

Java EE webalkalmazások

Alapfogalmak, felépítés, szervletek

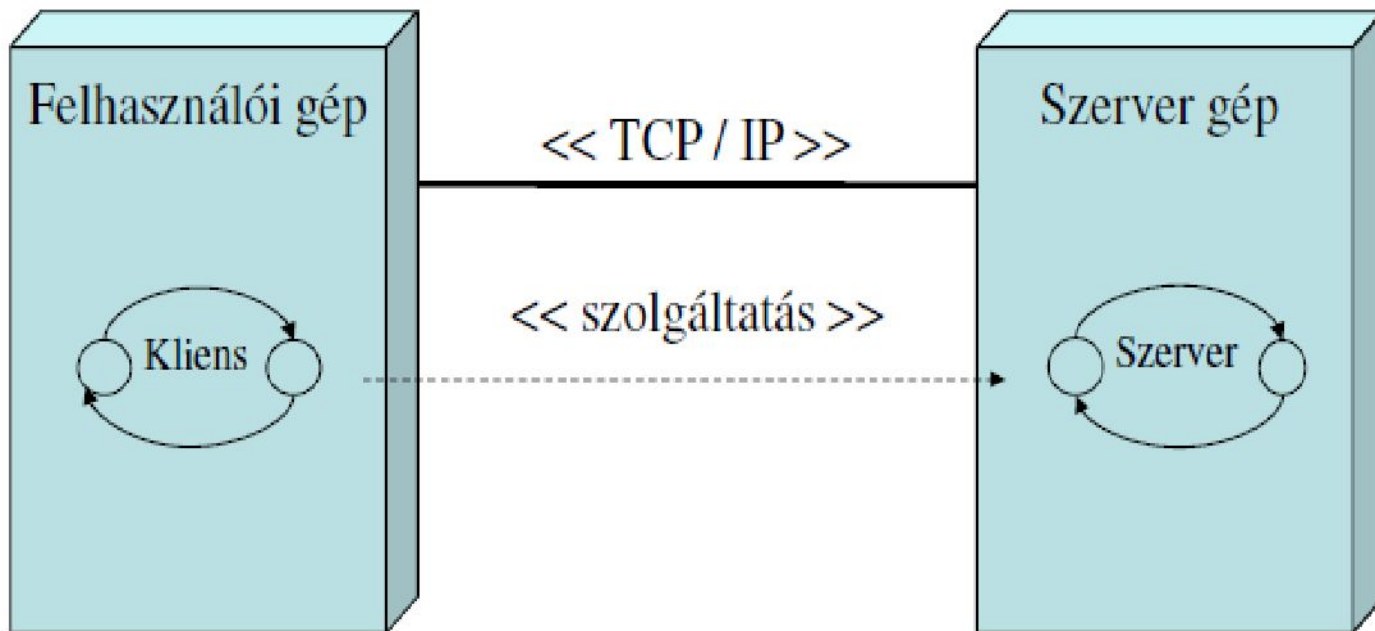


Tartalom

- ▶ Webalkalmazások alapfogalmai
- ▶ Java EE webalkalmazás definíciója, felépítése
- ▶ Szervlet fogalma
- ▶ Request, Response, Session, Filter
- ▶ URL leképezés
- ▶ Telepítésleíró felépítése

Webes alkalmazások

- ▶ Általános internetes alkalmazás



Java EE webalkalmazás definíció

- ▶ Def: Egy telepíthető csomag
 - Webkomponensek (szervlet, JSP, JSF)
 - Statikus erőforrás-fájlok (pl. képek)
 - Segédosztályok, osztálykönyvtárak (.jar)
 - Telepítés-leíró (deployment-descriptor) – web.xml
- ▶ Csomagolás:
 - war file
 - Alkalmazás részeként (ear), de külön is telepíthető

Java EE webalkalmazás felépítése

- ▶ Csomagolás: war, ear
- ▶ Mappastruktúra felépítése:

+java

+ - src (Servletek, JSF Beanek)

+resources

+ - META-INF

+webapp

+ - [/jsp, xhtml] -> oldalak

+ - /resources -> statikus tartalmak (képek, css)

+ - /WEB-INF -> konfigurációs adatok, település leíró

+ - /classes -> lefordított osztályok

+ - /lib -> osztálykönyvtárak (.jar)

+ - /web.xml -> konfigurációs file

+ - /faces-config.xml (optional)

+ - /*.taglib.xml (optional)

+ - /jboss-web.xml -> alkalmazáserver specifikus telepítés leíró

Servlet fogalma

- ▶ Egy Java objektum, amely egy keretrendszerre és API-ra (servlet- vagy webkonténer) építve kibővíti a HTTP (és más) szerver funkcionálisát
- ▶ A Servlet API a Java EE specifikáció része
- ▶ Dinamikus tartalomgenerálás (~ CGI)
- ▶ Az elkészített komponens alkalmazáserverek között hordozható
- ▶ Kéréseket fogad, és a kérésre válaszokat generál (request-response)

Szervlet fogalma

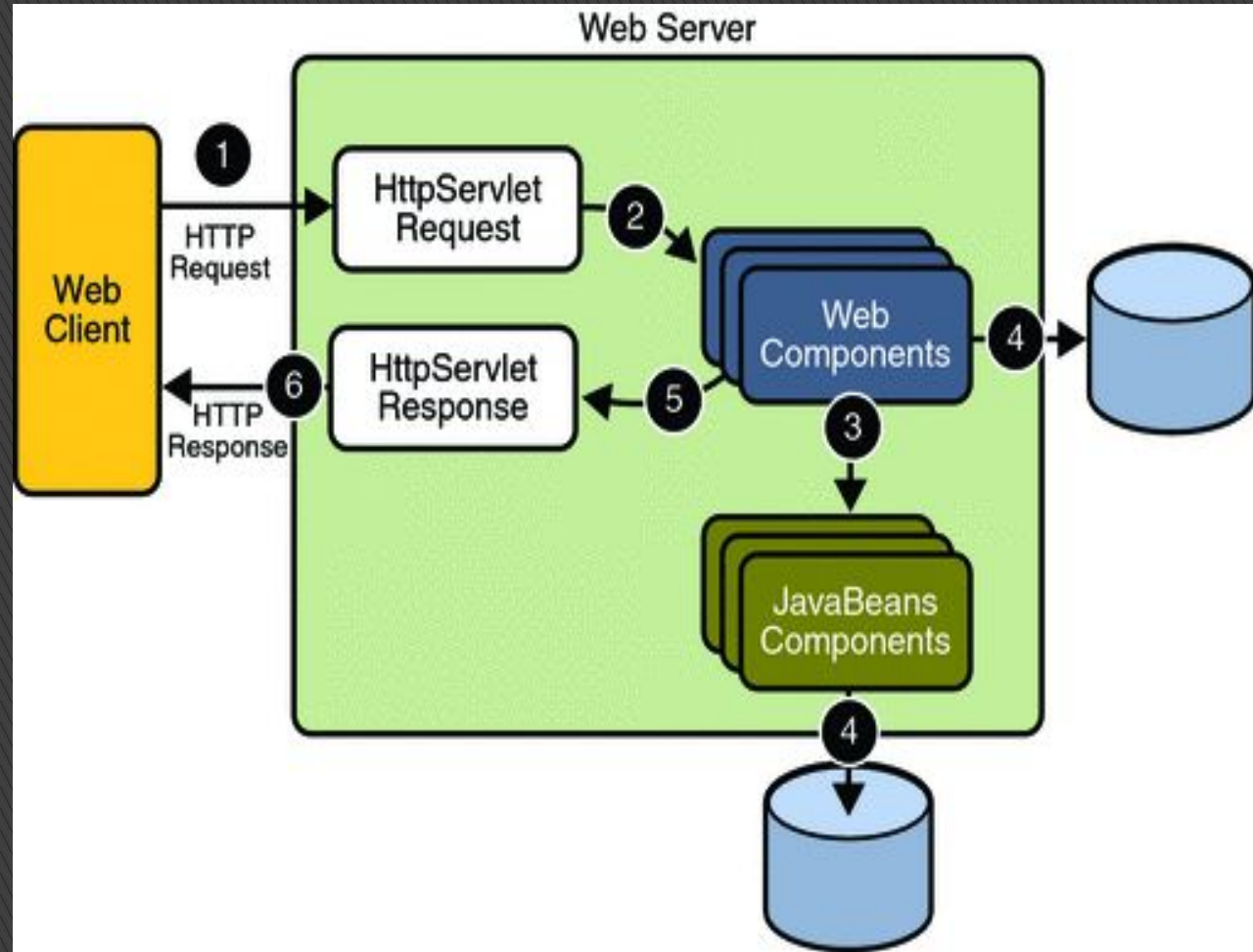
- ▶ Általános célú komponens tetszőleges protokollra
- ▶ Általában HTTP-re használjuk, de pl. FTP-re is implementálták

Folyamat:

- 1. Fogadja a kliens kérését (Request)
- 2. Kinyeri a megfelelő információkat (Parameters)
- 3. Az alkalmazáslogika alapján valamilyen adat, tartalom keletkezik
- 4. Visszaküldi a választ a kliensnek (Response)

Http protokoll és szervletek

- A felhasználó statikus és dinamikus tartalmakat kap eredményként (html, xhtml, xml...)
- http: állapotmentes protokoll



Szervlet példa

```
public class HelloServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse  
        response) {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println("<html><head>");  
        out.println("<title>Hello World!</title>");  
        out.println("</head></html>");  
        out.close();  
    }  
    ...  
}
```

- ▶ HTML kód Java-n belül
- ▶ Bármilyen adatfeldolgozás
- ▶ Egyszerű a webes kérésekhez
- ▶ Nem az egyszerűen karbantartható kimenetet szolgálja

URL leképzés

A Java EE webkonténer (többek között) HTTP szerverként működik:

- ▶ A HTTP kérések URL-jét dekódolja
- ▶ Statikus tartalmat statikusként szolgál ki
- ▶ Dinamikus tartalmat a konfigurációból kiolvasott szervletnek továbbít
 - 1. Alkalmazás megállapítása
 - 2. Szervlet kikeresése
- ▶ Context root:
 - Egy alkalmazásra jellemző egyedi elérési út
 - Telepítésleíróban állítható, alapértelmezett a .war fájl neve (http://szerver.com/gyak_7)

Telepítésleíró (web.xml) felépítése

- ▶ Életciklus-kezelők(alkalmazás indítás, leállítás)

```
<listener>  
  <listener-class>com.myapp.WebListener</listenerclass>  
</listener>
```

- ▶ Szervletek regisztrálása

```
<servlet>  
  <servlet-name>GreetingServlet</servlet-name>  
  <servlet-class>com.myapp.GreetingServlet</servletclass>  
  <load-on-startup>1</load-on-startup>  
</servlet>  
<servlet-mapping>  
  <servlet-name>GreetingServlet</servlet-name>  
  <url-pattern>*.xhtml</url-pattern>  
</servlet-mapping>
```

Telepítésleíró (web.xml) felépítése

► Szűrők

```
<filter>  
<filter-name>LoginFilter</filter-name>  
<filter-class>hu.ppke.web.LoginFilter</filter-class>  
</filter>  
<filter-mapping>  
<filter-name>LoginFilter</filter-name>  
<url-pattern>*.xhtml</url-pattern>  
</filter-mapping>
```

Telepítésleíró (web.xml) felépítése

- ▶ Error mapping

```
<error-page>  
<error-code>404</error-code>  
<location>/notfound.xhtml</location>  
</error-page>  
<error-page>  
<exception-type>java.lang.Throwable</exception-type>  
<location>/error.xhtml</location>  
</error-page>
```

Telepítésleíró (web.xml) felépítése

- ▶ Biztonsági beállítások
 - KÉSŐBB
- ▶ Session élettartam
 - KÉSŐBB

Szervlet szűrők

- ▶ Java Servlet Filters
- ▶ Cél: megfigyelni, módosítani, közbeavatkozni a kérés illetve a válasz alapján
- ▶ A szűrőket láncba szervezhetjük
- ▶ Használati esetek:
 - Hozzáférés biztosítása, blokkolása
 - Cache, tömörítés, loggolás, titkosítás
 - Tartalom transzformációja
- ▶ Szűrők sorrendje:
 - A web.xml-ben a `<filter>` definíciók sorrendje