

【手把手】JavaWeb 入门级项目实战 -- 文章发布系统（第七节）

01

国庆长假终于结束了，博主想到以往这个时候，自己就已经回到学校，和一群基友扯扯犊子，顺便吹吹牛，好不快活，可惜这种生活不会再有了。虽说如此，但是参加了工作以后，毕竟有更多的时间去做自己想做的事情，没有了学校里的那么多约束，也不再需要为了考试忙活个半天，想来也是不错的。

好的，再次回到这个系列。

不知不觉，我发现自己在简书已经写了好几万字了。当我无聊的时候，就会去看看自己之前写的文章，然后感到很奇怪，似乎那些东西根本不是自己写的一样。我仿佛是在看别人的文章，好长时间我都有这种感觉。

这真的是挺奇怪的，不过我转念又一想，当我看自己几个月以前的代码的时候，也经常有看不懂的情况，于是我也就释然了。

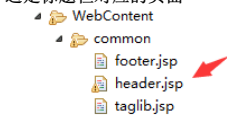
管他呢，先往下写吧。

上次登录工作做得差不多了，当我们在登录页面输入账号和密码，后台会进行一系列的验证，如果验证无误，就跳转到首页。

现在我们对首页进行一个小小的调整，就是说，如果登录成功了，右上角的登录图标就换成登录人的名字。



这是标题栏对应的页面 —— header.jsp



代码

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<div class="header">
    <div class='logo'>原创文字</div>
    <ul>
        <li class='first'><a href="index.jsp">首页</a></li>
        <li class='item'><a href="javascript:void(0)">原创故事</a></li>
        <li class='item'><a href="javascript:void(0)">热门专题</li>
        <li class='item'><a href="javascript:void(0)">欣赏美文</li>
        <li class='item'><a href="javascript:void(0)">赞助</a></li>
    </ul>

    <div class='login'>
        <span><a href="login.jsp">登录</a></span>
        <span>|</span>
        <span><a href="javascript:void(0)">注册</a></span>
    </div>
</div>
```

02

在上一节，我们判断当用户名和密码全部匹配的时候，就把user对象，还有username放到session中。

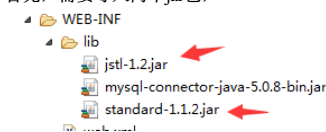
现在，要实现标题栏判断的功能，我们可能会这么想，就是能不能在header.jsp里面做一个简单的if判断，如果username存在，那么就把 登录|注册 这个字样给换掉。如果不存在，就维持原状。

对的，毫无疑问，这个思路肯定是可以的。

俗话说得好，只要思想不滑坡，办法总比困难多。写程序也是一样，怕就怕没思路，有了思路，啥都好说。实在不行，就百度呗。

嗯，我百度了一下（好吧，其实我也不太会。。。），解决了一点点小问题。现在开始进行代码的实现。

首先，需要导入两个jar包，



分别是 **jstl.jar** 和 **standard-1.1.2.jar**

接着，在taglib.jsp中引入这个标签库

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

这是JSTL的一个核心标签库，有了它，我们就可以为所欲为，哦不，可以在JSP页面上进行一些简单的控制了，比如说逻辑判断。

于是，代码就成了这样。

```
<div class='login'>
    <c:choose>
        <c:when test="${empty sessionScope.username}">
            <span><a href="login.jsp">登录</a></span>
            <span>|</span>
            <span><a href="javascript:void(0)">注册</a></span>
        </c:when>
        <c:otherwise>
            <span><a href="javascript:void(0)">欢迎您， ${sessionScope.username}</a></span>
            <span> | </span>
            <span><a href="javascript:void(0)">登出</a></span>
        </c:otherwise>
    </c:choose>
</div>
```

其实引入标签库的时候，我有点疑惑，因为我引入的明显是一个网址，也就是说，这个标签库应该是远程的。既然是远程的，为什么我还必须要添加两个jar包不可呢？可是如果我不添加的话，编译又报错了。难道和tomcat的版本有关系？我用的是tomcat7.0。额，如果有大神知道的话，求告知一下，这个我真的不太清楚为什么。

不管了，先继续往下写吧。

好的，让我们测试一下。



用户名: 张三
密码: 123456

点击登录



成功了。

JSP作为一个被广泛使用的模板引擎，有着很多很多的标签库，也有不少第三方的标签库。当然，我们也可以自己编写标签，这个有时间的话，我也打算来写一写，自定义标签还是很有趣的。

03

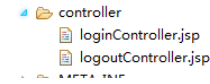
有了登录，肯定还需要有登出功能，总不可能每次都让用户关掉浏览器吧。

登出的话，我们只需要把session中的信息都删掉就行了，然后自动跳转到登录页面。

登出按钮：

```
<span><a href="{basePath}/controller/logoutController.jsp">登出</a></span>
```

我们再把controller文件夹内新建一个logoutController:



登出操作比较简单，我就直接上代码了：

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
    //注销
    session.removeAttribute("username");
    session.invalidate();
    String path = request.getContextPath();
    //获取端口
    int port = request.getServerPort();
    String basePath = null;
    if(port==80){
        basePath = request.getScheme()+ "://" + request.getServerName() + path;
    }else{
        basePath = request.getScheme()+ "://" + request.getServerName() + ":" + request.getServerPort() + path;
    }
    response.sendRedirect(basePath + "/login.jsp");
%>
```

注销后，页面就自动跳转到登录页面。

然后，即便你再次访问首页，右上角也不会显示用户信息了。



登陆 | 注册

因为我们已经将session中的用户信息给删除了。

04

接下来，我们继续画页面，之前首页的内容区一直没有做，现在我们来画一下吧。

既然是文章发布系统，主角就是各种文章，对不对？好的，文章又分很多种类，比如情感类，励志类，科技类等。于是，我们需要在首页对不同的文章进行分类展示。当然，只会展示前几条数据。

index.jsp

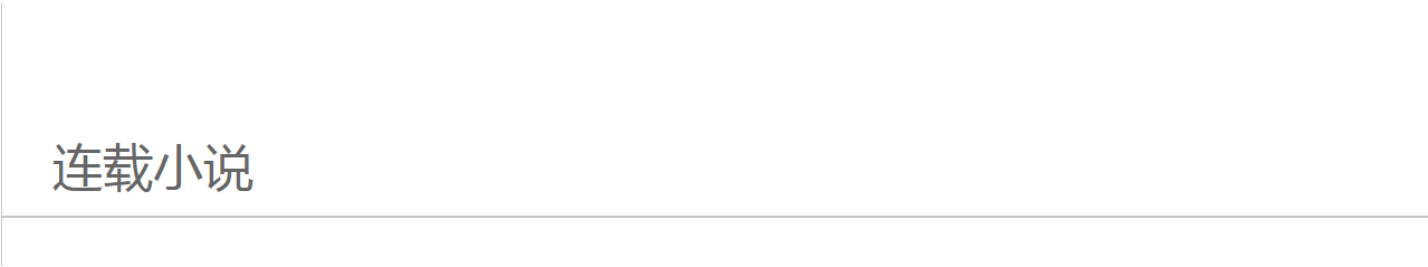
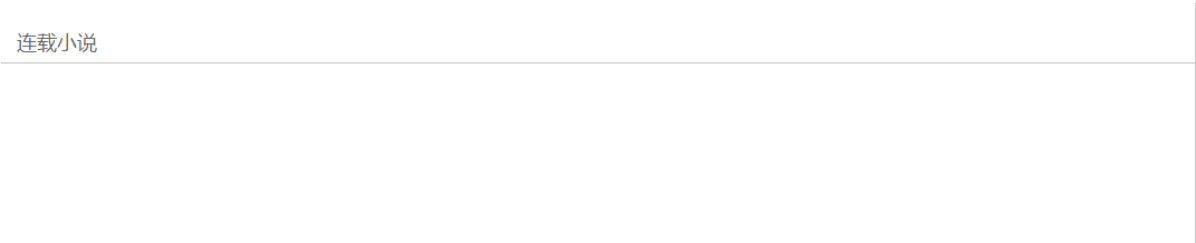
内容区代码：

```
<div class='h600' style='border:1px solid #ccc'>
  <div class='category'>
    <div class='title'>连载小说</div>
    <ul class='items'>
      <li class='item'></li>
      <li class='item'></li>
      <li class='item'></li>
    </ul>
  </div>
</div>
```

category有类型，种类的意思。

css

```
.category .title {
  margin-bottom: 10px;
  margin-top: 30px;
  border-bottom: 1px solid #cac5c5;
  height: 30px;
  text-indent:1em;
  font-size:18px;
  color:#666;
}
```

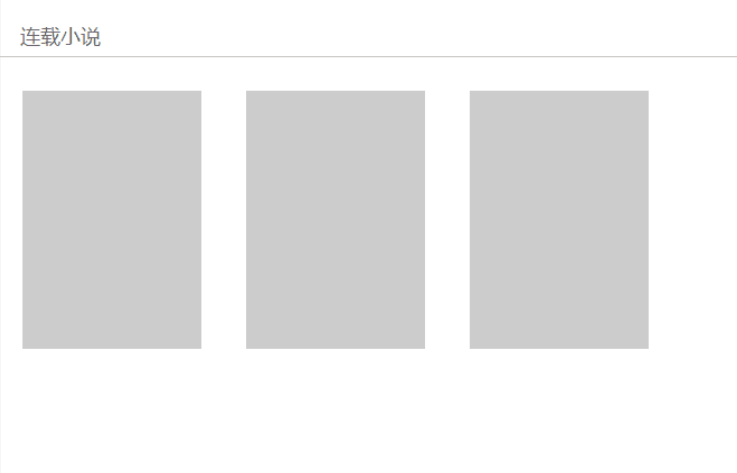


OK，大概有一个样子了。

接下来，我们给每一个item设置一个宽高，背景色，再加上一个左浮动。

```
.category .items .item {
  width: 160px;
  height: 231px;
  background: #ccc;
  margin: 20px;
  float: left;
}
```

就成了这样：



因为每一个item都是左浮动，所以别忘了清除浮动。

```
<div style='border:1px solid #ccc'>
  <br/><br/>
  <div class='category'>
    <div class='title'>连载小说</div>
    <ul class='items'>
      <li class='item'></li>
      <li class='item'></li>
      <li class='item'></li>
      <div style='clear:both'></div>
    </ul>
  </div>

  <div class='category'>
    <div class='title'>编程代码类</div>
    <ul class='items'>
      <li class='item'></li>
      <li class='item'></li>
      <div style='clear:both'></div>
    </ul>
  </div>
</div>
```

基于这样的思路，我们现在可以把页面模型搭出来了。

html

```
<!-- 内容区域 -->
<div style='border:1px solid #ccc'>
  <br/><br/>
  <div class='category'>
    <div class='title'>连载小说</div>
    <ul class='items'>
      <li class='item'></li>
      <li class='item'></li>
      <li class='item'></li>
      <div style='clear:both'></div>
    </ul>
  </div>

  <div class='category'>
    <div class='title'>编程代码类</div>
    <ul class='items'>
      <li class='item'></li>
      <li class='item'></li>
      <div style='clear:both'></div>
    </ul>
  </div>
</div>
```

css

```
.category {
  margin-top: 10px;
  margin-bottom:20px;
}

.category .title {
  margin-bottom: 10px;
  border-bottom: 1px solid #cac5c5;
  height: 30px;
  text-indent:1em;
  font-size:18px;
  color:#666;
```

```
}
.category .items {
  margin-left:10px;
}
.category .items .item {
  width: 230px;
  height: 320px;
  background: #ccc;
  margin: 20px;
  float: left;
  margin-right:20px;
  cursor:pointer;
}
```

现在的首页大概就是这个样子：

