Node.js JXcore 打包 Node.js 连接 MongoDB

Node.js 连接 MySQL

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

如果你还没有 MySQL 的基本知识，可以参考我们的教程：MySQL 教程。

本教程使用到的 Websites 表 SQL 文件：websites.sql。

安装驱动

本教程使用了淘宝定制的 cnpm 命令进行安装：

```
$ cnpm install mysql
```

连接数据库

在以下实例中根据你的实际配置修改数据库用户名、及密码及数据库名：

test.js 文件代码：

```
var mysql = require('mysql');
var connection = mysql.createConnection({
host : 'localhost',
user : 'root',
password : '123456',
database : 'test'
});
connection.connect();
connection.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
if (error) throw error;
console.log('The solution is: ', results[0].solution);
});
```

执行以下命令输出结果为：

```
$ node test.js
The solution is: 2
```

数据库连接参数说明：

参数	描述
host	主机地址（默认：localhost）
user	用户名
password	密码
port	端口号（默认：3306）
database	数据库名

charset	连接字符集（默认：'UTF8_GENERAL_CI'，注意字符集的字母都要大写）
localAddress	此IP用于TCP连接（可选）
socketPath	连接到unix域路径，当使用 host 和 port 时会被忽略
timezone	时区（默认：'local'）
connectTimeout	连接超时（默认：不限制；单位：毫秒）
stringifyObjects	是否序列化对象
typeCast	是否将列值转化为本地JavaScript类型值 （默认：true）
queryFormat	自定义query语句格式化方法
supportBigNumbers	数据库支持bigint或decimal类型列时，需要设此option为true （默认：false）
bigNumberStrings	supportBigNumbers和bigNumberStrings启用 强制bigint或decimal列以JavaScript字符串类型返回（默认：false）
dateStrings	强制timestamp,datetime,data类型以字符串类型返回，而不是JavaScript Date类型（默认：false）
debug	开启调试（默认：false）
multipleStatements	是否许一个query中有多个MySQL语句 （默认：false）
flags	用于修改连接标志
ssl	使用ssl参数（与crypto.createCredenitals参数格式一至）或一个包含ssl配置文件名称的字符串，目前只捆绑Amazon RDS的配置文件

更多说明可参见：https://github.com/mysqljs/mysql

数据库操作(CURD)

在进行数据库操作前，你需要将本站提供的 Websites 表 SQL 文件websites.sql 导入到你的 MySQL 数据库中。

本教程测试的 MySQL 用户名为 root，密码为 123456，数据库为 test，你需要根据自己配置情况修改。

查询数据

将上面我们提供的 SQL 文件导入数据库后，执行以下代码即可查询出数据：

查询数据

```
var mysql = require('mysql');
var connection = mysql.createConnection({
host : 'localhost',
user : 'root',
password : '123456',
port: '3306',
database: 'test'
});
connection.connect();
var sql = 'SELECT * FROM websites';
//查
connection.query(sql,function (err, result) {
if(err){
console.log('[SELECT ERROR] - ',err.message);
return;
}
console.log('--------------------------SELECT----------------------------');
console.log(result);
console.log('------------------------------------------------------------\n\n');
});
connection.end();
```

执行以下命令输出就结果为：

```
$ node test.js
--------------------------SELECT----------------------------
[ RowDataPacket {
    id: 1,
    name: 'Google',
    url: 'https://www.google.cm/',
    alexa: 1,
    country: 'USA' },
  RowDataPacket {
    id: 2,
    name: '淘宝',
    url: 'https://www.taobao.com/',
    alexa: 13,
    country: 'CN' },
  RowDataPacket {
    id: 3,
    name: '菜鸟教程',
    url: 'http://www.runoob.com/',
    alexa: 4689,
    country: 'CN' },
  RowDataPacket {
    id: 4,
    name: '微博',
    url: 'http://weibo.com/',
    alexa: 20,
    country: 'CN' },
```

```
  RowDataPacket {
    id: 5,
    name: 'Facebook',
    url: 'https://www.facebook.com/',
    alexa: 3,
    country: 'USA' } ]
 ------------------------------------------------------------
```

插入数据

我们可以向数据表 websties 插入数据：

插入数据

```
var mysql = require('mysql');
var connection = mysql.createConnection({
host : 'localhost',
user : 'root',
password : '123456',
port: '3306',
database: 'test'
});
connection.connect();
var addSql = 'INSERT INTO websites(Id,name,url,alexa,country) VALUES(0,?,?,?,?)';
var addSqlParams = ['菜鸟工具', 'https://c.runoob.com','23453', 'CN'];
//增
connection.query(addSql,addSqlParams,function (err, result) {
if(err){
console.log('[INSERT ERROR] - ',err.message);
return;
}
console.log('--------------------------INSERT----------------------------');
//console.log('INSERT ID:',result.insertId);
console.log('INSERT ID:',result);
console.log('-----------------------------------------------------------------\n\n');
});
connection.end();
```

执行以下命令输出就结果为：

```
$ node test.js
--------------------------INSERT----------------------------
INSERT ID: OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 6,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
```

```
  changedRows: 0 }
---------------------------------------------------------------
```

执行成功后，查看数据表，即可以看到添加的数据：

更新数据

我们也可以对数据库的数据进行修改：

更新数据

```
var mysql = require('mysql');
var connection = mysql.createConnection({
host : 'localhost',
user : 'root',
password : '123456',
port: '3306',
database: 'test'
});
connection.connect();
var modSql = 'UPDATE websites SET name = ?,url = ? WHERE Id = ?';
var modSqlParams = ['菜鸟移动站', 'https://m.runoob.com',6];
//改
connection.query(modSql,modSqlParams,function (err, result) {
if(err){
console.log('[UPDATE ERROR] - ',err.message);
return;
}
console.log('--------------------------UPDATE----------------------------');
console.log('UPDATE affectedRows',result.affectedRows);
console.log('-----------------------------------------------------------------\n\n');
});
connection.end();
```

执行以下命令输出就结果为：

```
--------------------------UPDATE----------------------------
UPDATE affectedRows 1
---------------------------------------------------------------
```

执行成功后，查看数据表，即可以看到更新的数据：

id	name	url	alexa	country
1	Google	https://www.google.cm/	1	USA
2	淘宝	https://www.taobao.com/	13	CN
3	菜鸟教程	http://www.runoob.com/	4689	CN
4	微博	http://weibo.com/	20	CN
5	Facebook	https://www.facebook.com/	3	USA
6	菜鸟移动站	https://m.runoob.com	23453	CN

删除数据

我们可以使用以下代码来删除 id 为 6 的数据:

删除数据

```
var mysql = require('mysql');
var connection = mysql.createConnection({
host : 'localhost',
user : 'root',
password : '123456',
port: '3306',
database: 'test'
});
connection.connect();
var delSql = 'DELETE FROM websites where id=6';
//删
connection.query(delSql,function (err, result) {
if(err){
console.log('[DELETE ERROR] - ',err.message);
return;
}
console.log('--------------------------DELETE----------------------------');
console.log('DELETE affectedRows',result.affectedRows);
console.log('-----------------------------------------------------------------\n\n');
});
connection.end();
```

执行以下命令输出就结果为：

```
--------------------------DELETE----------------------------
DELETE affectedRows 1
------------------------------------------------------------
```

执行成功后，查看数据表，即可以看到 id=6 的数据已被删除：

1	Google	https://www.google.cm/	1	USA
2	淘宝	https://www.taobao.com/	13	CN
3	菜鸟教程	http://www.runoob.com/	4689	CN
4	微博	http://weibo.com/	20	CN
5	Facebook	https://www.facebook.com/	3	USA

id 为 6 的数据已被删除

← Node.js JXcore 打包

Node.js 连接 MongoDB →

点我分享笔记