

Słuchacze zdarzeń



Obiekt	Zdarzenia	Klasa słuchacza
KONTEKST	inicjalizacja i niszczenie	<code>ServletContextListener</code>
	dodawanie, usuwanie, zmiana atrybutów	<code>ServletContextAttributeListener</code>
SESJA	Tworzenie, niszczenie, aktywacja, przejście do stanu pasywnego, timeout	<code>HttpSessionListener</code> , <code>HttpSessionActivationListener</code>
	dodawanie, usuwanie, zmiana atrybutu	<code>HttpSessionAttributeListener</code>
ŻĄDANIE	Żądanie zaczyna być przetwarzane przez komponent	<code>ServletRequestListener</code>
	Dodanie, usuwanie, zmiana atrybutu	<code>ServletRequestAttributeListener</code>

Słuchacze zdarzeń



Słuchacze zdarzeń:

`ServletContextAttributeListener` – obserwuje kolekcję atrybutów kontekstu.



metody:

- `attributeAdded()` – dodanie atrybutu,
- `attributeRemoved()` – usunięcie atrybutu,
- `attributeReplaced()` – zmiana atrybutu.



Metody przyjmują parametr typu

`ServletContextAttributeEvent` – umożliwia on pobranie nazwy modyfikowanego atrybutu (`getName()`) oraz jego wartości (`getValue()`).

183

Słuchacze zdarzeń



Słuchacze zdarzeń:

`ServletContextListener` – Słuchacz wykorzystywany przy tworzeniu lub usuwaniu kontekstu aplikacji.



Obiekt zawiera dwie metody:

`contextInitialized()` – wykorzystywana w momencie utworzenia kontekstu aplikacji,



`contextDestroyed()` – wykorzystywana w momencie usunięcia kontekstu aplikacji



Obydwie metody posiadają parametr typu `ServletContextEvent` (metoda `getServletContext()` pobiera kontekst aplikacji).

182

index.jsp:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8">
    <script>
      function dodaj(){
        document.getElementById("wybor").value="add";
      }
      function usun(){
        document.getElementById("wybor").value="del";
      }
    </script>
  </head>
  <body>
    <h1>Atrybuty kontekstu - co zrobić:</h1>
    <form action="con" method="get">
      <input type="button" value="dodaj" onclick="dodaj()" />
      <input type="button" value="usun" onclick="usun()" />
      <input type="hidden" value="add" id="wybor"
        name="wybor" /><br/>
      <input type="submit" value="OK" />
    </form>
  </body>
</html>
```



parametr1.java:

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    String akcja = request.getParameter("wybor");
    ServletContext c=getServletContext();
    Random ran = new Random();
    if(akcja.equals("add")){
        c.setAttribute("nazwa", ran.nextInt());
    }else if(akcja.equals("del")){
        c.removeAttribute("nazwa");
    }
    RequestDispatcher r = request.getRequestDispatcher("/");
    r.forward(request, response);
}
```

web.xml:

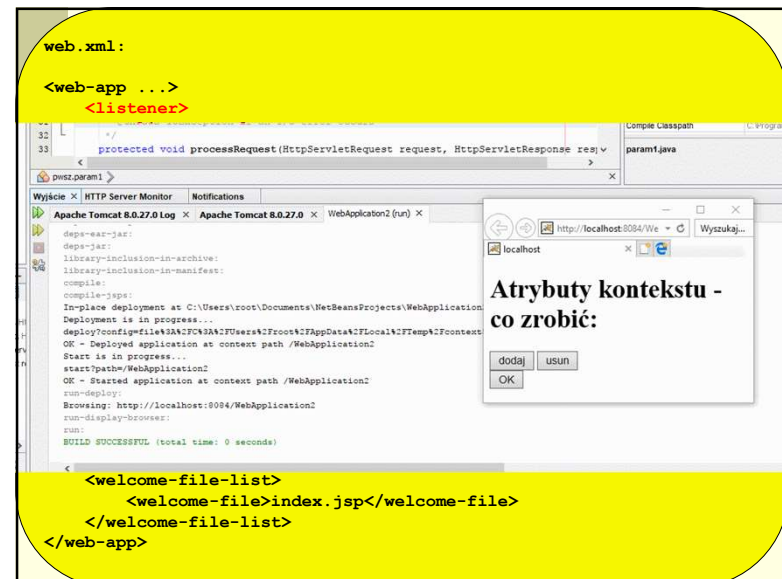
```
<web-app ...>
  <listener>
    <listener-class>
      pwsz.ConList
    </listener-class>
  </listener>
  <servlet>
    <servlet-name>param1</servlet-name>
    <servlet-class>pwsz.param1</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>param1</servlet-name>
    <url-pattern>/con</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

ConList.java:

```
public class ConList implements ServletContextAttributeListener {
    @Override
    public void attributeAdded(ServletContextAttributeEvent event) {
        System.out.println("Dodawanie:");
        System.out.println(event.getName());
        System.out.println(event.getValue());
    }
    @Override
    public void attributeRemoved(ServletContextAttributeEvent event)
    {
        System.out.println("Usuwanie:");
        System.out.println(event.getName());
        System.out.println(event.getValue());
    }
    @Override
    public void attributeReplaced(ServletContextAttributeEvent event)
    {
        System.out.println("Zmiana:");
        System.out.println(event.getName());
        System.out.println(event.getValue());
    }
}
```

web.xml:

```
<web-app ...>
  <listener>
    <listener-class>
      pwsz.ConList
    </listener-class>
  </listener>
  <servlet>
    <servlet-name>param1</servlet-name>
    <servlet-class>pwsz.param1</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>param1</servlet-name>
    <url-pattern>/con</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```



Słuchacze zdarzeń

ServletRequestListener

metody:

- `requestInitialized()`,
- `requestDestroyed()`.

Argument: `ServletRequestEvent` posiada metodę `getServletRequest()` zwracającą obiekt żądania.

ServletRequestAttributeListener

metody:

- `attributeAdded()` – dodanie atrybutu,
- `attributeRemoved()` – usunięcie atrybutu,
- `attributeReplaced()` – zmiana atrybutu.



189

index.jsp:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=UTF-8">
</head>
<body>
  <h1>Zamow produkt:</h1>
  <form action="con" method="get">
    Wybierz produkt:
    <select name="wybor">
      <option value="monitor">monitor</option>
      <option value="komputer">komputer</option>
      <option value="drukarka">drukarka</option>
    </select>
    <br>
    dodaj: <input type="radio" name="co" value="add"/>
    usun: <input type="radio" name="co" value="del"/>
    <br>
    <input type="submit" value="OK"/>
  </form>
</body>
</html>
```

Słuchacze zdarzeń

HttpSessionListener – wykorzystywany przy tworzeniu i kończeniu sesji. metody:

- `sessionCreated()`,
- `sessionDestroyed()`.

Argument `HttpSessionEvent` – metoda `getSession()` daje dostęp do utworzonej/kończonej sesji.

HttpSessionAttributeListener – metody:

- `attributeAdded`,
- `attributeRemoved`,
- `attributeReplaced`.



Argument `HttpSessionBindingEvent` – umożliwia pobranie obiektu sesji i nazwy lub wartości atrybutów.

190

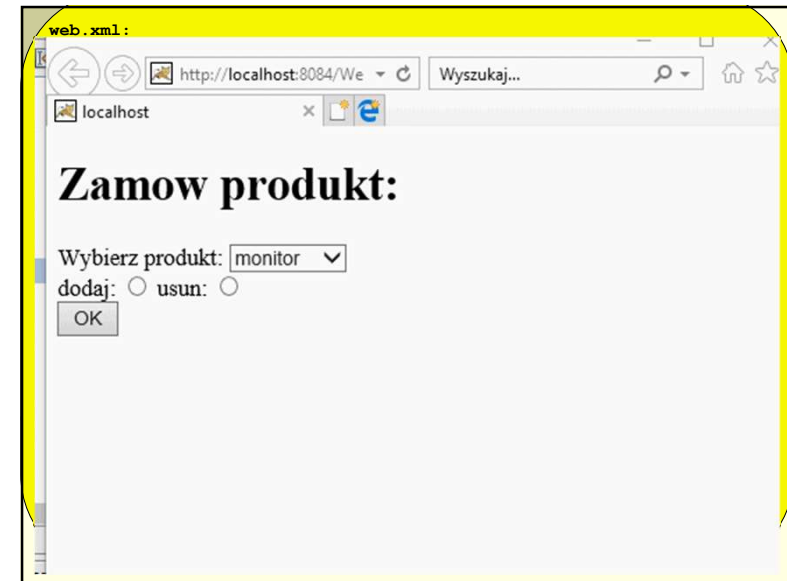
servlet paraml.java:

```
protected void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    HttpSession ses= request.getSession();
    String produkt = request.getParameter("wybor");
    String operacja=request.getParameter("co");
    if(ses.getAttribute(produkt)==null){
        ses.setAttribute(produkt, 0);
    }
    if(operacja.equals("add")){
        ses.setAttribute(produkt,
            (int)ses.getAttribute(produkt)+1);
    } else{
        ses.setAttribute(produkt,
            (int)ses.getAttribute(produkt)-1);
    }
    if((int)ses.getAttribute(produkt)<=0){
        ses.removeAttribute(produkt);
    }
    PrintWriter p = response.getWriter();
    p.println("LICZBA <B>ROZNYCH</B> PRODUKTOW W KOSZYKU:<br>");
    p.println(
        (ses.getAttribute("liczba")==null)? 0:
        ses.getAttribute("liczba").toString());
    RequestDispatcher r = request.getRequestDispatcher("/");
    r.include(request, response);
}
```

```

servlet SesList.java:
public class SesList implements HttpSessionAttributeListener {
    @Override
    public void attributeAdded(HttpSessionBindingEvent event) {
        HttpSession s=event.getSession();
        if(!event.getName().equals("liczba")){
            if(s.getAttribute("liczba")==null){
                s.setAttribute("liczba", 1);
            }else{
                s.setAttribute("liczba",
                    (int) s.getAttribute("liczba")+1);
            }
        }
    }
    @Override
    public void attributeRemoved(HttpSessionBindingEvent event) {
        HttpSession s=event.getSession();
        if(s.getAttribute("liczba")==null){
        }else{
            s.setAttribute("liczba",
                (int) s.getAttribute("liczba")-1);
        }
    }
    @Override
    public void attributeReplaced(HttpSessionBindingEvent event) {
    }
}

```



```

web.xml:
<web-app . . . >
  <listener>
    <listener-class>
      pwsz.SesList
    </listener-class>
  </listener>
  <servlet>
    <servlet-name>param1</servlet-name>
    <servlet-class>pwsz.param1</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>param1</servlet-name>
    <url-pattern>/con</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>

```