

7. Java_EE (основы работы с веб - сервлетами)

- Работа с Tomcat сервером
- Создание, объявление, конфигурация сервлетов
- Параметры Get, Post запросов
- Методы redirect(), forward()
- Использование сессий и cookies
- Шаблонизатор JSP, отображение веб-страниц

File pom.xml: добавление зависимостей

```
<dependencies>
  <dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-api</artifactId>
    <version>8.0.1</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>3.3.0</version>
    </plugin>
  </plugins>
</build>
</project>
```

File first.jsp: импорт классов, вывод данных на экран, использование переменных, обработка данных циклом, вызов методов класса

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <title>First JSP</title>
</head>
<body>
<h1>Test JSP</h1>
<p>
  <%
    java.util.Date now = new java.util.Date();
    String some = "Текущая дата: " + now;
  %>
  <%=some%>
</p>
```

```

<br>
<p>
    <%= new java.util.Date() %>
</p>
<br>
<p>
    <%
        out.println("Hello world");
    %>
</p>
<br>
<p>
    <%
        out.println("Hello world");
        for (int i = 0; i < 5; i++) {
            out.println("<p>" + "Hello: " + i + "</p>");
        }
    %>
</p>
<br>
<p>
    <%@ page import="java.util.Date, logic.TestClass" %>
    <%= new Date() %>
</p>
<br>
<p>
    <%
        TestClass testClass = new TestClass();
    %>
    <%=testClass.getInfo() %>
</p>
<br>
</body>
</html>

```

Запуск Tomcat сервера и отображение страницы в браузере по адресу /first.jsp:



Class HelloWorld: аннотация `@WebServlet`, параметры `name`, `value`, методы `init()` и `destroy()`

```
@WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloWorld extends HttpServlet {
    private String message;

    public void init() {
        message = "Hello World!";
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    }

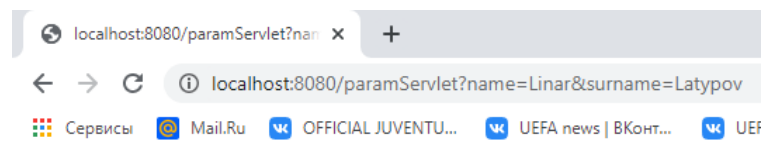
    public void destroy() {
    }
}
```

Class ParamServlet: использование параметров для `get` запроса

```
@WebServlet(name = "ParamServlet", value = "/paramServlet")
public class ParamServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String name = request.getParameter("name");
        String surname = request.getParameter("surname");
        PrintWriter pw = response.getWriter();
        pw.println("<html>");
        pw.println("<h1> Hello, " + name + " " + surname + "</h1>");
        pw.println("</html>");
    }
}
```

Отправка запроса:

`/paramServlet?name=Linar&surname=Latypov`



Hello, Linar Latypov

Class Sessions: использование сессий, объект `Session`, методы `getSession()`, `getAttribute()`, `setAttribute()`

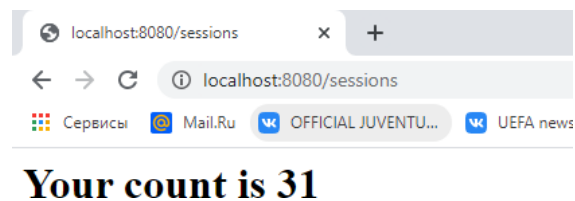
```
@WebServlet(name = "Sessions", value = "/sessions")
public class Sessions extends HttpServlet {
```

```

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    HttpSession session = request.getSession();
    Integer count = (Integer) session.getAttribute("count");
    if (count == null) {
        session.setAttribute("count", 1);
        count = 1;
    } else {
        session.setAttribute("count", count + 1);
    }
    PrintWriter pw = response.getWriter();
    pw.println("<html>");
    pw.println("<h1> Your count is " + count + "</h1>");
    pw.println("</html>");
}

```

При повторных обращениях к странице, счетчик будет увеличиваться:



Class RedirectServlet: использование метода `redirect()`, перенаправление на страницу `MyPage.jsp`:

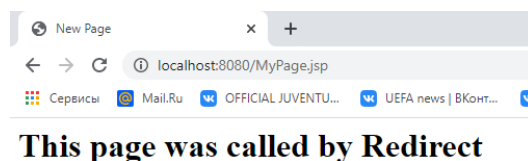
```

@WebServlet(name = "RedirectServlet", value = "/redirect")
public class RedirectServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        response.sendRedirect("/MyPage.jsp");
    }
}
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>New Page</title>
</head>
<body>
<H1>This page was called by Redirect</H1>

</body>
</html>

```

Отображение в браузере при переходе на страницу `/redirect`:



Class ForwardServlet: использование метода forward(). Объекты ServletContext, Dispatcher. Для демонстрации используется вспомогательный класс Cart (наименование, количество), имитирующий корзину в онлайн магазине, а также файл showCart.jsp, для отображения этой корзины

```
@WebServlet(name = "ForwardServlet", value = "/forward")
public class ForwardServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        Cart cart = (Cart) session.getAttribute("cart");

        String name = request.getParameter("name");
        int quantity = Integer.parseInt(request.getParameter("quantity"));

        if (cart == null) {
            cart = new Cart();
            cart.setName(name);
            cart.setQuantity(quantity);
        }
        session.setAttribute("cart", cart);

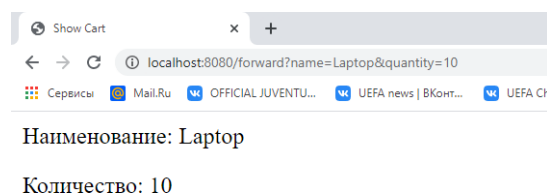
        getServletContext().getRequestDispatcher("/showCart.jsp").forward(request, response);
    }
}
```

File showCart.jsp:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Show Cart</title>
</head>
<body>
<%
    Cart cart = (Cart) session.getAttribute("cart");
%>
<p>Наименование: <%= cart.getName() %></p>
<p>Количество: <%= cart.getQuantity() %></p>
</body>
</html>
```

Отправка запроса:

/forward?name=Laptop&quantity=10



File web.xml: конфигурация сервлетов, параметры servlet-mapping, url-pattern

```
<servlet>
    <servlet-name>SetCookiesServlet</servlet-name>
    <servlet-class>Cookies.SetCookiesServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>SetCookiesServlet</servlet-name>
    <url-pattern>/set-cookies</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>GetCookiesServlet</servlet-name>
    <servlet-class>Cookies.GetCookiesServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>GetCookiesServlet</servlet-name>
    <url-pattern>/get-cookies</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>DeleteCookiesServlet</servlet-name>
    <servlet-class>Cookies.DeleteCookiesServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>DeleteCookiesServlet</servlet-name>
    <url-pattern>/delete-cookies</url-pattern>
</servlet-mapping>
</web-app>
```

Class SetCookiesServlet: создание и добавление новых cookies, объект Cookie, методы addCookie(), setMaxAge().

```
public class SetCookiesServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie cookie1 = new Cookie("some_id", "123");
        cookie1.setMaxAge(24*60*60);
        response.addCookie(cookie1);

        Cookie cookie2 = new Cookie("some_name", "Tom");
        cookie2.setMaxAge(24*60*60);
        response.addCookie(cookie2);
    }
}
```

Class GetCookiesServlet: получение списка Cookies и отображение в браузере

```
public class GetCookiesServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie[] cookies = request.getCookies();
    }
}
```

```

        PrintWriter pw = response.getWriter();
        pw.println("<html>");
        for (Cookie cookie:cookies){
            pw.println("<h1>" +cookie.getName()+ " : " + cookie.getValue() +
"</h1>");
        }
        pw.println("</html>");
    }
}

```

Class DeleteCookiesServlet: удаление Cookies

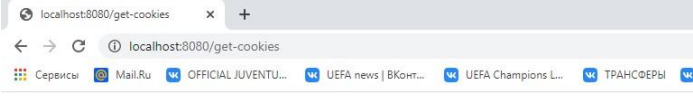
```

public class DeleteCookiesServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie cookie = new Cookie("some_id", "");
        cookie.setMaxAge(0);
        //cookie.setMaxAge(-1); - удалить куки после закрытия

        response.addCookie(cookie);
    }
}

```

Поочередная отправка запросов /set-cookies, /get-cookies:



localhost:8080/get-cookies

Idea-3179c73e : a6d3443f-056e-447b-b3e2-4033f6294888

jenkins-timestampers-offset : -28800000

JSESSIONID : 23CFB529017685282A00651A4594999C

some_id : 123

some_name : Tom

Поочередная отправка запросов /delete-cookies, /get-cookies:



localhost:8080/get-cookies

Idea-3179c73e : a6d3443f-056e-447b-b3e2-4033f6294888

jenkins-timestampers-offset : -28800000

JSESSIONID : 23CFB529017685282A00651A4594999C

some_name : Tom