今天学习的内容

1:复习昨天的内容

2:作业

2.1:完善用户登录

用户登录操作:

-在脚手架登录陆组件表单,让用户输入用户名和密码

点击登录按钮后在服务器验证,如果验证成功

登录自动跳转Home组件,在服务器记录用户登录成功

-session 对象生存周期

当用户登录成功后,创建session对象保存当前用户.

登录成功后将uid保存session对象

当我们查询购物车,获取session对象uid

当我们关闭浏览器 session对象消失

2.2:如果实现session存储uid并且其它功能使用uid功能

node.js

(1)验证跨域配置正常

const cors = require("cors");

app.use(cors({

origin:["http://127.0.0.1:8080",

"http://localhost:8080"],

credentials:true

}));

(2)下载第三方模块 express-session并且对其配置

const session = require("express-session");

app.use(session({

secret:"128位随机字符", //自定义安全字符串

resave:false, //每次请求都需要更新session

saveUninitialized:true, //初始化是否保存数据

cookie:{

maxAge:1000\*60\*60\*8 //依靠cookie保存8小时

}

}))

(3)下面操作***顺序***

3.1:先完成用户登录 req.session.uid = 2

3.2:再完成其它功能 "购物车列表".. var uid = req.session.uid;

#所有uid 参数不需从脚手架发送,服务器端己经保存uid

3.3:添加购物车 req.session.uid

脚手架

main.js 配置跨域访问保存session即可

axios.defaults.withCredentials=true;

其它功能一样

--常见错误

experss-session deprecated req.secret ..

secret 结果单词..

--常见错误:顺序错误

2.3:拖动删除小飞机

3:今天的目标

3.1:html5新特性---web Worker

程序:指可以被CPU执行代码存储磁盘中 1.js 1.html

进程:指程序被OS调用内存中并且分配执行空间

线程:线程是进程内部执行单位

chrome浏览器线程模型

一个chrome进程内至少有6个线程，可以向web服务器发起

请求 -- 资源请求线程

还有一个线程负责将所有资源绘制浏览器上并且执行 js程序

--UI主线程

<button></button>

<script src="1.js"></script>

<button></button>

现象:js扩行过程中，按钮1可见按钮2不可见

原因:浏览器执行js 的绘制按钮同一个线程

解决方法:创建一个新线程由它执行耗时js任务,UI线程

负责网页渲染.

<button></button>

<script>

new Worker("1.js");

#以上代码完成二个任务

#(1)创建Worker对象

#(2)创建新线程执行1.js

</script>

<button></button>

注意事项1:Worker不能双击网页运行,报错

Failed to construct 'Worker'

解决问题:必须在网络情况下运行 http://127..

注意事项2:浏览器不允许Worker执行任何DOM/BOM

对象.

原因:浏览器只允许UI主线程操作DOM/BOM,类似JQUERY

程序不能在Worker执行.

3.2: html5新特性---web Worker--数据传递

-Worker线程可以发送消息给UI线程

--Worker发送消息

postMessage(stringMsg)

--UI 接收消息

var w = new Worker("1.js");

w.onmessage = function(e){e.data}

-UI线程可以发送消息绘Worker线程

3.x:html5新特性---web Storage

3.x:html5新特性---web Socket(...)