今天学习的内容

1:复习昨天知识重点

NodeJS 模块分类

(1)官方提供原生模块 querystring url Buffer fs http

(2)第三方模块 npm mysql

(3)自定义模块

文件模块:user.js exports.XX require("./user");

目录模块:node\_modules/db/package.json require("db");

http模块的使用

const http = require("http");

var server = http.createServer();

server.listen(8080);

server.on("request",(req,res)=>{

});

mysql模块的使用

#npm i mysql ./node\_modules/mysql/...

const mysql = require("mysql");

var pool = mysql.createPool({

host:"",user:"",password:"",database:"",connectionLimit:10

});

pool.getConnection((err,conn)=>{

conn.query("SELECT ...?,?",[x,y],(err,result)=>{

....

});

});

练习:9:35--9:50

(1)编写sql: tmooc.sql 数据库名为tmooc,表 stu

(sid,sname,score,schoolTime-入学时间)

(2)创建NodeJS应用，01\_app.js 创建web服务器，监听端口8080

接收如下请求地址

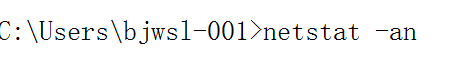
/stu/add 向客户端输出/public/stu/add.html内容

/stu/add.do 接收客户端提交sname/score/schoolTime,保存tmooc/stu表中，并返回 "添加成功！新记录编号为 5"

/stu/list 向客户端输出数据库stu表中所有学生信息，以JSON格式.

提示:将js数据转换JSON字符串 JSON.stringify(result)

下面指定用来查询端口是否启动



2:作业

3:NodeJS框架 Express

使用官方提供http模块可以创建一个web服务器应用，但是此模块非常底层，要处各种情形，比较繁琐。推荐使http模块进一步封装简化模块---express---第三方模块。是一个基于NodeJS http模块而编写高层模块，简化web服务器应用的开发，V4版本官方

该模块还有自己专用官方网站: http://expressjs.com

该模块有一个中文镜像网: http://www.expressjs.com.cn

使用方式

(1)下载express npm i express

(2)在自己的模块中引入express模块

const express = require("express");

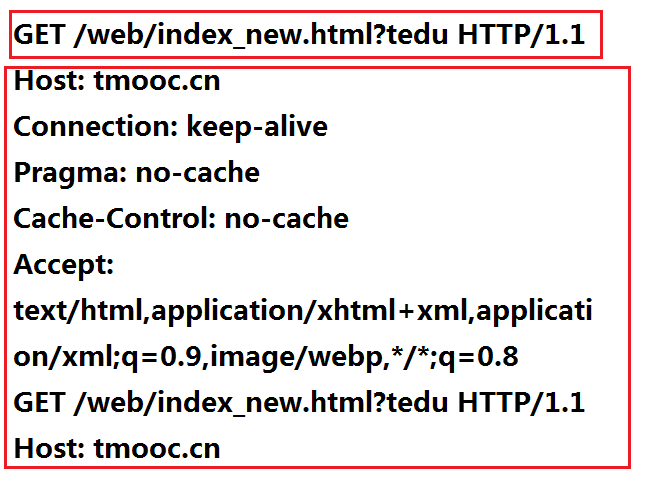
(3)使用http模块创建web服务器对象,让express处理请求

var app = express();

var server = http.createServer(app);

server.listen(8080);

HTTP协议--请求和响应



请求方法(**方式**):用于标明此次请求目的

GET 表客户端想"获得"指定的资源

POST 表客户端想"**添加**/上传"指定数据给服务器，相关数据在请求主体中.

PUT 表客户端想"放置/**更新**"服务器上指定资源，相关数据在请求主体中.

DELETE 表客户端想"删除"服务器上指定资源

|  |
| --- |
| POST和PUT语义区别:  所有请求方法中只有二个者有请求主体,都可以向服务器传数据  POST:一般情况表示"添加",每次POST请求都在服务器上产生一条数据记录.  PUT:一般情况表示"更新",多次请求也只能产生一条影响. |

|  |
| --- |
| GET /user 表示客户端想获取所有用户数据 |
| GET /user?pno=2&psize=10 表示客户端想分页获取数据 |
| GET /user/pno/2/psize/10 表示客户端想分页获取数据 |
| GET /user/10 表示客户端想获取编号为10数据 |
| POST /user 表示客户端想添加一条记录  uname=tom&age=10 |
| PUT /user 表示客户端想更新一条记录  uname=james&uid=1 |
| DELETE /user 表示客户端想删除服务器所有用户数据 |
| DELETE /user?uid=9 表示客户端想删除服务器编号为9用户 |
| DELETE /user/9 表示客户端想删除服务器编号为9用户 |

浏览器中如何发起GET请求

地址栏输入URL/超链接/js跳转/src/href/表单GET/AJAX-GET

浏览器中如何发起POST请求

表单-POST/AJAX-POST

浏览器中如何发起PUT请求

AJAX-PUT

浏览器中如何发起DELETE 请求

AJAX-DELETE

练习:使用Express创建一个web服务器,可以向客户端提供一个 /index

响应,内容是一个HTML页面.其中还使用一个css文件/图片/jquery.js文件

<html>

<head>

<link.. href="/index.css">

</head>

<body>

<img src="/logo" />

<script src="/jquery" />

</body>

</html>

使用 Express处理客户端的GET请求

app.get(url,(req,res)=>{

res.send("向客户端返回HTML字符串响应");

res.sendFile(\_\_dirname+"/向客户端输出指定文件内容");

res.json(obj); //将obj转化为JSON字符串，输出给客户端

})

|  |
| --- |
| Express路由  //路由(Route)=Method+Path+fn |

参数:

(1)接收GET请求中查询字符串

//GET /user?uid=10&loc=bj

app.get("/user",(req,res)=>{

//express为每一个req对象添加属性query

req.query.uid;

req.query.loc

});

(2)接收GET 请求中请求参数

//GET /user/jsj/60

app.get("/user/:type/:id",(req,res)=>{

//express为每个req对象添加属性 params属性

console.log(req.params.type);//jsj

console.log(req.params.id);//60

});

练习:创建Express服务器，配置合mysql模块,接收如下请求

jd/jd\_user

GET /stu/3 向客户端输出用户编号3信息,以JSON格式

GET /stu/5 向客户端输出用户编号 5信息,以JSON格式

疑问:如何使express处理静态资源(jpg/css/js...)

express 4.0 官方提供函数(中间件)

app.use(express.static('public'));

#若客户请求了/public目录某个指定资源,它可以直接向客户端返回，

#不会再调用后续路由

#使用Express处理客户端的POST请求

app.post(url,(req,res)=>{

//接收请求主体的数据

req.on("data",(data)=>{

var obj = qs.parse(data.toString());

})

});

#使用Express处理客户端的PUT请求

app.put(url,(req,res)=>{

//接收请求主体的数据

req.on("data",(data)=>{

var obj = qs.parse(data.toString());

})

});

#使用Express处理客户端的DELETE请求

app.delete(url,(req,res)=>{});

练习内容:

1:编写SQL：创建数据库dangdang,书信息表book(bid/title

price,pubdate,intro-内容简介)并插入4行记录

2:使用express创建web服务器应用,可接收如下的静态请求

/public/booklist.html

/public/jquery.js

/public/book\_add.html

/public/book\_detail.html

以及如下动态资地址--注意:下述地址使用AJAX请求

1) GET /book

向客户端输出书籍表中 所有记录,以JSON格式

2) GET /book/:bid

向客户端输出指定编号书籍全部信息,以JSON格式

3)POST /book

接客户端提交请求主体数据(title/price/pubdate/intro)

执行书籍添加操作保存数据库，向客户端{code:1,bid:xx}