官方文档地址

http://mongodb.github.io/node-mongodb-native/

1. **安装MongoDB并实例化**

npm install mongodb --save

**const** MongoClient = require(**'mongodb'**).MongoClient;

1. **连接数据库**

*// 数据库地址(无密码)*

**const** url = **'mongodb://localhost:27017'**;

*// 数据库地址(有密码)*

const url ='mongodb://admin:jack@localhost:27017/dbname?authSource=admin';

*// 数据库名称*

**const** dbName = **dbname**;

*// 连接数据库*

MongoClient.connect(url, **function**(err, client) {

If (err){

Console.log(err);

Return false;

}

console.log(**"数据库连接成功"**);

**const** db = client.db(dbName);

// 执行语句的最后写

client.close();

});

1. **操作**
2. **新增**

**const** collection = db.collection(**'documents'**);

*// 插入一些文档，或插入一条文档 insertOne()*

collection.insertMany([

{a : 1}, {a : 2}, {a : 3}

], **function**(err, result) {

If (err){

Console.log(err);

} else {

Console.log(result);

}

});

1. **修改**

**const** collection = db.collection(**'documents'**);

*//查找键名是a值为2的记录，将其b的值改成1；多条更新使用 updateMany()*

collection.updateOne({a : 2 }

, {$set: { b : 1}}, **function**(err, result) {

If (err){

Console.log(err);

} else {

Console.log(result);

}

});

1. **删除**

**const** collection = db.collection(**'documents'**);

*// 删除键为a值为3的记录*

collection.removeOne({ a : 3 }, **function**(err, result) {

If (err){

Console.log(err);

} else {

Console.log(result);

}

});

1. **查询**
   1. 查找全部

**const** collection = db.collection(**'documents'**);

*// Find some documents*

collection.find({}).toArray(**function**(err, docs) {

If (err){

Console.log(err);

} else {

Console.log(docs);

}

});

4.2 查找条件

**const** collection = db.collection(**'documents'**);

*// Find some documents*

collection.find({**'a'**: 3}).toArray(**function**(err, docs) {

If (err){

Console.log(err);

} else {

Console.log(docs);

}

});

* 1. 更多查询条件

| **操作** | **格式** | **范例** |
| --- | --- | --- |
| 等于 | {<key>:<value>} | db.col.find({"name":"jack"}) |
| 小于 | {<key>:{$lt:<value>}} | db.col.find({"likes":{$lt:50}}) |
| 小于或等于 | {<key>:{$lte:<value>}} | db.col.find({"likes":{$lte:50}}) |
| 大于 | {<key>:{$gt:<value>}} | db.col.find({"likes":{$gt:50}}) |
| 大于或等于 | {<key>:{$gte:<value>}} | db.col.find({"likes":{$gte:50}}) |
| 不等于 | {<key>:{$ne:<value>}} | db.col.find({"likes":{$ne:50}}) |
| 模糊 | {<key>:/<value>/}}  {<key>:/^<value>/}}  {<key>:/<value>$/}} | db.col.find({"likes":/ja/}) |

4.4 排序

collection.find().sort({"name":-1} // -1倒序，1正序

4.5 分页

collection.find().sort({"name":-1}).skip(1).limit(2).toArray((err,res)=>{

If (err) throw err

Console.log(res)

})

4.6 多字段查询

实现 （state1=11 and state2=22）

db.getCollection('testOrAnd').find(

{$and:[{"state1":11},{"state2":22}]}

)

使用or形式实现 value >300

db.getCollection('testOrAnd'). find(

{ $or:[{"value":{$gte:300}}] }

)

将步骤一参数拼接到步骤二or条件

db.getCollection('testOrAnd').

find({$or:

[

{$and:[{"state1":11},{"state2":22}]},{"value":{$gte:300}}

]

})