**1，express**

**基于Node.js平台，快速，开放，极简的web开发框架。**

**下载安装: npm install express**

**(1)创建web服务器**

|  |
| --- |
| **const express=require('express'); //引入模块**  **const app=express(); //创建web服务器**  **app.listen(8800); //设置端口** |

**(2)路由**

**用来处理特定的请求，根据请求的方法和请求的URL来作出的响应，一旦匹配自动执行回调函数**

|  |
| --- |
| **app.get('/login', (req,res)=>{**  **req 请求的对象**  **req.url 获取请求的URL**  **req.method 获取请求的方法**  **req.query 获取查询字符串传递的数据**  **req.params 获取路由传参的数据**  **req.body 获取post传递(流的方式)的数据，需要使用插件才可以。**  **res 响应的对象**  **res.send() 设置响应内容并发送**  **res.sendFile() 设置响应文件并发送，需要使用绝对路径 \_\_dirname**  **res.redirect() 响应的重定向，执行跳转**  **})** |

**express中请求的对象和响应的对象和之前http模块中的两个不一样，它比之前的功能更为强大**

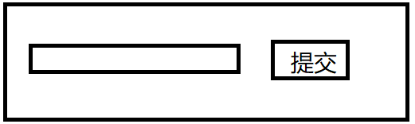
练习：创建文件02\_express.js，使用express创建web服务器并设置端口为8080，添加商品列表的路由(get，/list)，响应'这是商品的列表'

**2.路由中获取传递的数据**

|  |  |  |
| --- | --- | --- |
| 传递方式 | 格式 | 获取 |
| post传递 | **流的方式** | 通过事件，一旦触发执行回调函数  req.on('data', (chunk)=>{  chunk格式为buffer  chunk.toString() //转字符串后为查询字符串，需要使用查询字符串模块解析为对象  }) |
| get传递 | 查询字符串 | req.query 结果为对象 |
| 路由传参 | /package/express | app.get('/package/:pname', (req,res)=>{  //pname是形参，用于接收实参  //形参格式是冒号（：）开头  req.params 获取传递数据，格式为对象  }) |

练习：编写文件03\_get.js，使用express创建web服务器，设置端口为8080；创建路由获取搜索的网页(get /search)，响应文件search.html；

点击提交向服务器端发送请求(get /mysearch)，响应 ‘搜索成功’



每次修改完，重启服务器，刷新浏览器。

练习：创建添加到购物车的路由，(get /shopping)，传递商品的编号lid和价格price，最后在浏览器端响应 '商品编号：xx 价格：xxx￥'

**3.对比post和get请求**

**get 请求传递的数据通过查询字符串，可见的，会被浏览器缓存，安全程度低，但是速度快，通常用在搜索等**

**post 请求传递的数据通过流的方式传递，不可见，安全程度高，但是速度慢，通常用在注册，登陆。**

**4.路由器**

如果所有的路由写在一个文件下，无法多个人同时开发项目；

不同的模块下的路由可能会出现相同的URL。

**路由器可以解决以上问题: 将每个功能模块单独的放到路由器下，给每个路由器下的路由添加前缀。**

商品模块

列表 删除 修改

/product/list /product /delete /product /update

用户模块

列表 删除 修改

/user/list /user/delete /user/update

|  |
| --- |
| **创建路由器**  //引入express，路由器是express下的一个功能  const express=require('express');  //创建路由器对象  **const router=express.Router();**  //往路由器中添加路由  router.get('/list',(req,res)=>{  res.send('这是用户的列表');  });  //这个路由器最终要给web服务器使用，需要导出路由器对象router  module.exports=router; |
| **在web服务器下引入并挂载**  const express=require('express');  //引入用户路由器user.js  const userRouter=require('./user.js');  //把用户路由器挂载到web服务器下，给所有路由中的url添加前缀/user  //参数1：添加的前缀 访问形式 /user/list  //参数2：要挂载的路由器  app.use('/user',userRouter); |

课后任务

(1)复习今天内容，整理思维导图

(2)练习

创建web服务器app.js和商品路由器product.js，在路由器下添加路由(get /list)，把路由器挂载到服务器下，并添加前缀/product

(3)预习express中间件