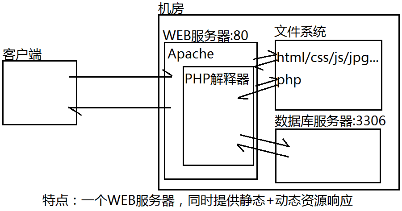
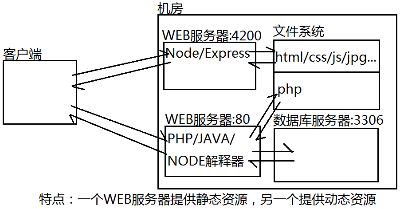
1.“学子商城前台系统”重构

**xz\_v1项目部署结构——前后端紧耦合的项目结构**



**xz\_v2项目部署结构——前后端分离的项目结构**



前提准备：准备后台数据API

**首页商品API：**

http://127.0.0.1/xz\_v1/data/product/index.php

**商品列表API：**

http://127.0.0.1/xz\_v1/data/product/list.php?pno=1&kw=apple

**商品详情API：**

http://127.0.0.1/xz\_v1/data/product/details.php?lid=9

**注册新用户API：**

http://127.0.0.1/xz\_v1/data/user/register.php?uname=x&upwd=x&email=x&phone=x

**用户登录API：**

http://127.0.0.1/xz\_v1/data/user/login.php?uname=yaya&upwd=123456

**注销登录API：**

http://127.0.0.1/xz\_v1/data/user/logout.php

**查询服务器端会话信息API：**

http://127.0.0.1/xz\_v1/data/user/session\_data.php

Angular项目实现步骤：

(1)创建Angular脚手架项目(Node+Webpack+Angular+TS编译器)

ng new xz\_v2

(2)拷贝项目静态资源到assets目录下： js/css/img/fonts

(3)在index.html中引入所有组件都共用的CSS和JS

提示：在index.html中引入全局CSS；再引入通用JS/与当前DOM树无关的JS；不能引入查找当前DOM树元素的JS！

(4)创建必需路由组件，配置路由词典，使得路由访问生效

app.component.html： <router-outlet></router-outlet>

app.module.ts： var routes = [ {path:'', component: ...} ]

app.module.ts： imports: [RouterModule.forRoot(routes)]

(5)详细完成每个路由组件

没有找到组件：ng g component not-found

页头组件：ng g component header 没有路由地址

页尾组件：ng g component footer 没有路由地址

首页组件：ng g component index

<ng-container \*ngIf=""> setTimeout( fn, 0 )

商品服务：ng g service product

商品列表组件：ng g component product-list

商品详情组件：ng g component product-detail

------------------------------------------------------------

用户注册组件：ng g component register 路由地址:/register

用户登录组件：ng g component login 路由地址:/login

登录/注册专用页头组件：ng g component header-user无需路由地址

用户服务对象：ng g service user

用户中心组件：ng g component user-center

我的信息组件：ng g component my-info

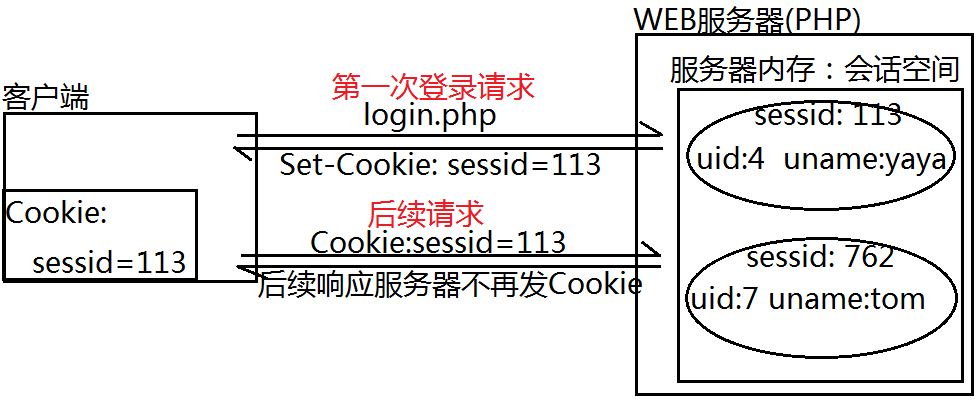
安全管理组件：ng g component security-management

2.使用HttpClient发起异步POST请求

|  |
| --- |
| 经典高阶错误：  Failed to load http://127.0.0.1/xz\_v1/data/user/register.php: Request header field Content-Type is not allowed by Access-Control-Allow-Headers *in preflight response*.  register.component.ts:23 用户注册异步请求失败  解释说明：客户端发了预取请求，而服务器给的预取响应中没有“客户端试探性提交的”内容。  解决方法：(1)要么服务器给出正确的预取响应 (2)要么客户端别发预取请求，直接发正式的请求即可 |
| **预取请求(preflight request)和预取响应(preflight response)：**  指在客户端与服务器进行正式的请求和响应之前，先进行的“试探性”的请求和响应。  一个典型的预取请求：  OPTIONS /xz\_v1/data/user/register.php HTTP/1.1  Access-Control-Request-Method: POST  Access-Control-Request-Headers: content-type  一个典型的预取预取响应：  HTTP/1.1 100 Continue  Access-Control-Allow-Methods: [get, post, put, delete]  Access-Control-Allow-Headers: [content-type, cache-control] |
| 跨域问题解决方案CORS相关的响应消息头部：  Access-Control-Allow-Origin: 服务器允许哪些客户端发起请求，如果客户端携带了身份认证信息，此处的值不允许使用\*  Access-Control-Allow-Methods: 服务器允许哪些请求方法  Access-Control-Allow-Headers: 服务器允许哪些请求头部  Access-Control-Allow-Credentials: true 服务器允许客户端提交身份认证信息 |

3.难点&面试题：跨域会话问题

服务器端会话的原理：



(1)客户端浏览器第一次请求时，请求消息中无Cookie:sessid；

(2)服务器在给出响应消息时，顺带在服务器端分配出Session空间并保存数据，响应消息头部中，使用Set-Cookie: sessid=xxx将会话编号发送给客户端

(3)浏览器将服务器端发来的会话编号保存在客户端

(4)浏览器再次发送请求时，在请求头部中会携带自己的会话编号Cookie:sessid=xxx

(5)服务器读取客户端提交的会话编号，识别出该编号之前存储的数据，输出响应消息，但不再发送新的SessionID

**结论**：服务器端Session可用于识别出不同的用户（用户识别），最关键字的点是客户端保存着Cookie:SESSID —— 身份认证信息(Credentials)

提示：异步跨域请求时，浏览器默认都不会向服务器发送任何Cookie——即身份认证信息不会发给服务器——Session即失效。

|  |
| --- |
| Angular中：  this.http.get(url, {  withCredentials: true  }); |
| jQuery中：  $.ajax(  url: url,  withCredentials: true,  success: fn  ) |
| 原生JS：  var xhr = new XMLHttpRequest();  xhr.withCredentials = true |

注意：客户端请求若“携带了身份认证信息”，则服务器端响应消息头“允许的跨域来源”就不允许用星号，例如：Access-Control-Allow-Origin: \*；必须明确指定哪些客户端允许跨域访问，例如：Access-Control-Allow-Origin: http://127.0.0.1:4200；同时还必须声明响应头：Access-Control-Allow-Credentials: true，以允许客户端请求携带身份认证信息。

课后任务：

继续完成“学子商城”Angular重构