Дубенков Семен, 9308

Цель работы: знакомство с методами передачи информации между соединениями, открываемыми в рамках одного сеанса работы пользователя

Передача данных через файл Cookie

Был реализована страница welcome.jsp, которая представляет из себя форму для заполнения значения зарплаты. В значение формы value помещается информация, которая была в предыдущем куке (если она там была).

```
%@page import="java.net.URLDecoder"%>
c%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Приветствую</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
      <form method = get action = "SalaryProcessor">
             <h2>Введите минимальную<br>зарплату футболиста</h2>
             <input type = "text" name = "salary" placeholder = "Пропуск, если не
надо"
             <%
                    Cookie [] c = request.getCookies();
                    if(c != null)
                          for(int i = 0; i < c.length; i++)</pre>
                                 if("salary".equals(c[i].getName())) {
                                        // Запись значения в поле ввода, если найден
Cookie
                                        out.print(" value='" +
URLDecoder.decode(c[i].getValue(), "UTF-8") + "' ");
                                        break;
                                 }
             %>
             > <br>
             <input type = "submit" value = "B6od">
      </form>
</body>
</html>
```

После заполнения формы информация передается сервлету SalaryProcessor.java. Полученную из формы информацию сохраняем в куку, которая потом отправляется на сервер.

```
import java.io.IOException;
import java.net.URLEncoder;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
```

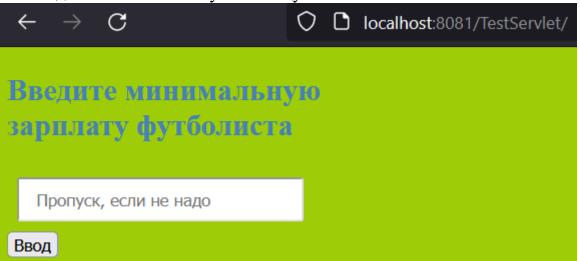
```
mport jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
 * Servlet implementation class SalaryProcessor
public class SalaryProcessor extends HttpServlet {
      private static final long serialVersionUID = 1L;
    * Default constructor.
   public SalaryProcessor() {
       // TODO Auto-generated constructor stub
   protected void processRequest(HttpServletRequest request, HttpServletResponse
            response) throws ServletException, IOException{
      response.setContentType("text/html; charset=UTF-8");
      request.setCharacterEncoding("utf-8");
      // Получение параметра из строки запроса
      String parameter = request.getParameter("salary"); //может быть null
      // Сохранение зарплаты в сессии
      request.getSession().setAttribute("salary", parameter);
      // Сохранение зарплаты в Cookie
      Cookie c = new Cookie("salary", URLEncoder.encode(parameter, "UTF-8"));
      // Установка времени жизни Cookie в секундах
      c.setMaxAge(100);
      response.addCookie(c);
      // Перенаправление на страницу
      response.sendRedirect(response.encodeRedirectURL(request.getContextPath() +
      "/TeamTitle.jsp"));
   }
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
      protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            // TODO Auto-generated method stub
            response.getWriter().append("Served at:
').append(request.getContextPath());
            Cookie cookie = new Cookie("salary", "17000");
            response.addCookie(cookie);
            processRequest(request, response);
      }
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            processRequest(request, response);
```

Затем происходит переход на следующую страницу TeamTitile.jsp, которая уже работает с кукой

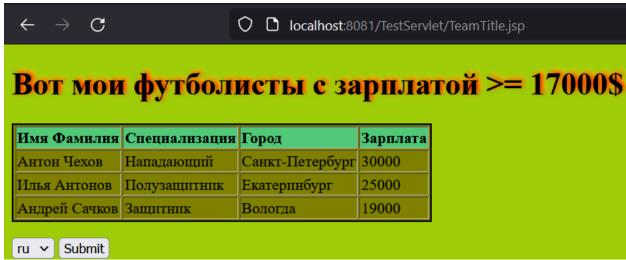
```
c%@ page import="java.util.ResourceBundle"%>
c%@ page import="java.util.Locale"%>
<%@ page import="java.util.Locale %>
<%@ page language="java" contentType="text/html; charset=UTF-8"</pre>
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
       <meta charset="UTF-8">
       <title>Список футболистов</title>
       k rel="stylesheet" href="style.css">
</head>
<body>
       request.setCharacterEncoding("UTF-8");
       //String salary = request.getParameter("salary");
       String salary = (String)session.getAttribute("salary");
       String lang = request.getParameter("lang");
       if (lang == null) lang = "ru";
       if (!"en".equalsIgnoreCase(lang) && !"ru".equalsIgnoreCase(lang)) {
               response.sendError(HttpServletResponse.SC_NOT_ACCEPTABLE,
               "Параметр lang может принимать значения ru или en вместо \"" + lang +
'\"");
              return;
       if (salary.equals("")) salary = null;
       ResourceBundle res = ResourceBundle.getBundle("team", new Locale(lang));
       %>
       <%//@include file="Header.jsp"%>
       <h1>
               <%=res.getString("title") %>
               <%=(salary == null)? " " : (res.getString("condition") + salary + "$")</pre>
%>
       </h1>
       Object[][] team = new Object[][] {
              {"Билли Херрингтон", 0, "Махачкала", 15000}, 
{"Антон Чехов", 1, "Санкт-Петербург", 30000}, 
{"Илья Антонов", 2, "Екатеринбург", 25000}, 
{"Андрей Сачков", 3, "Вологда", 19000}
       String[] roles = new String[] {"Вратарь", "Нападающий", "Полузащитник",
"Защитник"};
<b><%=res.getString("name") %></b>
              <b</s>ess.getString("spec") %></b><b</s>ess.getString("city") %></b>
              <b><%=res.getString("salary") %></b>
       <%
       for (Object[] temp : team)
```

Демонстрация результатов

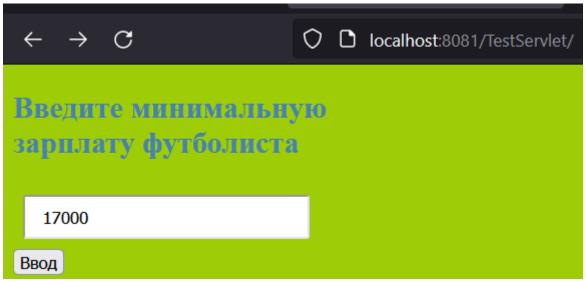
Приложение приветствует формой, которую можно не заполнять, если необходимо вывести полную таблицу



После ввода значения 17000 выводится таблица с игроками, чья зарплата больше 17000



Если закрыть и снова открыть стартовую страницу, то в форме появляется предыдущее вводимое значение, значит информация в виде куки сохранилась



Остальной функционал работает так же, как и в предыдущих лабораторных: например, если ввести неверный формат зарплаты, то выдаст ошибку, и если поменять язык, то он успешно поменяется

Некорректно введенная зарплата

Значение должно быть целым числом без литералов

