Mongoose

Node操作mongodb数据库有很多模块,这里先讲讲mongoose这个模块,这一节用到了mvc的设计思想,代码分层比较多，也许需要多看几遍

首先是链接mongodb的数据的JS文件 。文件路径node\_dian\models\mongoose.js

var mongoose = require('mongoose');

mongoose.connect('mongodb://127.0.0.1:27017/user');

require('./user');

exports.User = mongoose.model('user');

这样封装起来的好处就是 程序启动后链接一次,其他对于数据库的请求可以直接用了,不需要做无数次的 open/close的操作了

然后再来看看model层

对于数据库的操作 首先得创建一个collection的 Model。文件路径node\_dian\models\user.js

var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var UserSchema = new Schema({

username: { type: String},

pwd: { type: String},

email: { type: String},

nickname: { type: String }

});

mongoose.model('user', UserSchema);

UserSchema 对象就是该collection的所有的字段

最后来看看调用

var mongoose=require('./models/mongoose.js');

var UserModel = mongoose.User;

UserModel.find({},function(err,user){

if(err){

logInfo.info(err.message);

}else{

logInfo.info(JSON.stringify(user));

res.render('test\_user.html',{user:user});

}

})

这里有个问题必须得注意一下,用mongogse 来操作mongodb的时候 ,如上面例子来操作的collection 名字是user. 开始我的本意是想用它来操作 已经存在于mongodb数据库中的user 集合,但是后来发现怎么查都显示数据的个数为0.后来知道了原来mongoose 在collection的数据名称后会自动加上一个s，也就是说我所写的这个访问mongodb的Model实际上是访问的users,但是此时我的数据库中并没有users这个数据集.所以查询的结果是0.

如果我们要查询现有的数据集合那么在定义model的时候就必须加上一个collection的参数

var UserSchema = new Schema({

username: { type: String},

pwd: { type: String},

email: { type: String},

nickname: { type: String }

},{ collection: 'user'});