**МИНОБРНАУКИ РОССИИ**

**САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**ОТЧЕТ**

**по лабораторным работам №2**

**по дисциплине «Программирование интернет-приложений на языке Java»**

**Тема: построение Web-приложений**

**с использованием технологии сервлетов**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 7308 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Фомченкова Н.А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Павловский М.Г. |

Санкт–Петербург

2021

**Цель работы**

Знакомство с технологией построения Web-приложений на основе сервлетов.

**Задание**

* Выберите предметную область для лабораторной работы и разработайте для нее Web-приложение.
* Откройте проект для Web-приложения.
* Разработайте HTML-страницу и сервлет для Web-приложения.
* Сервлет должен принимать параметры инициализации и один или более динамических параметров.
* Создайте war-архив и поместите его на сервере Apache Tomcat.
* Запустите Web-сервер и обратитесь к сервлету из браузера, проверьте правильность отображения HTML-страницы и работу сервлета.
* Сгенерируйте документацию для сервлета с помощью Javadoc и просмотрите ее в браузере.

**Ход работы**

Разработаем Web-приложение «Список врачей». Как и в примере с книгами, возможно передать динамический параметр (в нашем случае имя пациента) или вызвать без параметра (имя пациента будет не указано).

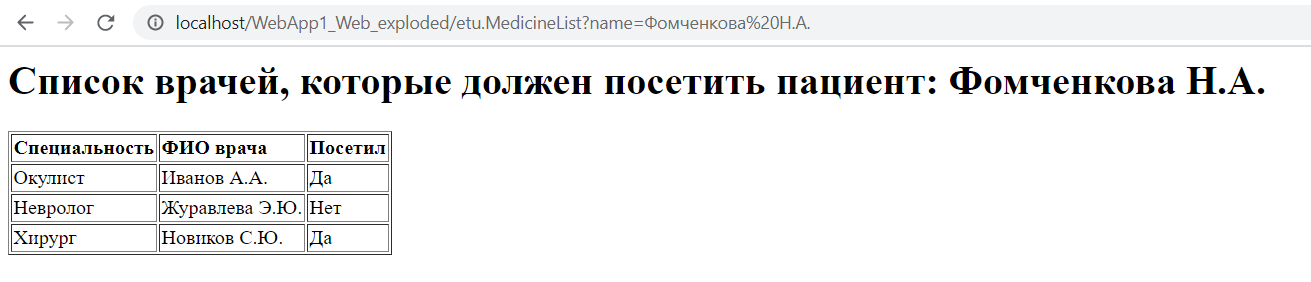


Рис.1 Вызов с динамическим параметром

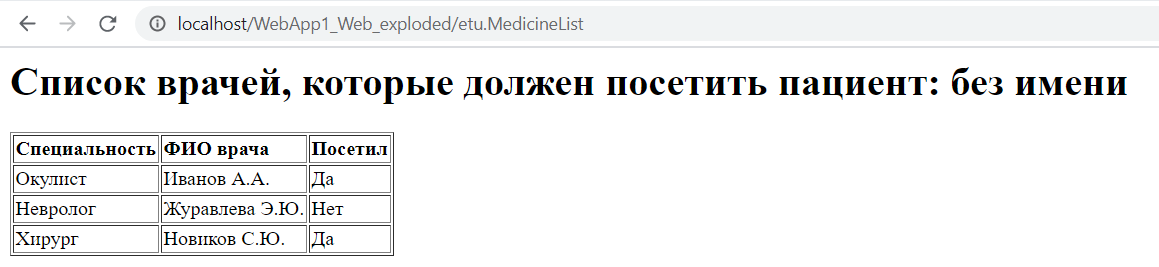


Рис.2 Вызов без параметра

Подключаем сервлеты с помощью зависимости, добавленной в pom.xml.

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

</dependency>

</dependencies>

В процессе выполнения л/р были опробованы два способа описания сервлета и задания пути: через web.xml и с помощью аннотации @WebServlet.

Первый способ: (в web.xml)

<servlet>

<servlet-name>MedicineList</servlet-name>

<servlet-class>etu.MedicineList</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MedicineList</servlet-name>

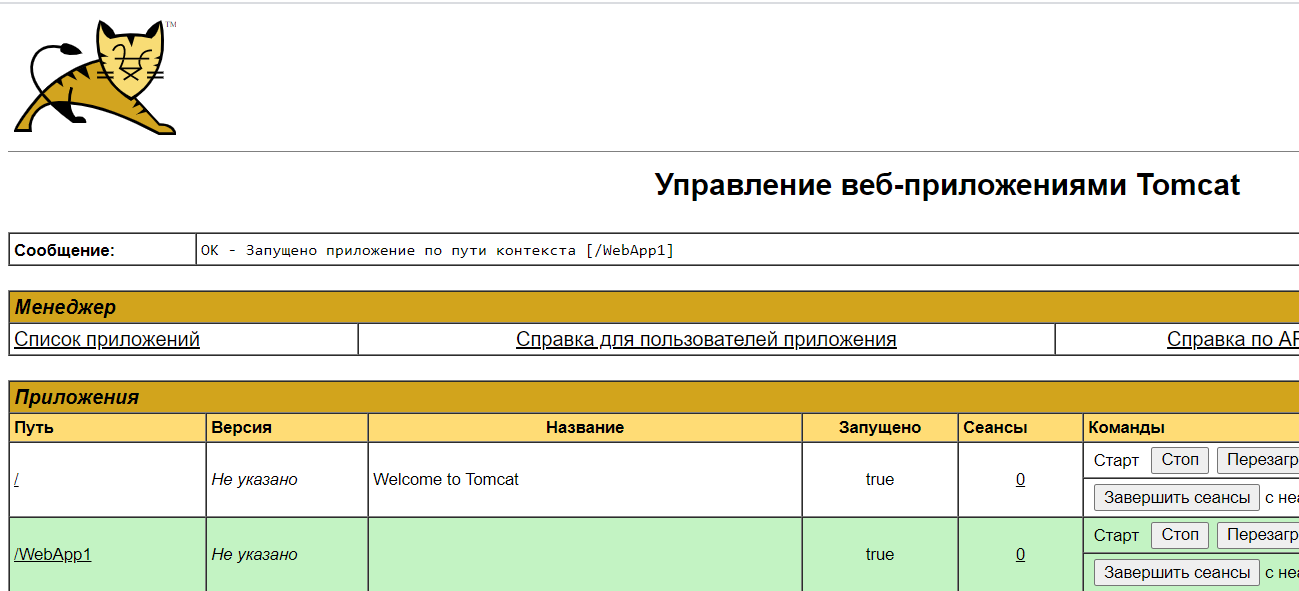
<url-pattern>/etu.MedicineList</url-pattern>

</servlet-mapping>

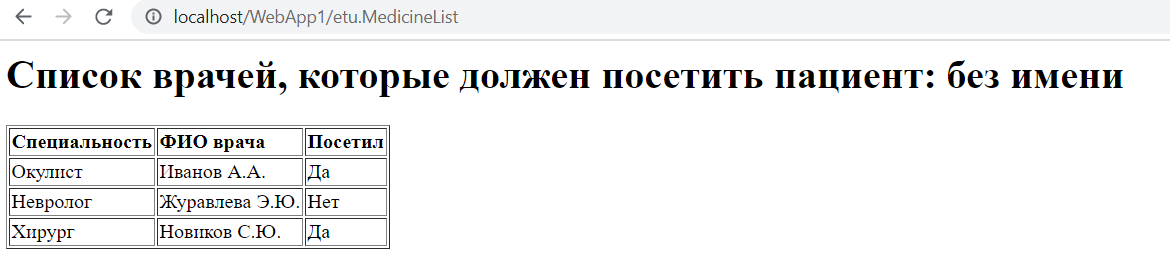
Второй способ: (Пиши перед классом сервлета)

@WebServlet("/etu.MedicineList")

Опытным путем принято решение, что второй способ удобнее. Также при использовании этой аннотации отпадает необходимость описывать каждый созданный сервлет в файле web.xml.  
Создадим war-архив и поместим его на сервер Apache Tomcat.



Как и в запуске из Idea все исправно работает.



**Вывод:** в процессе выполнения лабораторной работы я познакомилась с тем, что из себя представляют сервлеты, для чего они нужны. Разработала сервлет «MedicineList». Научилась создавать war-архивы и запускать из сервиса Apache Tomcat.

Лстинг

MedicineList.java

package etu;  
  
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
// библиотечный класс для работы с потоками вывода  
import java.io.PrintWriter;  
*/\*\*  
 \** ***@author*** *Fomchenkova Nataliya 7308  
 \** ***@date*** *11.02.2021  
 \*/*@WebServlet("/etu.MedicineList")  
public class MedicineList extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 public MedicineList() {  
 super();  
 }protected void processRequest(HttpServletRequest request, HttpServletResponse  
 response)  
 throws ServletException, IOException {  
// Задание типа кодировки для параметров запроса  
 request.setCharacterEncoding("utf-8");  
 // Чтение параметра name из запроса  
 String name = request.getParameter("name");  
 // Задание типа содержимого для ответа (в том числе кодировки)  
 response.setContentType("text/html;charset=UTF-8");  
 // Получение потока для вывода ответа  
 PrintWriter out = response.getWriter();  
 try {  
 // Создание HTML-страницы  
 out.println("<html>");  
 out.println("<head><title>Список врачей</title></head>");  
 out.println("<body>");  
 out.println("<h1>Список врачей, которые должен посетить пациент: " + (name != null ? name :  
 "без имени") + "</h1>");  
 out.println("<table border='1'>");  
 out.println("<tr><td><b>Специальность</b></td><td><b>ФИО врача </b></td><td><b>Посетил</b></td></tr>");  
 out.println("<tr><td>Окулист </td><td>Иванов А.А. </td><td>Да</td></tr>");  
 out.println("<tr><td>Невролог </td><td>Журавлева Э.Ю. </td><td>Нет</td></tr>");  
 out.println("<tr><td>Хирург </td><td>Новиков С.Ю. </td><td>Да</td></tr>");  
 out.println("</table>");  
 out.println("</body>");  
 out.println("</html>");  
 } finally {  
 // Закрытие потока вывода  
 out.close();  
 }  
 }  
@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);  
 }  
  
 @Override  
 public void destroy() {  
 super.destroy();  
 }  
}

pom.xml

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>servlets</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>war</packaging>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
 <dependencies>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 </dependency>  
 </dependencies>  
</project>