

今日内容

1. JSP:

1. 指令
 2. 注释
 3. 内置对象
2. MVC开发模式
 3. EL表达式
 4. JSTL标签
 5. 三层架构

JSP:

指令

- 作用：用于配置JSP页面，导入资源文件
- 格式：

```
<%@ 指令名称 属性名1=属性值1 属性名2=属性值2 ... %>
```

- 分类：

page：配置JSP页面的

- contentType：等同于response.setContentType()
 1. 设置响应体的mime类型以及字符集
 2. 设置当前jsp页面的编码（只能是高级的IDE才能生效，如果使用低级工具，则需要设置pageEncoding属性设置当前页面的字符集）
- import：导包
- errorPage：当前页面发生异常后，会自动跳转到指定的错误页面
- isErrorPage：标识当前也是是否是错误页面。
- true：是，可以使用内置对象exception
- false：否。默认值。不可以使用内置对象exception

<%--	
page contentType="text/html; charset=UTF-8"	设置浏览器解析时使用的编码
pageEncoding="utf-8"	设置本jsp界面使用的字符集
language="java"	在这个jsp上使用的语言
buffer="8kb"	设置out输出流可以使用的, 默认为8kb
import="com"	导包
errorPage="index.jsp"	页面发生错误之后自动跳转到指定的界面
isErrorPage="true"	(true/false) 标注本界面是否是一个错误界面
面, 如果是错误界面, 那么可以使用exception.getMessage() 来获取	
--%>	

include

页面包含的。导入页面的资源文件

```
<%@include file="top.jsp"%>
```

使用这个指令就可以把之前的界面引进来, 相当于重新写了一遍引进来的界面

taglib

导入资源, 导入标签库, 这这里面一开始只能使用基本的html标签, 不过引进来之后就可以使用了

- <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
- prefix: 前缀, 自定义的

注释:

html注释:

<!-- -->: 只能注释html代码片段

jsp注释: 推荐使用

<%-- --%>: 可以注释所有

内置对象

- 在jsp页面中不需要创建, 直接使用的对象
- 一共有9个:

变量名	真实类型	作用
pageContext	PageContext	当前页面共享数据, 还可以获取其他八个内置对象
request	HttpServletRequest	一次请求访问的多个资源(转发)

变量名	真实类型	作用
session	HttpSession	一次会话的多个请求间
application	ServletContext	所有用户间共享数据
response	Response	响应对象
page	Object	当前页面(Servlet)的对象 this
out	JspWriter	输出对象，数据输出到页面上
config	ServletConfig	Servlet的配置对象
exception	Throwable	异常对象

MVC：开发模式

jsp演变历史

1. 早期只有servlet，只能使用response输出标签数据，非常麻烦
2. 后来又jsp，简化了Servlet的开发，如果过度使用jsp，在jsp中即写大量的java代码，又写html表，造成难于维护，难于分工协作
3. 再后来，java的web开发，借鉴mvc开发模式，使得程序的设计更加合理性

MVC：

M：Model，模型。JavaBean

- 完成具体的业务操作，如：查询数据库，封装对象

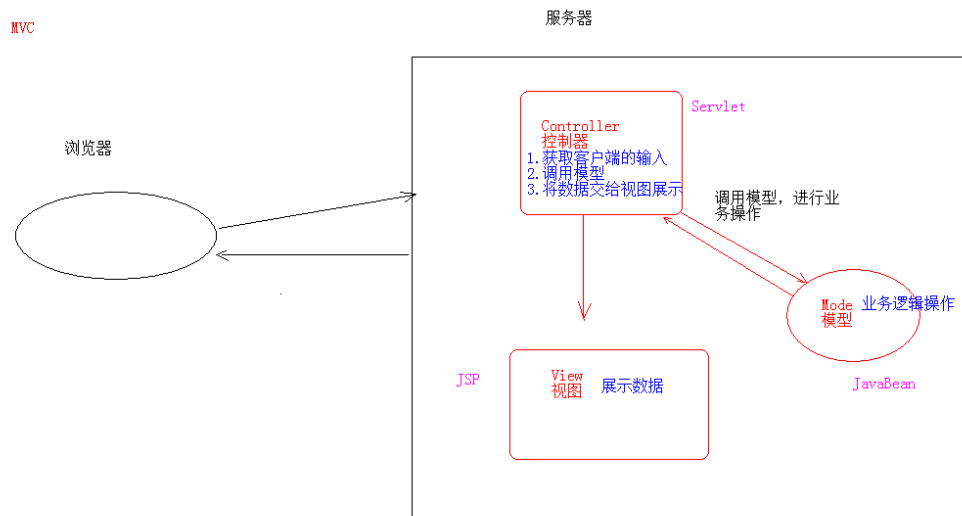
V：View，视图。JSP

- 展示数据

C：Controller，控制器。Servlet

- 获取用户的输入
- 调用模型
- 将数据交给视图进行展示
- 优缺点：
 1. 优点：
 1. 耦合性低，方便维护，可以利于分工协作
 2. 重用性高
 2. 缺点：

1. 使得项目架构变得复杂，对开发人员要求高



EL表达式

概念: Expression Language 表达式语言

作用: 替换和简化jsp页面中java代码的编写

语法: `${表达式}`

注意:

- jsp默认支持el表达式的。如果要忽略el表达式
 - 设置jsp中page指令中: `isELIgnored="true"` 忽略当前jsp页面中所有的el表达式
 - `\${表达式}` (反斜线+`${表达式}`): 忽略当前这个el表达式

使用:

运算:

- 运算符:
 - 算数运算符: `+`, `-`, `*`, `/`(div), `%`(mod)
 - 注意, 除法出来的结果是小数, 不再是整数了
 - 比较运算符: `>` `<` `>=` `<=` `==` `!=`
 - 逻辑运算符: `&&`(and) `||`(or) `!`(not)
 - 空运算符: `empty`
 - 功能: 用于判断字符串、集合、数组对象是否为null或者长度是否为0
 - `${empty list}`: 判断字符串、集合、数组对象是否为null或者长度为0
 - `${not empty str}`: 表示判断字符串、集合、数组对象是否不为null 并且 长度>0

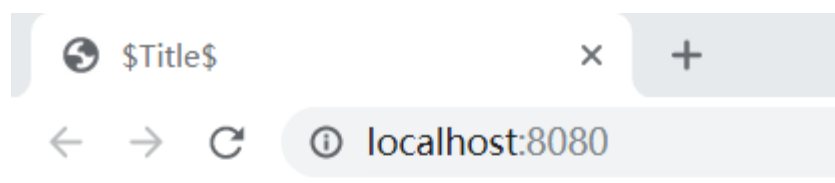
```
<%@page contentType="text/html; charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    <h3>算数运算符</h3>
    ${3+4}<br>
    ${3/4}<br>
    ${3%4}<br>

    <h3>比较运算符</h3>
    ${3==4}<br>

    <h3>逻辑运算符</h3>
    ${3>4} && 3<4><br>
    ${3>4} and 3<4><br>

  </body>
</html>
```



算数运算符

7

0.75

3

比较运算符

false

逻辑运算符

false

false

el表达式只能从域对象中获取值

语法:

1. `${域名.键名}`: 从指定域中获取指定键的值
 - 域名称:
 1. `pageScope --> pageContext`
 2. `requestScope --> request`
 3. `sessionScope --> session`
 4. `applicationScope --> application (ServletContext)`
 - 举例: 在`request`域中存储了`name=张三`
 - 获取: `${requestScope.name}`, 如果找不到, 那么会使用空字符串来展示, 不会打乱布局

```
<%@page contentType="text/html; charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    <%
      request.setAttribute("name", "张三");
      session.setAttribute("age", "23");
    %>

    ${requestScope.name}
    ${sessionScope.age}

  </body>
</html>
```

张三 23

2. `${键名}`: 表示依次从最小的域中查找是否有该键对应的值, 直到找到为止, 找到之后就不会向下寻找了

寻找的顺序为: `pageContext-->request-->session-->ServletContext`

```
<%@page contentType="text/html; charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
  <head>
    <title>${Title}</title>
```

```

</head>
<body>

<%
    session.setAttribute("name", "李四");
    request.setAttribute("name", "张三");
    session.setAttribute("age", "23");
%>

    ${name}
    ${age}

</body>
</html>

```

张三 23

获取对象、List集合、Map集合的值

1. 对象：\${域名称.键名.属性名}
- 本质上会去调用对象的getter方法

```

<%@ page import="cn.User.User" %>
<%@ page import="java.util.Date" %>

<%@page contentType="text/html;charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
    <head>
        <title>${Title}</title>
    </head>
    <body>

        <%
            User user = new User();
            user.setName("张三");
            user.setAge(23);
            user.setBirthday(new Date());

            request.setAttribute("user", user);
        %>

        ${requestScope.user}<br> <!--这样会获取到hashCode值:
cn.User.User@7dd110b1-->
        <!--
            我们一般情况下不需要这个，那么如何获取值？
            通过setter或者getter方法获取，具体步骤为：

```


1. 去掉get或者set
2. 将剩余字母变小写, 变为属性
3. 通过名称.属性来获取, 如: `${requestScope.user.birthday}`

如:

```
getName() --> Name --> name
```

注意:

1. 属性虽然看着和成员变量很相像, 但其实并不是成员变量, 这个换掉也可以

如: `getName(){ return age; }`

在这里, 成员变量为age, 但是属性却是name, 可见不一样, 虽然大部分都一样罢了

2. 既然是使用getter和setter方法调用的, 那么将来如果缺少getter和setter方法就会报错

```
--%>
    ${requestScope.user.birthday} <%--Thu May 16 17:32:10 CST 2019--%>
</body>
</html>
```

2. List集合: `${域名.键名[索引]}`

3. Map集合:

- `${域名.键名.key名称}`
- `${域名.键名["key名称"]}`

```
<%@ page import="cn.User.User" %>
<%@ page import="java.util.*" %>

<%@page contentType="text/html; charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
    <head>
        <title>${Title}</title>
    </head>
    <body>

        <%
            User user = new User();
            user.setName("张三");
            user.setAge(23);
            user.setBirthday(new Date());

            List list = new ArrayList<>();
            list.add("LIST");
            list.add(user);

            Map map = new HashMap<>();
            map.put("MAP", "HashMap");
            map.put("USER", user);
        %>
    </body>
</html>
```

```

        request.setAttribute("List", list);
        request.setAttribute("Map", map);
    %>

    ${List}<br>    <!--直接获取List, 当然是hashCode--%>
    ${List[0]}<br>    <!--获取list集合添加进去的第一个元素--%>
    ${List[1]}<br>    <!--获取list集合添加进去的第二个元素--%>
    ${List[1].name}<br>    <!--获取list集合添加进去的第二个元素的name值, 注意要有get方法才可以--%>

    ${Map}<br>
    ${Map["MAP"]}<br>
    ${Map.MAP}<br>    <!--map集合可以直接点出来, 当然也可以使用像上面的方法--%>
    ${Map.USER}<br>
    ${Map.USER.age}<br>

</body>
</html>

```

- empty

```

<%@ page import="cn.User.User" %>
<%@ page import="java.util.*" %>

<%@page contentType="text/html;charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
    <head>
        <title>${Title}</title>
    </head>
    <body>

        <%

            List list = new ArrayList<>();
            list.add("LIST");

            Map map = new HashMap<>();
            map.put("MAP", "HashMap");

            request.setAttribute("List", list);
            request.setAttribute("Map", map);

```

```

    %>

    ${empty List}<br> <!--false-->

    ${not empty Map}<br> <!--true-->

</body>
</html>

```

隐式对象：

不用创建直接用，类似与内置对象

- el表达式中有11个隐式对象
- 学一个隐式对象：pageContext
 - 功能：
 - 获取jsp其他八个内置对象
 - 比如：\${pageContext.request.contextPath}：动态获取虚拟目录

```

<%@ page import="cn.User.User" %>
<%@ page import="java.util.*" %>

<%@page contentType="text/html; charset=UTF-8" pageEncoding="utf-8"
language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    ${pageContext.request.contextPath} <!--动态获取虚拟目录，用于以后的编写-->

  </body>
</html>

```

JSTL

概念：JavaServer Pages Tag Library JSP标准标签库

- 是由Apache组织提供的开源的免费的jsp标签 <标签>

作用：用于简化和替换jsp页面上的java代码

使用步骤：

1. 导入jstl相关jar包
2. 引入标签库：taglib指令： <%@ taglib %>
3. 使用标签

报错解决方案

解决方法一：

- 在WEB-INF下面的lib添加文件，这个目录不能错，注意是WEB-INF和lib，不能叫其他的名字

解决方法二：

在IntelliJ Idea中使用JSTL标签库原网址：<https://www.cnblogs.com/junhuawang/p/6953177.html>

习惯了eclipse和myeclipse开发的我们总是依赖于系统的插件，而当我想当然的以为IntelliJ IDEA 的jstl 的使用应该和myeclipse一样，当时使用起来却到处碰壁，完全找不到思路，网上查阅资料也是缺少一部分，下面我来总结了Idea jstl标签的引入和使用的全过程：

如果直接用jstl标签会报错：

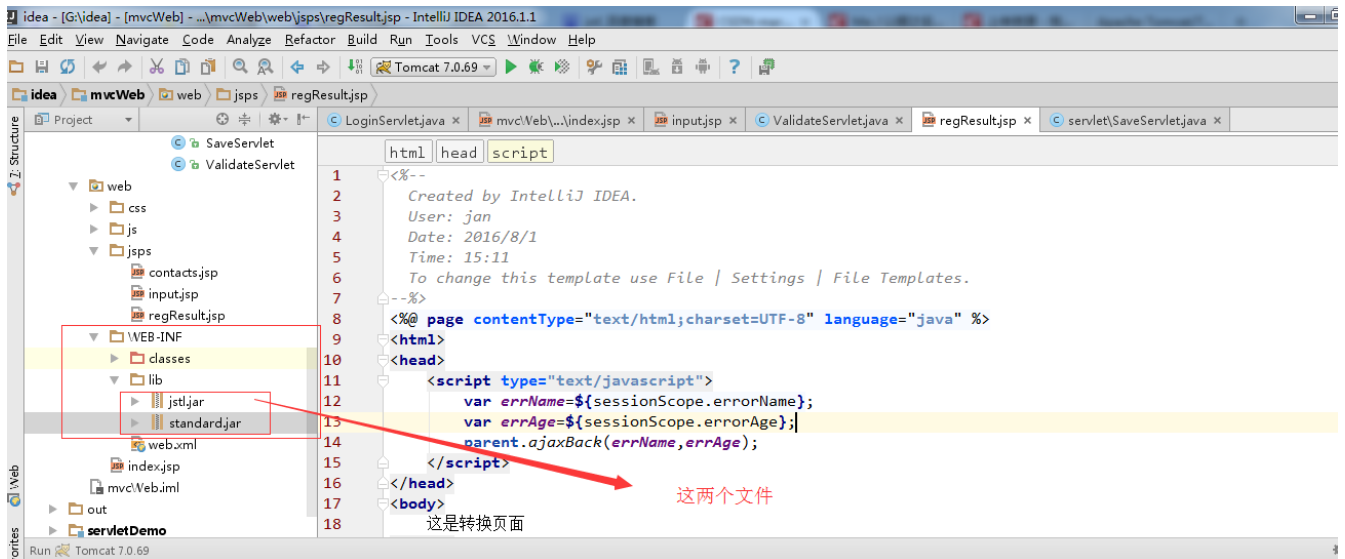
```
org.apache.jasper.JasperException: The absolute uri:
http://java.sun.com/jsp/jstl/core cannot be resolved in either web.xml or the
jar files deployed with this application
```

所以需要引入标签

一、从网上下载jakarta-taglibs-standard-1.1.2.zip解压并把standard.jar和jstl.jar加入到你的项目中，

1、下载JSTL标签库的DTD文件 下载地址：<http://archive.apache.org/dist/jakarta/taglibs/standard/binaries/> 拉到最下面，找到最新的那一个也就是 "jakarta-taglibs-standard-1.1.2.zip"，它下面的asc文件不用下载 或者直接点的下载地址：<http://archive.apache.org/dist/jakarta/taglibs/standard/binaries/jakarta-taglibs-standard-1.1.2.zip>

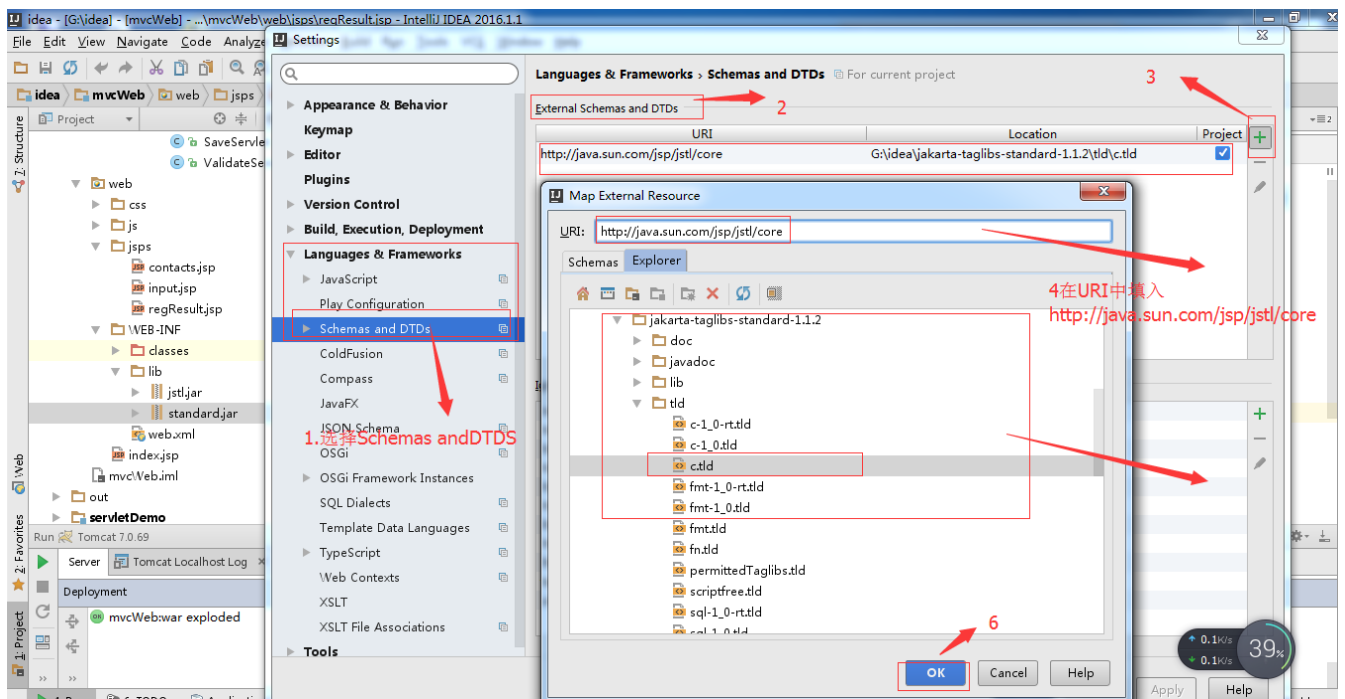
2、找到lib目录下的standard.jar和jstl.jar文件,复制文件，把它们粘贴到你的WEB_INF/lib目录下如图



下载页面[官方下载 jakarta-taglibs-standard-1.1.2.zip](#)

第一步解决了: org.apache.jasper.JasperException: Unable to load class for JSP这个报错

二、第一步会让你能使用jstl标签,但是idea还是会报错,因为它不认识这个标签,所以需要在设置里进行添加。选择File—>Setting在Setting下找到languages&Frameworks—>Schemas and DTDs,在右边的External Schemas and DTDs框下点击+,然后在URI中填入<http://java.sun.com/jsp/jstl/core> 在explorer中找到解压文件的tld目录下的c.tld文件,并选择,到这里,你就可以结局idea不识别人jstl的问题



三、在jsp页面头部加如下代码

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

这时候再在jsp 使用是没有问题的

常用的JSTL标签

if:相当于java代码的if语句

属性:

- test 必须属性（必须写上），接受boolean表达式
- 如果表达式为true，则显示if标签体内容，如果为false，则不显示标签体内容
- 一般情况下，test属性值会结合el表达式一起使用

注意:

- c:if标签没有else情况，想要else情况，则可以在定义一个c:if标签

```
<%@ page import="cn.User.User" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.List" %>
<%@taglib uri = "http://java.sun.com/jsp/jstl/core" prefix ="c" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    <%
      User user = new User();
      user.setName("张三");
      user.setAge(23);

      List list = new ArrayList<>();
      list.add(1);
      list.add(user);

      request.setAttribute("LIST",list);
    %>

    <c:if test="true">
      <h4>能看见的标签</h4>
    </c:if>
    <c:if test="false">
      <h4>看不见的标签</h4>
    </c:if>
    <c:if test="${not empty LIST}">
      <h4>有点东西</h4>
    </c:if>

  </body>
</html>
```

choose:相当于java代码的switch语句

1. 使用choose标签声明 相当于switch声明, when标签不可以省略
2. 使用when标签做判断 相当于case, test标签不能省略
3. 使用otherwise标签做其他情况的声明 相当于default

```
<%@ page import="cn.User.User" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.List" %>
<%@taglib uri = "http://java.sun.com/jsp/jstl/core" prefix ="c" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    <%
      request.setAttribute("number",1);
    %>

    <c:choose>
      <c:when test="${number==1}">星期1</c:when>
      <c:when test="${number==2}">星期2</c:when>
      <c:when test="${number==3}">星期3</c:when>
      <c:when test="${number==4}">星期4</c:when>
      <c:when test="${number==5}">星期5</c:when>
      <c:when test="${number==6}">星期6</c:when>
      <c:when test="${number==7}">星期7</c:when>
      <c:otherwise>数字输入有误</c:otherwise>
    </c:choose>

  </body>
</html>
```

foreach:相当于java代码的for语句

1. 完成重复的操作
 - 属性
 - begin: 开始值
 - end: 结束值
 - var: 临时变量
 - step: 每一次临时变量增加多少, 1就是var++, 2就是var+=2,
 - varStatus: 循环的状态
 - index: 容器中元素的索引, 从0开始
 - count: 循环的次数(不受step的影响), 从1开始
2. 遍历容器

```
for(User user : list){
    xxx
}
```

- items: 容器对象, 相当于list
- var: 容器中元素的临时变量user
- varStatus: 循环的状态
 - index: 容器中元素的索引, 从0开始
 - count: 循环的次数(不受step的影响), 从1开始

```
<%@ page import="cn.User.User" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.List" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
  <head>
    <title>${Title}</title>
  </head>
  <body>

    <%

      List list = new ArrayList();
      list.add(1);
      list.add(2);
      list.add(3);
      list.add(4);
      list.add(5);

      request.setAttribute("number", 1);
      request.setAttribute("List", list);
    %>

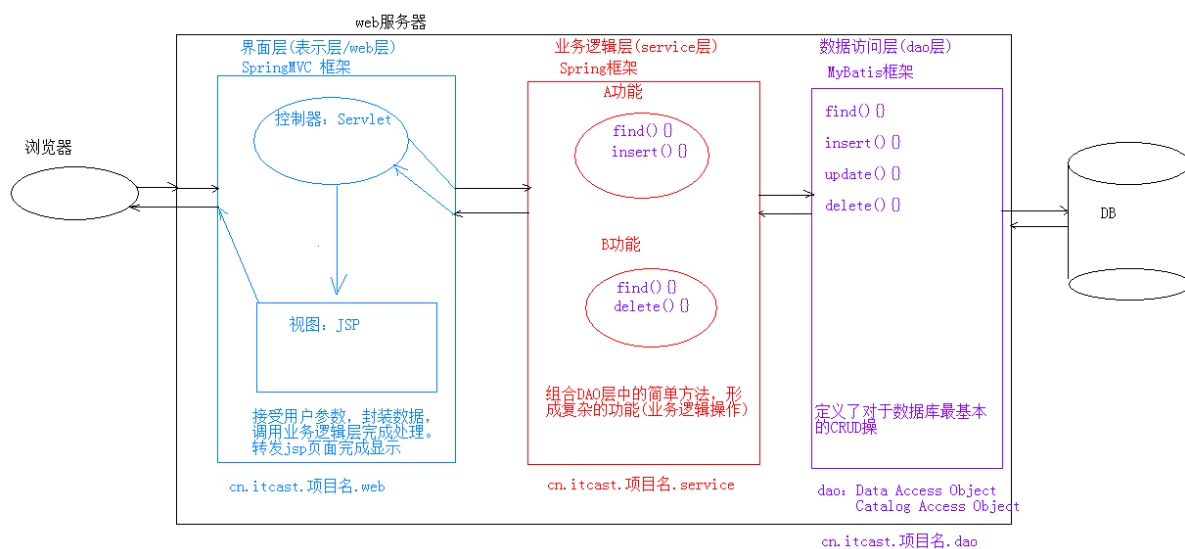
    <c:forEach begin="1" end="10" var="i" step="2">${i}</c:forEach><!--输出
1,3,5,7,9--%>
    <br>
    <c:forEach items="${List}" var="i" varStatus="s">${i} ${s.index} ${s.count}
  </c:forEach>

  </body>
</html>
```

三层架构：软件设计架构

1. 界面层(表示层): 用户看得界面。用户可以通过界面上的组件和服务进行交互

2. 业务逻辑层：处理业务逻辑的。比如登陆，比如注册。
3. 数据访问层(dao: Data Access Object)：操作数据存储文件。

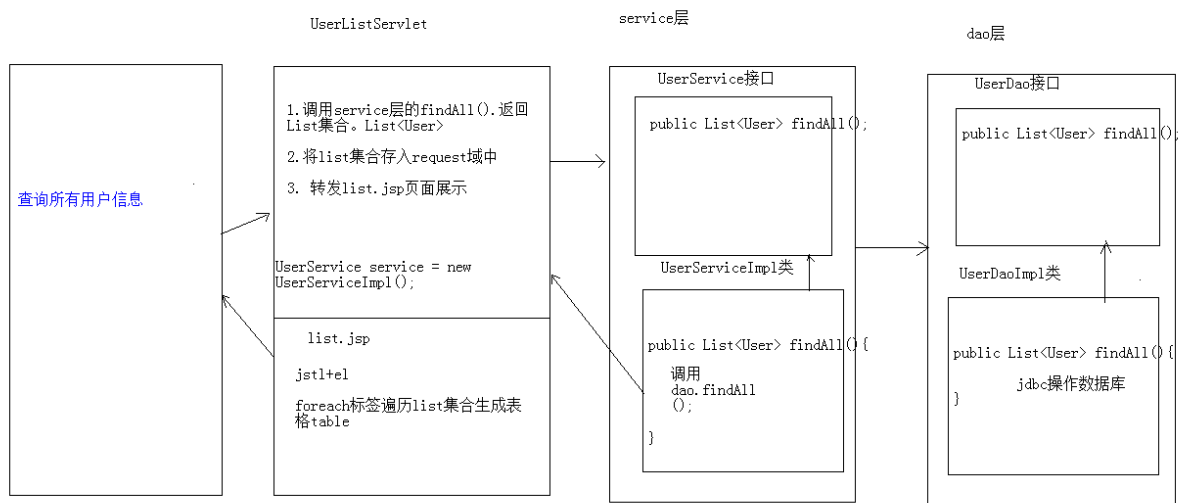


案例：用户信息列表展示

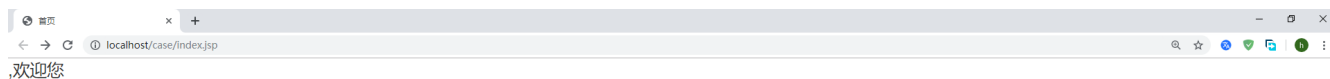
1. 需求：用户信息的增删改查操作
2. 设计：
 1. 技术选型：Servlet+JSP+MySQL+JDBCTemplate+DuiD+BeanUtilS+tomcat
 2. 数据库设计：

```
create database day17; -- 创建数据库
use day17;           -- 使用数据库
create table user(   -- 创建表
  id int primary key auto_increment,
  name varchar(20) not null,
  gender varchar(5),
  age int,
  address varchar(32),
  qq varchar(20),
  email varchar(50)
);
```

3. 开发：
4. 环境搭建
5. 创建数据库环境
6. 创建项目，导入需要的jar包
7. 编码
8. 测试
9. 部署运维



先看一下效果图：



用户信息列表

姓名	<input type="text"/>	籍贯	<input type="text"/>	邮箱	<input type="text"/>	查询	添加联系人	删除选中
<input type="checkbox"/>	编号	姓名	性别	年龄	籍贯	QQ	邮箱	操作
<input type="checkbox"/>	1	张三	男	13	陕西	12345	zhangsan@itcast.cn	<input type="button" value="修改"/> <input type="button" value="删除"/>
<input type="checkbox"/>	2	李四	女	15	北京	88888	ls@itcast.cn	<input type="button" value="修改"/> <input type="button" value="删除"/>

« 1 2 3 4 5 » 共16条记录，共4页

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <title>首页</title>

    <!-- 1. 导入CSS的全局样式 -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- 2. jQuery导入, 建议使用1.9以上的版本 -->
    <script src="js/jquery-2.1.0.min.js"></script>
    <!-- 3. 导入bootstrap的js文件 -->
    <script src="js/bootstrap.min.js"></script>
    <script type="text/javascript">
    </script>
</head>
<body>

<div >${user.name},欢迎您</div>
<div align="center">
    <a
        href="${pageContext.request.contextPath}/findUserServlet"
        style="text-decoration:none;font-size:33px">查询所有用户信息
    </a>
</div>
```

```
</body>
</html>
```

```
package web.servlet;

import domain.User;
import service.Impl.UserServiceImpl;
import service.UserService;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.util.List;

@WebServlet("/findUserServlet")
public class FindUserServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        UserServiceImpl userService = new UserServiceImpl();

        List<User> list = userService.findAll();

        HttpSession session = request.getSession();

        session.setAttribute("users", list);

        request.getRequestDispatcher("/list.jsp").forward(request, response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        this.doPost(request, response);
    }
}
```

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
```

```

<title>表单界面</title>

<!-- 1. 导入css的全局样式 -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- 2. jQuery导入, 建议使用1.9以上的版本 -->
<script src="js/jquery-2.1.0.min.js"></script>
<!-- 3. 导入bootstrap的js文件 -->
<script src="js/bootstrap.min.js"></script>
<script type="text/javascript">
</script>
</head>
<style type="text/css">
    td, th {
        text-align: center;
    }
</style>
</head>
<body>
<div class="container">
    <h3 style="text-align: center;">用户信息列表</h3>

    <div>
        <form style="float: left;" class="form-inline form-horizontal">
            <div class="form-group">
                <label for="exampleInputName2">姓名</label>
                <input type="text" class="form-control" id="exampleInputName2" >
            </div>
            <div class="form-group">
                <label for="exampleInputName3">籍贯</label>
                <input type="text" class="form-control" id="exampleInputName3" >
            </div>

            <div class="form-group">
                <label for="exampleInputEmail2">邮箱</label>
                <input type="email" class="form-control" id="exampleInputEmail2"
            >
            </div>
            <button type="submit" class="btn btn-default">查询</button>
        </form>

    </div>

    <div style="float: right;margin: 5px;">

        <a class="btn btn-primary" href="add.html">添加联系人</a>
        <a class="btn btn-primary" href="add.html">删除选中</a>

    </div>

    <table border="1" class="table table-bordered table-hover">
        <tr class="success">

```

```

        <th><input type="checkbox"></th>
        <th>编号</th>
        <th>姓名</th>
        <th>性别</th>
        <th>年龄</th>
        <th>籍贯</th>
        <th>QQ</th>
        <th>邮箱</th>
        <th>操作</th>
    </tr>

    <c:forEach items="${sessionScope.users}" var="user" varStatus="s">
        <tr>
            <td><input type="checkbox"></td>
            <td>${s.count}</td>
            <td>${user.name}</td>
            <td>${user.gender}</td>
            <td>${user.age}</td>
            <td>${user.address}</td>
            <td>${user.qq}</td>
            <td>${user.email}</td>
            <td><a class="btn btn-default btn-sm" href="update.html">修改
        </a>&nbsp;<a class="btn btn-default btn-sm" href="">删除</a></td>
        </tr>

    </c:forEach>

</table>
<div>
    <nav aria-label="Page navigation">
        <ul class="pagination">
            <li>
                <a href="#" aria-label="Previous">
                    <span aria-hidden="true">&laquo;</span>
                </a>
            </li>
            <li><a href="#">1</a></li>
            <li><a href="#">2</a></li>
            <li><a href="#">3</a></li>
            <li><a href="#">4</a></li>
            <li><a href="#">5</a></li>
            <li>
                <a href="#" aria-label="Next">
                    <span aria-hidden="true">&raquo;</span>
                </a>
            </li>
            <span style="font-size: 25px;margin-left: 5px;">
                共16条记录, 共4页
            </span>
        </ul>
    </nav>
</div>

```

```
        </ul>
    </nav>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```