Основи на Java EE прв дел

Веб програмирање



Сервлет контејнер и сервлети

- Јава платформи JVM + колекција од API за развивање апликации
 - Java Standard Edition (JSE)
 - Јава библиотеки (Јава типови и класи, вмрежување, безбедност, пристап до бази на податоци, ...)
 - Java Enterprise Edition (Java EE)
 - Првично J2EE, па преименувана во Java EE
 - Екстензија на JSE за креирање на скалабилни, повеќеслојни, безбедни и доверливи апликации кои се извршуваат на сервер
 - Воведува управувани компоненти како сервлети, филтри и обсервери од страна на сервлет контејнери



Сервлет

- Java класа која имплементира бизнис логика потребна за креирање динамички веб страници
 - Се повикува во зависност од побараната URL адреса
 - Hema main()
 - Контролиран од сервлет контејнер
 - Имплементира одредена функционалност на апликацијата
 - Една апликација е составена од колекција од сервлети



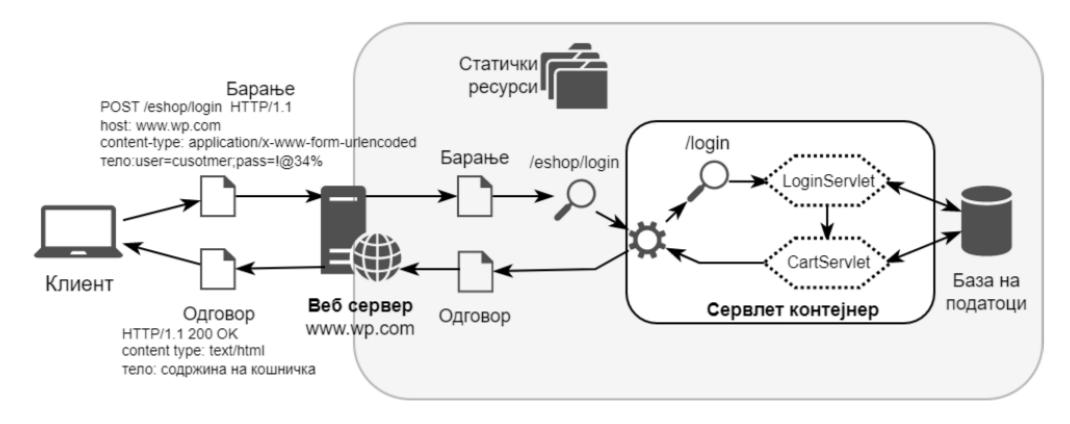
Сервлет контејнер

- Java процес кој управува со сервлети
 - Дел од Java базиран веб сервер (Apache Tomcat)
 - Ја олеснува комуникацијата помеѓу веб серверот и клиентот
 - Прима HTTP барање и го конвертира во инстанца од HttpServletRequest
 - Креира инстанца од HttpServletResponse
 - Ги предава објектите на еден од сервлетите
 - Креира HTTP одговор од инстанцата на HttpServletResponse
 - откако сервлетот ќе заврши со обработка
 - Управува со животниот циклус на сервлетите, филтрите и обсерверите
 - Нуди паралелизација
 - Обработка на повеќе барања одеднаш
 - Нуди заеднички простор за споделување на податоци



Сервлет контејнер и сервлети

• Комуникација помеѓу клиент, веб сервер и контејнер





Филтер и обсервер

- Филтер (Filter) компонента која пресретнува и модифицира барања и одговори пред да бидат предадени на сервлет
- Сервлет контејнерот може да повика повеќе филтри пред да го предаде барањето на сервлетот
- Обсервер компонента која се користи за регистрирање за следење на одредени настани во рамките на сервлет контејнерот и дефиниција на соодветна акција
 - Креирање на барање
 - Креирање на корисничка сесија
 - •

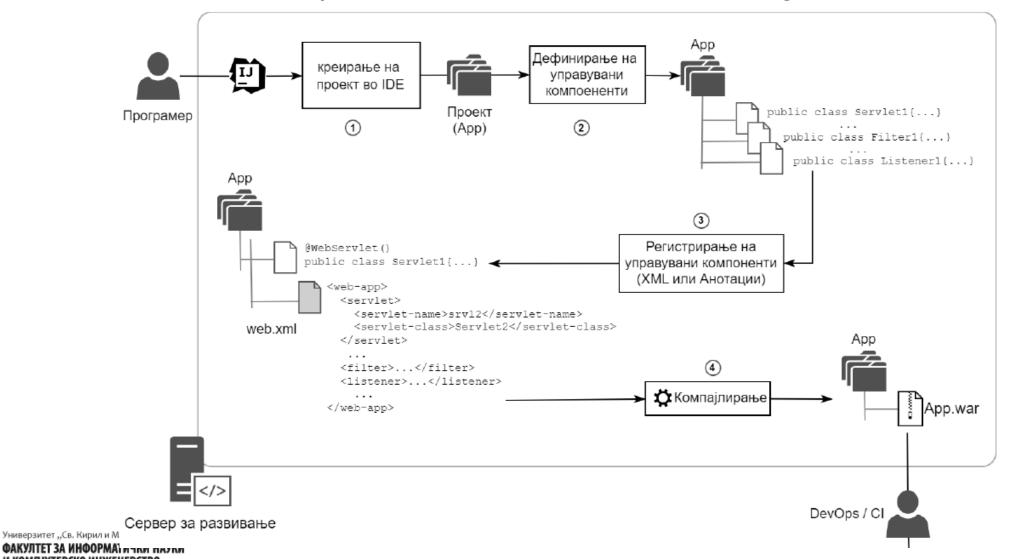


Регистрација на Java EE компоненти

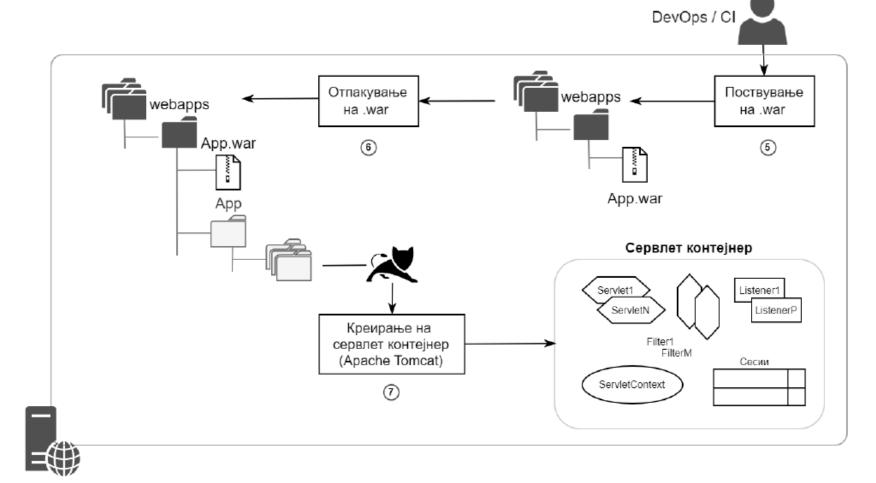
- Управуваните компоненти од типот сервлети, филтри и обсервери се класи кои имплементираат интерфејси од Java EE API
- Сите дефинирани компоненти се регистрираат кај сервлет контејнерот преку
 - Дескриптор за поставување (Deployment descriptor)
 - web.xml
 - Анотации во самите компоненти
- Сите регистрирани компоненти преставуваат дел од апликацијата
- По имплементацијата, заедно со останатите ресурси се пакуваат во архива .war



Животен циклус на веб апликација



Животен циклус на веб апликација





Веб сервер

Веб апликација со Spring Boot

- Содржи интегриран Apache Tomcat сервер
- Апликацијата најчесто се пакува во .jar
- Извршувањето на .jar го подига и вградениот Apache Tomcat сервер
 - Апликацијата е самостојна и не е потребен дополнителен апликациски сервер
- Лесно создавање на продукциска средина
 - Фокус на развој на апликациите
 - Минимален напор за нивно поставување



Сервлет

• Дефиниција на генерички сервлет

 GenericServlet го имплементира интерфејсот Servlet со методи init(), service() и destroy()



Сервлет

• Дефиниција на HttpServlet (за HTTP комуникација)

- Постојат и doPut(), doDelete(), doHead(), doOptions() и doTrace()
- Сите претходни методи се повикани од страна на service()



HttpServletRequest

- String getMethod()
- String getRemoteAddr()
- String getHeader(String header)
- Enumeration getHeaders(String header)
- Enumeration getHeaderNames()
- Enumeration getHeaders()
- String getContentType()
- int getContentLength()
- Cookie[] getCookies()
- String getParameter(String name)
- Enumeration getParameterNames()
- String[] getParameterValues(String name)
- ServletInputStream getInputStream()
- BufferedReader getReader()



HttpServletResponse

- void setHeader(String header, String value)String getRemoteAddr()
- String getHeader(String header)
- void addHeader(String header, String value)
- boolean containsHeader(String header)
- Enumeration<String> getHeaders(String header)
- Enumeration<String> getHeaderNames()
- void setStatus(int status)
- int getStatus(int status)
- void sendError(int status, String message)
- void addCookie(Cookie cookie)
- void setContentType()
- ServletOutputStream getOutputStream()
- PrintWriter getWriter()



Регистрација на сервлети - XML

- За секој сервлет кој се регистрира кај сервлет контејнерот во web.xml се специфицира
 - Име
 - Класа
 - URL пресликување



Регистрација на сервлети - XML

• Пример

```
<web-app>
    <servlet>
        <servlet-name>login</servlet-name>
        <servlet-class>mk.ukim.finki.Login</servlet-class>
   </servlet>
    <servlet>
        <servlet-name>cart</servlet-name>
        <servlet-class>mk.ukim.finki.ShopingCart</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>login</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>cart</servlet-name>
        <url-pattern>/show-cart</url-pattern>
    </servlet-mapping>
</web-app>
```

Пресликување на сервлети - XML

- Регуларни изрази
 - За изразот "/products/*"
 - "/products/e-books" се поклоплува
 - "/products/software?page=2" се поклопува
 - "/products/software/ide" не се поклопува
- Повеќекратни пресликувања
 - Можни се повеќе пресликувања за ист сервлет
 - Една URL адреса може да одговора на повеќе пресликувања
 - Кое пресликување да се одбере?



Пресликување на сервлети - XML

- Правила за повеќекратни пресликувања
 - 1. Се преферира егзактно поклопување наспроти поклопување со регуларен израз.
 - 2. Се преферираат подолгите поклопувања.
 - 3. Ако не се пронајде ниту едно поклопување, се прави предефинирано пресликување на сервлетот одговорен за коренот на апликацијата /.
- Пример
 - Пресликувања "/products/*" и "/products/software/*"
 - Барање со URL "/products/software?name=spring"
 - Се одбира второто пресликување
- Потребно е да постои предефиниран сервлет



Регистрација на сервлети - анотации

• Пример за единечно пресликување

```
@WebServlet(name = "cart", value = "/show-cart")
public class ShoppingCart extends HttpServlet {
    ...
}
```

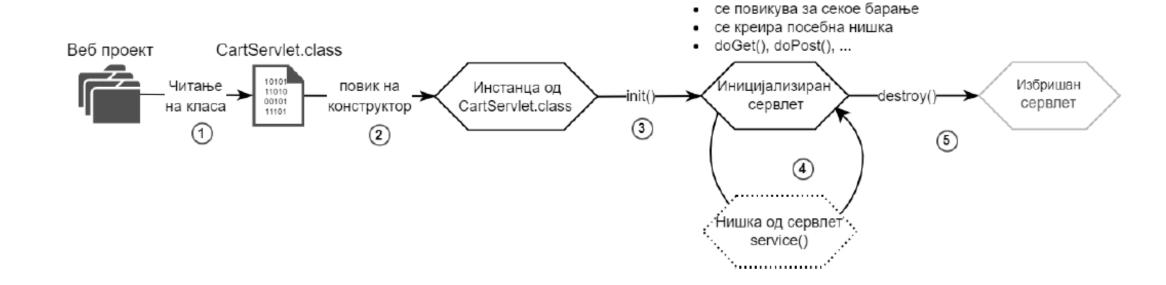
- Пример за повеќекратно пресликување
 - пате може да се изостави (и во горниот пример)

```
@WebServlet(ulrPatterns = {"/show-cart", "/cart"})
public class ShoppingCart extends HttpServlet {
    ...
}
```

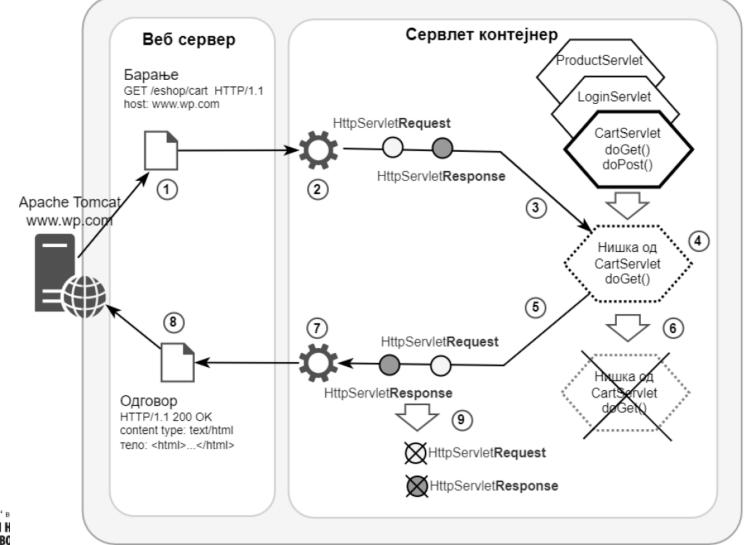
web.xml vs. @WebServlet?



Животен циклус на сервлет

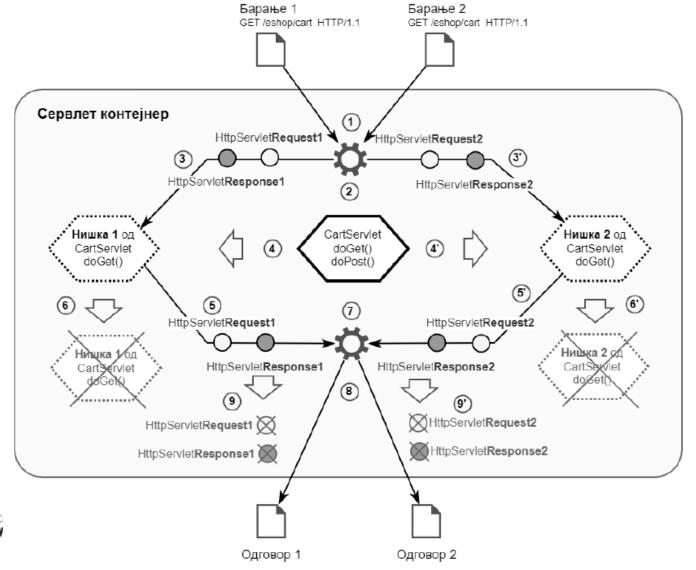


Обработка на барање од сервлет





Обработка на повеќе барања од сервлет







Редослед на вчитување

- Предефинирано однесување: во моментот на првото барање за сервлетот
- Дефиниција на редослед

• Негативна вредност: контејнерот одлучува за редоследот



index.html

Create empty GET request
Create GET request with query string

Insert a value

WP submit



SimpleServlet.class

```
@WebServlet(name = "SimpleServlet", value = "/servlet")
public class SimpleServlet extends HttpServlet {
    public void init() {
        System.out.println("SimpleServlet is initialized!");
    public void doGet(HttpServletRequest request, HttpServletResponse

→ response) throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.print(getHeaders(request));
        out.println("<h5>" + "Message from doGet() method in SimpleSerlvet."

→ + "</h5>");

        if ( request.getParameter("q")!=null)
            out.println("<h6>" + "Received data:" +

→ request.getParameter("q") + "</h6>");
        else
            out.println("<h5>No data received!</h5>");
        out.println("</body></html>");
        int rnd = new Random().nextInt();
        response.addCookie(new Cookie("firstCookie", String.valueOf(rnd)));
        System.out.println("SimpleServlet doGet() invoked!");
```

```
public void doPost(HttpServletRequest request, HttpServletResponse

→ response)

        throws IOException {
   response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.print(getHeaders(request));
    out.println("<h5>" + "Message from doPost in SimpleSerlvet." +
    out.println("<h6>Received data: " +

→ request.getParameter("input_data") + "</h6>");
    out.println("</body></html>");
   System.out.println("SimpleServlet doPost() invoked!");
public void destroy() {
   System.out.println("SimpleServlet is destroyed!");
private String getHeaders(HttpServletRequest request){
    StringBuilder builder = new StringBuilder();
    Enumeration<String> headerNames = request.getHeaderNames();
   while(headerNames.hasMoreElements())
        String header = headerNames.nextElement();
        builder.append(header+": "+request.getHeader(header)+" <br > ");
   return builder.toString();
```

• Резултат од прв линк

host: localhost:8080 connection: keep-alive

sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google

Chrome";v="102" sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows" upgrade-insecure-requests: 1

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0

Safari/537.36

accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web

exchange;v=b3;q=0.9

sec-fetch-site: same-origin sec-fetch-mode: navigate

sec-fetch-user: ?1

sec-fetch-dest: document

referer: http://localhost:8080/ch2 handson 1 war exploded/

accept-encoding: gzip, deflate, br accept-language: en-GB,en;q=0.9

Message from doGet() method in SimpleSerlvet.

No data received!



• Резултат од втор линк

• Конзола во IDE

SimpleServlet is initialized!
SimpleServlet doGet() invoked!

host: localhost:8080 connection: keep-alive

sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google

Chrome";v="102" sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows" upgrade-insecure-requests: 1

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0

Safari/537.36

accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web

exchange;v=b3;q=0.9

sec-fetch-site: same-origin sec-fetch-mode: navigate

sec-fetch-user: ?1

sec-fetch-dest: document

referer: http://localhost:8080/ch2_handson_1_war_exploded/

accept-encoding: gzip, deflate, br accept-language: en-GB,en;q=0.9

Message from doGet() method in SimpleSerlvet.

No data received!



• Резултат од втор линк

• Конзола во IDE

SimpleServlet doGet()invoked!

host: localhost:8080 connection: keep-alive

sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google

Chrome";v="102" sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows" upgrade-insecure-requests: 1

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0

Safari/537.36

accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web

exchange;v=b3;q=0.9

sec-fetch-site: same-origin sec-fetch-mode: navigate

sec-fetch-user: ?1

sec-fetch-dest: document

referer: http://localhost:8080/ch2 handson 1 war exploded/

accept-encoding: gzip, deflate, br cookie: firsCookie=-1193959466

Message from doGet() method in SimpleSerlvet. t.



- Резултат од submit
 - после внесен текст post-data

• Конзола во IDE

SimpleServlet doPost()invoked!

• Ако се прекине апликацијата

SimpleServlet is destroyed!

host: localhost:8080 connection: keep-alive content-length: 20

cache-control: max-age=0

sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google

Chrome";v="102" sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows" upgrade-insecure-requests: 1 origin: http://localhost:8080

content-type: application/x-www-form-urlencoded

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0

Safari/537.36

accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/wel

exchange;v=b3;q=0.9 sec-fetch-site: same-origin sec-fetch-mode: navigate

sec-fetch-user: ?1

sec-fetch-dest: document

referer: http://localhost:8080/ch2 handson 1 war exploded/

accept-encoding: gzip, deflate, br accept-language: en-GB,en;q=0.9 cookie: firsCookie=-1193959466

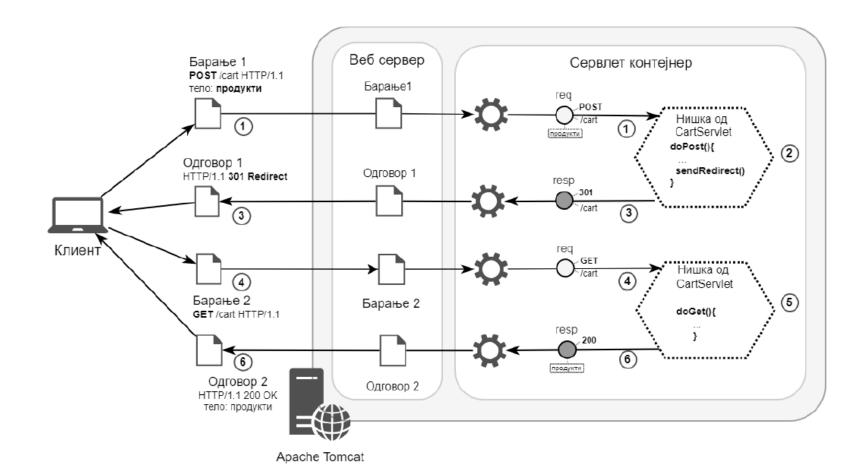
Message from doPost in SimpleSerlvet.

Универзитет "Св. Кирил и Методиј" во Скопје ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Received data: post-data

- Проследување на барање од еден сервлет до друг
 - SendRedirect
 - Forward
 - Include

SendRedirect





Forward и include

```
@WebServlet(name = "login", value = "/login")
public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
    → response) throws IOException, ServletException {
        // retrieve username and password from request
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        if (username.equals("admin") && password.equals("!@34")){
            RequestDispatcher dispatcher =

→ request.getRequestDispatcher("cart");
            dispatcher.forward(request, response);
        else {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.print("Invalid user name or password!");
            RequestDispatcher
                dispatcher=request.getRequestDispatcher("/index.html");
            dispatcher.include(request, response);
```



