



chapter2- jSP数据交互（一）

提问

- 在JSP页面中计算两个数的和，将结果输出显示，横线处应为：

```
<%@ page language="java"
import="java.util.*,java.text.*" contentType="text/html;
charset=utf-8" %>
<html>
  <head><title>计算求和</title></head>
  <body>
    两个数的求和结果为:
    <%
      int numA = 4, numB = 5 ;
      int result = numA+numB;
    %>
    _____
  </body>
</html>
```

<%=result %>

<% out.print(result); %>

提问

- JSP如何处理客户端的请求？
- 使用session的什么方法存取对象值，取到的值是什么类型？

- 实现在JSP页面中获取注册信息
- 实现企业邮箱登录验证功能
- 实现新闻发布系统页面访问控制



本章目标



南京网博

专业专注 全心服务

- 熟练应用request对象获取用户请求
- 熟练应用response对象处理响应
- 熟练应用转发与重定向控制页面跳转
- 掌握session的原理及应用
- 掌握include指令的应用

重点

重点

重点

重点

难点

什么是JSP内置对象



南京网博

专业专注 全心服务

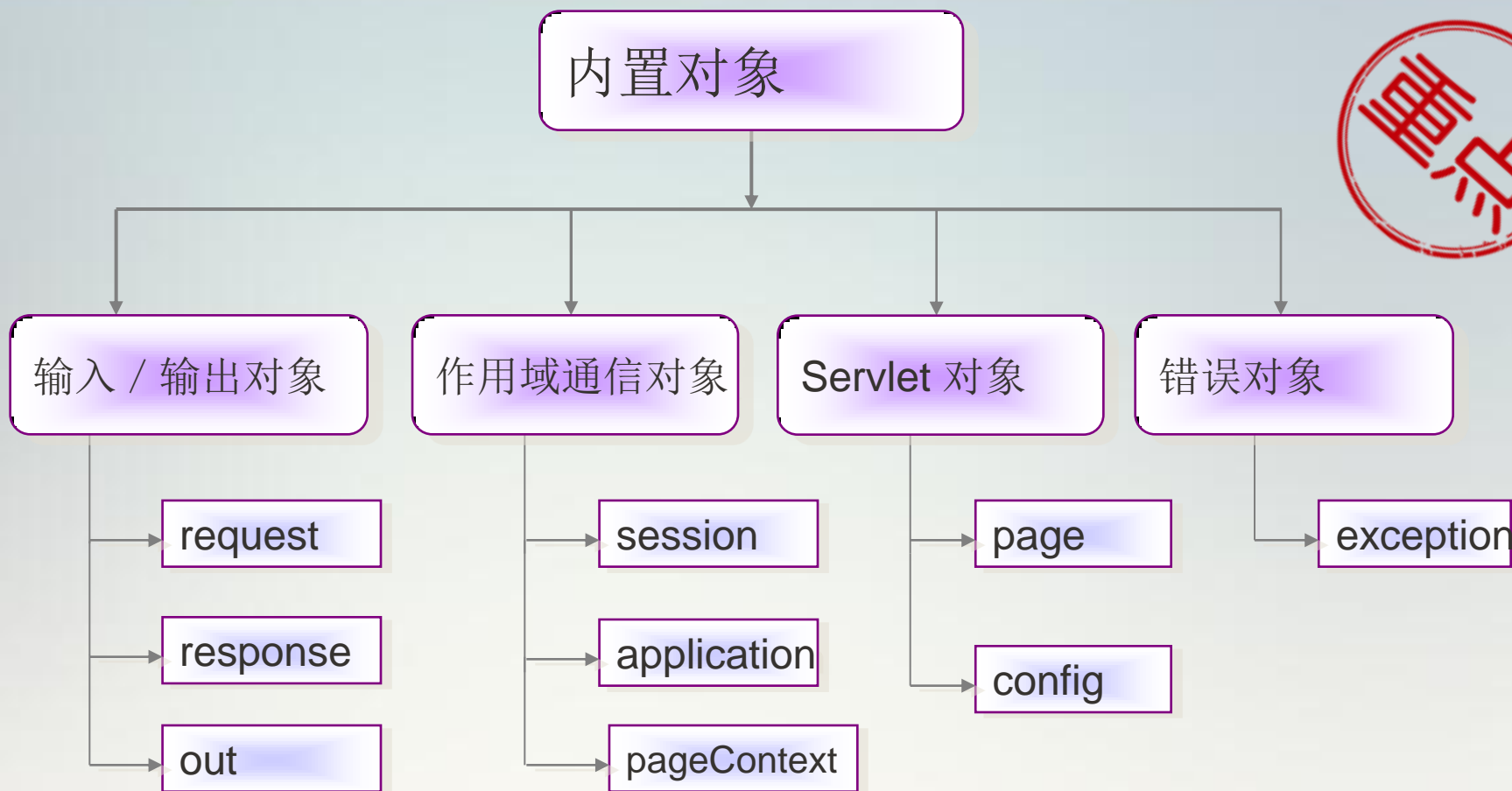
- JSP内置对象是 Web 容器创建的一组对象

没有出现new关键字，但却可以使用out对象

```
<%  
int[ ] value = { 60, 70, 80 };  
    for (int i : value) {  
        out.println(i);  
    }  
%>
```



- JSP 内置对象是Web容器加载的一组类的实例
- 无需用“new”去创建实例，而是可以直接在JSP页面使用的对象，也称JSP隐式对象
- 内置对象的名称是JSP的保留字
- JSP 使用 Java 定义的内置对象来访问网页的动态内容
- 共分为四类9个内置对象



内置对象 2-2



南京网博
专业专注 全心服务

表 3.1 JSP 内置对象映射表

对象名	类型	作用域
request	javax.servlet.HttpServletRequest 的子类	Request
response	javax.servlet.HttpServletResponse 的子类	Page
pageContext	javax.servlet.jsp.PageContext	Page
session	javax.servlet.http.HttpSession	Session
application	javax.servlet.ServletContext	Application
Out	javax.servlet.jsp.JspWriter	Page
config	javax.servlet.ServletConfig	Page
Page	java.lang.Object	Page
exception	java.lang.Throwable	Page

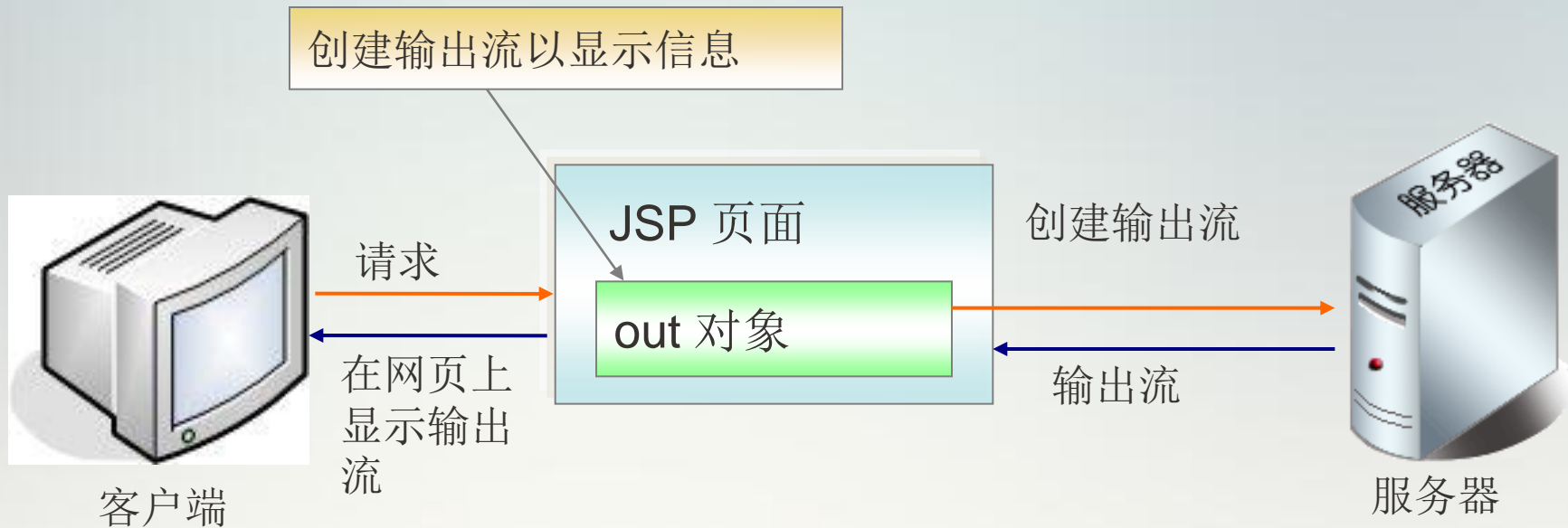


内置对象分类

- JSP内置对象的分类及组成
- **输入/输出对象**：request、response 和 out
- **作用域对象**：session、application和pageContext
- **Servlet 相关对象**：page 和 config
- **错误对象**：exception

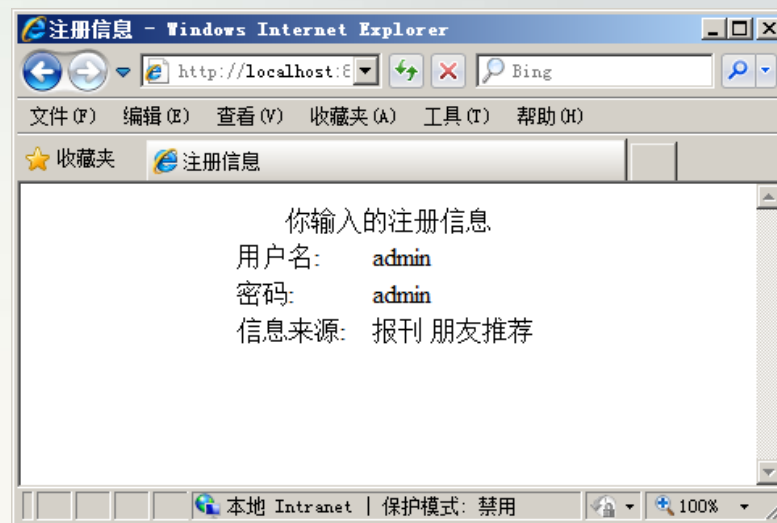
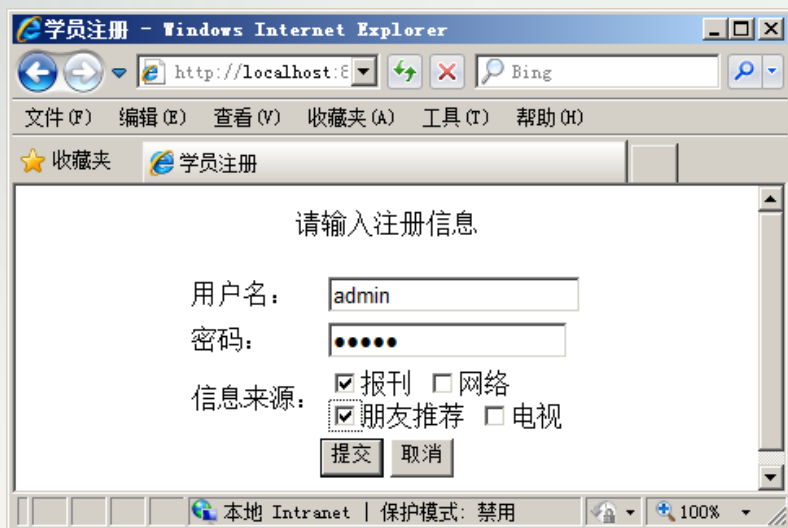
- 控制页面的输入和输出
- 访问与所有请求和响应有关的数据
- 输入和输出对象包括request、response 和 out

- 表示输出流
- `javax.servlet.jsp.JspWriter` 类的实例
- 使用 `write()`、`print()` 和 `println()` 方法

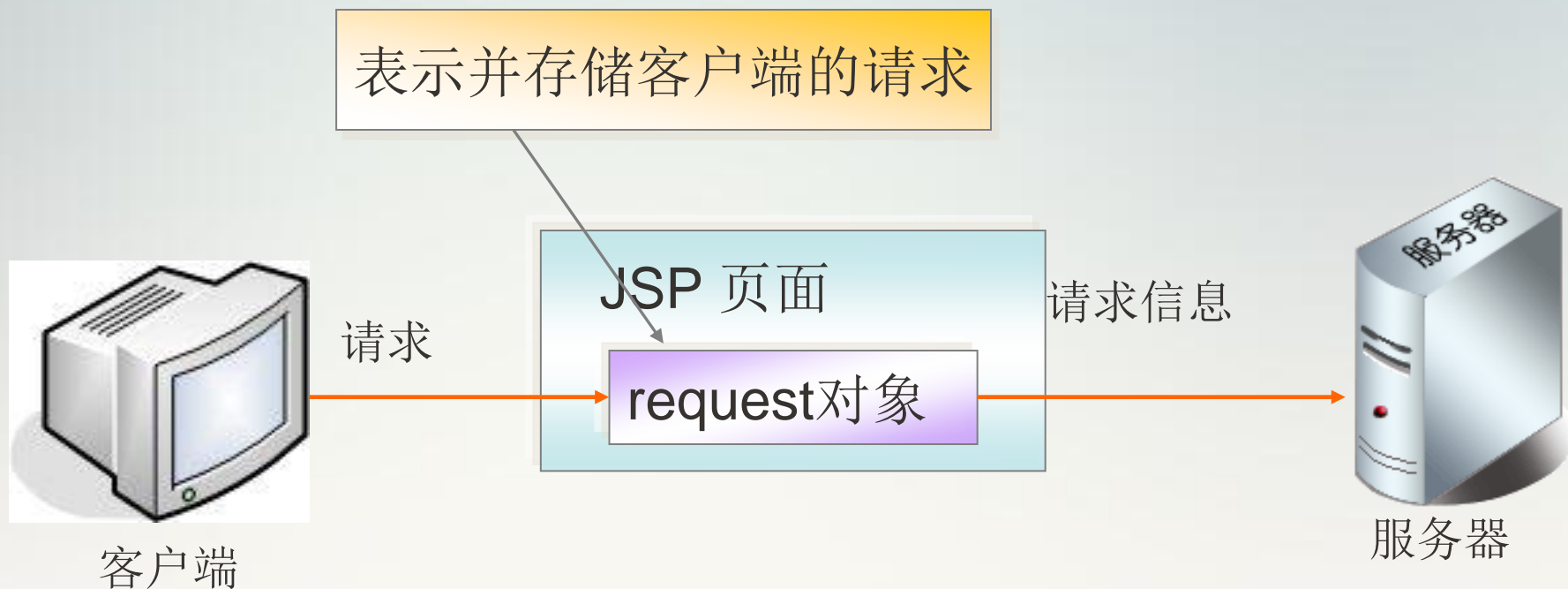


? 问题

- 如何实现学员的注册功能？
 - 注册信息包括：用户名、密码、信息来源
 - 页面提交后，显示学员输入的数据



- 表示客户端对网页的请求
- 实现 `javax.servlet.http.HttpServletRequest` 接口
- 使用 **HTTP** 协议处理客户端的请求



● request对象常用方法

方法名称	说明
String getParameter(String name)	根据表单组件名称获取提交数据
String[] getParameterValues(String name)	获取表单组件对应多个值时的请求数据
void setCharacterEncoding(String charset)	指定每个请求的编码
RequestDispatcher getRequestDispatcher(String path)	返回一个RequestDispatcher对象，该对象的forward()方法用于转发请求
String getMethod()	获取请求方式
getParameterNames ()	获取所有请求的参数名



- `Cookie[] getCookies()`
- `String getHeader(String name)`
- `Enumeration getHeaderNames()`
- `String getRequestURI()`
- `HttpSession getSession()`



● 学员注册页面

```
<form name="form1" method="post" action="reginfo.jsp">
  <table border="0" align="center">
    <tr> <td>用户名</td><td> <input type="text" name="name"></td></tr>
    <tr><td>密码</td><td> <input type="password" name="pwd"> </td></tr>
    <tr> <td>信息来源</td><td>
      <input type="checkbox" name="channel" value="报刊">报刊
      <input type="checkbox" name="channel" value="网络">网络
      <input type="checkbox" name="channel" value="朋友推荐"> 朋友推荐
      <input type="checkbox" name="channel" value="电视"> 电视
    </td></tr>
    <!--省略提交、取消按钮 -->
  </table>
</form>
```

- 信息读取显示页面

```
<%
```

```
//读取用户名和密码
```

```
String name = request.getParameter("name");
```

```
String pwd = request.getParameter("pwd");
```

```
//...HTML页面代码
```

```
//读取复选框
```

```
String[] ch
```

```
if (channel
```

```
for (Strin
```

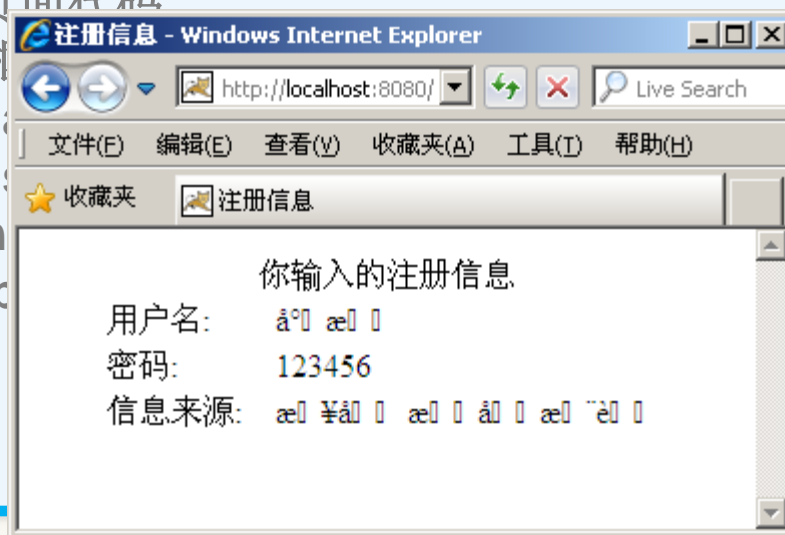
```
out.p
```

```
}
```

```
}
```

```
%>
```

- 为什么我们输入的中文都是乱码显示?



```
es("channel");
```



演示示例1：学员注册

网博汇智课程研发

- 在页面设置支持中文字符的字符集

方式3

在Tomcat目录结构\conf\server.xml中设置字符集

```
<Connector port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" URIEncoding="UTF-8"  
>
```

```
String pwd = request.getParameter("pwd");  
%>
```





练习

● 需求说明

- 通过表单提交注册信息
- 使用request对象获取表单提交数据
- 将获取的数据输出显示



提示

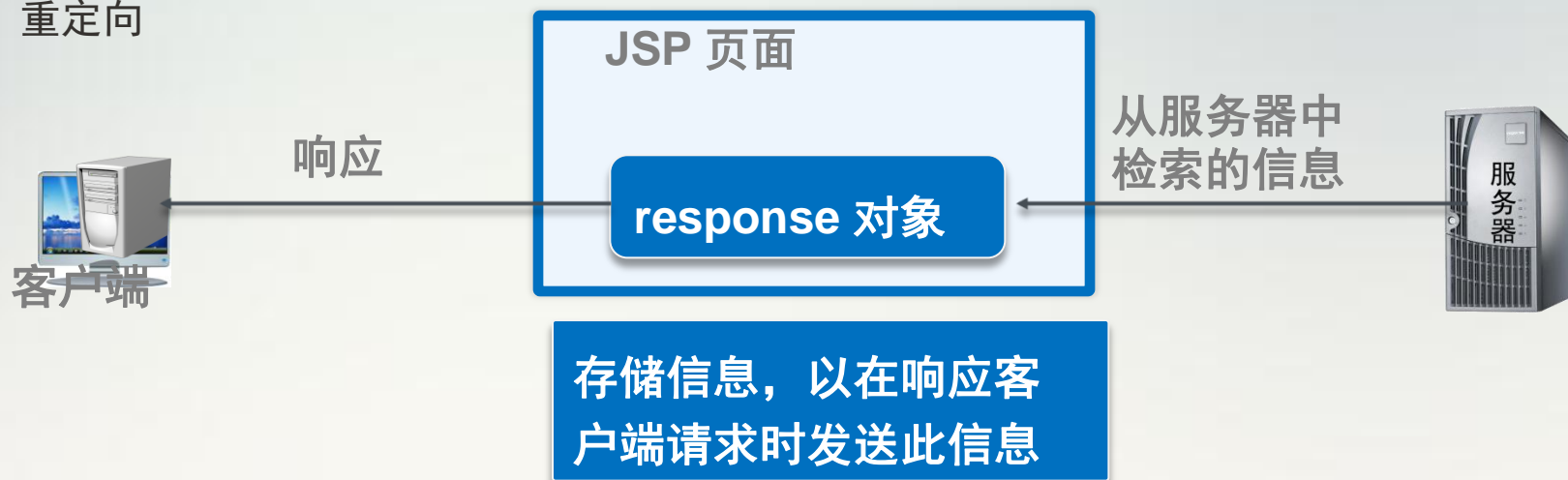
● 用request.get

的数据

完成时间：20分钟

- response对象用于响应客户请求并向客户端输出信息
- 使用 HTTP 协议将响应发送给客户端
- 实现javax.servlet.http.HttpServletResponse 接口
- response 对象常用方法

■ void sendRedirect (String location)：将请求重新定位到一个不同的URL，即页面重定向

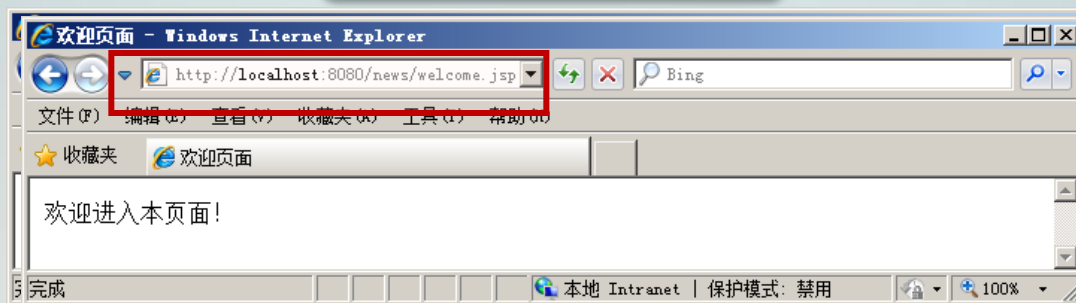


● response 对象常用方法

- void `setContentType (String name)`
设置作为响应生成的内容的类型和字符编码
- void `sendRedirect (String name)`
发送一个响应给浏览器，指示其应请求另一个URL
- String `encodeURL(java.lang.String url)`
构建一个包含会话信息的URL地址
- void `sendError(int sc)`
发送一个错误状态信息

- 根据业务逻辑实现

客户端重新建立链接



control.jsp

```
<%  
request.setCharacterEncoding("utf-8");  
String name = request.getParameter("userName");  
String pwd = request.getParameter("pwd");  
if(name.equals("sa") && pwd.equals("sa"))  
    response.sendRedirect("welcome.jsp");  
%>
```

跳转至欢迎页面



演示示例3: response对象

? 问题

- 页面实现跳转了，请求的信息是否也一起转移呢？



演示示例4：页面跳转后获取请求信息

- 那么如何才能实现页面跳转后，请求信息不丢失呢？

使用转发取代重定向实现页面跳转

- 转发的作用

- 在多个页面交互过程中实现请求数据的共享

- 转发的实现

- RequestDispatcher对象.forward()方法

```
<%  
RequestDispatcher rd=request.getRequestDispatcher("welcome.jsp");  
rd.forward(request,response);  
%>
```

将当前接收的用户请求，发送给服务器的其他资源使用



演示示例5：页面转发后获取请求信息

转发

转发是在服务器端发挥作用，通过forward()将提交信息在多个页面间进行传递

客户端浏览器的地址栏不会显示出转向后的地址

重定向



重定向是在客户端发挥作用，通过请求新的地址实现页面转向

在地址栏中可以显示转向后的地址

练习

● 需求说明

■ 实现企业邮箱登录验证功能

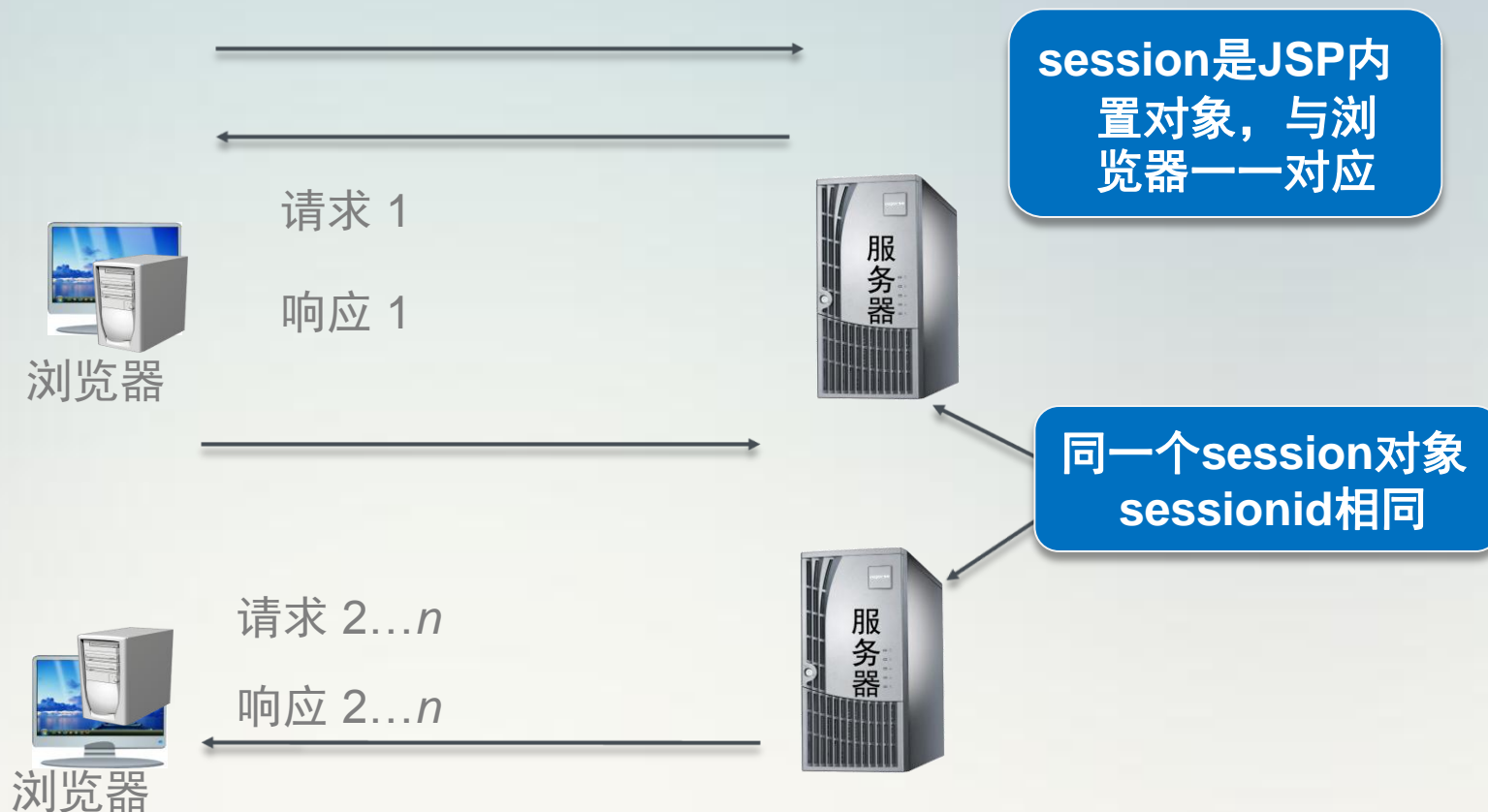
- ◆ 用户通过JSP页面输入用户名和密码
- ◆ 如果用户名为lucky，密码为123456，在欢迎页面显示“你好：lucky！”
- ◆ 如果验证登录失败，则返回登录页面重新登录



完成时间：25分钟

什么是会话

- 一个会话就是浏览器与服务器之间的一次通话



- session对象常用方法:

方法名称	说明
<code>void setAttribute(String key,Object value)</code>	以key/value的形式保存对象值
<code>Object getAttribute(String key)</code>	通过key获取对象值
<code>void invalidate()</code>	设置session对象失效
<code>String getId()</code>	获取sessionid
<code>void setMaxInactiveInterval(int interval)</code>	设定session的非活动时间
<code>int getMaxInactiveInterval()</code>	获取session的有效非活动时间(以秒为单位)
<code>void removeAttribute(String key)</code>	从session中删除指定名称(key)所对应的对象

问题

- 一个session对应一个窗口，那么通过超链接打开的窗口是否也是新的session呢？

分析

- 每个session对象都与浏览器一一对应，重新开启一个浏览器，相当于重新创建一个session对象（版本不同可能有所差别）
- 通过超链接打开的新窗口，新窗口的session与其父窗口的session相同



演示示例6：获取session的ID

- 在新闻发布系统中，增加访问控制功能

- 提供用户登录功能
- 管理员有权对新闻实现增、删、改的操作



session中无法
提取到用户信息



登录处理页面

- 1、获得登录信息
- 2、查询数据库，判断该用户是否注册
- 3、如果该用户已注册，在**session**中保存该用户的登录信息
- 4、如果用户是管理员就跳转到管理员界面；否则跳转到新闻发布系统的首页面

管理员界面

- 1、从**session**中提取该用户信息
- 2、如果用户信息存在，显示管理员界面内容
- 3、如果用户信息不存在，跳转到登录页面

用户已注册

- 在控制页面获取用户请求的登录信息进行验证

```
<%  
if ("admin".equals(name)&&"admin".equals(pwd){ //如果是已注册用户  
    session.setAttribute("login", name);  
    //设置session过期时间  
    session.setMaxInactiveInterval(10*60);  
    request.getRequestDispatcher("admin.jsp").forward(request,response);  
} else {  
    response.sendRedirect("index.jsp");  
}  
%>
```

在session中存放
用户登录信息

- 在新闻发布系统新闻发布页面增加登录验证

如果session中不存在该用户的登录信息，转入登录页面

```
<%  
String login = (String) session.getAttribute("login");  
if (login == null){  
    response.sendRedirect("index.jsp");  
} %>
```



演示示例7：使用session实现访问控制

- 访问控制效果验证，步骤如下：





练习

● 需求说明

- 新闻发布系统只允许管理员能够进入后台操作页面
- 普通用户只有浏览新闻和发布评论的权限



提示

- `session.getAttribute(String key)` 方法的返回值是一个 `Object`，必须进行强制类型转换

完成时间：25分钟

语法

- 手动设置失效: `invalidate()`
- 超时失效
 - 通过 `setMaxInactiveInterval()` 方法, 单位是秒

■ 通过

```
<%  
session.setAttribute("login","admin");  
session.setMaxInactiveInterval(600);  
response.sendRedirect("admin.jsp");  
%>
```

单位是分钟

```
<session-config>  
  <session-timeout>10</session-timeout>  
</session-config>
```

? 问题

- 除了首页面，其它页面中同样需要加入登录验证，有没有办法避免冗余代码的出现？

可以将一些共性的内容写入一个单独的文件中，然后通过include指令引用该文件

- 创建登录验证文件 loginControl.jsp

```
<%  
String login = (String) session.getAttribute("login");  
if (login == null)  
    response.sendRedirect("index.jsp");  
%>
```

- 在后台首页面中使用include指令引用登录验证文件

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

练习

- 训练要点
 - 使用 include
- 需求说明
 - 修改新闻发布系统后台管理员操作页面
 - 使用 include 指令引用登录验证文件





- 内置对象

- 由Web容器加载的一组类的实例

- request与response对象

- request处理客户端请求
- response响应客户端请求

- 会话对象session

- 可以保持每个用户的会话信息，为不同的用户保存自己的数据，存储在客户端

● 转发与重定向

转发

转发是在服务器端发挥作用，通过forward方法将提交信息在多个页面间进行传递

客户端浏览器的地址栏不会显示出转向后的地址

重定向

重定向是在客户端发挥作用，通过请求新的地址实现页面转向

在地址栏中可以显示转向后的地址