nodejs+express+mongodb简单的例子

发表于1年前(2014-09-11 11:45)   阅读（21693） | 评论（[8](http://my.oschina.net/chenhao901007/blog/312367#comments)） **22**人收藏此文章, [我要收藏](javascript:add_to_favor(312367,3))

赞5

摘要 刚学习不久，在网上找了不少资料，遇到不少问题，都解决了。感觉收获蛮多的。但是还是觉得菜鸟的路还有很远很远... 文中写的不对的地方，欢迎指出~~~

**简单的介绍下node+express+mongodb这三个东西。**  
 **node**：是运行在服务器端的程序语言，表面上看过去就是javascript一样的东西，但是呢，确实就是服务器语言，个人觉得在一定层次上比c灵活，java就不提了。反正你只要认为node可以干很多事就行了，绝对不只是web开发。 **express**：这货呢，就是node的一种框架，node有很多的开源框架，express是一个大神开发的（这尊神已经移驾到go语言的开发去了）。express可以让你更方便的操作node（因为原生的node写起来比较麻烦，而且因为node是事件驱动的，所以有很多异步回调，写多了就看着晕...）  
**mongodb**：这是一种非关系数据库（nosql），太深的东西我也不清楚，反正这货也有很强大的地方，缺点就是不适合数据一致性要求高的比如金融方面的开发。但是优点就快。  
总结：也就是说node和mongodb组合起来特别适合一个应用场景——速度快，处理量大的情况。

下面先说说准备工作：（以windows8.1系统环境为例）  
1.node：先下载安装nodejs，下载地址<http://www.nodejs.org/>，安装好了之后检查是否在系统环境变量里自动配置好了path，如果没有，请把node的安装地址配置到path里去。然后在cmd里敲入node，如果可以，那么这一步ok了，很简单吧。  
2.express：这个安装分两种，一种是全局安装，一种是本地安装。网络上说的玄乎玄乎的，但是实际操作中你会发现不同的系统环境会出现问题，比如win8.1中文用户名情况下，你得改变npm的全局路径（npm是node模块包管理程序，在你安装好node的时候已经自带npm了），操作：  
prefix = 自定义模块目录  
cache = 自定义缓存目录  
接着安装express（新版本4.X以上要安装express-generator）  
npm i -g express（旧版本npm i -g express+版本号）  
npm i -g express-generator  
（如果想本地安装在和npm同一个目录下，先cd到node\_modules目录，再使用npm i express-generator命令，然后再将node\_modules目录下的./bin配置到环境变量path里去）  
测试：  
express -V 成功的话会显示版本号  
3.mongodb：这个数据库安装很简单，下载地址<http://www.mongodb.org/>。安装好了，就在cmd里cd到mongodb安装目录下的bin目录，然后敲命令：  
mongod --dbpath="mongodb安装目录\data" --logpath="mongodb安装目录\log\log.txt" --install --serviceName MongoDB --serviceDisplayName MongoDB  
操作完，你会发现，你的电脑的服务里多了一个MongoDB服务，没错，就是它，然后你运行这个服务就行了。

**正题：搭建简单的node+express+mongodb项目**  
  
先在cmd控制台里cd到一个目录下面，记住这你的workspace，然后是用是用express创建一个app项目  
express hello-world -e （-e表示支持ejs模板引擎，默认是jaden。什么事模板引擎，比如jsp...太深的我也不懂。本人比较擅长html原生的东西，像这种模板引擎我也是第一次使用，也蛮方便的哦，不过在我看来，没啥用, 我不需要，但是可能你需要...）  
然后我们再下载依赖包  
npm i （这样就会自动将项目需要的依赖modules安装到项目的modules里去了）  
我们cd到hello-world目录下，是用命令  
npm start 启动项目（也可以是node ./bin/www，旧版本直接node app.js，因为具体要看package.json里的启动配置了）  
我们可以在浏览器地址栏里敲入 <http://127.0.0.1:3000/> 这就是你的第一个express创建的node app。  
是不是很哈皮。。。  
  
我们研究下express创建项目  
你需要了解的项目主要目录为：routes和views，你最好再在项目里新建一个目录叫models（作用后面讲）  
routes里index.js配置的都是get和post请求的路径映射关系，很简单的哦。  
views里index.ejs就相当于一个html文件，里面就是一些html标签和<%%>标签，感觉和jsp差不多哦。  
看起来不错的样子，标准的MVC框架（models里放模型，views里面放展示，routes里面放控制）  
  
上面我们已经生成好了app原型，接着我们设计数据库  
cmd命令行里：  
mongo //进入数据库  
use hello-world //创建项目数据库  
db.addUser("shuaige", "123456") //给这个数据库创建了一个叫帅哥的账号，密码123456 （但是我觉得可能我理解的不到位，你也可以不做这个操作）  
然后，我们就为这个hello-world数据库创建collection（collection就相当于oracle和mysql里的table）  
db.createCollection("users") //创建一个集合，也就是表  
db.users.insert({userid: "admin", password: "123456"}) //给users里添加一个文档，也就是一条记录账号admin，密码123456  
ok，现在检查一下：  
db.users.find() //如果看到你刚刚添加的文档记录，就ok咯  
  
好简单的数据库集合以及文档设置好，我们就回到express创建的node项目里，我们需要：     
  
在models下创建一个user.js,作为实体类映射数据库的users集合   
在views下做几个页面（可以用ejs也可以用html，我就用ejs吧）  
在routes下的index.js配置路由，也就是请求映射处理

**1在models下创建一个user.js,作为实体类映射数据库的users集合**  
user.js

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var mongoose = require("mongoose");  //  顶会议用户组件  var Schema = mongoose.Schema;    //  创建模型  var userScheMa = new Schema({      userid: String,      password: String  }); //  定义了一个新的模型，但是此模式还未和users集合有关联  exports.user = mongoose.model('users', userScheMa); //  与users集合关联 |

**2在views下面建index.ejs, errors.ejs, login.ejs, logout.ejs, homepage.ejs。** **（index是自带的，不用建）**  
  
index.ejs

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <!DOCTYPE html>  <html>    <head>      <title><%= title %></title>      <link rel='stylesheet' href='/stylesheets/style.css' />    </head>    <body>      <h1>Hello World</h1>      <p>Welcome to <%= title %></p>      <p><a href="login">登陆</a></p>    </body>  </html> |

login.ejs

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <!DOCTYPE html>  <html>    <head>      <title><%= title %></title>      <link rel='stylesheet' href='/stylesheets/style.css' />    </head>    <body>      <h1>Hello World</h1>      <p>Welcome to <%= title %></p>      <form action="homepage" method="post">          <p>              <span>userId:</span>              <br>              <input id="userid" name="userid" type="text">          </p>          <p>              <span>password:</span>              <br>              <input id="password" name="password" type="password">          </p>          <p><input type="submit" value="submit"></p>      </form>    </body>  </html> |

logout.ejs

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <!DOCTYPE html>  <html>    <head>      <title><%= title %></title>      <link rel='stylesheet' href='/stylesheets/style.css' />    </head>    <body>      <h1>Hello World</h1>      <p>Welcome to <%= title %></p>      <p>正在登出...</p>      <script type="text/javascript">          setTimeout(function(){              window.location.href = "/";          }, 500);      </script>    </body>  </html> |

homepage.ejs

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <!DOCTYPE html>  <html>    <head>      <title><%= title %></title>      <link rel='stylesheet' href='/stylesheets/style.css' />    </head>    <body>      <h1>Hello World</h1>      <p>Welcome to <%= title %></p>      <p><a href="logout">登出</a></p>    </body>  </html> |

error.ejs 出错页面，我没做，你有兴趣可以自己试试玩玩。   
  
3 在routes下的index.js配置路由，也就是请求映射处理  
  
index.js

[?](http://my.oschina.net/chenhao901007/blog/312367)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | var express = require('express');  var router = express.Router();  var mongoose = require('mongoose');  var user = require('../models/user').user;  mongoose.connect('mongodb://localhost/hello-world');    /\* GET home page. \*/  router.get('/', function(req, res) {        res.render('index', { title: 'index' });  });    /\*login\*/  router.get('/login', function(req, res) {      res.render('login', { title: 'login' });  });    /\*logout\*/  router.get('/logout', function(req, res) {        res.render('logout', { title: 'logout' });  });    /\*hompage\*/  router.post('/homepage', function(req, res) {      var query\_doc = {userid: req.body.userid, password: req.body.password};      (function(){          user.count(query\_doc, function(err, doc){              if(doc == 1){                  console.log(query\_doc.userid + ": login success in " + new Date());                  res.render('homepage', { title: 'homepage' });              }else{                  console.log(query\_doc.userid + ": login failed in " + new Date());                  res.redirect('/');              }          });      })(query\_doc);  });    module.exports = router; |

ok，基本上大功告成，可以试试咯。

**下面讲讲如何调试服务器端的代码：**  
我们最好借助一个叫node-inspector的工具包  
npm i -g node-inspector //安装node-inspector  
然后在cmd里运行  
node-inspector  
再新打开一个cmd，cd到项目hello-world目录下  
node --debug ./bin/www （或者 node --debug-brk ./bin/www , 旧版本express创建的node程序请使用 node --debug app.js）  
在浏览器里打开http://127.0.0.1:8080/debug?port=5858  
再新建窗口打开http://127.0.0.1:3000/  
就在浏览器可以调试服务器端代码。