**Node.js 连接 MongoDB**

MongoDB是一种文档导向数据库管理系统，由C++撰写而成。

本章节我们将为大家介绍如何使用 Node.js 来连接 MongoDB，并对数据库进行操作。

如果你还没有 MongoDB 的基本知识，可以参考我们的教程：[MongoDB 教程](http://www.runoob.com/mongodb/mongodb-tutorial.html)。

**安装驱动**

本教程使用了[淘宝定制的 cnpm 命令](http://www.runoob.com/nodejs/nodejs-npm.html#taobaonpm)进行安装：

$ cnpm install mongodb

接下来我们来实现增删改查功能。

**数据库操作( CURD )**

与 MySQL 不同的是 MongoDB 会自动创建数据库和集合，所以使用前我们不需要手动去创建。

**插入数据**

以下实例我们连接数据库 runoob 的 site 表，并插入两条数据：

**插入数据**

var MongoClient = require('mongodb').MongoClient;

var DB\_CONN\_STR = 'mongodb://localhost:27017/runoob';

# 数据库为 runoob

var insertData = function(db, callback) {

//连接到表 site var collection = db.collection('site');

//插入数据 var data = [{"name":"菜鸟教程","url":"www.runoob.com"},{"name":"菜鸟工具","url":"c.runoob.com"}];

collection.insert(data, function(err, result) { if(err) { console.log('Error:'+ err); return; } callback(result); }); }

MongoClient.connect(DB\_CONN\_STR, function(err, db) { console.log("连接成功！");

insertData(db, function(result) { console.log(result); db.close(); }); });

执行以下命令输出就结果为：

$ node test.js

连接成功！

{ result: { ok: 1, n: 2 },

ops:

[ { name: '菜鸟教程',

url: 'www.runoob.com',

\_id: 58c25e13a08de70d3b9d4116 },

{ name: '菜鸟工具',

url: 'c.runoob.com',

\_id: 58c25e13a08de70d3b9d4117 } ],

insertedCount: 2,

insertedIds: [ 58c25e13a08de70d3b9d4116, 58c25e13a08de70d3b9d4117 ] }

从输出结果来看，数据已插入成功。

我们也可以打开 MongoDB 的客户端查看数据，如：

> show dbs

admin 0.000GB

local 0.000GB

runoob 0.000GB # 自动创建了 runoob 数据库

> show tables

site # 自动创建了 site 集合（数据表）

> db.site.find() # 查看集合中的数据

{ "\_id" : ObjectId("58c25f300cd56e0d7ddfc0c8"), "name" : "菜鸟教程", "url" : "www.runoob.com" }

{ "\_id" : ObjectId("58c25f300cd56e0d7ddfc0c9"), "name" : "菜鸟工具", "url" : "c.runoob.com" }

>

**查询数据**

以下实例检索 name 为 "菜鸟教程" 的实例：

**查询数据**

var MongoClient = require('mongodb').MongoClient; var DB\_CONN\_STR = 'mongodb://localhost:27017/runoob'; var selectData = function(db, callback) { //连接到表 var collection = db.collection('site'); //查询数据 var whereStr = {"name":'菜鸟教程'}; collection.find(whereStr).toArray(function(err, result) { if(err) { console.log('Error:'+ err); return; } callback(result); }); } MongoClient.connect(DB\_CONN\_STR, function(err, db) { console.log("连接成功！"); selectData(db, function(result) { console.log(result); db.close(); }); });

执行以下命令输出就结果为：

连接成功！

[ { \_id: 58c25f300cd56e0d7ddfc0c8,

name: '菜鸟教程',

url: 'www.runoob.com' } ]

**更新数据**

我们也可以对数据库的数据进行修改，以下实例将 name 为 "菜鸟教程" 的 url 改为 https://www.runoob.com：

**更新数据**

var MongoClient = require('mongodb').MongoClient; var DB\_CONN\_STR = 'mongodb://localhost:27017/runoob'; var updateData = function(db, callback) { //连接到表 var collection = db.collection('site'); //更新数据 var whereStr = {"name":'菜鸟教程'}; var updateStr = {$set: { "url" : "https://www.runoob.com" }}; collection.update(whereStr,updateStr, function(err, result) { if(err) { console.log('Error:'+ err); return; } callback(result); }); } MongoClient.connect(DB\_CONN\_STR, function(err, db) { console.log("连接成功！"); updateData(db, function(result) { console.log(result); db.close(); }); });

执行成功后，进入 mongo 管理工具查看数据已修改：

> db.site.find()

{ "\_id" : ObjectId("58c25f300cd56e0d7ddfc0c8"), "name" : "菜鸟教程", "url" : "https://www.runoob.com" }

{ "\_id" : ObjectId("58c25f300cd56e0d7ddfc0c9"), "name" : "菜鸟工具", "url" : "c.runoob.com" }

**删除数据**

以下实例将 name 为 "菜鸟工具" 的数据删除 :

**删除数据**

var MongoClient = require('mongodb').MongoClient; var DB\_CONN\_STR = 'mongodb://localhost:27017/runoob'; var delData = function(db, callback) { //连接到表 var collection = db.collection('site'); //删除数据 var whereStr = {"name":'菜鸟工具'}; collection.remove(whereStr, function(err, result) { if(err) { console.log('Error:'+ err); return; } callback(result); }); } MongoClient.connect(DB\_CONN\_STR, function(err, db) { console.log("连接成功！"); delData(db, function(result) { console.log(result); db.close(); }); });

执行成功后，进入 mongo 管理工具查看数据已删除：

> db.site.find()

{ "\_id" : ObjectId("58c25f300cd56e0d7ddfc0c8"), "name" : "菜鸟教程", "url" : "https://www.runoob.com" }

>