今天学习的内容

1:复习昨天内容

复习node.js

原理:node.js对文件模块自动代码添加

(function(){})

五个参数:

exports

module.exports

exports.age = 10; 向外导出变量

module.exports.age = 10;

module.exports = {name:”tom”}; ok

exports = {name:”tom”}; error

require();

2:今天的目标

2.1: 中间件-(路由级中间件)

Express 是一个自身功能极简，完全是中间件构成web框架

从本质上讲一个express应用就是由中间件组成系统.

2.2:中间件(Middleware)是一个函数,它可以访问请求对象和响应对象

(req request,res response),可以控制请求-响应流程,有一个参数

next变量 下一个中间件

2.3:中间件功能包括

执行任何代码

修改请求和响应对象

终结请求-响应循环 [拦载器]

调用下一个中间件

2.4:中间件分类

应用级中间件(\*\*\*)

路由级中间件(\*\*\*)

错误处理中间件

内置中间件

2.5:标准语法

app.use(url,(req,res,next)=>{

});

url:拦载地址(触发地址)

req:请求对象

res:响应对象

next:调用下一个中间件或路由

2.6:路由中间件

路由 = 请求方法+请求地址+处理函数

示例: app.get(“/list”,(req,res)=>{…})

语法

var router = express.Router(); //创建路由中间件对象

router.get(“/find”,(req,res)=>{});

router.post(“/list”,(req,res)=>{});

module.exports = router;

app.use(“/user”,router);

http://127.0.0.1:3000/user/find

http://127.0.0.1:3000/user/list

(1)创建三个路由文件

router/userRouter.js 保存所有用户功能

router/orderRouter.js 保存所有订单功能

router/productRouter.js 保存所有产品功能

app.js

var userRouter = require(“./router/userRouter”);

var orderRouter = require(“./router/orderRouter”);

var productRouter = require(“./router/productRouter”);

app.use(“/user”,userRouter);

app.use(“/order”, orderRouter);

app.use(“/product”, productRouter);

http://127.0.0.1:3000/user/list

http://127.0.0.1:3000/order/list

2.7: 项目

课程准备

知识点{思维导图/github}

JS 基础[原型/原型链;闭包;ES6;面试题]

dom/bom

JQuery

Canvas[小鱼大鱼]

Node.js

Vue[概念背]

Rect[概念背]

项目

参考www.codeboy.com/pro

最少二个{特效多,精彩效果} 原生版/vue版

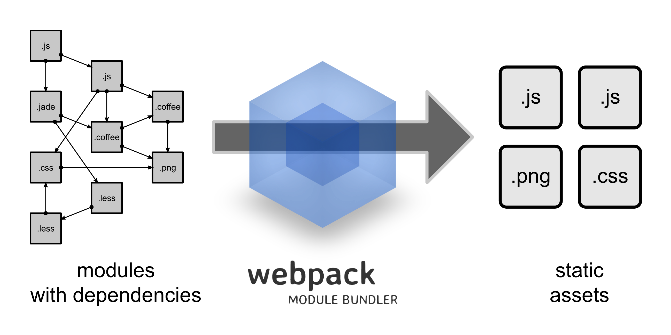
功能不用太多;细致

微信登录/微博登录

动画

商城:购物车/多选产品

2.8: 打包工具 webpack-----(演示)



webpack是一款流行打包工具，将各种资源

(.coffee/.less/.sass/图片)作为模块进行处理，生成一个或一组文件

安装配置使用

(1)创建目录 E:\>mkdir webpack-test

(2)进入指定目录 E:\>cd webpack-test

(3)初始化目录 E:\webpack-test>cnpm init

(4)询问一些问题:按回车选择默认值 自动生成文件

package.json

package name: (webpack-test) 包名

version: (1.0.0) 版本

description: 描述

entry point: (index.js) 入口程序

test command: 测试指令

git repository: git仓库地址

keywords: 关键字

author: 作者

license: (ISC) MIT MIT

Is this ok? (yes) yes

|  |
| --- |
| 如果下载npm第三方模块:  采用淘宝镜像  1:全局安装cnpm  npm install -g cnpm --registry=https://registry.npm.taobao.org  2:再安装其它工具 将npm 切换 cnpm |

(5)安装依赖工具

cnpm install webpack -save-dev

(6)全局安装webpack 指令直接操作

cnpm install webpack -g

1:创建test.js文件

webpack test.js test.bundle.js

webpack word.js word.bundle.js

2:创建1.css文件

webpack word.js word.bundle.js

3:安装二个css加载模块

cnpm install css-loader

cnpm install style-loader

webpack word.js word.bundle.js

4:监控项目,实时修改/过程/哪些模块

webpack word.js word.bundle.js --progress

webpack word.js word.bundle.js --watch

错误集锦:

1:node.js 404

原因1:node.js 服务器出错故障/没有启动

原因2:请求地址不正确

服务器: app.post(“/userupdate”,(req,res)=>);

客户端:ajax type:”POST”

url:”http://127.0.0.1:3000/userupdate”