**实验四 Servlet容器模型**

学生姓名： 张小锋 学 号： 8002119102 专业班级： 计软192班

实验类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期：2021.10.27 实验成绩：

一、实验项目名称

Servlet容器模型

二、实验目的

1. 掌握Web应用程序的DD文件web.xml的配置方法和作用；

2. 掌握如何通过ServletConfig获取Servlet的初始化参数；

3. 掌握ServletContext的概念，如何获得ServletContext对象；

4. 掌握如何通过RequestDispatcher转发请求。

三、实验基本原理

Web应用程序有严格定义的目录结构，它需要一个称为部署描述文件的文件web.xml来配置。对应用程序的每个Servlet，可以在web.xml中定义若干个初始化参数，然后通过ServletConfig对象获取这些参数。

ServletContext对象表示该应用程序的环境，通过该对象可以获得应用程序范围的初始化参数、容器版本信息、资源的URL、在其中存放共享数据以及登录日志等。

四、主要仪器设备及耗材

计算机、Idea、Tomcat9.0、JDK1.8

五、实验步骤（完整内容见光盘）

（一）建立一个名为bookstore的Web应用程序

在Tomcat的webapps目录下，建立一个如下所示的目录结构：

|- bookstore

|- html (包含所有的HTML文件)

|- jsp (包含所有的JSP文件)

|- images (包含所有的GIF、JPEG和BMP图象文件)

|- index.html (默认的HTML文件)

|- WEB-INF

|-classes

├ lib

├ web.xml

图4.1 bookstore应用程序的目录结构

这样就建立了一个名为bookstore的Web应用程序，本实验后面的题目都在该应用程序下面完成。

（二）使用ServletConfig检索Servlet的初始化参数

【步骤1】编写ConfigDemoServlet程序，内容如下：

package com.demo;

import java.io.\*;

import java.sql.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class ConfigDemoServlet extends HttpServlet{

String servletName=null;

ServletConfig config=null;

String email=null;

String telephone=null;

public void init() {

config = **getServletConfig();**

servletName = config.getServletName();

email = **config.getInitParameter("email");**

telephone = **config.getInitParameter("telephone");**

}

public void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException,IOException{

response.setContentType("text/html;charset=gb2312");

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("Servlet名称："+servletName+"<br>");

out.println("Email地址："+email+"<br>");

out.println("电话："+telephone);

out.println("</body></html>");

}

}

【步骤2】修改web.xml文件，在其中添加下面Servlet定义和初始化参数定义

<servlet>

<servlet-name>configDemoServlet</servlet-name>

<servlet-class>com.demo.ConfigDemoServlet</servlet-class>

<init-param>

<param-name>**email**</param-name>

<param-value>**hacker@163.com**</param-value>

</init-param>

<init-param>

<param-name>**telephone**</param-name>

<param-value>**8899123**</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>configDemoServlet</servlet-name>

<url-pattern>/configDemo.do</url-pattern>

</servlet-mapping>

【步骤3】在浏览器的地址栏中输入下面URL访问该Servlet：

http://localhost:8080/bookstore/configDemo.do

程序运行结果如下图所示：



ConfigDemoServlet程序的运行结果

（三）ServletContext的使用

修改上面程序，将email地址和电话使用ServletContext初始化参数传递给ConfigDemoServlet。使用ServletContext对象将email地址和电话信息写入日志文件中，并查看日志文件。

（五）下面是一个Servlet，它可以获得有关服务器的信息。

package com.demo;

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

public class ServerInfo extends GenericServlet {

public void service(ServletRequest request, ServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=GB2312");

PrintWriter out = response.getWriter();

ServletContext sc = getServletContext();

out.println("<html>");

out.println("<body>");

out.println("<head>");

out.println("<title>获取服务器端信息</title>");

out.println("</head>");

out.println("<body>");

out.println("<h2>服务器端信息：</h2>");

out.println("<table width=\"500\" border=\"1\">");

out.println("<tr>");

out.println("<td width=\"175\">站点名</td>");

out.println("<td width=\"325\">" + request.getServerName() + "</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>端口号</td>");

out.println("<td>" + request.getServerPort() + "</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>服务器类型</td>");

out.println("<td>" + sc.getServerInfo() + "</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>支持Servlet版本</td>");

out.println("<td>"+sc.getMajorVersion()+"."+sc.getMinorVersion()+"</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>服务器属性</td>");

out.println("<td>");

// 获得服务器属性集合

Enumeration attributes = sc.getAttributeNames();

while (attributes.hasMoreElements()) {

String name = (String)attributes.nextElement();

out.println(name);

}

out.println("</td>");

out.println("</tr>");

out.println("</table>");

out.println("</body>");

out.println("</html>");

}

}

（六）简单的综合应用

【步骤1】创建一个名为input.html的HTML页面，其中包括一个表单，表单中包含两个文本域，分别供用户输入学号和姓名，该页面也包含提交和重置按钮。

【步骤2】定义一个名为com.demo.Student类，其中包括学号sno和姓名name两个private的成员变量，定义访问和修改sno和name的方法。

【步骤3】编写名为FirstServlet的Servlet，要求当用户在input.html中输入信息后点击“提交”按钮，请求FirstServlet对其处理。在FirstServlet中使用表单传递的参数（学号和姓名）创建一个Student对象并将其作为属性存储在ServletContext对象中，然后获得通过ServletContext的getRequestDispatcher()方法获得RequestDispatcher()对象，将请求转发到SecondServlet。

【步骤4】在SecondServlet中取出ServletContext上存储的Student对象，并显示输出该学生的学号和姓名。在SecondServlet的输出中应该包含一个超链接，点击该连接可以返回input.html页面。



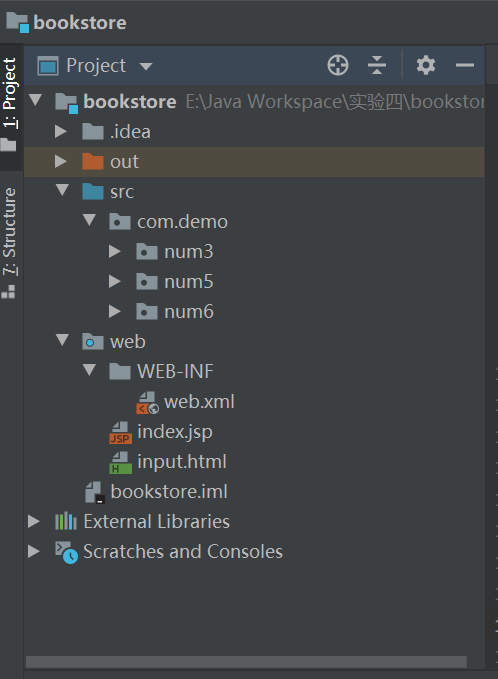
input.html页面显示结果



SecondServlet显示结果

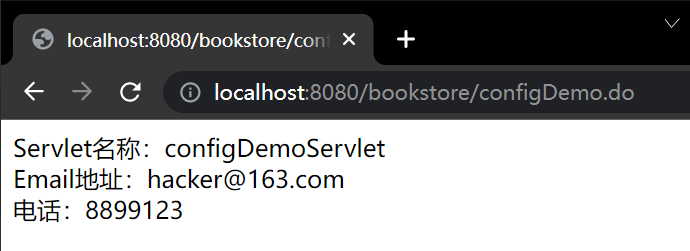
六、实验数据及处理结果

（一）建立的名为bookstore的项目的目录结构如下：



（二）使用ServletConfig检索Servlet的初始化参数，

访问<http://localhost:8080/bookstore/configDemo.do>结果如下：



（三）ServletContext的使用，使用ServletContext传递参数修改后得到的代码如下：

源代码：

web.xml

1. <?xml version="1.0" encoding="UTF-8"?>
2. <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"
5. version="4.0">
6. <context-param>
7. <param-name>email</param-name>
8. <param-value>hacker@163.com</param-value>
9. </context-param>
10. <context-param>
11. <param-name>telephone</param-name>
12. <param-value>8899123</param-value>
13. </context-param>
14. <servlet>
15. <servlet-name>configDemoServlet</servlet-name>
16. <servlet-class>com.demo.num3.ConfigDemoServlet</servlet-class>
17. <init-param>
18. <param-name>email</param-name>
19. <param-value>hacker@163.com</param-value>
20. </init-param>
21. <init-param>
22. <param-name>telephone</param-name>
23. <param-value>8899123</param-value>
24. </init-param>
25. <load-on-startup>1</load-on-startup>
26. </servlet>
27. <servlet-mapping>
28. <servlet-name>configDemoServlet</servlet-name>
29. <url-pattern>/configDemo.do</url-pattern>
30. </servlet-mapping>
31. </web-app>

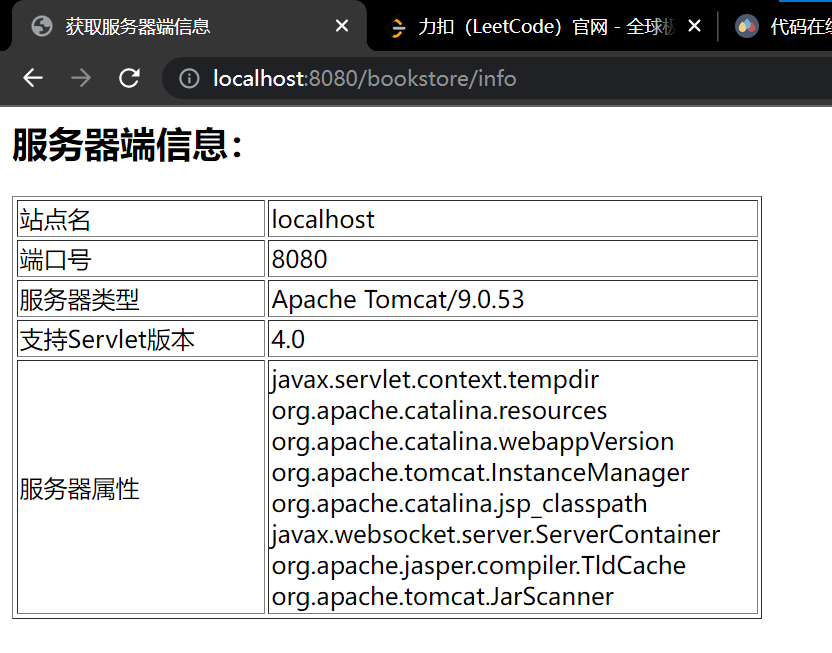
**ConfigDemoServlet**

1. package com.demo.num3;
2. import java.io.\*;
3. import java.sql.\*;
4. import javax.servlet.\*;
5. import javax.servlet.http.\*;
6. public class ConfigDemoServlet extends HttpServlet{
7. String servletName=null;
8. ServletContext servletContext=null;
9. String email=null;
10. String telephone=null;
11. public void init() {
12. servletContext=this.getServletContext();
13. servletName = servletContext.getServletContextName();
14. email = servletContext.getInitParameter("email");
15. telephone = servletContext.getInitParameter("telephone");
16. servletContext.log("email:"+email);
17. servletContext.log("telephone:"+telephone);
18. }
19. public void service(HttpServletRequest request, HttpServletResponse response)
20. throws ServletException,IOException{
21. response.setContentType("text/html;charset=gb2312");
22. PrintWriter out = response.getWriter();
23. out.println("<html><body>");
24. out.println("Servlet名称："+servletName+"<br>");
25. out.println("Email地址："+email+"<br>");
26. out.println("电话："+telephone);
27. out.println("</body></html>");
28. }
29. }

查看输出的日志文件如下：



（五）获得有关服务器的信息，运行结果如下：



（六）简单的综合应用

源代码：

**input.html**

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>提交</title>
6. </head>
7. <body>
8. <form action="FirstServlet">
9. 学号：<input type="text" name="number"><br/>
10. 姓名：<input type="text" name="name"><br/>
11. <input type="submit" value="登录" >
12. <input type="submit" value="取消">
13. </form>
14. </body>
15. </html>

**FirstServlet**

1. package com.demo.num6;
2. import javax.servlet.ServletContext;
3. import javax.servlet.ServletException;
4. import javax.servlet.annotation.WebServlet;
5. import javax.servlet.http.HttpServlet;
6. import javax.servlet.http.HttpServletRequest;
7. import javax.servlet.http.HttpServletResponse;
8. import java.io.IOException;
9. @WebServlet("/FirstServlet")
10. public class FirstServlet extends HttpServlet{
11. @Override
12. protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
13. req.setCharacterEncoding("UTF-8");
14. resp.setContentType("text/html;charset=utf-8");
15. Student student=new Student();
16. student.setName(req.getParameter("name"));
17. student.setNumber(req.getParameter("number"));
18. ServletContext servletContext=this.getServletContext();
19. servletContext.setAttribute("student",student);
20. servletContext.getRequestDispatcher("/SecondServlet").forward(req,resp);
21. }
22. }

**SecondServlet**

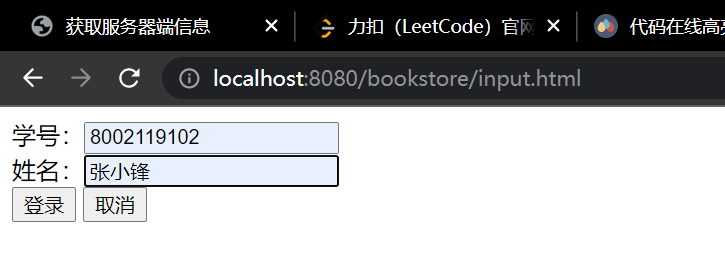
1. package com.demo.num6;
2. import javax.servlet.ServletContext;
3. import javax.servlet.ServletException;
4. import javax.servlet.annotation.WebServlet;
5. import javax.servlet.http.HttpServlet;
6. import javax.servlet.http.HttpServletRequest;
7. import javax.servlet.http.HttpServletResponse;
8. import java.io.IOException;
9. import java.io.PrintWriter;
10. @WebServlet("/SecondServlet")
11. public class SecondServlet extends HttpServlet {
12. @Override
13. protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
14. PrintWriter out = resp.getWriter();
15. ServletContext servletContext=getServletContext();
16. Student student = (Student) servletContext.getAttribute("student");
17. out.write("<h2>学号："+student.getNumber()+"<h2/>");
18. out.write("<h2>姓名："+student.getName()+"<h2/>");
19. out.write("<a href=\"http://localhost:8080/bookstore/input.html\">回到提交页面</a>");
20. }
21. }

**Student**

1. package com.demo.num6;
2. public class Student {
3. private String name;
4. private String number;
5. public void setName(String name) {
6. this.name = name;
7. }
8. public void setNumber(String number) {
9. this.number = number;
10. }
11. public String getName() {
12. return name;
13. }
14. public String getNumber() {
15. return number;
16. }
17. }

运行结果：

输入界面如下：



提交后界面如下，点击蓝色字体超链接可回到原先界面：



七、思考讨论题或体会或对改进实验的建议

1. 如何通过ServletConfig获得Servlet参数？

通过ServletConfig获取参数的代码片段如下：

config = getServletConfig();

servletName = config.getServletName();

email = config.getInitParameter("email");

telephone = config.getInitParameter("telephone");

可知通过getInitParameter函数能够获取指定名字的参数。

2. 如何获得ServletContext对象，如何通过该对象获得上下文参数？

获取ServletContext对象并通过该对象获得上下文参数的代码片段如下：

servletContext=this.getServletContext();

servletName = servletContext.getServletContextName();

email = servletContext.getInitParameter("email");

telephone = servletContext.getInitParameter("telephone");

可知可以通过this.getServletContext()获得ServletContext对象，通过getInitParameter获得指定名字的参数。

3. 如何使用ServletContext共享数据？

只要在一个Servlet中设置了ServletContext属性值，那么本项目的中任意一个Servlet中都可以获取到设置的属性值。作用范围是所有Servlet，而有效期是服务器关闭之前。通过ServletContext存储属性值的代码如下：

ServletContext context = this.getServletContext();

// 设置属性值setAttribute(String name, Object object)

context.setAttribute("name", "张三");

获取代码如下：

String name = (String)context.getAttribute("name");

这样便能够实现使用ServletContext共享数据。

八、参考资料

1.Java Web程序设计