

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

Szervletek

- „Kis szerverprogram”
- Kliens-szerver architektúrát feltételezünk
- Szerveroldalon futó kisalkalmazás
- Mint ahogy az applet egy kliens oldali kisalkalmazás
- Például egy HTTP szerverbe integrált program
- A CGI programozás Java megfelelője

CGI programok

- CGI = Common Gateway Interface
- A HTTP szerver nem egy „igazi” HTML oldalt ad vissza
- Egy program generálja az oldalt
- Dinamikus HTML oldal: tartalma paramétereiktől függhet
- A programot egy speciális URL-re történő hivatkozáskor indítja a HTTP szerver
- Pl. adatbázisból vehet adatokat, hogy egy HTML form-ra reagáljon

Ugyanez Java-ban: szervletekkel

- A HTTP szerverbe be van integrálva egy Java Virtuális Gép
- Ez tud szervleteket futtatni, amikor erre igény van
- Egy speciális (szervletet azonosító) URL-re történő hivatkozáskor a szervlet állítja elő a HTTP szerver választ

Miért jobb, mint a CGI?

- Platform-független megoldás
- Hatékonyabb
 - Igaz, hogy a Java lassú, de...
 - A WEB szerveren belül állandóan fut a JVM: gyorsabb, mint egy külső programot indítani (általában a CGI programok szkript nyelven íródtak, ugyanúgy interpretáltak/lassúak, mint a Java)
- Konkurens kéréskezelések szinkronizációja
- Azonos helyről jövő kérések kiszolgálása között információ megőrzése viszonylag könnyű

Mi ez a technológia?

- Régebben: Server Development Kit segítségével (ezt fogjuk használni most)
- Ma már integrált más technológiákkal
 - JSP (Java Server Pages)
 - A Java Enterprise Technológia szerves része
- Megfelelő WEB szerver kell a futtatáshoz
 - SUN referencia implementáció: használja a Tomcat-et (Jakarta projekt, Apache SW Found.)
 - Persze a mai profi eszközök is fel vannak már erre készítve (pl. az Apache, MS IIS, stb.)
 - Applikáció szerverek (BEA, IBM, iPlanet, stb.)



Java Servlet Development Kit

- JSDK 2.1 - 1999 április
- <http://java.sun.com/products/servlet/>
- Ez már csak az archívumban található meg
- Tartalma
 - szervlet programozói könyvtár (API)
servlet.jar
 - egy játék Web-szerver fejlesztéshez/teszteléshez
 - példaprogramok

A Web szerver

- Indítás: **startserver(.bat)**
- Alapértelmezésben nem a szokványos 80-as TCP portot használja, hanem a 8080-asat
- Egy böngészővel próbáljuk ki:
http://localhost:8080/
Vagy akár egy másik gépen futót...
- Leállítás: **stopserver(.bat)**
- Van egy speciális könyvtár, ahova a szervleteinket kell pakolni

A szerver programozói könyvtár

- A servlet.jar fájl tartalmazza
- javax.servlet és javax.servlet.http csomagok
 - szabványos kiterjesztések (standard extension)
- Ezt a jar fájlt bele kell tenni a classpath-ba
- Sokszor nem megy így, csak ha kicsomagolom

Szervletek és HTTP szervletek

- Leggyakrabban HTTP protokollt használó szervleteket szoktak írni (pl. mi is azt fogunk)
- javax.servlet.http.HttpServlet**
- Lehetőség van más protokoll használatára is
javax.servlet.Servlet interfész
javax.servlet.GenericServlet

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Hello extends HttpServlet {
    public void doGet( HttpServletRequest req,
                      HttpServletResponse res ){
        doPost(req, res);
    }
    public void doPost( HttpServletRequest req,
                       HttpServletResponse res ){
        try {
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();
            out.println("<html><head><title>Hi</title>" +
                        "</head><body>Hello</body></html>");
            out.close();
        } catch (IOException e) {}
    }
}
```

Első szervletünk: Hello

- Importáljuk a javax.servlet, a javax.servlet.http és a java.io tartalmát
- Írunk egy doGet metódust, ami meghívja a doPost metódust
- A doPost paramétere HttpServletRequest és HttpServletResponse
- Az utóbbira kinyomtatunk egy HTML oldalt, amit a böngésző kap majd

Kipróbálás

- Fordítsuk le! Használjuk a szervlet API-t!
- A .class fájlt másoljuk a **webpages/WEB-INF/servlets** könyvtárba
- Indítsuk el/újra a web-szervert
- Állítsuk a böngészőt erre az oldalra:
http://localhost:8080/servlet/Hello
- Próbáljuk elérni a más számítógépeken futó web-szervereket is...

Értékelés

- Ez a szervlet mindig ugyanazt az oldalt adja vissza
- Az igazi haszna a szervleteknek a dinamikus HTML oldalak generálásában rejlik
- Sokszor a szervlet aktiválása a böngészőből paraméterek átadásával is jár
- Például egy HTML form a szervlettől vár választ

Feladat

- Írjunk olyan szervletet, amely egy, a felhasználó által begépelt szöveget megfordít.
Használjunk HTML form-ot! A form egy szövegbeviteli mezőt tartalmazzon, melynek tartalmát (Enter hatására) elküldjük a szervletnek.

Java tutorial

Copyright © 2000-2002, Kozsik Tamás

A HTML form

```
<html>
<head><title>Hi</title></head>
<body>
<form action =
    "http://localhost:8080/servlet/Fordit"
    method = post>
Szöveg: <input type=text name=szo size=30>
</form>
</body>
</html>
```

Mit csinál ez a form?

- A szövegbeviteli mező neve: szo
- A form kitöltése után (Enter hatására) a böngésző a web-szervertől ezt az oldalt kéri el:
http://localhost:8080/servlet/Fordit
- Ez egy szervlet címe, az ő válaszát jeleníti meg.
- A kéressel együtt elküldi a form-ba bevitt információkat is, melyekhez a szervlet hozzáfér.
- Az információ „csomagolási módja” most „post”.
- Azaz a szervlet a doPost metódusával reagál rá.
 - A másik lehetőség a get, doGet lenne.

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Hello extends HttpServlet {
    public void doGet( ... ) {...}
    public void doPost( HttpServletRequest req,
                       HttpServletResponse res ){

        try {
            StringBuffer s = new StringBuffer(
                req.getParameter("szo") );
            res.setContentType("text/html");
            PrintWriter out = res.getWriter();
            out.println("<html><head><title>Hi</title>" +
                "</head><body>" + s.reverse().toString() +
                "</body></html>");
            out.close();
        } catch (IOException e) {}
    }
}
```

A szervlet működése

- A szervlet majdnem ugyanolyan, mint az előző példában
- Lekéri a „szo” nevű paraméterének értékét.
- Ez a „szo” szövegbeviteli mező tartalma
- Ezt az értéket megfordítja, és beleszövi a generált HTML oldalba.
- A szervlet fordítása, installálása és futtatása a már megismert módon történik. A HTML oldal akárhol is lehet, csak rá tudunk állítani egy web-böngészőt.

Java tutorial

Copyright © 2000-2002, Kozsik Tamás