.com/cnblogs_com/hitandrew/node1.png

示例：

  1 var mongodb = require('mongodb');

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7       console.log('connect');  
 8       db.createCollection('mycoll',{safe:true},function(err,collection){  
 9           if(err){  
10               console.log(err);  
11           }     
12       });   
13   
14     }else{  
15         console.log(err);  
16     }     
17   
18 });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 结果如图所示：

http://images.cnblogs.com/cnblogs_com/hitandrew/node2.png

删除collection则使用dropCollection函数即可：

示例：

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 1 var mongodb = require('mongodb');  
 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7       console.log('connect');  
 8       db.dropCollection('mycoll',{safe:true},function(err,result){  
 9          console.log(result);  
10       });   
11   
12     }else{  
13         console.log(err);  
14     }     
15

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

16 });

结果如图所示：

http://images.cnblogs.com/cnblogs_com/hitandrew/node3.png

对collection进行增删改查

向collection添加数据使用insert函数

示例：

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 1 var mongodb = require('mongodb');  
 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7       db.collection('mycoll',{safe:true},function(err,collection){  
 8           var tmp1 = {title:'hello',number:1};  
 9           collection.insert(tmp1,{safe:true},function(err,result){  
10               console.log(result);  
11           });   
12 　　　　});  
13     }else{  
14         console.log(err);  
15     }     
16   
17 });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

结果如图：

http://images.cnblogs.com/cnblogs_com/hitandrew/node4.png

对数据进行更新：

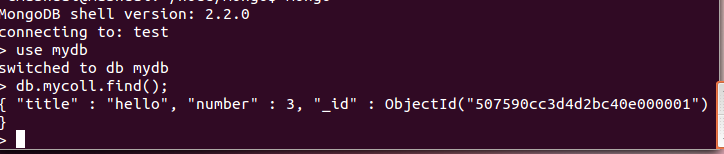
示例：

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 1 var mongodb = require('mongodb');  
 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {  
 7       db.collection('mycoll',{safe:true},function(err,collection){  
 8           collection.update({title:'hello'},{$set:{number:3}},{safe:true},function(err,result){  
 9               console.log(result);  
10           });  
11   
12     }else{  
13         console.log(err);  
14     }  
15   
16 });  
17

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

结果如图所示：



对数据进行删除使用remove函数

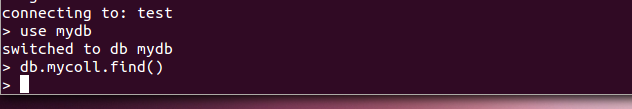
 示例：

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 1 var mongodb = require('mongodb');  
 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {  
 7       db.collection('mycoll',{safe:true},function(err,collection){  
 8           collection.remove({title:'hello'},{safe:true},function(err,result){  
 9               console.log(result);  
10           });  
11           
12     }else{  
13         console.log(err);  
14     }           
15                                   
16 });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

结果如图：



如果remove没有任何的参数，则删除全部。

查找操作，查找操作有两个方法一个是find，一个是findOne

示例：

  1 var mongodb = require('mongodb');

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)

 2 var server = new mongodb.Server('localhost',27017,{auto\_reconnect:true});  
 3 var db = new mongodb.Db('mydb',server,{safe:true});  
 4 db.open(function(err,db){  
 5     if(!err)  
 6     {     
 7       db.collection('mycoll',{safe:true},function(err,collection){  
 8           var tmp1 = {title:'hello'};  
 9           var tmp2 = {title:'world'};  
10           collection.insert([tmp1,tmp2],{safe:true},function(err,result){  
11               console.log(result);  
12           });   
13           collection.find().toArray(function(err,docs){  
14               console.log('find');  
15               console.log(docs);  
16           });   
17           collection.findOne(function(err,doc){  
18               console.log('findOne');  
19               console.log(doc);  
20           });   
21       });

[复制代码](http://blog.csdn.net/tk1905345/article/details/8773943)