

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И.УЛЬЯНОВА (ЛЕНИНА)**

Кафедра ВТ

**ОТЧЕТ
по лабораторной работе №4
по дисциплине «Java программирование
интернет-приложений»
Тема: Интернационализация WEB-приложений**

Репин С.А.

Преподаватель

Павловский М.Г.

**Санкт-Петербург
2022**

СОДЕРЖАНИЕ

1. Цель работы	3
2. Выполнение работы	3
3. Вывод	4
Исходный код сервлета	4
Исходный код ресурсов (английский язык)	8
Исходный код ресурсов (русский язык)	8

1. ЦЕЛЬ РАБОТЫ

Знакомство со способами отображения данных на различных языках при использовании файлов ресурсов.

2. ВЫПОЛНЕНИЕ РАБОТЫ

В приложение, разработанное в предыдущей работе, были добавлены ресурсы, необходимые для интернационализации приложения на русский и английский языки.

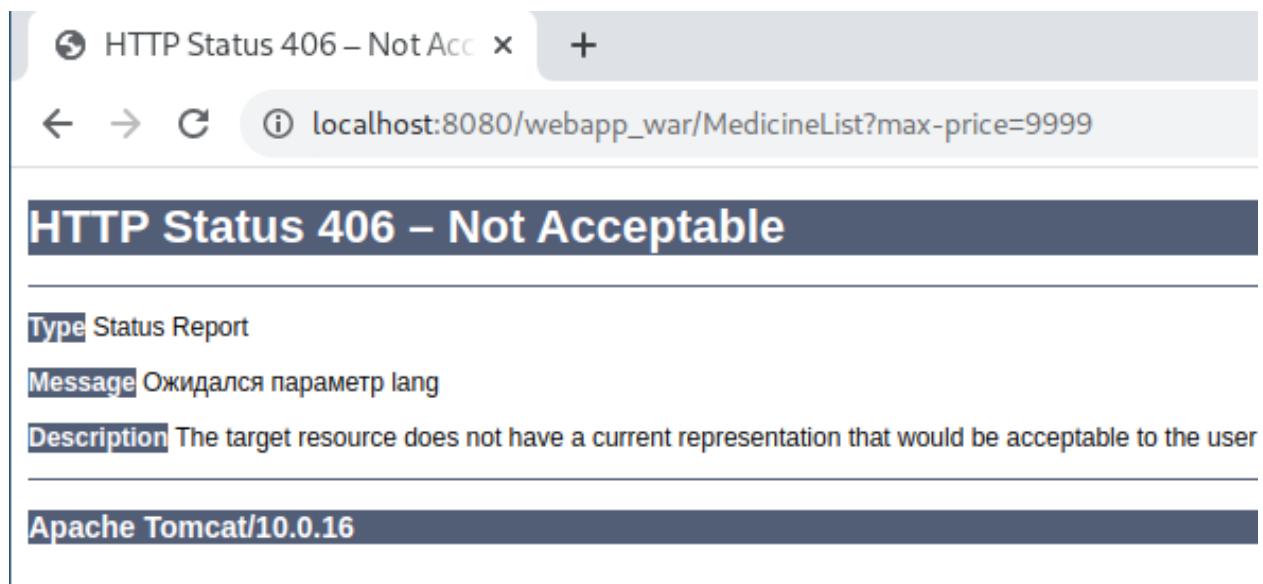


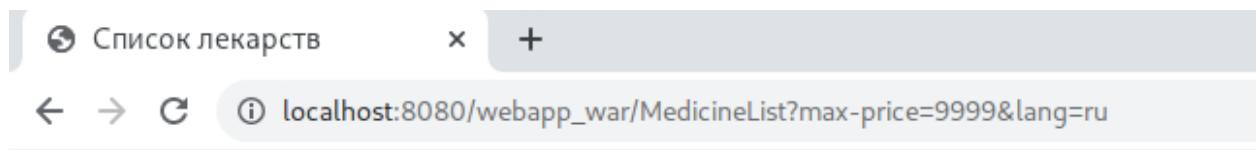
Рис. 2.1 Указание неверного языка

The screenshot shows a web browser window with the following details:

- Address bar: localhost:8080/webapp_war/MedicineList?max-price=9999&lang=en
- Content:
 - List of medicines cheaper than 9999 rub/unit**
 - A table with the following data:

ID	Name	Quantity	Unit Price
0	AAA	10	100
1	BBB	32	200
2	CCC	1	220
3	DDD	56	1340
4	EEE	23	50

Рис. 2.2 Приложение на английском языке



Список лекарств дешевле 9999 руб/шт

Идентификатор	Имя	Количество	Цена за ед
0	AAA	10	100
1	BBB	32	200
2	CCC	1	220
3	DDD	56	1340
4	EEE	23	50

Рис. 2.3 Приложение на русском языке

3. ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки работы с ресурсами приложений и выполнения на их основе интернационализации веб-приложений.

ИСХОДНЫЙ КОД СЕРВЛЕТА

```
package app.servlet;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
import java.util.ResourceBundle;

/**
 * Лекарство в аптеке.
 */
class Medicine {
    public int id;
    public String name;
```

```

public int quantity;
public int unitPrice;

/**
 * Конструктор.
 * @param id ID
 * @param name Имя
 * @param quantity Количество на складе
 * @param unitPrice Цена единицы
 */
public Medicine(int id, String name, int quantity, int unitPrice) {
    this.id = id;
    this.name = name;
    this.quantity = quantity;
    this.unitPrice = unitPrice;
}

/**
 * Возвращает список строковых значений полей класса.
 * @return Список строк
 */
public List<String> fieldsToStringList() {
    return List.of(
        String.valueOf(this.id),
        this.name,
        String.valueOf(this.quantity),
        String.valueOf(this.unitPrice)
    );
}
}

/**
 * Сервлет, отображающий список лекарств.
*/
@WebServlet(name = "MedicineList", value = "/MedicineList")
public class MedicineList extends HttpServlet {
    /**
     * Список лекарств.
     */
    private final List<Medicine> medicines;

    /**
     * Конструктор.
     */
    public MedicineList() {
        super();
        medicines = new ArrayList<>();
        medicines.add(new Medicine(0, "AAA", 10, 100));
        medicines.add(new Medicine(1, "BBB", 32, 200));
        medicines.add(new Medicine(2, "CCC", 1, 220));
        medicines.add(new Medicine(3, "DDD", 56, 1340));
        medicines.add(new Medicine(4, "EEE", 23, 50));
    }
}

```

```

}

/**
 * Выполняет GET и POST HTTP-запросы.
 *
 * @param request Запрос к сервлету
 * @param response Ответ сервлета
 * @throws ServletException Внутренняя ошибка
 * @throws IOException Ошибка ввода вывода
 */
protected void processRequest(HttpServletRequest request,
                             HttpServletResponse response)
    throws ServletException, IOException {

    var lang = request.getParameter("lang");
    if (lang == null) {
        response.sendError(HttpServletResponse.SC_NOT_ACCEPTABLE,
                          "Ожидался параметр lang");
        return;
    }
    if (!"en".equalsIgnoreCase(lang) && !"ru".equalsIgnoreCase(lang))
        response.sendError(HttpServletResponse.SC_NOT_ACCEPTABLE,
                          "Параметр lang может принимать значения en или ru");
    return;
}

var res = ResourceBundle.getBundle(
    "/MedicineList",
    new Locale(lang));
request.setCharacterEncoding("utf-8");
response.setContentType("text/html;charset=UTF-8");

var max_price_str = request.getParameter("max-price");
var max_price = (max_price_str == null)
    ? 0
    : Integer.parseInt(max_price_str);
var table_content = buildTableHtml(max_price);

var head_title = res.getString("head.title");
var body_title = res.getString("body.title");
var price_units = res.getString("price_units");
var id = res.getString("table.id");
var name = res.getString("table.name");
var quantity = res.getString("table.quantity");
var unit_price = res.getString("table.unit_price");

try (PrintWriter out = response.getWriter()) {
    out.println(
        "<html>" +
        "<head><title>" + head_title + "</title></head>" +
        "<body>" +
        "<h1>"
```

```

                + body_title + " "
                + max_price + " " + price_units
            + "</h1>"
            + "<table border='1'>"
            + "<tr>"
                + "<td><b>" + id + "</b></td>"
                + "<td><b>" + name + "</b></td>"
                + "<td><b>" + quantity + "</b></td>"
                + "<td><b>" + unit_price + "</b></td>"
            + "</tr>"
            + table_content
            + "</table>"
        + "</body> +
    "</html>"
);
}
}

/**
 * Создает содержимое таблицы для всех лекарств, стоимость единицы
 * которых меньше, чем 'max_price'.
 * @param max_price Максимальная цена (не включая)
 * @return HTML-содержимое таблицы
 */
private String buildTableHtml(int max_price) {
    StringBuilder sb = new StringBuilder();
    for (var med : this.medicines) {
        if (med.unitPrice >= max_price) {
            continue;
        }
        sb.append("<tr>");
        for (String field : med.fieldsToStringList()) {
            sb.append("<td>");
            sb.append(field);
            sb.append("</td>");
        }
        sb.append("</tr>");
    }
    return sb.toString();
}

/**
 * Выполняет GET HTTP-запрос.
 *
 * @param request Запрос к серверу
 * @param response Ответ сервера
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
        throws ServletException, IOException {
    processRequest(request, response);
}

```

```

    /**
     * Выполняет POST HTTP-запрос.
     *
     * @param request Запрос к сервлету
     * @param response Ответ сервлета
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

ИСХОДНЫЙ КОД РЕСУРСОВ (АНГЛИЙСКИЙ ЯЗЫК)

```

head.title = Medicine List
body.title = List of medicines cheaper than
price_units = rub/unit
table.id = ID
table.name = Name
table.quantity = Quantity
table.unit_price = Unit Price

```

ИСХОДНЫЙ КОД РЕСУРСОВ (РУССКИЙ ЯЗЫК)

```

head.title = Список лекарств
body.title = Список лекарств дешевле
price_units = руб/шт
table.id = Идентификатор
table.name = Имя
table.quantity = Количество
table.unit_price = Цена за ед

```