JavaWeb学习笔记

## 说明

2022年3月31日23:08:48开始记录 目标9天学习完JavaWeb

再次学习javaweb的时候感慨颇多啊，想起了很多往事

## 前端基础

2022年3月31日23:10:25

### 知识

#### html总述

html是解释性语言而不是编译型语言

浏览器是容错的 会尽量解释

#### html

页面中由一对标签组成 <html> </html>

<html> 开始标签

</html> 结束标签

#### 3)title 网页的标题

#### 4)meta标签

可以在meta标签中设置编码格式

#### 5)<br/>标签

标签表示换行 br标签指的是单标签，单标签：开始和结束标签是同一个 斜杠放在单词后面

#### 6)p标签

表示段落标签

#### 7)img 标签

表示图片标签 单标签

src属性表示图片的路径

width 和 height表示大小

alt表示图片的提示

#### 8)路径问题

1 相对路径

2 绝对路径

#### 9)h1-h6标题标签

#### 10)列表标签 ：ul ol

- ol 是有序列表

start 表示从\*开始 type表示显示的类型: A a I i 1(default)

- ul 是无序列表

type disc(default),circle square

#### 11) u 下划线 b 粗体 i 斜体

#### 12）sub下标 sup是上标

#### 13) html中的实体 小于 &lt; 大于&gt;版权@copy; 可以百度

Html实体不用记 用到可以查

#### 14) span 不换行的

#### 15) a 表示超链接

href 链接的地址

target :

\_self:

\_blank:

\_parent:

\_top:

#### 16) div 层级结构 具有空间的概念

#### 17） 表格 table

行 tr

列 td

表头 th

table中有如下属性 已经被淘汰

- border 表格边框

- width 表格的宽度

- cellspacing 单元格间距

- cellpadding 单元格填充

tr align->center left right

只要和显示有关的都已经淘汰

rowspan 行合并

colspan 列合并

#### 18) 表单 from

#### 19) input

input type = "text" 表示文本框 其中name属性必须要指定 否则这个文本框数据不会发送给服务器

input type = "password" 密码框

input type = "radio" 表示单选按钮 需要指定name属性值保持一致才能有互斥的效果，可以通过属性checked设置默认选中的状态

input type = "checkbox" 表示复选框 name属性值建议保持一致，这样服务器获取属性值是一个数组，方便处理 以通过属性checked设置默认选中的状态

select 表示下拉列表 每一个选项是一个option 其中value属性是发送给服务器的值，selected表示默认选中的项

textarea 表示多行文本框 或者文本域 他的value值就是开始和结束标签之间的内容

input type = "submit" 表示提交按钮

input type = "reset" 表示重置按钮

input type = "button" 表示普通按钮

经过实验证明 name字段是必须的

<http://localhost:63342/day01/demo04.html?nickName=111&pwd=222&gender=female&hobby=java&hobby=python&hobby=cpp&star=%E5%8F%8C%E5%AD%90%E5%BA%A7&remark=%E5%A4%87%E6%B3%A8%EF%BC%9A34f3g5>

#### 20)frameset frame

frameset 表示页面框架 这个标签已经被淘汰 不需要掌握

frame 表示具体引用

#### 21)iframe

在一个标签中嵌入一个小的页面

#### 总结

1 html是解释性的标记语言 不区分大小写

2 html,head,title,meta,body,br,p,div,img,table,form,sup,sub,&nbsp;,ul,ol,tr,th,td,h1h6,input select

2.1

2.2 table tr th td

2.3 from(action = , method = "post") input type = "text,password,radio,checkbox,submit,button,reset" select textarea

### 代码

这里只给出我觉得比较重要的代码

|  |
| --- |
| <form action="demo01.html" method="get">  昵称：<input type="text" name="nickName"> <br/>  密码：<input type="password" name="pwd"> <br/>  性别：<input type="radio" name="gender" value="male" checked="checked"> 男  <input type="radio" name="gender" value="female"> 女<br/>  爱好：<input type="checkbox" name="hobby" value="java" checked> Java  <input type="checkbox" name="hobby" value="python" checked="checked"> Python  <input type="checkbox" name="hobby" value="cpp"> CPP <br/>  星座：<select name="star">  <option value="1">双子座</option>  <option value="2">白羊座</option>  <option value="3" selected>金牛座</option>  <option value="4">天蝎座</option>  <option value="5">天秤座</option>  <option value="6">射手座</option>  </select><br/>  备注：<textarea name="remark" rows="4" cols="25">备注：</textarea><br/>  <input type="submit" value="register">  <input type="reset" value="reset">  <input type="button" value="我很普通">  </form> |

### 重点

#### 1 From标签中的input字段等一定要带上name

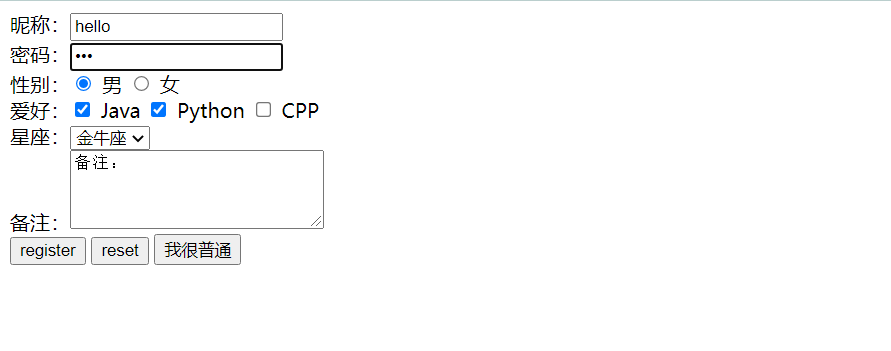
比较重要的是表单from标签

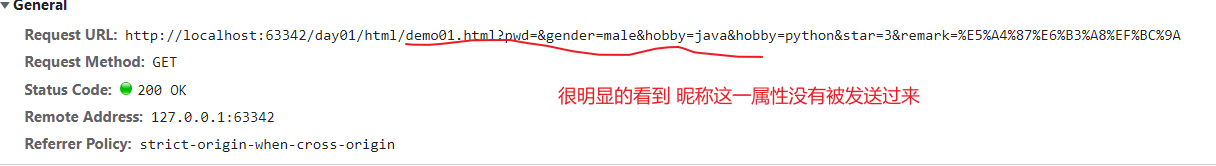
这里需要注意的是 name属性必须要指定 否则这个文本框数据不会发送给服务器

下面进行一个演示

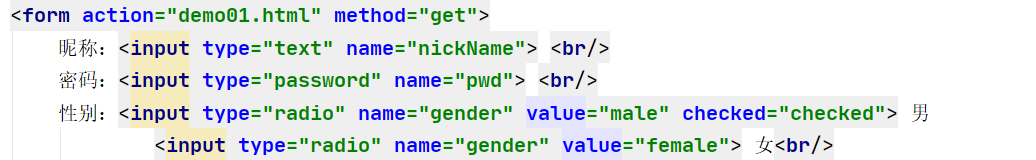


填入字段后发送数据

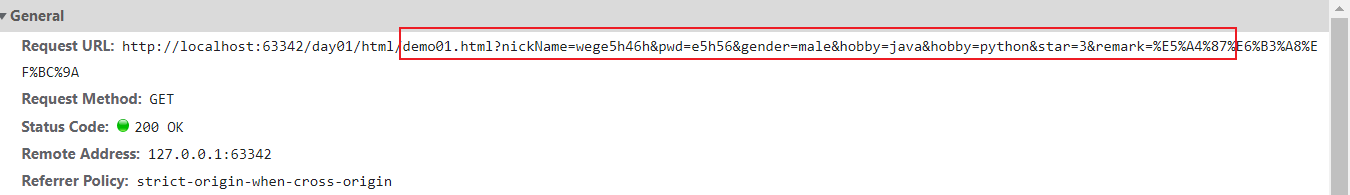




添加字段



重新提交 并且查看信息



很明显的可以看到数据已经被发送过来

注意另外一个地方就是 hobby这个字段，可以发现有多个值（复选）

#### 2 对于属性名和属性值一致的情况可以不写属性值

如图下面都可以省略，而且不会报错



#### 3 建议使用post方法

1 get方法是默认

2 get不安全

3 get传送的数据有限 URL 的最大长度是 2048 个字符



#### 4 不要在html中书写样式有关的属性

因为已经过时

### Css

时间

#### 很重要的内容

在html文件中书写css

<style type="text/css"></style>

引入外部的样式

<link rel="stylesheet" href="css/style.css">

### Js

在html文件中书写JavaScript

<script language="JavaScript">

引入外部的js文件

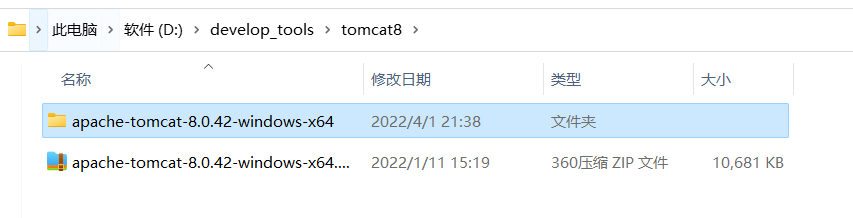
<script type="text/javascript" src="js/demo02.js"></script>

## 后端基础

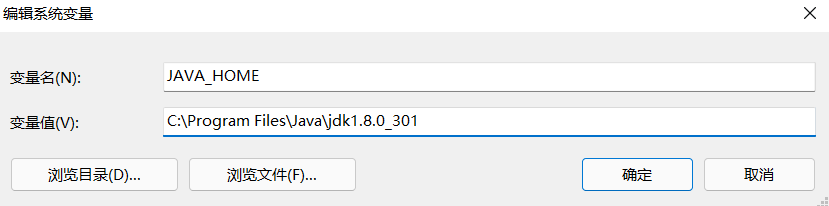
### 00 如何新建一个tomcat工程

#### Tomcat引入

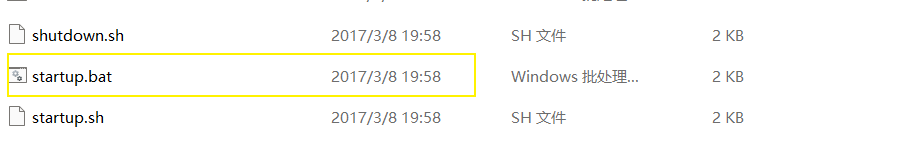
##### 1 解压tomcat包 无需安装



##### 2 配置JAVA\_HOME 对应到安装的bin目录下



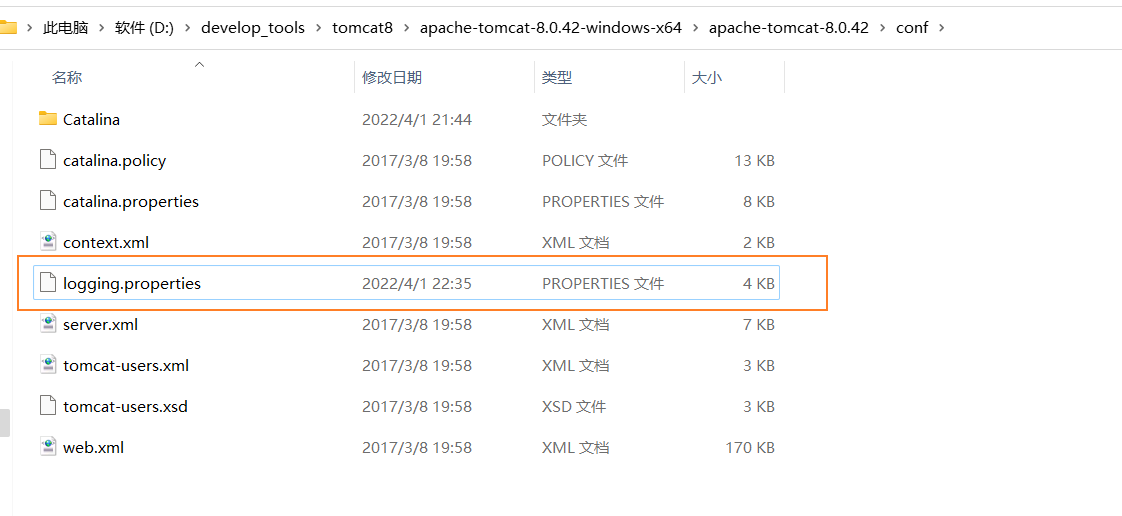
如果没有配置 双击 startup会一闪而过



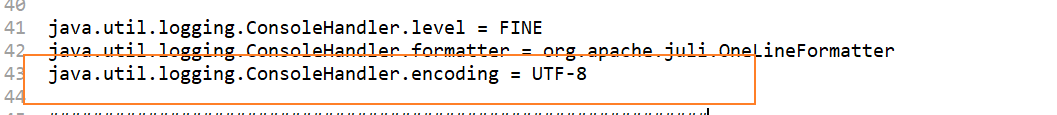
##### 3 说明tomcat的文件架构



##### 4 控制台输出乱码怎么办



双击打开



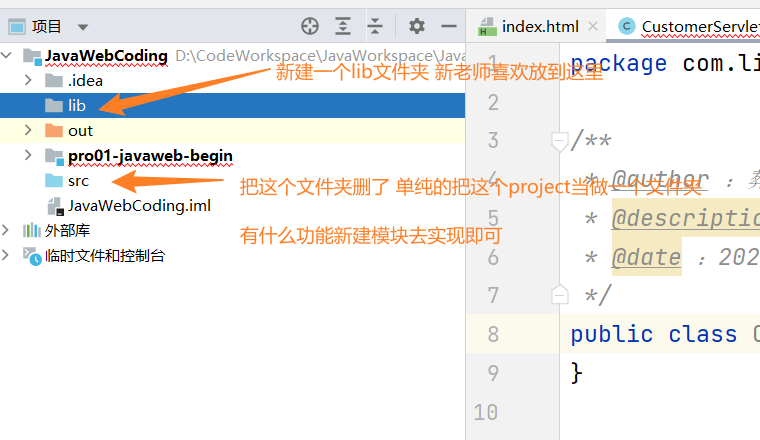
没有这一行就加上 如果UTF-8不行就GBK

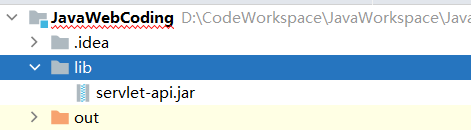
#### 在idea中创建一个tomcat工程

### 02 没有HttpServlet怎么办

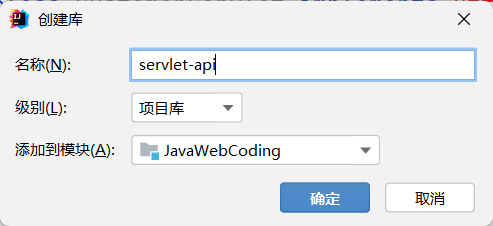




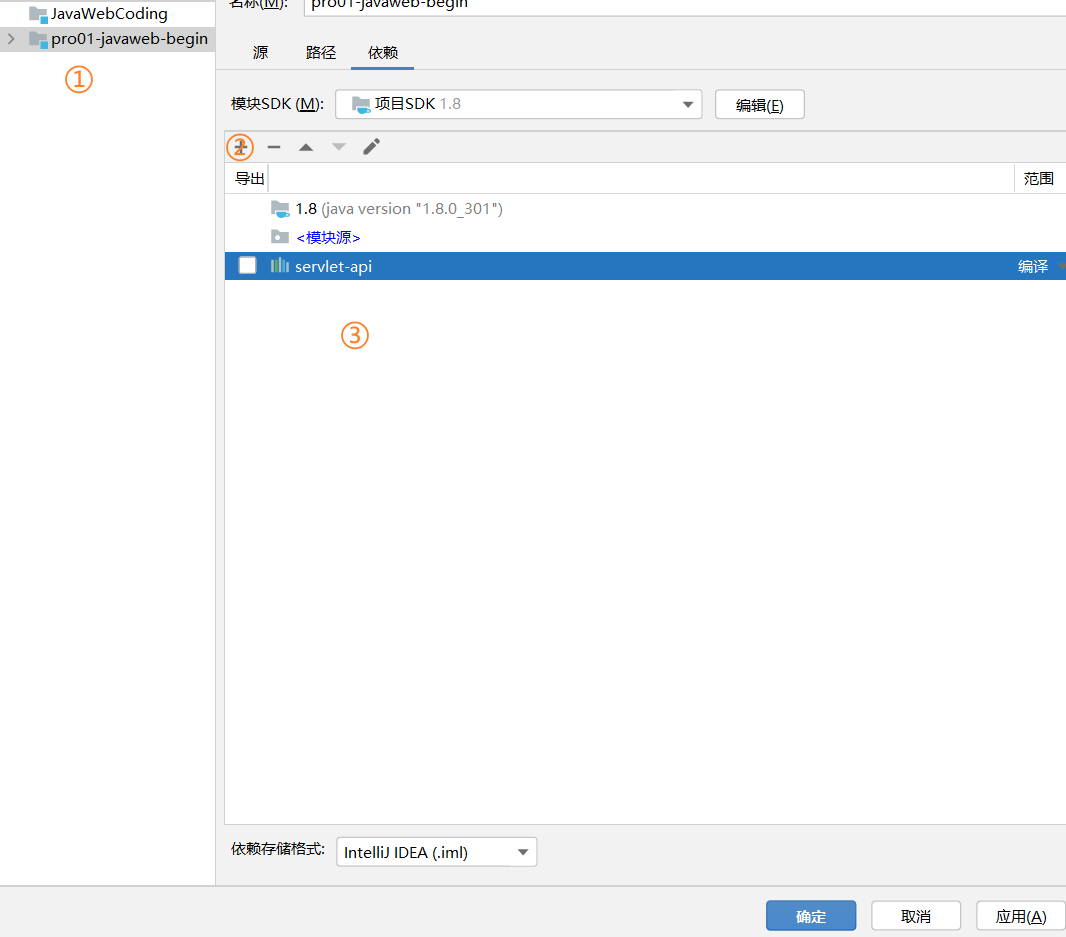


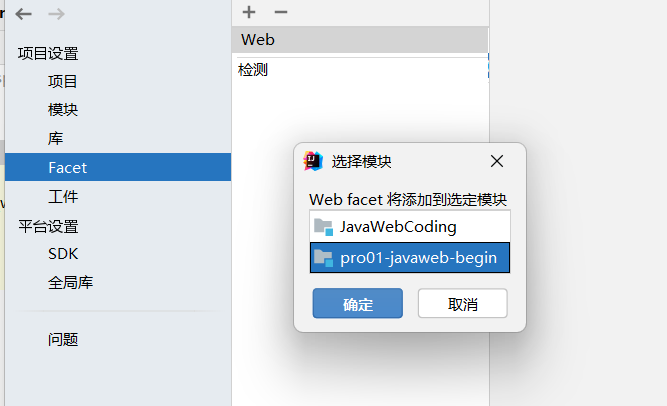


作为项目库添加

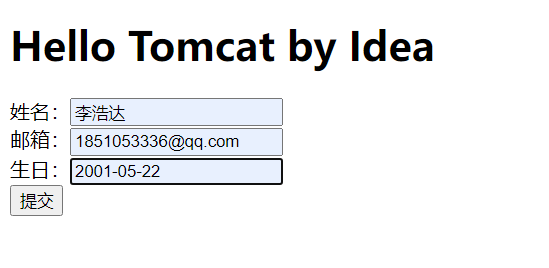


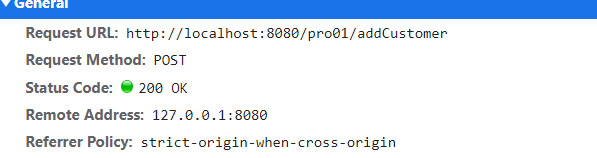
去模块下把jar包依赖导入

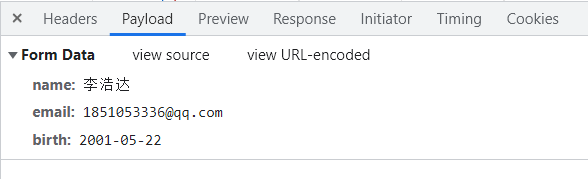


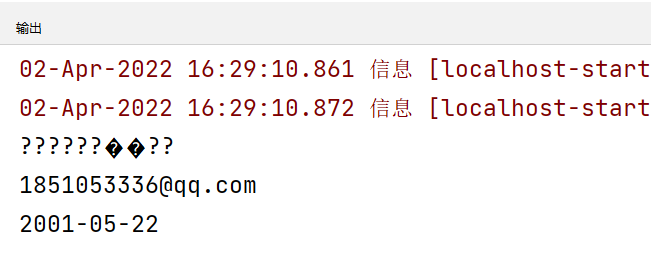


### 03 解决中文乱码



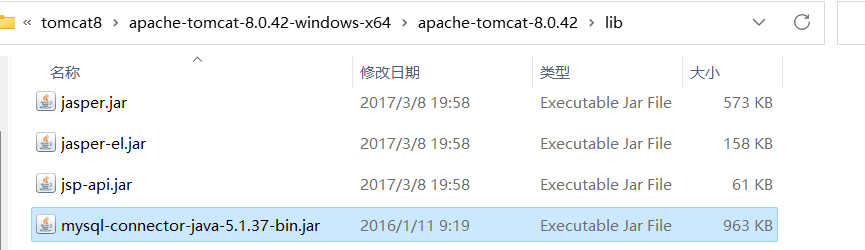




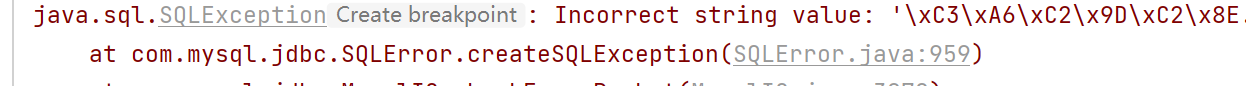


在控制台就已经乱码

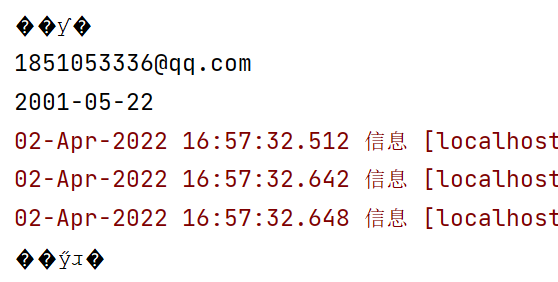
找不到



乱码甚至导致插入到数据库失败 现在就解决这个问题

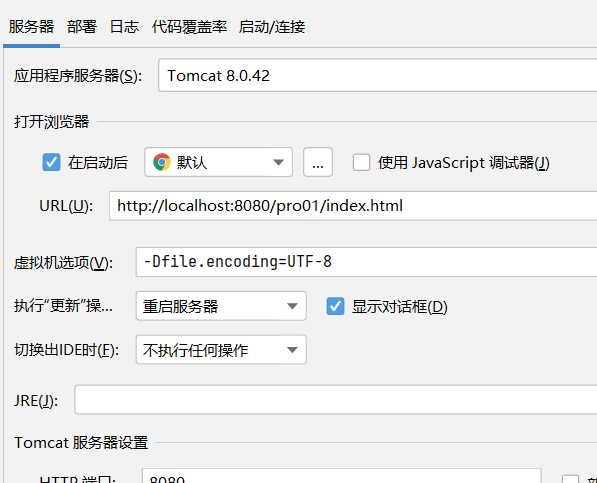


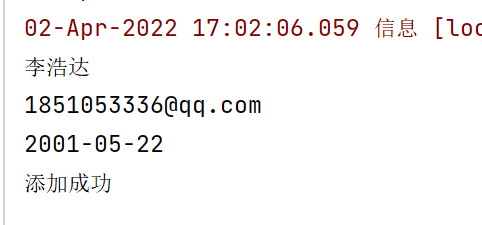
此时数据库不乱码 但是

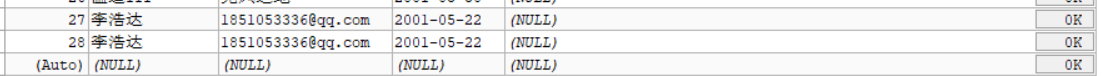




这时候调整配置

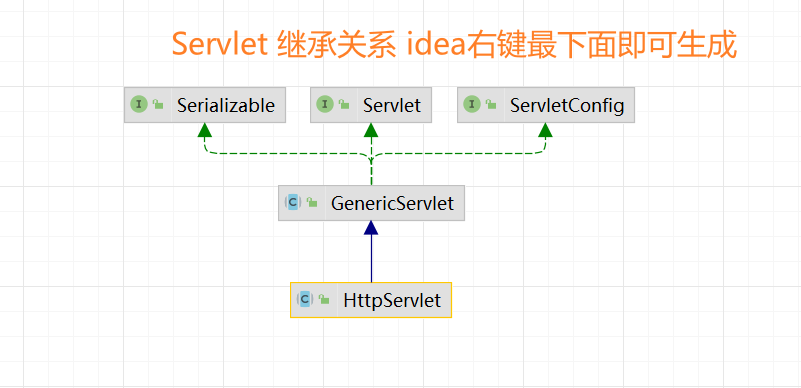






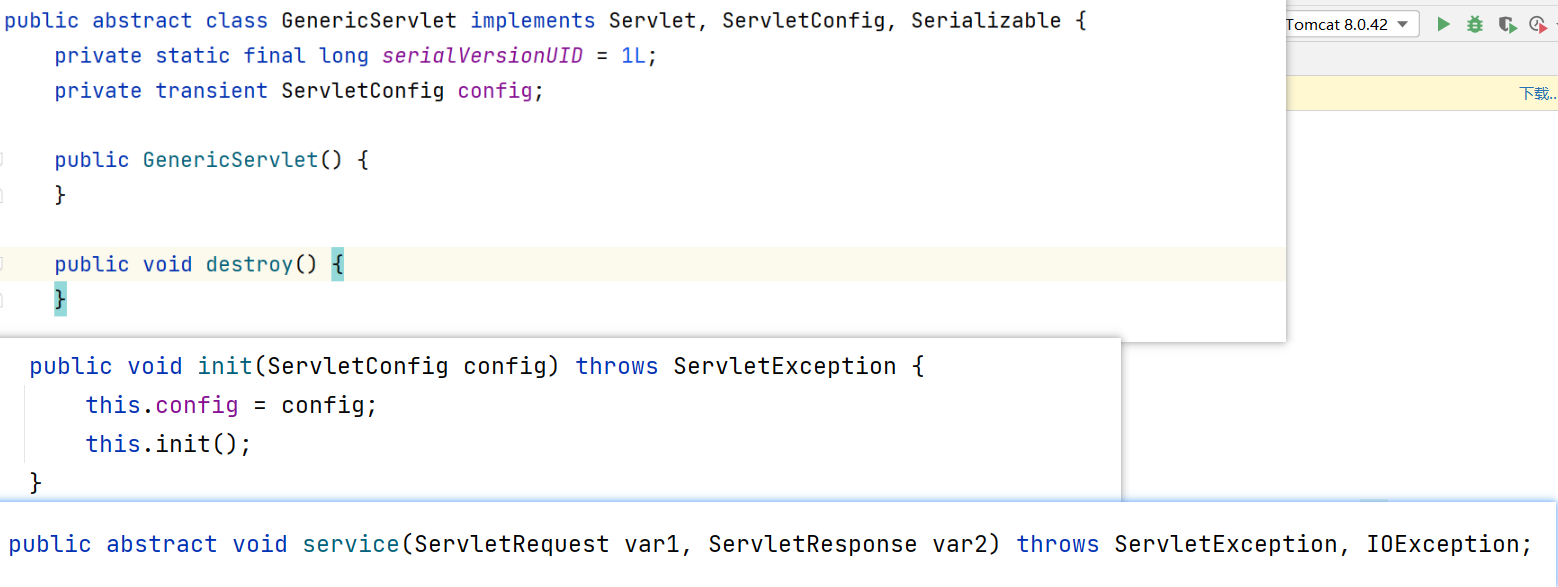
我只能说真tmd的一波三折

### 04Servlet继承关系





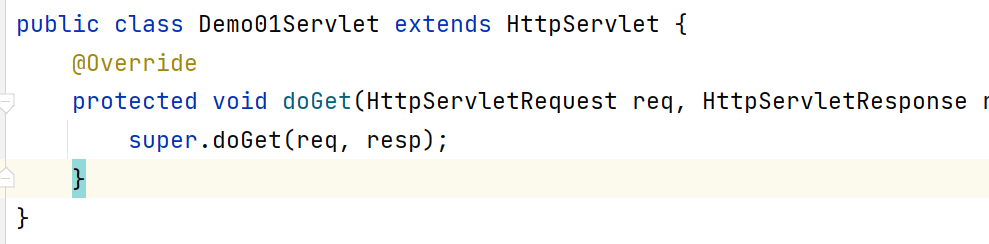
可以看到 GenericServlet并没有重写 service方法



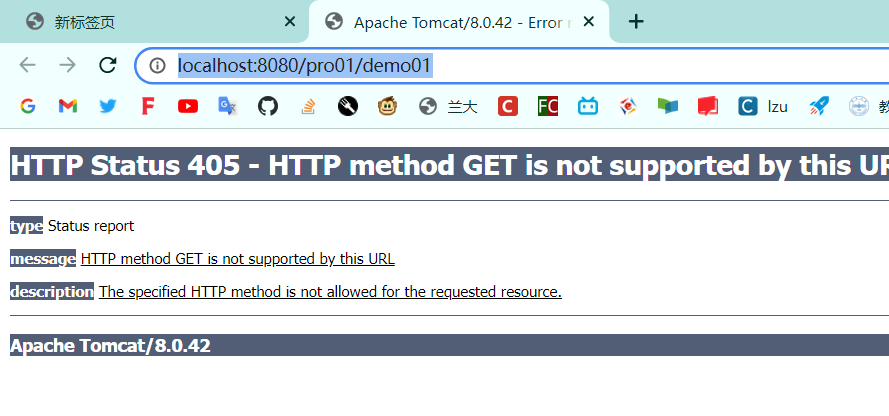




我们自定义一个servlet



通过浏览器直接访问的就是get方法 调用父类的方法 直接报异常



### 05Servlet的生命周期



第一次访问会创建（反射）并且初始化然后服务，之后的每次访问都只是服务

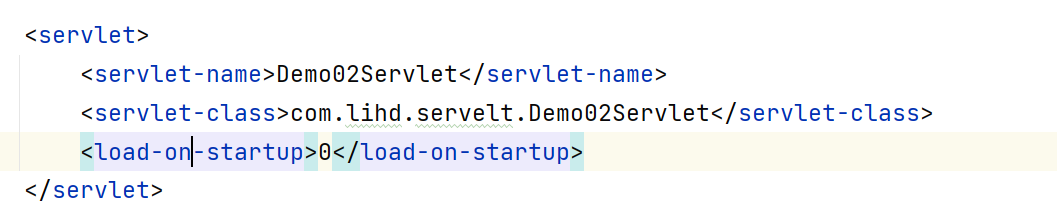


Tomcat停止服务时 销毁对象



这样做的好处是 服务器启动快 带来的问题是 第一访问资源时比较慢

可以在web.xml配置创建的时机 这个值越小 创建时机越靠前 没有负数







上面已经提到这个对象是通过反射创建的

于是我们把构造器的权限设置为private

(反射实例化newInstance要求有一个公共无参的构造器)

这样访问servlet时就会报错

另外还有一点是 ：servlet 是单例的 线程不安全的

因此不要在Servlet中设置属性，如果设置了属性不要根据属性值去做一些操作，也不要修改属性值

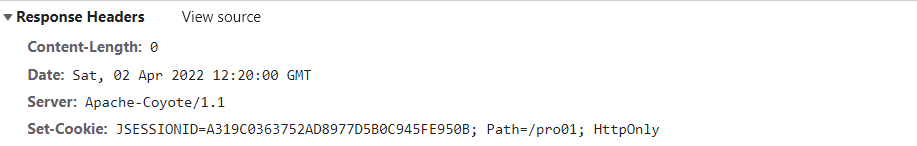
### 06 http协议

### 07session

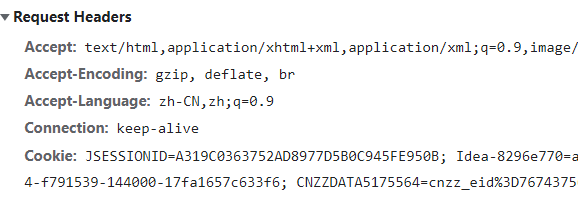
第一次获取session



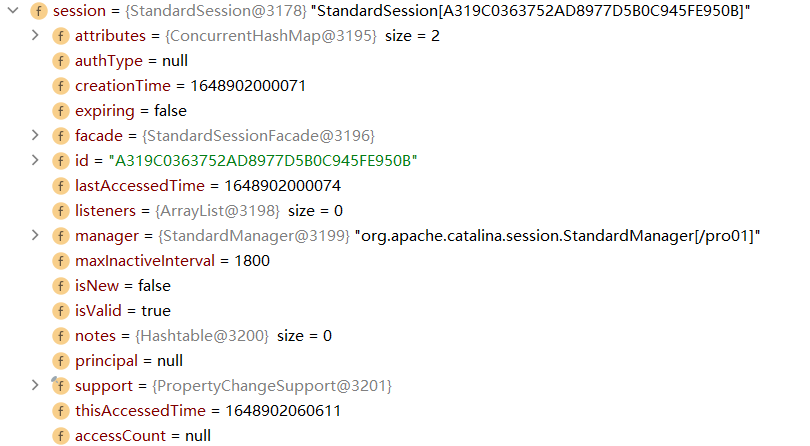
此时服务器会给浏览器setCookie 其中包含session

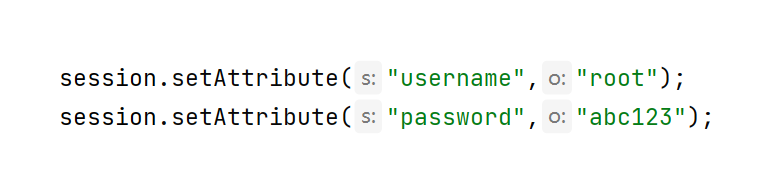


当第二次获取session时 浏览器会把这个sessionid发送给服务器



在idea调试中查看session的消息 如果是第一次isNew = true









浏览器 存放cookie 里面只会保存sessionId 并不存放session数据

服务器 存放session信息 在堆中，只有主动销毁和超时（默认1800秒）才会使session失效

但是关闭浏览器会丢失sessionID导致session看起来好像被删除了

session不会因为浏览器的关闭而删除。但是存有session ID的cookie的默认过期时间是会话级别。也就是用户关闭了浏览器，那么存储在客户端的session ID便会丢失，但是存储在服务器端的session数据并不会被立即删除。从客户端即浏览器看来，好像session被删除了一样（因为我们丢失了session ID，找不到原来的session数据了）。

Java中的session

sessionid是一个会话的key，浏览器第一次访问服务器会在服务器端生成一个session，有一个sessionid和它对应。tomcat生成的sessionid叫做jsessionid。

　　session在访问tomcat服务器HttpServletRequest的getSession(true)的时候创建，tomcat的ManagerBase类提供创建sessionid的方法：随机数+时间+jvmid。

　　存储在服务器的内存中，tomcat的StandardManager类将session存储在内存中，也可以持久化到file，数据库，memcache，redis等。客户端只保存sessionid到cookie中，而不会保存session，session销毁只能通过invalidate或超时，关掉浏览器并不会关闭session。

　　那么Session在何时创建呢？当然还是在服务器端程序运行的过程中创建的，不同语言实现的应用程序有不同创建Session的方法，而在Java中是通过调用HttpServletRequest的getSession方法（使用true作为参数）创建的。【４】在创建了Session的同时，服务器会为该Session生成唯一的Session id，而这个Session id在随后的请求中会被用来重新获得已经创建的Session；在Session被创建之后，就可以调用Session相关的方法往Session中增加内容了，而这些内容只会保存在服务器中，发到客户端的只有Session id；当客户端再次发送请求的时候，会将这个Session id带上，服务器接受到请求之后就会依据Session id找到相应的Session，从而再次使用之。

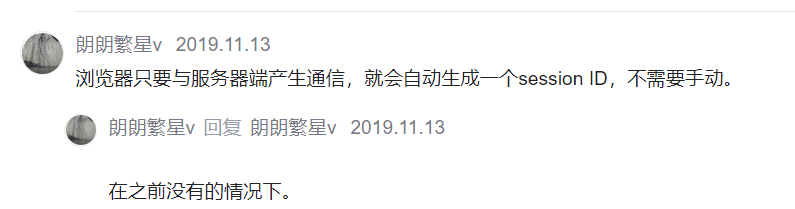
　　创建：sessionid第一次产生是在直到某server端程序调用 HttpServletRequest.getSession(true)这样的语句时才被创建。

　　删除：超时；程序调用HttpSession.invalidate()；程序关闭。

　　session存放在哪里：服务器端的内存中。【5】不过session可以通过特殊的方式做持久化管理（memcache，redis）。

　　session的id是从哪里来的，sessionID是如何使用的：当客户端第一次请求session对象时候，服务器会为客户端创建一个session，并将通过特殊算法算出一个session的ID，用来标识该session对象。

session会因为浏览器的关闭而删除吗？不会，session只会通过上面提到的方式去关闭。



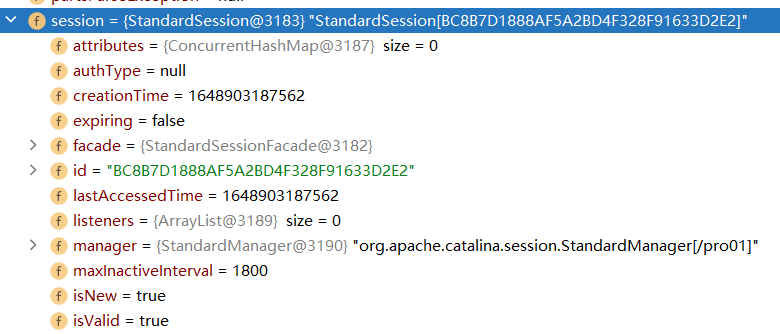
有个大佬这么说

主动调用方法才会创建一个session 至于session id的情况就不得而知了

我又去查了一下资料 结合对idea的断点调试

上面的话是错的，只有getSession时才会生成 访问jsp时也会生成，这是jsp导致的

对于后端来说确实是什么时候什么时候创建session

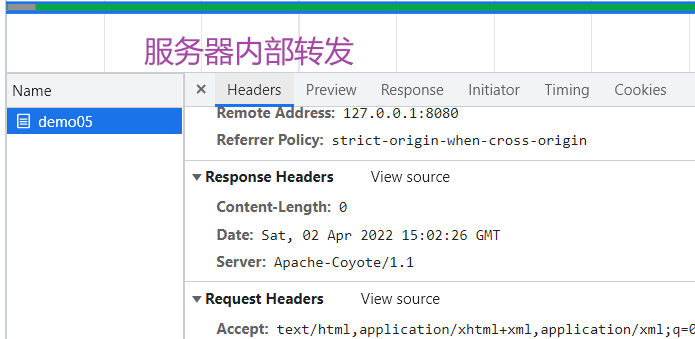


### 08 转发与重定向

对于服务器内部转发 地址栏没有变化 是一次请求（req.）

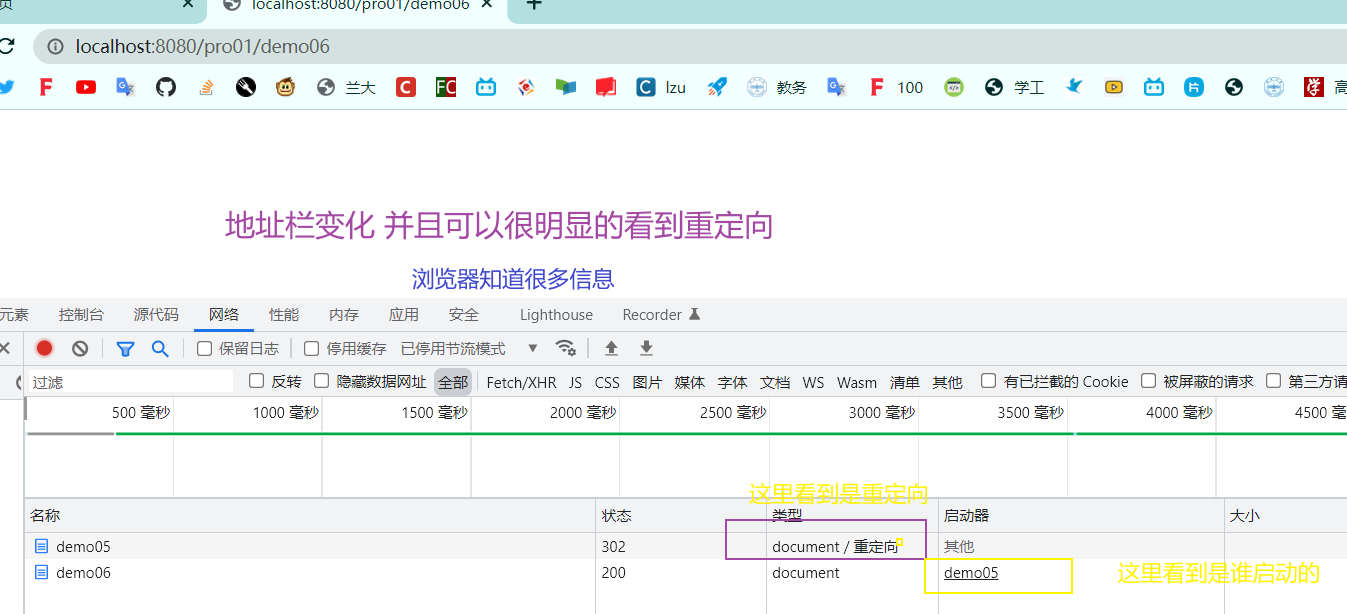






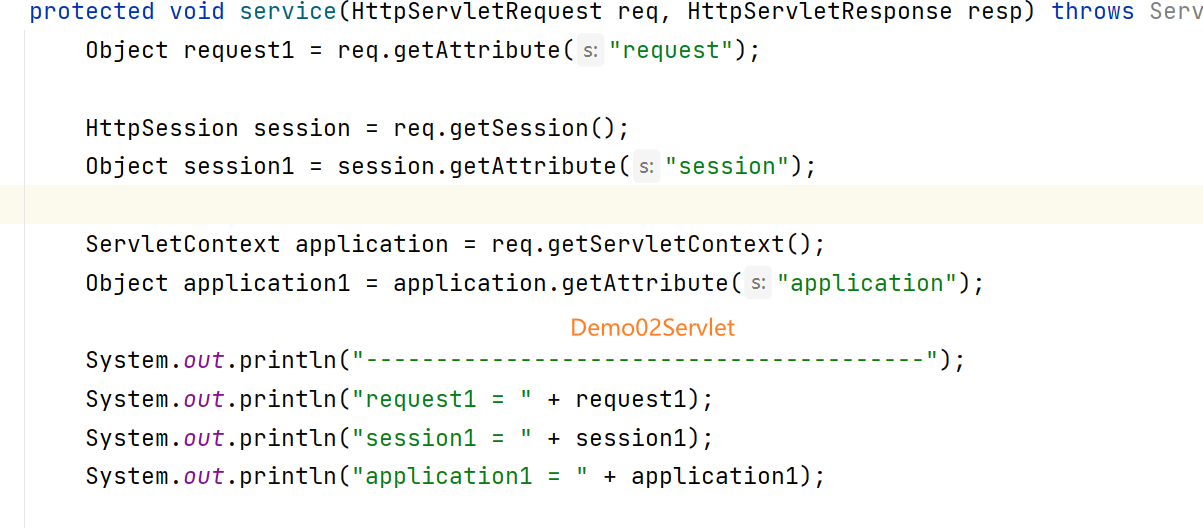
对于客户端重定向(resp)





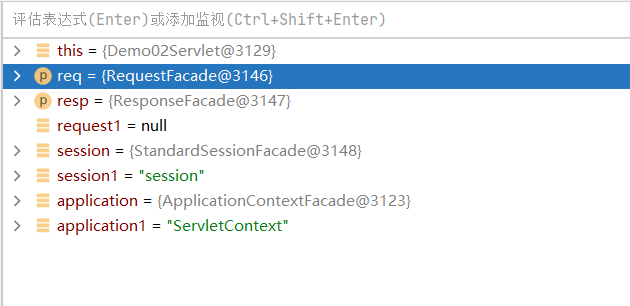
### 09 四种域对象



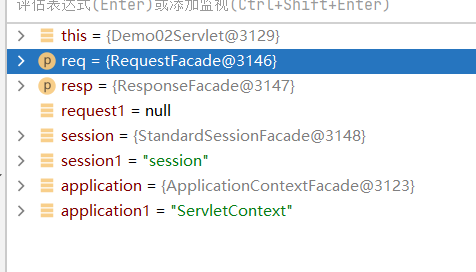


#### 对于浏览器重定向

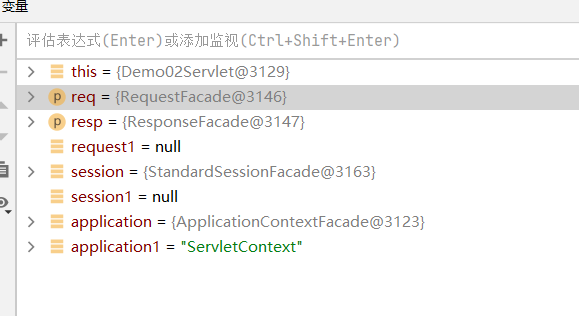
demo01



使用一个浏览器访问 demo02



更换浏览器访问 demo02



#### 对于服务器内部转发

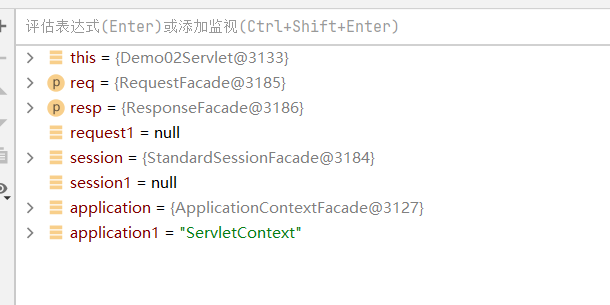
访问demo01



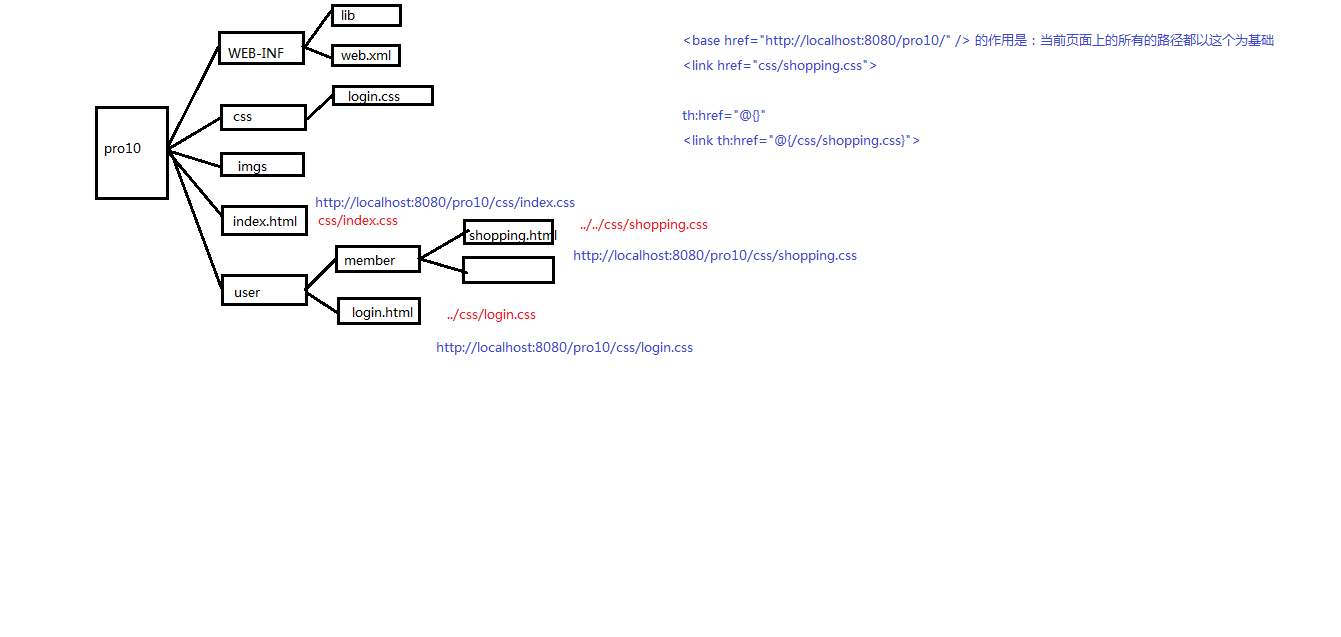
使用同一个浏览器访问 demo02



换一个浏览器 访问 demo02

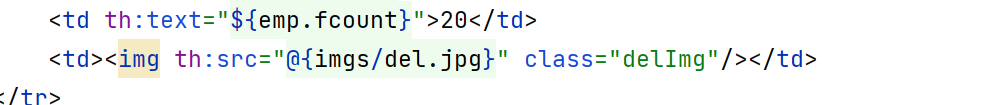


### 10 相对路径与绝对路径



对之前的代码进行修改 使用绝对路径





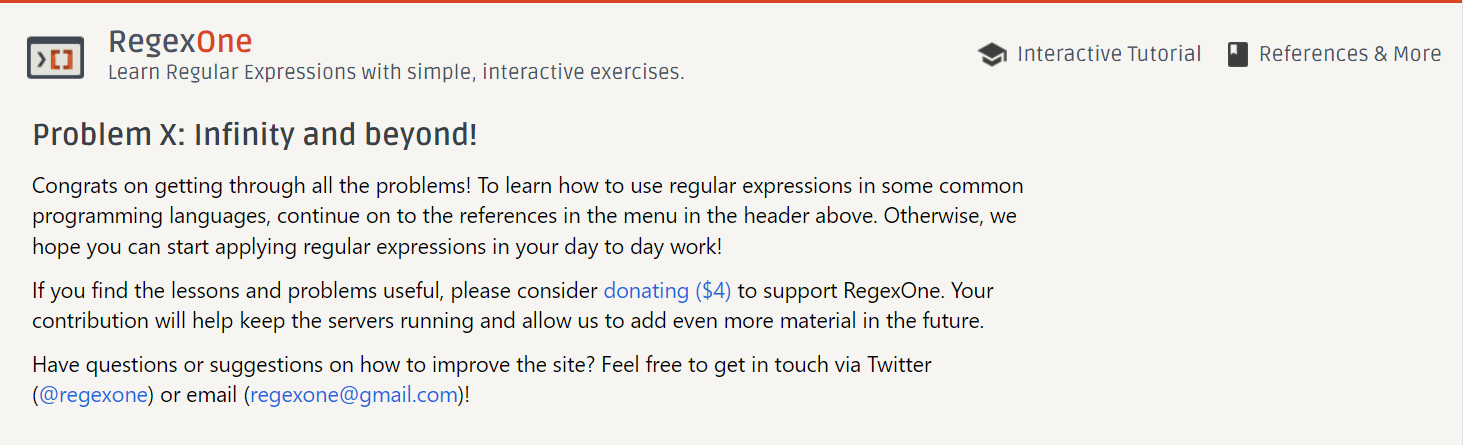
## 后端高级

## 前端高级

### JS EXP

今天（2022年4月9日21:52:54）虽然花了不少时间 但是看到这个页面还是很开心的，唯一的不足是我没有4美元

<https://regexone.com/problem>



# 总结

这里总结一些小错误吧

总之是一些很小很小的错误，但是这些错误会带来严重的后果！

很可能因为一个println 或者 print的不同 ，或者是多了一个单引号，

就有可能使你浪费3个小时 ，比这更难的是你会感觉自己很废物 会怀疑自己

因此结论如下

1 代码一定要细心，仔细比对。

2 多使用debug 以后就只用debug启动 包括浏览器的debug

3 相信自己 我就是太阳

忘了东西不要怕 多查一查 但是不是说忘了就忘了 一定要快速复习

1 json用：不用=

2 setContentType中字符不要加 单引号

3

从3月31日晚上开始看视频 4月10日看完 并且照着做了一遍

感觉比之前的要好

中间有三个晚上晚睡至3点 导致早上也起不来 要不然最少提前一天完成任务

感觉自己太赶了，有很多东西根本没有记住

今天看了看宋红康的总结 高手的是慢 都是扎实的，我突然就感觉有点破防

时间不多了 我还有很多事情没做 特别是leetcode剑指offer

我是不能慢下来的呀 我必须前进

