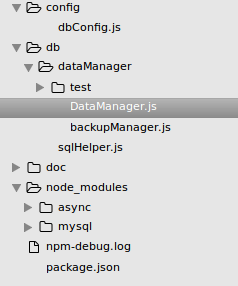
# Nodejs 连接 Mysql 实例说明

## 文档结构:

 图1.

## 文件内容：

### 1.config文件夹

如图1所示,初步文档结构如图所示， config文件夹中包含了mysql数据库的相关信息，结构如下：

module.exports={

local:{

host:'localhost',//本地连接

user:'root',//root用户

password:'123456',

charset:'utf8\_general\_ci'//显示中文

},

test:{

host:'\*\*\*.\*\*\*.\*\*\*.\*\*\*',//主机ip地址

user:'username',//共享的远程用户名

password:'123456',/密码

charset:'utf8\_general\_ci'

}

}

通过将mysql信息作为模块导出，在连接部分引用即可。

### 2.db文件夹：

此文件夹主要存放nodejs连接mysql的部分，也是本文最主要的代码所在。

**sqlHelper.js内容如下：**

var mysql=require('mysql');//导入mysql模块，需要npm install mysql导入

module.exports=SqlHelper;

function SqlHelper(dbConnection){//dbConnection是上面代码段的dbConfig中的信息，之后可以看到引用方法

//this.connection=mysql.createConnection(dbConnection);//用Connection方法也可以连接

this.pool=mysql.createPool(dbConnection);

}

SqlHelper.prototype.query=function(sqlText,callBack){//sqlText即是相应的mysql语句

this.pool.getConnection(function(err,connection){

if(err){

console.log(err);//错误处理，连接失败，可以更改为更好的方式，本文只是打印出错误信息。

}

connection.query(sqlText,function(err,result){

if(err){

result = err;//插入失败的话返回错误信息，因为本文中新建的表结构包含了唯一的数据，可能会报错，用来分析错误信息来判断是更新还是插入新数据。

}

callBack(result);

connection.release();//释放连接

});

});

}

**backupManager.js内容：**

module.exports=BackupManager;

function BackupManager(helper){

this.sqlHelper=helper;

}

//创建表格的代码，backupFiles为我的数据库名。Ostype为表名

BackupManager.prototype.createtable=function(ostype,callback){

var sqlText = "create table backupFiles.@ostype(\n"+

"Id int(4) NOT NULL PRIMARY KEY AUTO\_INCREMENT,\n"+

"Filename varchar(128) NOT NULL UNIQUE,\n"+

"Filepath varchar(255) NOT NULL,\n"+

"Filesize int NOT NULL,\n"+

"File mediumtext Not NULL,\n"+

"Filemd5 varchar(255) NOT NULL,\n"+

"Ostype varchar(128) NOT NULL,\n"+

"Date DATETIME NOT NULL\n"+

");";

sqlText = sqlText.replace('@ostype',ostype);

this.sqlHelper.query(sqlText,callback);

}

//用来检验表中是否已经存在文件名为filename的数据了

BackupManager.prototype.check=function(data,callback){

var findText = "select \* from backupFiles.@ostype where filename='@filename'";

findText = findText.replace('@ostype',data.ostype);

findText = findText.replace('@filename',data.filename);

this.sqlHelper.query(findText,callback);

}

//用来上传新的数据

BackupManager.prototype.upload=function(data,callback){

var sqlText="insert into backupFiles.@ostype values(\"\",\""+ data.filename+"\",\""+data.filepath+"\",\""+data.filesize+"\",\""+data.filetext+"\",\""+data.filemd5+"\",\""+data.ostype+"\",now())";

console.log(sqlText);

sqlText = sqlText.replace('@ostype',data.ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

//用来更新数据

BackupManager.prototype.update= function(data,callback){

var sqlText="update backupFiles.@ostype set Filepath='@filepath',Filesize=@filesize,File='@filetext',Filemd5='@filemd5',Ostype='@ostype',Date=now() where Filename='@filename'";

sqlText = sqlText.replace('@filename',data.filename);

sqlText = sqlText.replace('@filepath',data.filepath);

sqlText = sqlText.replace('@filesize',data.filesize);

sqlText = sqlText.replace('@filetext',data.filetext);

sqlText = sqlText.replace('@filemd5',data.filemd5);

sqlText = sqlText.replace(/@ostype/g,data.ostype.toLowerCase());

//console.log(sqlText);

this.sqlHelper.query(sqlText,callback);

}

//返回表中所有的数据

BackupManager.prototype.getAllData=function(ostype,callback){

var sqlText='select \* from backupFiles.@ostype';

sqlText = sqlText.replace('@ostype',ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

//返回对应表中对应文件名的文件路径

BackupManager.prototype.getFilepath=function(ostype,filename,callback){

var sqlText='select Filepath from backupFiles.@ostype where Filename="@filename" AND Ostype="@ostype"';

sqlText = sqlText.replace('@filename',filename);

sqlText = sqlText.replace(/@ostype/g,ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

//返回对应表中对应文件名的文件大小

BackupManager.prototype.getFilesize=function(ostype,filename,callback){

var sqlText='select Filesize from backupFiles.@ostype where Filename="@filename"';

sqlText = sqlText.replace('@filename',filename);

sqlText = sqlText.replace('@ostype',ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

//返回对应表中对应文件名的文件内容

BackupManager.prototype.getFiletext = function(ostype,filename,callback){

var sqlText='select File from backupFiles.@ostype where Filename="@filename"';

sqlText = sqlText.replace('@filename',filename);

sqlText = sqlText.replace('@ostype',ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

//返回对应表中对应文件名的文件md5

BackupManager.prototype.getFilemd5=function(ostype,filename,callback){

var sqlText='select Filemd5 from manage.backupFiles where Filename="@filename"';

sqlText = sqlText.replace('@filename',filename);

sqlText = sqlText.replace('@ostype',ostype.toLowerCase());

this.sqlHelper.query(sqlText,callback);

}

在DataManager.js中进一步封装各种方法。并返回需要的结果

**DataManager.js内容：**

var BackupManager = require('./backupManager');

var dbConfig=require('../../config/dbConfig');

var SqlHelper=require('../../db/sqlHelper');

var async = require('async');//异步控制流程模块

var fs = require('fs');

var sqlHelper=new SqlHelper(dbConfig.local);

var backupManager = new BackupManager(sqlHelper);

var DataManager = function(){

}

//进一步封装的上传方法，

//根据上传json格式的data数据判断，若data.ostype对应的表不存在，则调用createTable

//新建一个表；

//若表中不存在data.filename的数据，则使用upload导入新数据

//若存在，则用update更新数据。

DataManager.prototype.Upload = function(data,callback){

var stat = fs.statSync(data.filepath);

var result = {

detail:"",

code:0 //默认为0,-1表示有错误，

//1表示表原来不存在，新建了并导入了数据

//2表示表存在，没有相同名称的数据文件，新建数据并导入

//3表示表存在，有相同名称的数据文件，更新数据并导入

}

data.filesize = stat.size;

if(!fs.existsSync(data.filepath)){

result.detail = "路径不存在";

result.code = -1;

callback(result);

}else{

var content = fs.readFileSync(data.filepath,"utf-8");

content = content.replace(/"/g,"#@#");

content = content.replace(/'/g,"#$#");

content = content.replace(/`/g,"#%#");

data.filetext = content;

backupManager.check(data,function(subdata){

//console.log(subdata);

if(subdata.code == "ER\_NO\_SUCH\_TABLE"){

var ostype = data.ostype.toLowerCase();

backupManager.createtable(ostype,function(data0){

//console.log(data0);

});

backupManager.upload(data,function(data1){

result.detail = JSON.stringify(data1);

result.code = 1;

callback(result);

})

}else{

if(subdata.length == 0){

//console.log(data.length);

backupManager.upload(data,function(data2){

result.detail = JSON.stringify(data2);

result.code = 2;

callback(result);

})

}else{

//console.log(data.length);

backupManager.update(data,function(data3){

result.detail = JSON.stringify(data3);

result.code = 3;

callback(result);

})

}

}

});

}

}

//调用一组数据，都是json格式的数组，这里用到了async.map方法来控制异步流程。

//结果仍是异步的，导入的数据顺序不定，若需要按照顺序执行，可调用async别的方法

DataManager.prototype.UploadMultiple = function(datalist,callback){

async.map(datalist,this.Upload,function(err,results){

callback(results);

})

}

//获取所有的数据

DataManager.prototype.GetAllData = function(ostype,callback){

backupManager.getAllData(ostype,function(data){

callback(data);

});

}

//获取文件路径

DataManager.prototype.GetFilePath = function(ostype,filename,callback){

backupManager.getFilepath(ostype,filename,function(data){

callback(data);

});

}

//获取文件大小

DataManager.prototype.GetFileSize = function(ostype,filename,callback){

backupManager.getFilesize(ostype,filename,function(data){

callback(data);

});

}

//获取文件内容，对文件内容进行了替换，因为存储文件时进行了一定的替换，此时为反替

//换，保证数据一致

DataManager.prototype.GetFileText = function(ostype,filename,callback){

backupManager.getFiletext(ostype,filename,function(data){

data[0].File = data[0].File.replace(/#@#/g,"\"");

data[0].File = data[0].File.replace(/#$#/g,"'");

data[0].File = data[0].File.replace(/#%#/g,"`");

callback(data[0].File);

});

}

module.exports = DataManager;

**test.js内容:**

var DataManager = require('../DataManager');

var fs = require('fs');

var datamanager = new DataManager();

var data = {

filename:"syslog-ng",

filepath:"/home/zt/syslog-ng",

filesize:0,

filetext:"",

filemd5:"",

ostype:"redhat"

}

datamanager.Upload(data,function(result){

console.log(result);

})

var datalist = [

{

filename:"change1",

filepath:"/home/zt/changelog",

filesize:0,

filetext:"",

filemd5:"",

ostype:"ubuntu"

},

{

filename:"change2",

filepath:"/home/zt/changelog",

filesize:0,

filetext:"",

filemd5:"",

ostype:" ubuntu "

},

{

filename:"change3",

filepath:"/home/zt/changelog",

filesize:0,

filetext:"",

filemd5:"",

ostype:" ubuntu "

}

]

datamanager.UploadMultiple(datalist,function(result){

console.log(result);

})

// console.log(content);

datamanager.GetAllData("ubuntu ",function(data){

console.log(data);

})

datamanager.GetFileSize("ubuntu ","syslog",function(data){

console.log(data);

})

datamanager.GetFilePath("ubuntu ","syslog","CDOS",function(data){

console.log(data);

})

datamanager.GetFileText("ubuntu ","syslog-ng",function(data){

console.log(data);

})