

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”

Выполнил студент гр.9308:

Яловега Н.В.

Проверил:

Павловский М.Г.

Оглавление

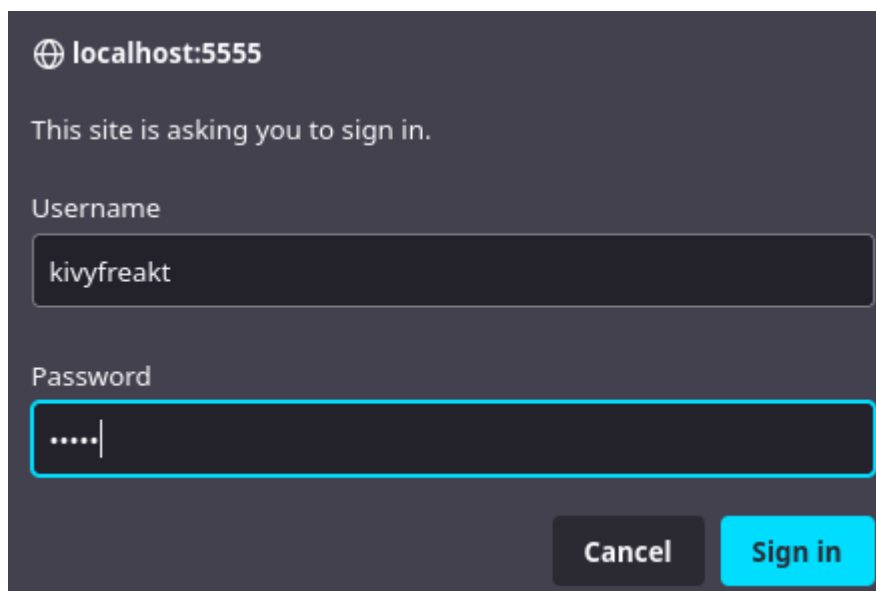
Введение.....	3
Скриншоты, иллюстрирующие работу приложения.....	3
Вывод.....	6
Приложение 1. (Post.java).....	7
Приложение 2. (Dbposts.java).....	9
Приложение 3. (FormProcess.java).....	10
Приложение 4. (Main.jsp).....	12
Приложение 5. (Post.jsp).....	14
Приложение 6. (User.jsp).....	15
Приложение 7. (UserFeed.jsp).....	17
Приложение 8. (UserFeed.jsp).....	19

Введение

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

Скриншоты, иллюстрирующие работу приложения

Авторизируемся под пользователем kivyfreakt



The screenshot shows a dark-themed login window. At the top left, there is a globe icon followed by the text 'localhost:5555'. Below this, the text 'This site is asking you to sign in.' is displayed. There are two input fields: 'Username' with the value 'kivyfreakt' and 'Password' with masked characters '.....'. At the bottom right, there are two buttons: 'Cancel' and 'Sign in'.

Рисунок 1: Авторизация

Добавим несколько новых записей, для того, чтоб протестировать поиск.

Записи

Поиск

[ru](#) / [en](#)

Введите свое сообщение...

Опубликовать

kivyfreakt

11-декабря-2021 15:53:10

Hello World!

asscar

11-декабря-2021 15:53:10

leninapaket

kivyfreakt

11-декабря-2021 15:53:10

тест

kivyfreakt

11-декабря-2021 15:53:10

123

kivyfreakt

11-декабря-2021 15:53:10

впаывпаывпаывпа

Рисунок 2: Добавление записей

В строку поиск введем наш никнейм:

Записи

Рисунок 3: Строка поиска

Получим следующую страницу:

Записи

kivyfreakt

[ru](#) / [en](#)

kivyfreakt

11-декабря-2021 15:53:38

Hello World!

kivyfreakt

11-декабря-2021 15:53:38

тест

kivyfreakt

11-декабря-2021 15:53:38

123

kivyfreakt

11-декабря-2021 15:53:38

впаывпаывпаывпа

Рисунок 4: Страница с результатами поиска

Как видим, поиск работает корректно. Теперь попробуем вернуться на предыдущую страницу и проверим, работают ли cookie файлы для поля поиска.

Записи

Рисунок 5: Проверка файлов Cookie

Видим, что никнейм, который мы вводили в предыдущий раз восстановился на странице, следовательно все работает корректно.

Вывод

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

Приложение 1. (Post.java)

```
package model;

import java.util.List;
import util.LocaleUtil;

public class Post
{
    private String username;
    private String text;
    private String date;
    private List<String> likes;

    public Post(String username, String text)
    {
        this.username = username;
        this.text = text;
        this.date = LocaleUtil.getTime();
        this.likes = null;
    }

    public String getUsername()
    {
        return username;
    }

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

    public String getText()
    {
        return text;
    }
}
```

```
public void setText(String text)
{
    this.text = text;
}

public String getDate()
{
    return date;
}
}
```


Приложение 2. (Dbposts.java)

```
package service;

import java.util.List;
import java.util.ArrayList;
import model.Post;

public class Dbposts
{
    private static List<Post> postList = new ArrayList<Post>();
    static
    {
        postList.add(new Post("kivyfreakt", "Hello World!"));
        postList.add(new Post("asscar", "leninapaket"));
    }

    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);
            }
        }
        return (result.size() > 0) ? result : null;
    }
}
```

Приложение 3. (FormProcess.java)

```
package servlet;

import service.Dbposts;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.net.URLEncoder;

public class FormProcess extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
    {
        response.setContentType("text/html; charset=UTF-8");
        request.setCharacterEncoding("utf-8");

        String parameter = request.getParameter("search_username");
        if(parameter == null)
        {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Не задано
имя пользователя");
            return;
        }
        // Проверка, что такой пользователь есть
        boolean haveComments = Dbposts.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("search_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);

        response.sendRedirect(response.encodeRedirectURL(request.getContextPath() +
"/user"));
    }
}
```

```
@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);
}
}
```

Приложение 4. (Main.jsp)

```
<%@ page import="util.LocaleUtil, java.util.Base64, java.net.URLDecoder" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" errorPage="/Error.jsp"%>
<!DOCTYPE html>
<html>
<%
    request.setCharacterEncoding("UTF-8");

    String authorization = request.getHeader("authorization");
    String enc = authorization.substring("Basic".length()).trim();
    String pair = new String(Base64.getDecoder().decode(enc));
    session.setAttribute("username", pair.split(":")[0]);

    String text = request.getParameter("text");
    if (text != null && text.length() > 140)
        throw new IllegalArgumentException();

    String lang = request.getParameter("lang");

    if (!LocaleUtil.setup(lang)) {
        response.sendError(HttpServletResponse.SC_NOT_ACCEPTABLE,
"Приложение не поддерживает запрашиваемый язык");
        return;
    }

%>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css/style.css">
<title><%=LocaleUtil.getString("title")%></title>
</head>
<body>
    <header>
        <div id="header-content">
            <div id="header">
                <h1><%=LocaleUtil.getString("header")%></h1>
            </div>
            <div id="search">
                <form method="get" action="FormProcess">
                    <input name="search_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;
        }
    %>>
    <button type="submit"><%=LocaleUtil.getString("search")%></button>
</form>
</div>
<div id="locale">
    <a href="/social/?lang=ru">ru</a>
    /
    <a href="/social/?lang=en">en</a>
</div>
</div>
</header>
<main>
    <%@include file="Post.jsp"%>
    <%@include file="Feed.jsp"%>
</main>
</body>
</html>

```

Приложение 5. (Post.jsp)

```
<%@ page import="util.LocaleUtil" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<article>
    <div class = "article-content">
        <form action="/social/" method="post">
            <textarea name="text" placeholder='<%=LocaleUtil.getString("text")%>'
rows="3"></textarea>
            <br>
            <hr>
            <button id="post-btn" type="submit"><%=LocaleUtil.getString("submit")%></button>
        </form>
    </div>
</article>
```

Приложение 6. (User.jsp)

```
package servlet;
import service.Dbposts;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.net.URLEncoder;

public class FormProcess extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
    {
        response.setContentType("text/html; charset=UTF-8");
        request.setCharacterEncoding("utf-8");

        String parameter = request.getParameter("search_username");
        if(parameter == null)
        {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Не задано
имя пользователя");
            return;
        }
        // Проверка, что такой пользователь есть
        boolean haveComments = Dbposts.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("search_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);

        response.sendRedirect(response.encodeRedirectURL(request.getContextPath() +
"/user"));
    }
}
```

```
@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);
}
}
```


Приложение 7. (UserFeed.jsp)

```
package servlet;

import service.Dbposts;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.net.URLEncoder;

public class FormProcess extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
    {
        response.setContentType("text/html; charset=UTF-8");
        request.setCharacterEncoding("utf-8");

        String parameter = request.getParameter("search_username");
        if(parameter == null)
        {
            response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Не задано
имя пользователя");
            return;
        }
        // Проверка, что такой пользователь есть
        boolean haveComments = Dbposts.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("search_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);

        response.sendRedirect(response.encodeRedirectURL(request.getContextPath() +
"/user"));
    }
}
```

```
@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);
}
}
```

Приложение 8. (UserFeed.jsp)

```
<%@ page import="util.LocaleUtil, service.Dbposts, model.Post" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<%

    String username = (String)session.getAttribute("username");
    String post_text = request.getParameter("text");
    Dbposts.save(username, post_text);

    for (Post post: Dbposts.getPostList() )
    {
        out.println("<article><div class = 'article-content'>" +
            "<h2 class = 'username'>" + post.getUsername() + "</h2>" +
            "<div class = 'date'>" + LocaleUtil.getTime() + "</div><hr>" +
            "<div class = 'post-text'>" + post.getText() + "</div>" +
            "</div></article>");
    }
%>
```