Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт по лабораторной работе № 6
на тему: “ОРГАНИЗАЦИЯ ПЕРЕДАЧИ ДАННЫХ МЕЖДУ ЗАПРОСАМИ ПОЛЬЗОВАТЕЛЕЙ”
по дисциплине
“Web-программирование”

Выполнили студенты гр.9308: Дементьев Д.П.

Проверил: Павловский М.Г.

Санкт-Петербург, 2021 г.

Оглавление

Введение

Целью работы является знакомство с методами передачи информации между соединениями, открываемыми в рамках одного сеанса работы пользователя.

Реализация поиска сообщений пользователя

В данной лабораторной работе реализуем поиск всех сообщений пользователя по заданному имени. Для удобства работы добавим сущность Записи (Post) и хранилища для всех записей (Storage):

```
package com.example.models;

import com.example.utils.LocaleManager;

public class Post implements Cloneable {
    private String username;
    private String comment;
    private String date;

    public Post(String username, String comment) {
        this.username = username;
        this.comment = comment;
        this.date = LocaleManager.getTime();
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getComment() {
        return comment;
    }

    public void setComment(String comment) {
        this.comment = comment;
    }

    private void setDate(String date) {
        this.date = date;
    }

    public String getDate() {
        return date;
    }

    public Post copy() {
        Post copy = new Post(this.username, this.comment);
        copy.setDate(this.date);
        return copy;
    }
}
```

Реализация простенького хранилища в оперативной памяти:

```
package com.example.storage;

import com.example.models.Post;

import java.util.ArrayList;
import java.util.List;
```

```
public class Storage {
    private static List<Post> postList = new ArrayList<>();

    static {
        postList.add(new Post("ДимонЛимон", "Всем привет в этом чатике"));
        postList.add(new Post("Чипибарум", "Жду обновлений приложения с нетерпе-
нием!"));
    }

    public static void save(String username, String comment) {
        if (username != null && comment != null)
            postList.add(new Post(username, comment));
    }

    public static List<Post> getPostList() {
        return postList;
    }

    public static List<Post> findByName(String username) {
        List<Post> result = new ArrayList<>();
        for (Post post: postList) {
            if (post.getUsername().equals(username)) {
                result.add(post.copy());
            }
        }
        return (result.size() > 0) ? result : null;
    }
}
```

Теперь мы можем создать форму search_form (/search через web.xml) для поиска всех сообщений пользователя с сохранением последнего запроса поиска при помощи Cookie файлов:

```
<%@page import="java.net.URLDecoder"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding=
        "UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Поиск сообщений пользователя</title>
</head>
<body>
<form METHOD=GET action="usernameProcessor">
    Введите имя пользователя <br>
    <INPUT TYPE=TEXT NAME="username"
        <%
        // Выбор всех Cookie
        Cookie [] c = request.getCookies();
        if(c != null)
            for(int i = 0; i < c.length; i++)
            if("post.author".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="Ввод">
</form>
</body>
</html>
```

Разработанная форма будет отсылать GET запрос на следующий сервлет, ответственный за обработку запроса поиска:

```
package com.example.servlets;

import com.example.storage.Storage;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
import java.net.URLEncoder;

@WebServlet(name = "userProcessorServlet", value = "/usernameProcessor")
public class UsernameProcessor extends HttpServlet {

    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("username");
        if(parameter == null) {
            // Сообщение об ошибке, если сервлет вызван без параметра
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Не задано имя
пользователя");
            return;
        }
        // Проверка, что такой пользователь есть
        boolean haveComments = Storage.findByName(parameter) != null;
        if(!haveComments) { // Сообщение об ошибке, если пользователь не найден
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Пользователь " +
parameter + " не найден");
            return;
        }
        // Сохранение автора в сессии
        request.getSession().setAttribute("username", parameter);
        // Сохранение автора в Cookie
        Cookie c = new Cookie("post.author", URLEncoder.encode(parameter, "UTF-8"));
        // Установка времени жизни Cookie в секундах
        c.setMaxAge(100);
        response.addCookie(c);
        // Перенаправление на страницу книг
        response.sendRedirect(response.encodeRedirectURL(request.getContextPath() +
                "/search_result"));
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

При соблюдении необходимых условий (заданном имени пользователя для поиска, последующем успешном нахождении комментариев от пользователя) совершается редирект на JSP страницу AskUserName (/search_result через web.xml), ответственную за отображение результатов поиска:

```
<%@ page import="com.example.utils.LocaleManager" %>
<%@ page import="com.example.storage.Storage" %>
<%@ page import="com.example.models.Post" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
         pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Список сообщений пользователя
<%=(String)session.getAttribute("username")%></title>
</head>
<body>
<h1>Список сообщений</h1>
<table border='1'>
    <tr>
        <td><b> <%=LocaleManager.getString("user")%> </b></td>
        <td><b> <%=LocaleManager.getString("comment")%> </b></td>
        <td><b> <%=LocaleManager.getString("date")%> </b></td>
    </tr>
    <%
        for (Post post: Storage.findByName((String)session.getAttribute("username")))
{
            out.println("<tr>"+
                    "<td>"+post.getUsername()+"</td>"+
                    "<td>"+post.getComment()+"</td>"+
                    "<td>"+post.getDate()+"</td>"+
                    "</tr>");
        }
    %>
</table>
<%-- вернуться на главную страницу --%>
<br>
<a href='/posts'>На главную страницу</a>

</body>
</html>
```

Демонстрация результатов

Работа начинается с главной страницы приложения, теперь время комментария отображается корректно, а не создается каждый раз заново, также была добавлена кнопка для поиска сообщений пользователя (переход на форму search_form.jsp):

Все записи сообщества

Пользователь	Комментарий	Дата
ДимонЛимон	Всем привет в этом чатике	12-сентября-2021 16:38:38
Чипибарум	Жду обновлений приложения с нетерпением!	12-сентября-2021 16:38:38

Новый пост
Поиск сообщений пользователя
Доступные языки: ru en de

Рис.1. Главная страница приложения

Создадим несколько новых записей для более полной демонстрации работы (процесс создания записей рассмотрен в прошлых лабораторных работах):

Все записи сообщества

Пользователь	Комментарий	Дата
ДимонЛимон	Всем привет в этом чатике	12-сентября-2021 16:38:38
Чипибарум	Жду обновлений приложения с нетерпением!	12-сентября-2021 16:38:38
ДимонЛимон	тестим лабу6	12-сентября-2021 16:41:16
Владимир	где донаты?	12-сентября-2021 16:41:47
ДимонЛимон	я тут	12-сентября-2021 16:41:57

Новый пост
Поиск сообщений пользователя
Доступные языки: ru en de

Рис.2. Добавленные комментарии

Теперь можем осуществить поиск по всем сообщения – жмём соответствующую кнопку и переходим на форму:

Введите имя пользователя

Ввод

Рис.3. Форма поиска

Введём имя пользователя «ДимонЛимон» и нажмём Ввод:

Список сообщений

Пользователь	Комментарий	Дата
ДимонЛимон	Всем привет в этом чатике	12-сентября-2021 16:38:38
ДимонЛимон	тестим лабу6	12-сентября-2021 16:41:16
ДимонЛимон	я тут	12-сентября-2021 16:41:57

На главную страницу

Рис.4. Результат поиска по запросу «ДимонЛимон»

Перейдем на главную страницу, затем снова попробуем осуществить поиск:

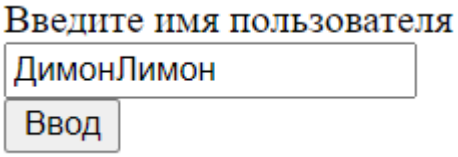

Рис.5. Сохранение последнего запроса в форме

Видим, что прошлый запрос успешно сохранен в Cookie и сразу записан в поле ввода, но сейчас попробуем вызвать ошибку, поэтому впишем несуществующее имя пользователя, например «НесуществующийПользователь»:

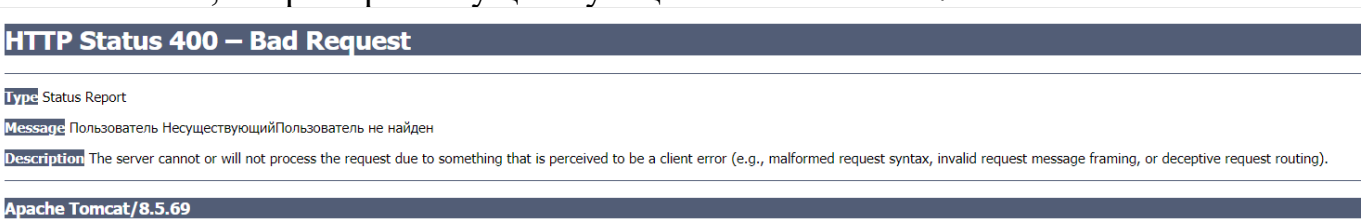

Рис.6. Ошибка поиска

Ожидаемо получаем ошибку, ведь пользователь с таким именем комментариев не оставлял.

Вывод

В ходе выполнения лабораторной работы была изучены методы передачи информации между соединениями, открываемыми в рамках одного сеанса работы пользователя, также была реализована работа с Cookie файлами.