第三次实验

实验要求

1. 使用servlet接收前台表单数据，表单包括用户姓名、年龄、所在学校
2. 用get和post两种方式传送到后台，用户姓名不允许为空
3. servlet1接收到数据以后，调用servlet2进行验证，如果验证年龄小于20，返回数据“欢迎你，少年”(利用forward和include来操作)
4. 如果年龄大于40岁，页面跳转至一个jsp（使用send redirect ），显示 “欢迎您，老先生

目录结构

├─src

│ └─main

│ ├─java

│ │ └─tech

│ │ └─haonan

│ │ ├─filter//过滤器用来拒绝掉一些非法请求,或是设置字符编码方式

│ │ │ CharacterEncodingFilter.java //设置传输字符集为utf-8

│ │ │ MyFilter.java // 用来拒绝姓名为空的请求

│ │ └─servlet

│ │ MyServlet.java //最先跳servlet,他会转发请求到Myservlet2

│ │ Myservlet2.java // 根据年龄跳转到不同jsp

│ └─webapp

│ │ error.jsp //filter中姓名为空的被拒绝连接的的请求被转发到这里

│ │ index.jsp //主界面 登录表单的的位置

│ │ old.jsp //年龄大于40的用户跳转的位置

│ └─WEB-INF

│ web.xml

实验说明

这次实验主要是学会使用servlet的基本使用,首先我们要知道servlet的基本原理,web.xml的配置是个重头戏,servlet里的注解帮助我们减轻配置负担,减少代码量,需要我们着重学习.其次我们需要知道到达servlet的时候有几种方式进行反馈,例如直接调用getwriter方法写入页面,或者是将携带信息的请求转发给jsp,在jsp中将信息展现出来,理解了这些,本次实验就非常容易实现了

核心代码

1. 以下是index.jsp文件内容

<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <title>yhn</title>

    <script type="text/javascript">

        window.onload = function (){  //javascript 代码用来提示在前端提升用户 姓名不能为空

            let oButton1 = document.getElementById("submit1");

            let oButton2 = document.getElementById("submit2");

            let oName1 = document.getElementById("name1");

            let oName2 = document.getElementById("name2");

            let name1 = oName1.value;

            let name2 = oName2.value;

            oButton1.onclick = function (){

                name1 = oName1.value;

                if(name1 === "")

                    alert("get提交中姓名不能为空");

            }

            oButton2.onclick = function (){

                name2 = oName2.value;

                if(name2 === "")

                    alert("post提交中姓名不能为空");

            }

        }

    </script>

</head>

<body>

    <form action="/hello" method="get">  <!--此表单提交方式为    get    -->

        姓名: <input type="text" id="name1" name="name"> <br>

年龄: <input type="text" name="age"> <br>

        所在学校: <input type="text" name="school"> <br>

        <input type="submit" value="提交" id="submit1" >

    </form>

<hr>

    <form action="/hello" method="post">  <!--此表单提交方式为    post    -->

        姓名: <input name="name" id="name2"> <br>

        年龄: <input name="age"> <br>

        所在学校: <input name="school"> <br>

        <input type="submit" value="提交" id="submit2">

    </form>

</body>

</html>

2.以下是old.jsp文件核心代码

<!--获取从后端得到的name值-->

<% String name = request.getParameter("name"); %>

<h1>欢迎你 老先生 <%=name%></h1>

3.以下是error.jsp文件核心代码

<h1>发生错误 名字不能为空</h1>

<a href="index.jsp">点击跳转重新输入</a>

4.以下是web.xml文件的配置

<filter>

    <!--配置拦截所有请求 将传输信息编码改为utf-8 防止乱码-->

    <filter-name>encodingFilter</filter-name>

    <filter-class>tech.haonan.filter.CharacterEncodingFilter</filter-class>

  </filter>

  <filter-mapping>

    <filter-name>encodingFilter</filter-name>

    <url-pattern>/\*</url-pattern>

  </filter-mapping>

  <filter>

    <!--配置拦截/hello接口 将名字为空的非法请求拦截下来-->

    <filter-name>MyFilter</filter-name>

    <filter-class>tech.haonan.filter.MyFilter</filter-class>

  </filter>

  <filter-mapping>

    <filter-name>MyFilter</filter-name>

    <url-pattern>/hello</url-pattern>

  </filter-mapping>

  <display-name>Archetype Created Web Application</display-name>

  <!--设置默认访问界面为index.jsp-->

  <welcome-file-list>

    <welcome-file>index.jsp</welcome-file>

  </welcome-file-list>

1. 以下是CharacterEncodingFilter .java文件核心代码

public class CharacterEncodingFilter implements Filter {

    @Override

    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

        // 设置信息编码格式为utf-8

        servletRequest.setCharacterEncoding("utf-8");

        servletResponse.setCharacterEncoding("utf-8");

        servletResponse.setContentType("text/html;charset=UTF-8");

        // 将所有的请求放行

        filterChain.doFilter(servletRequest, servletResponse);

    }

}

1. 以下是MyFilter.java的核心代码

public class MyFilter implements Filter {

    @Override

    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

        //验证姓名是否为空

        if(servletRequest.getParameter("name").length() == 0)

            //如果为空 转发到error.jsp 拒绝连接

            servletRequest.getRequestDispatcher("error.jsp").forward(servletRequest,servletResponse);

        else

            //如果不为空 放行

            filterChain.doFilter(servletRequest,servletResponse);

    }

}

1. 以下是MyServlet.java的核心代码

@WebServlet("/hello")

public class MyServlet extends HttpServlet {

    @Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// 全部跳转到Mysevlet2.java中去

        req.getRequestDispatcher("/insideInterface").forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        doGet(req,resp);

    }

}

1. 以下是Myservlet2.java的文件内容

@WebServlet("/insideInterface")

public class Myservlet2 extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        //拿到name的值

        String name = req.getParameter("name");

        // 如果name值大于40 那么就跳转到old.jsp 文件中去

        if(Integer.parseInt((String) req.getParameter("age")) > 40 )

            resp.sendRedirect("old.jsp?name=" +name );

        // 如果name值小于40 那么就直接写到网页中 欢迎年轻人

        else if(Integer.parseInt(  (String) req.getParameter("age")) < 20 )

            resp.getWriter().println("<h1>欢迎你 年轻人</h1> ");

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        doGet(req,resp);

    }}