

Hello World Servlet

Készítsünk egy szervletet, amellyel összeadhatunk két számot, és meghívásakor üdvözlőszöveget ír a konzolra.

Hozzunk létre egy Dynamic Web projectet

File → New → Other → ... itt a következőket kell választani: Web → Dynamic Web Project
A megjelenő párbeszédablakban adjuk meg a projekt nevét, ez a példában ServletExample.

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location
 Use default location
Location:

Target runtime

Dynamic web module version

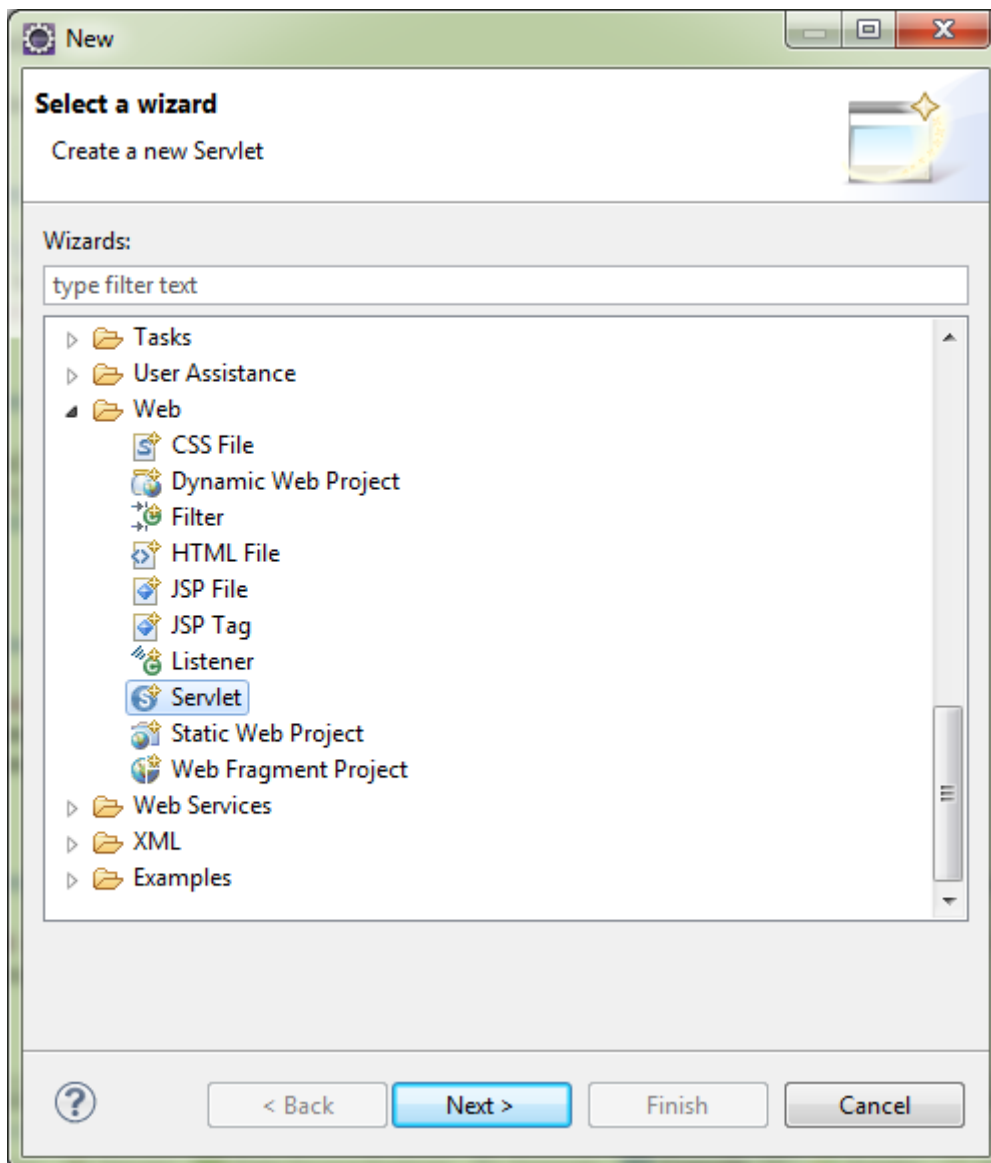
Configuration

A good starting point for working with JBoss 7.1 Runtime runtime. Additional facets can later be installed to add new functionality to the project.

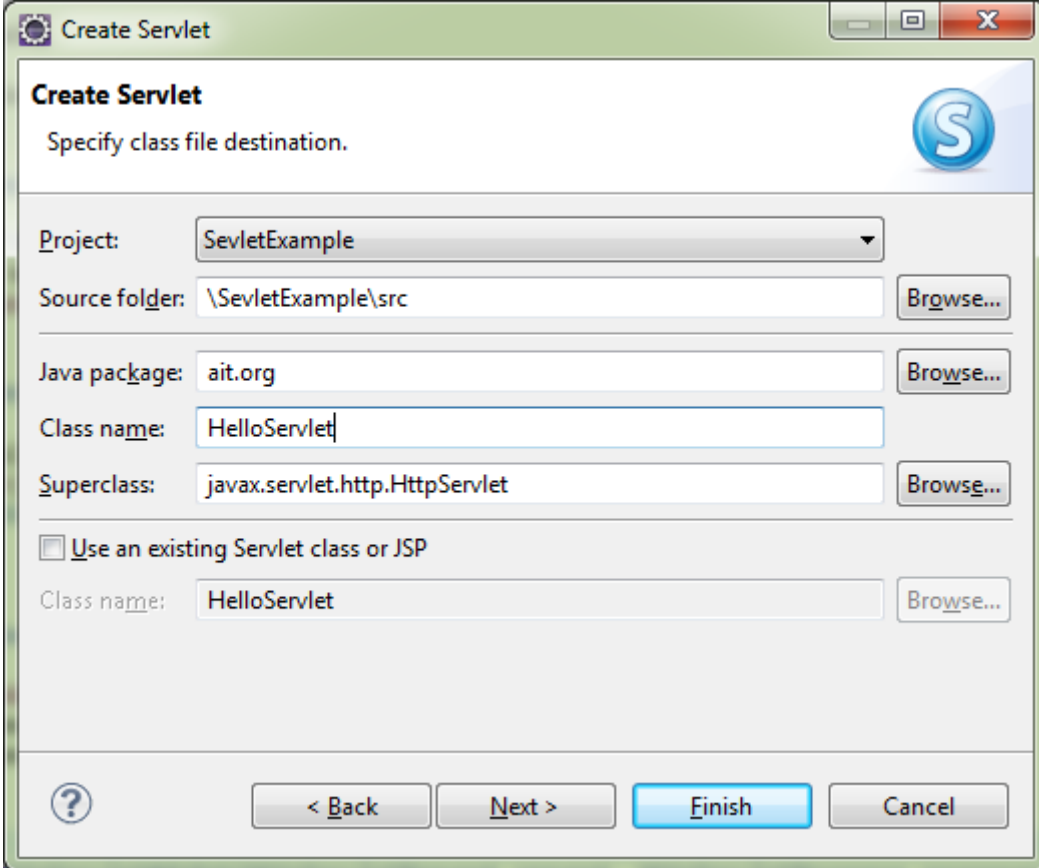
EAR membership
 Add project to an EAR
EAR project name:

Working sets
 Add project to working sets
Working sets:

Új HTTP Servlet létrehozását a következő menüpontban lehet megtenni:
File → New → Other ... itt a következőket kell választani: Web → Servlet



A megjelenő párbeszédablakban a Servlet paramétereit állíthatók be. A példában a csomag név org.ait, az osztály neve pedig HelloServlet.



Create Servlet

Specify class file destination.

Project:

Source folder:

Java package:

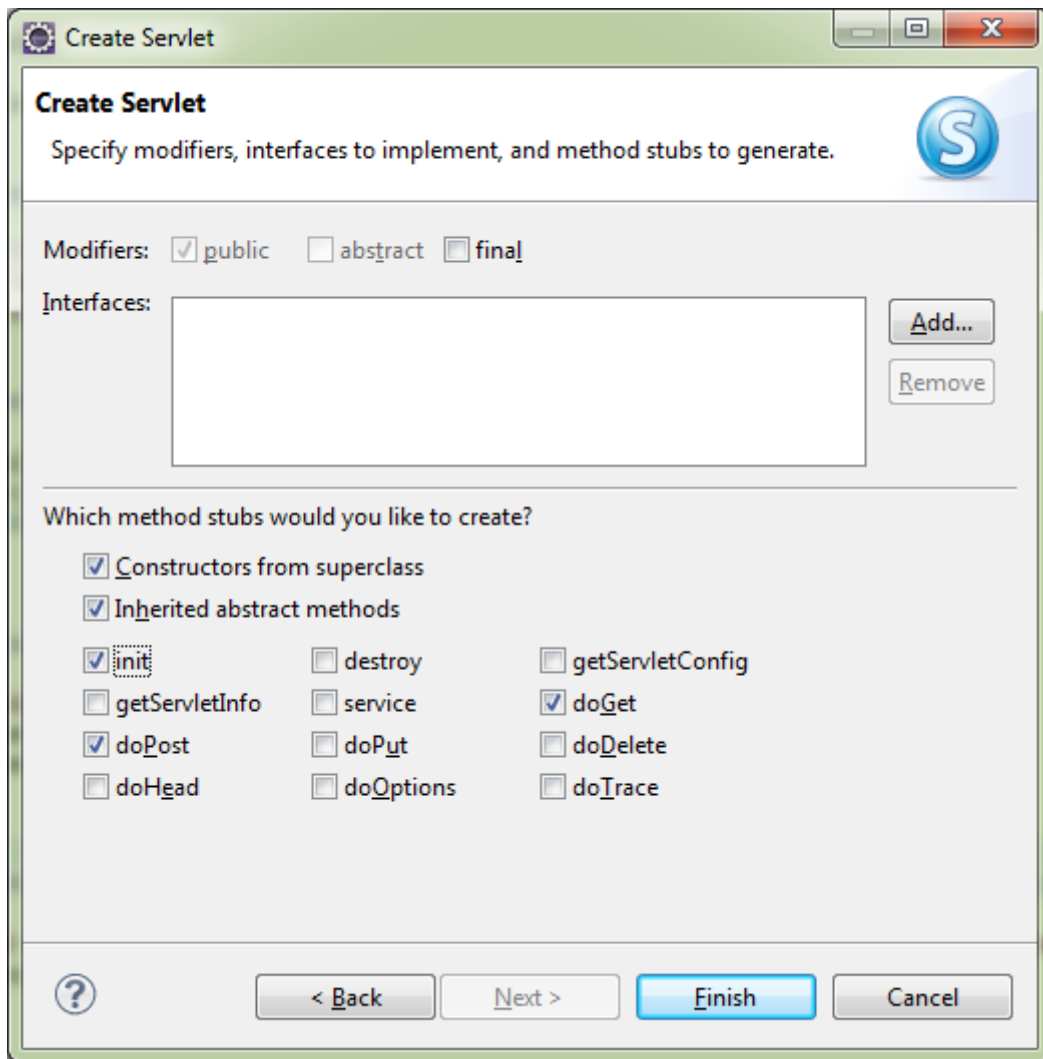
Class name:

Superclass:

Use an existing Servlet class or JSP

Class name:

A következő ablakban bejelölhetjük, hogy milyen metódusokat szeretnénk létrehozni. Jelöljük be az *init*, *doPost* és a *doGet* metódusokat.



Forráskód:

A szervlet meghívásakor az inicializálás során az *init()* metódus fog lefutni. Ezt módosítva írassuk ki a konzolra a „Hello World!” szöveget és az időpontot.

```
public void init(ServletConfig config) throws ServletException {
    System.out.println("Hello World!");
    System.out.println(new Date());
}
```

Két szám összeadásához készítsünk egy metódust, amit a *doGet()* és a *doPost()* metódusok meghívhatnak majd. Ez a metódus legyen a *process* metódus, amely paraméterében megkapja a HTTP kéréshez és válaszhoz szükséges objektumokat.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    process(request, response);
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    process(request, response);
}
```

A *process* metódus az *első* és *második* nevű paraméterekben megkapott két számot összeadja és egy html kódot küld válaszként az eredménnyel, amely majd a weboldalon megjelenik.

```
private void process(HttpServletRequest request, HttpServletResponse response) {
    String param1 = request.getParameter("első");
    String param2 = request.getParameter("második");

    int szam1 = Integer.parseInt(param1);
    int szam2 = Integer.parseInt(param2);

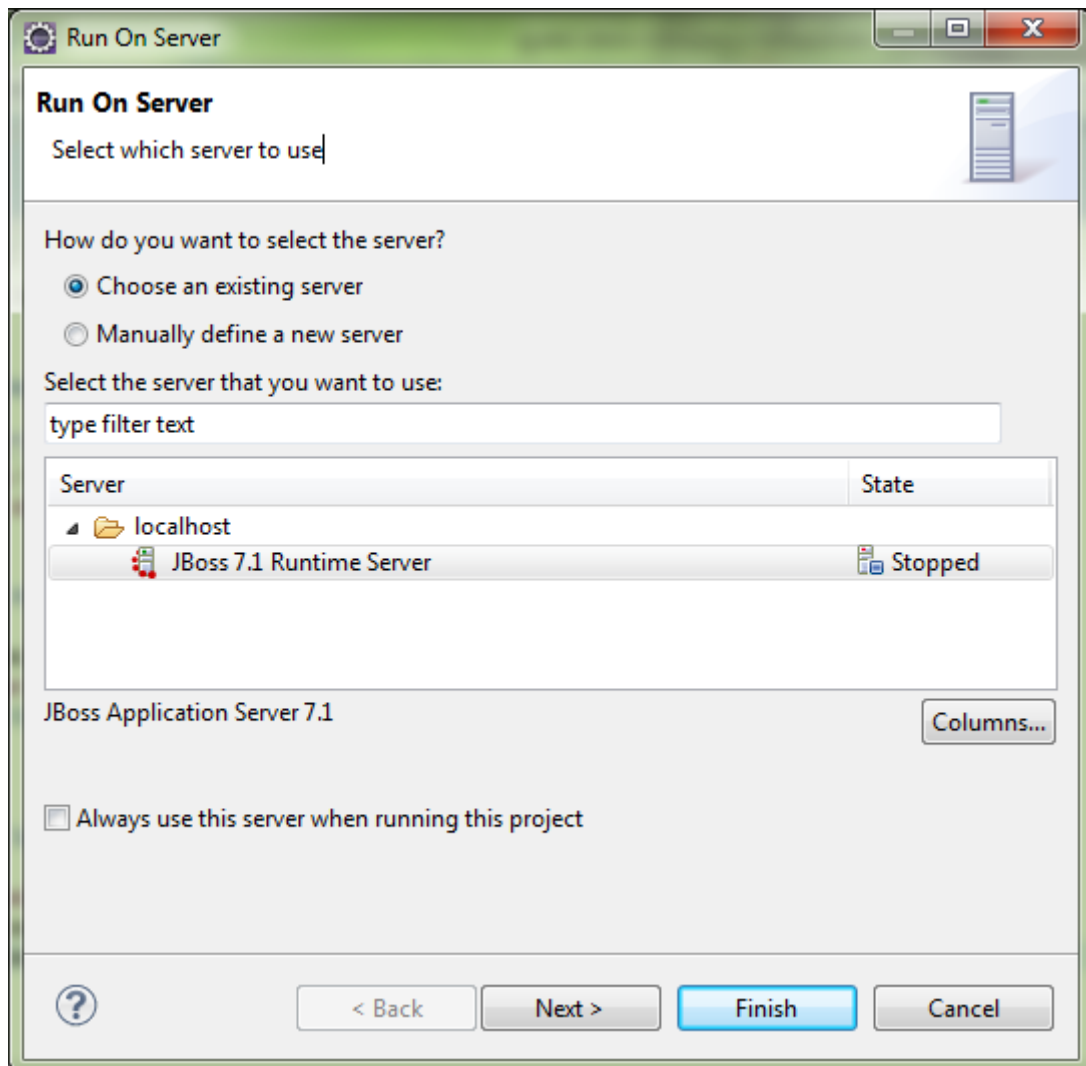
    PrintWriter out = null;

    try {
        out = response.getWriter();
    } catch (IOException ioex) {
        ioex.printStackTrace();
    }

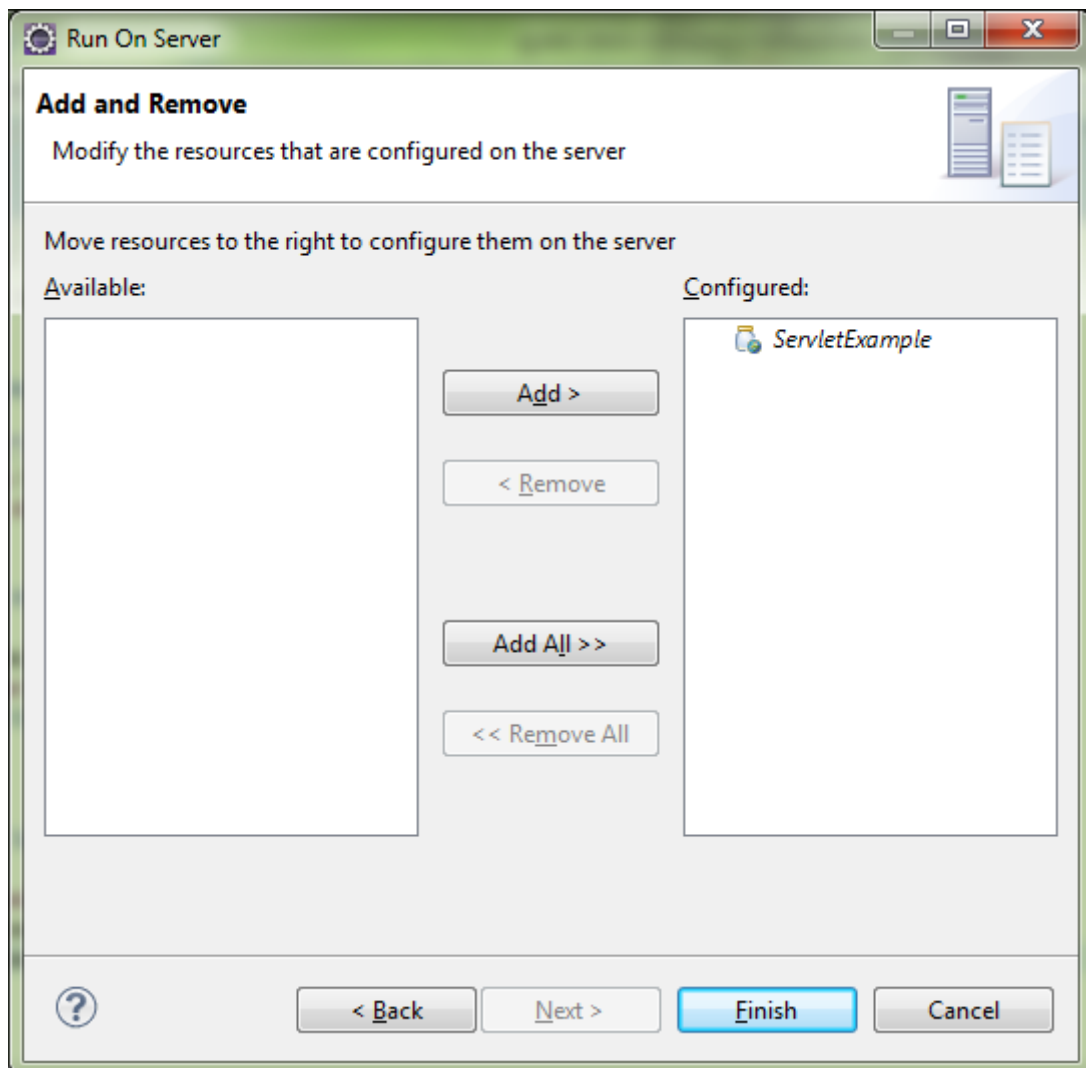
    out.println("<html><head><title>Egyszeru szervlet</title></head><body>");
    out.println("A két szám összege= " + (szam1+szam2));
    out.println("</doby></html>");
    out.close();
}
```

A projektet futtatása szerveren

Ru As -> Run on Server, majd a megjelenő párbeszéd ablakban válasszuk ki a kívánt szervert. Jelen esetben ez a JBoss 7.1 Runtime Server.

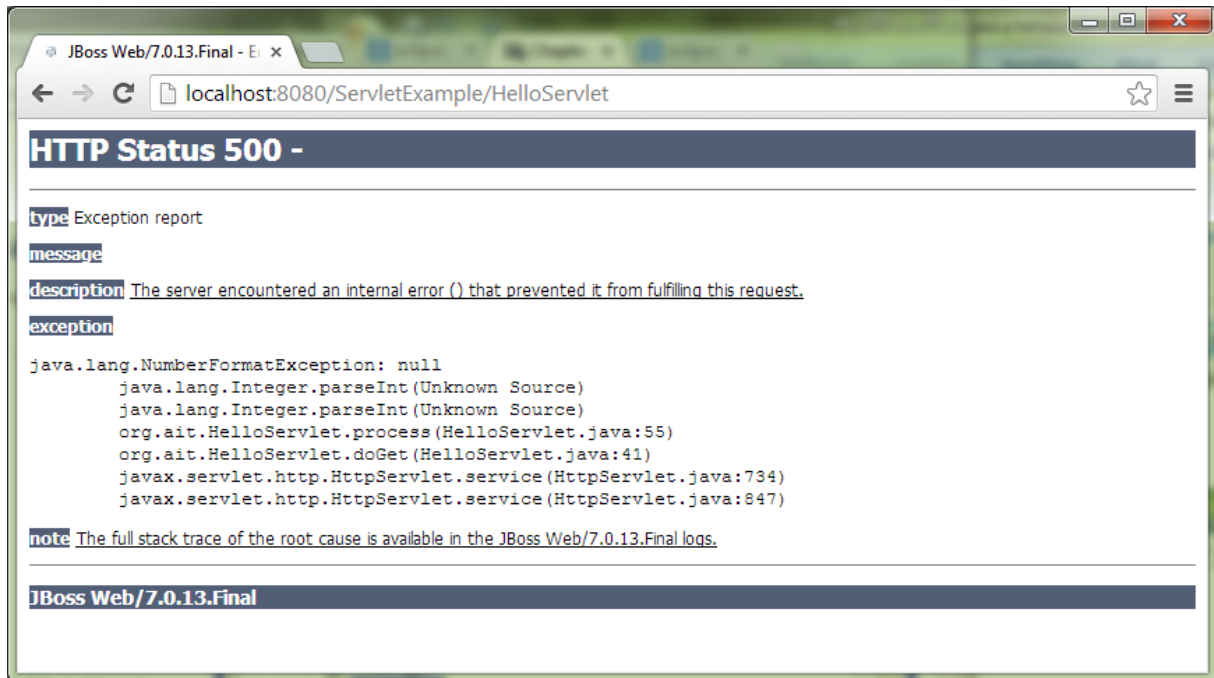


A következő megjelenő párbeszédablakban más projektet is hozzáadhatunk a szerverhez.

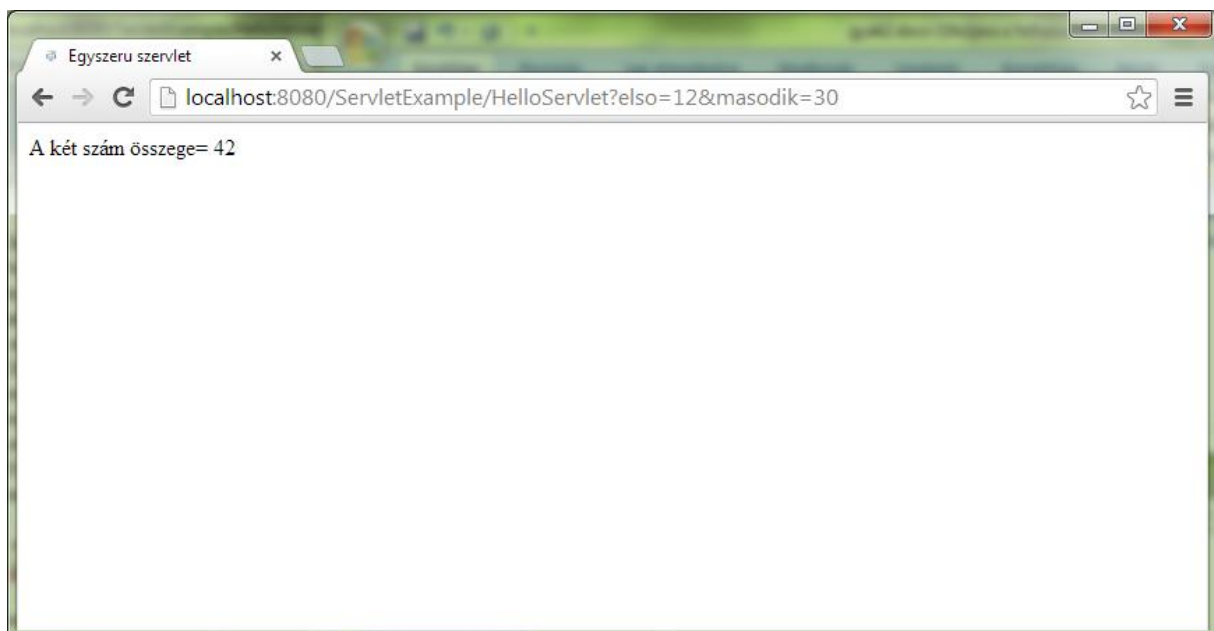


A szervlet meghívásakor a konzolban megjelenik a „Hello World!” üzenet és a szervlet meghívása kori időpont.

Azonban, paraméter nélkül meghívva a szervlet hibaüzenetet jelenít meg.



A számok összeadásához meg kell adnunk a két paramétert, melyek neve *első* és *második*. Ebben az esetben a két szám összege fog megjelenni a weboldalon.



Hello World JSP

Projektlétrehozás

Egy Dynamic Web Project-et kell létrehozni minimális konfigurációval.

- A menüben válasszuk a File → New → Other... pontot majd Web → Dynamic Web Project
- A megjelenő párbeszédablakban adjuk meg a projekt nevét (a példában JSPEXample)
- A konfiguráció beállításánál válasszuk a Minimal Configuration-t
- A Next gombra való kétszeri kattintás után megjelenő ablakban jelöljük be a Generate web.xml jelölőnégyzetet.

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: JSPEXample

Project location
 Use default location
Location: C:\Users\Simon\workspaceEE\JSPEXample

Target runtime
JBoss 7.1 Runtime

Dynamic web module version
3.0

Configuration
Minimal Configuration

EAR membership
 Add project to an EAR
EAR project name: EAR

Working sets
 Add project to working sets
Working sets:

< Back Next > Finish Cancel

New Dynamic Web Project


Web Module

Configure web module settings.

Context root:

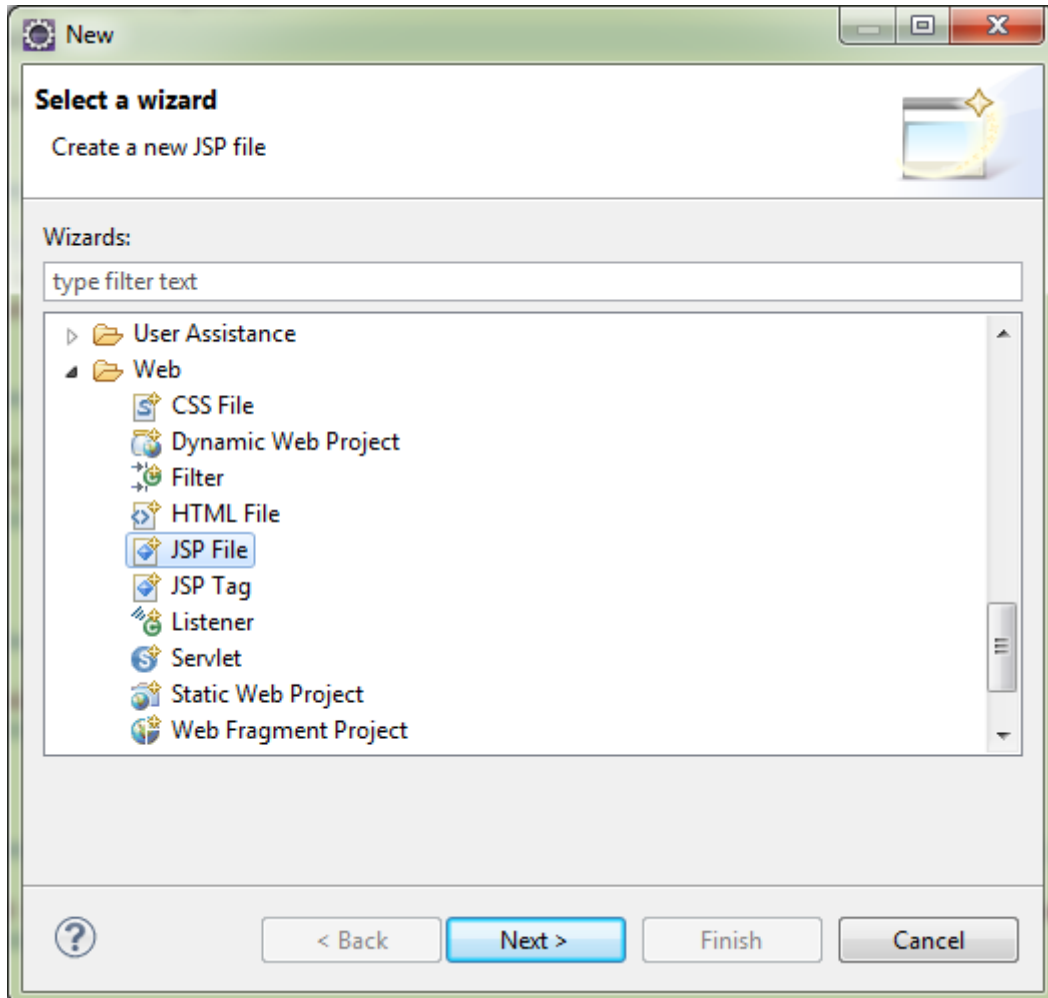
Content directory:

Generate web.xml deployment descriptor

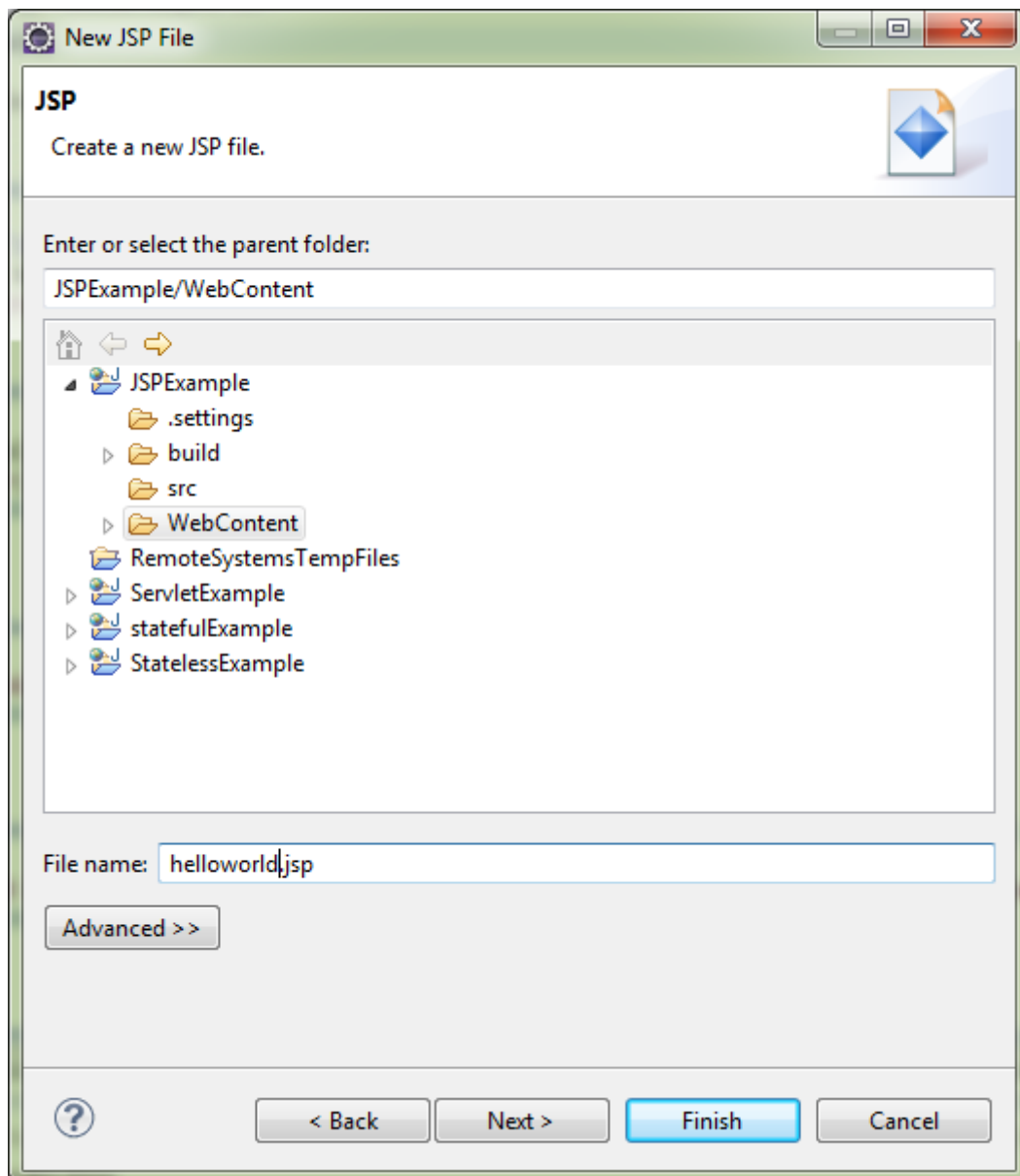


HelloWorld JSP oldal létrehozása

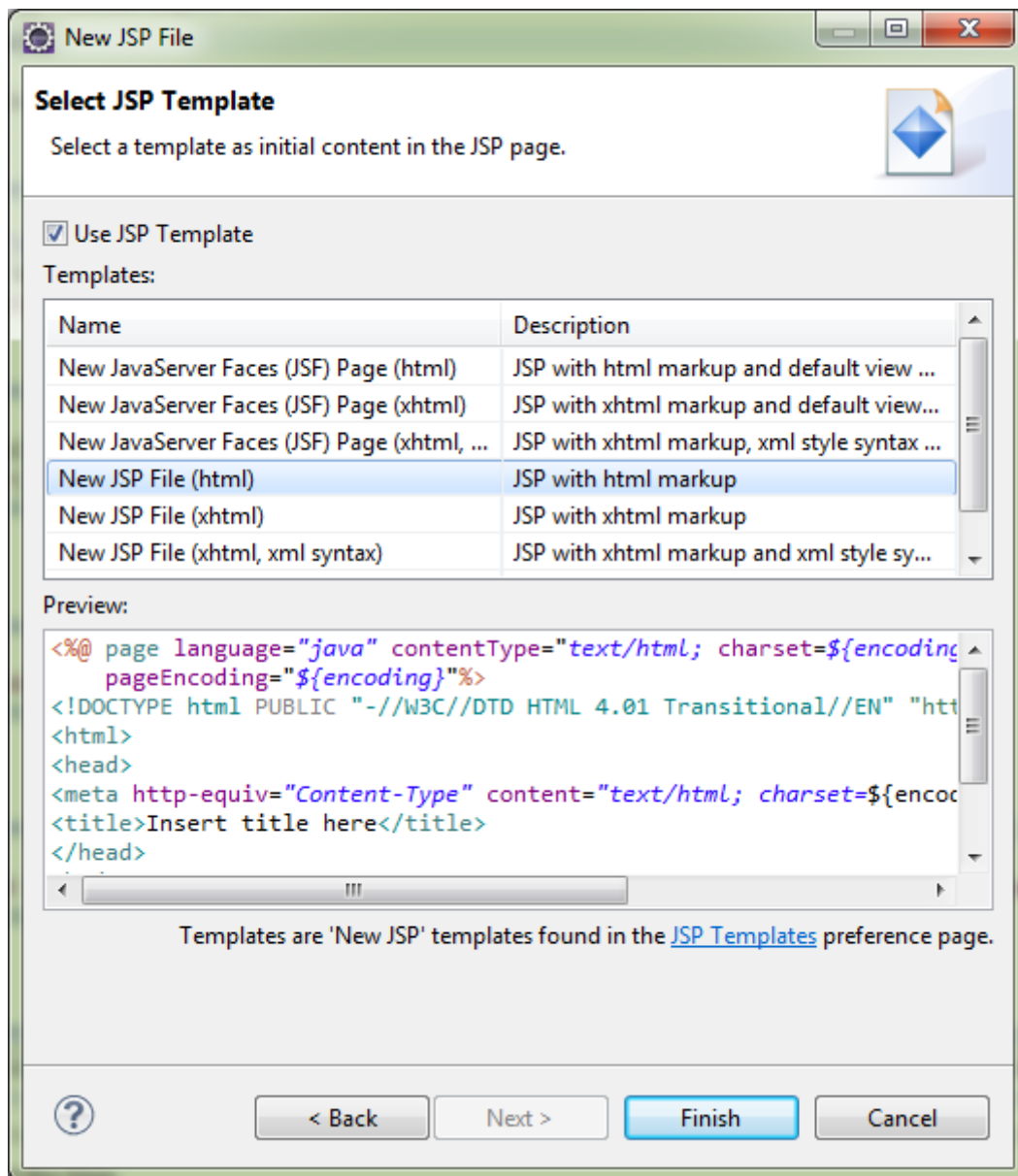
A következőkben egy JSP oldalt hozunk létre, amely a „Hello World!” üzenetet jeleníti meg. JSP oldal létrehozásához a menüben válasszuk a New → Other... pontot, majd a Web → JSP File pontot.



A megjelenő párbeszédablakban adjuk meg a jsp fájl nevét, a példában helloworld.jsp.



A következő ablakban válasszuk a New JSP File (html) templétet.



JSP oldal szerkesztése

A „Hello World!” üzenet megjelenítéséhez a következő sort kell a <body></body> tagek közé szúrni.

```
<% System.out.println("Hello World!"); %>
```

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <%System.out.println("Hello World!");%>
11 </body>
12 </html>
```

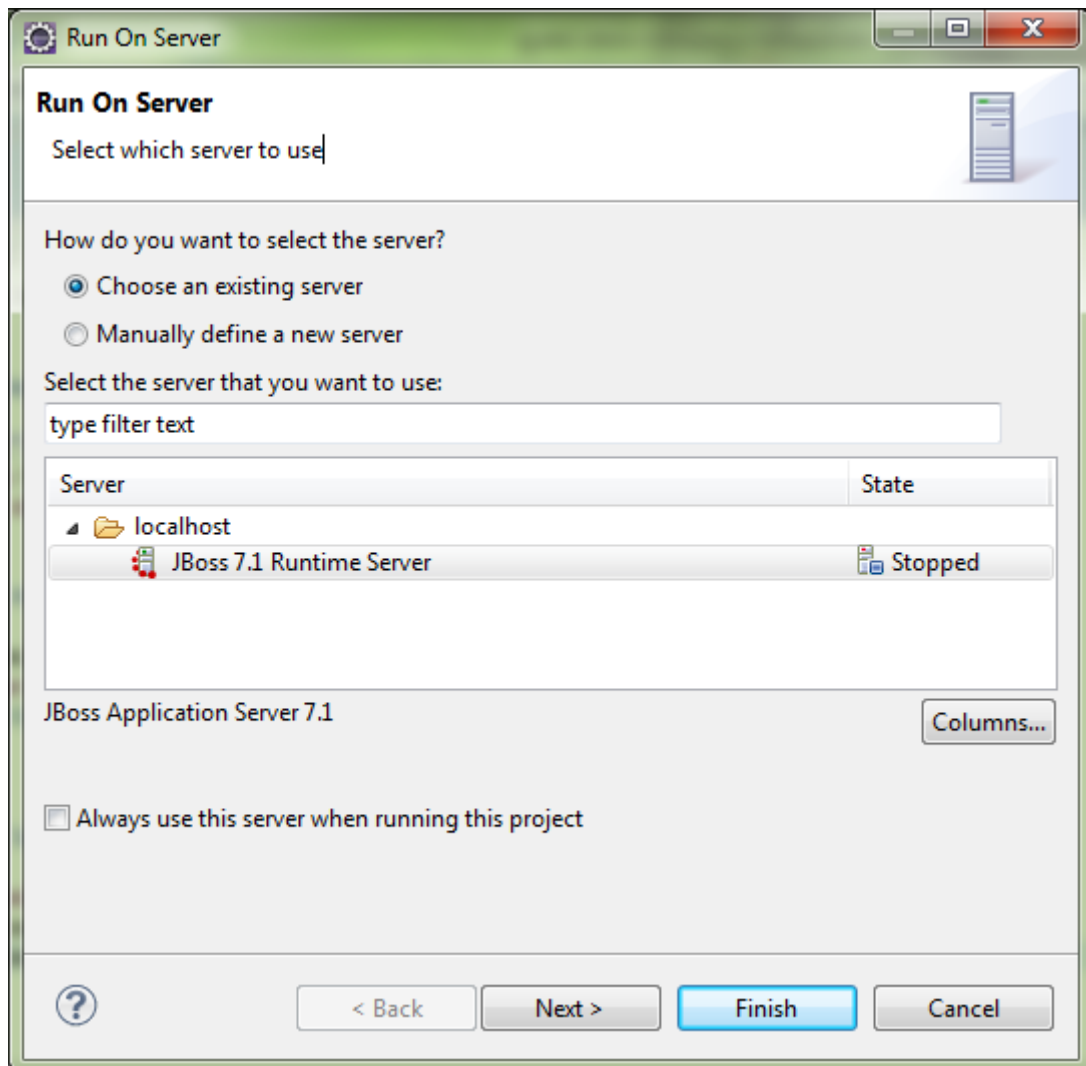
html ▾ body ▾ jsp:scriptlet ▾ #text

Visual/Source Source Preview

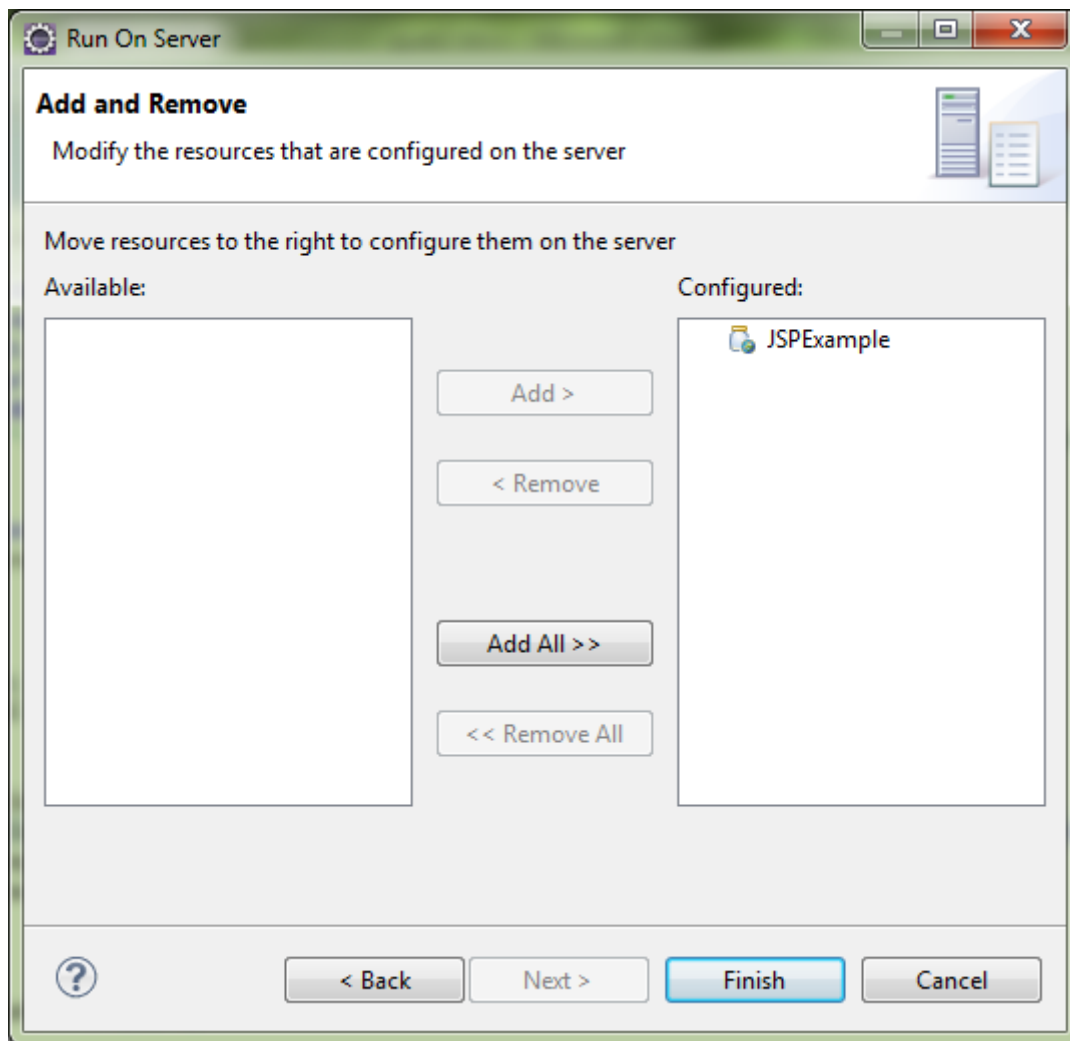
Ez a sor a „Hello World!” üzenetet a konzolra írja. Ahhoz, hogy ez az üzenet a weboldalon jelenjen meg, egyszerűen a „Hello World!” sorral kell helyettesíteni.

A projektet futtatása szerveren

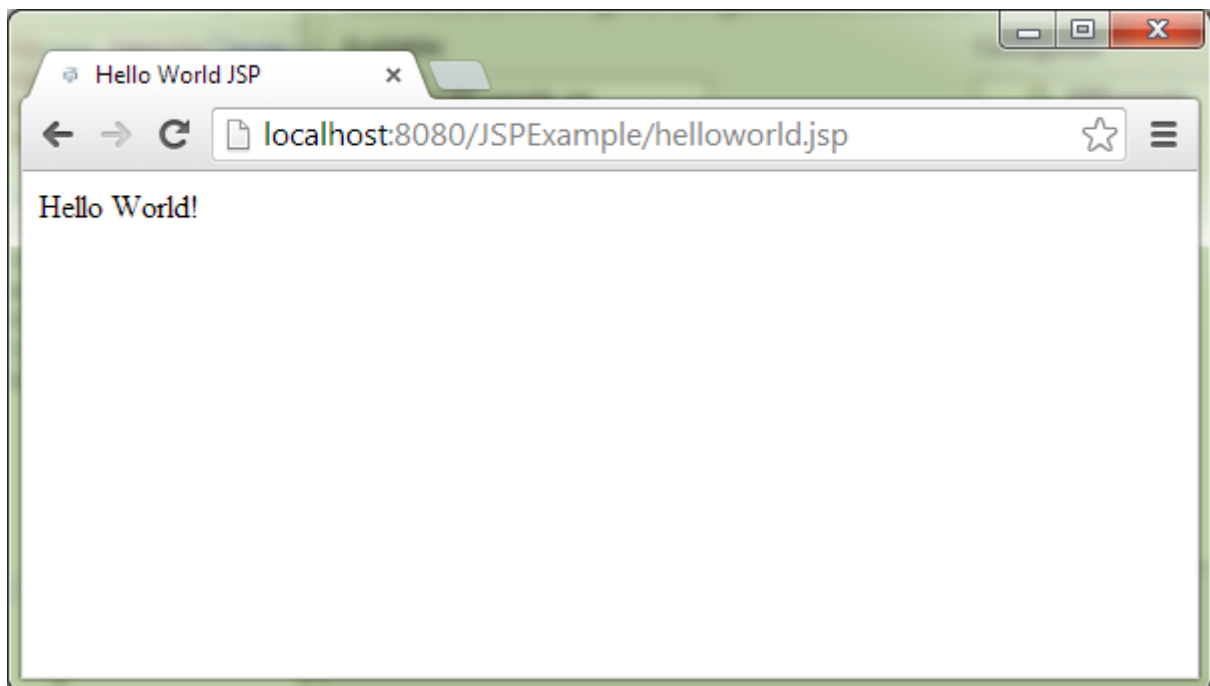
Ru As -> Run on Server, majd a megjelenő párbeszéd ablakban válasszuk ki a kívánt szervert. Jelen esetben ez a JBoss 7.1 Runtime Server.



A következő megjelenő párbeszédablakban más projektet is hozzáadhatunk a szerverhez.

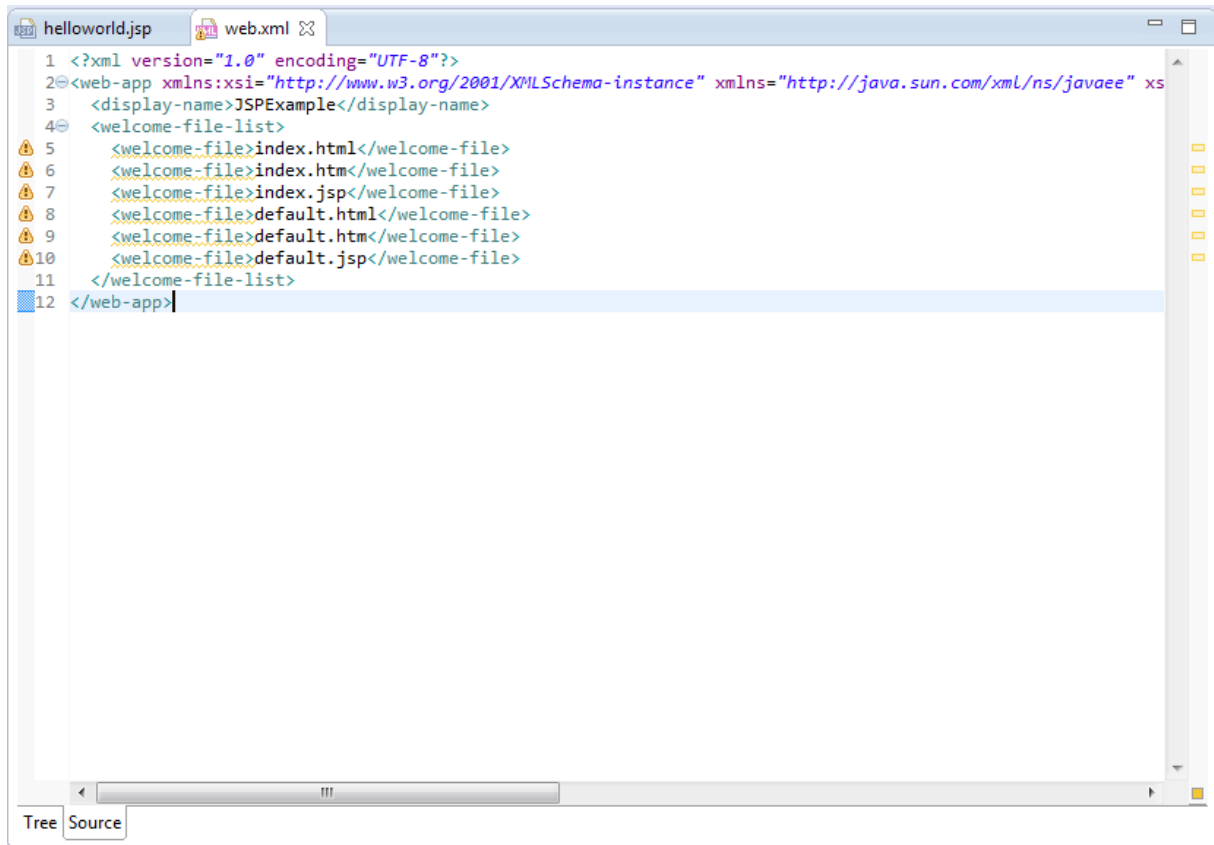


Böngészőben megnyitva az oldalt megjelenik a Hello World üzenet.



A web.xml fájl

Egy projekt létrehozásakor egy web.xml fájlt is lehet automatikusan generáltatni. Ez a fájl tartalmazza a projekt lehetséges kezdőoldalának neveit. Itt adhatjuk hozzá a létrehozott jsp oldalunkat mint kezdőoldalt.

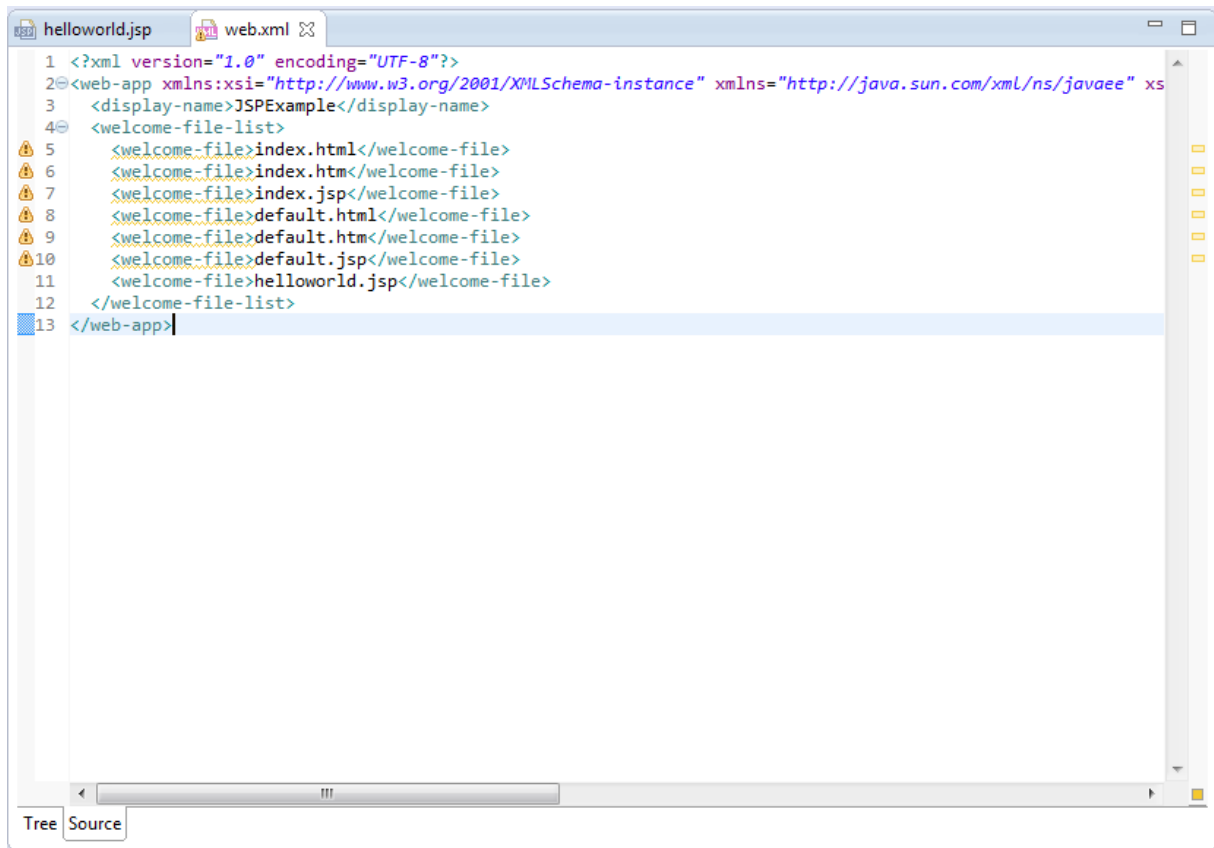


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xs
3 <display-name>JSPExample</display-name>
4 <welcome-file-list>
5 <welcome-file>index.html</welcome-file>
6 <welcome-file>index.htm</welcome-file>
7 <welcome-file>index.jsp</welcome-file>
8 <welcome-file>default.html</welcome-file>
9 <welcome-file>default.htm</welcome-file>
10 <welcome-file>default.jsp</welcome-file>
11 </welcome-file-list>
12 </web-app>
```

Az xml fájl megnyitása után kétféle szerkesztésre is mód van. A grafikus szerkesztés a Tree lap, míg a forrás szerkesztése a Source lapon érhető el.

Ahhoz hogy a helloworld.jsp oldal legyen az üdvözlő oldal, az xml fájlba a következő sort kell beilleszteni a <welcome-file-list> ... </welcome-file-list> tagek közé.

```
<welcome-file>helloworld.jsp</welcome-file>
```



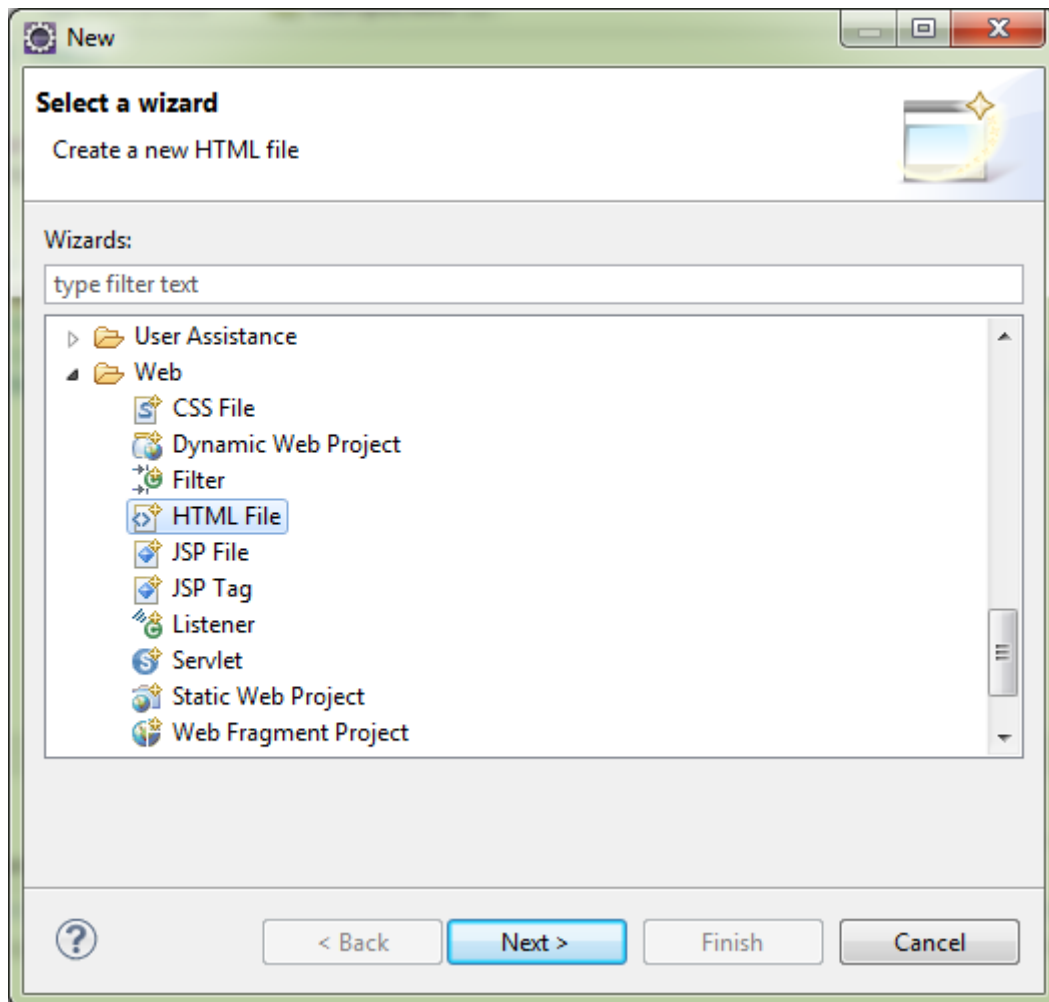
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xs
3 <display-name>JSPExample</display-name>
4 <welcome-file-list>
5 <welcome-file>index.html</welcome-file>
6 <welcome-file>index.htm</welcome-file>
7 <welcome-file>index.jsp</welcome-file>
8 <welcome-file>default.html</welcome-file>
9 <welcome-file>default.htm</welcome-file>
10 <welcome-file>default.jsp</welcome-file>
11 <welcome-file>helloworld.jsp</welcome-file>
12 </welcome-file-list>
13 </web-app>
```

Ezután a Tree lapra kattintva látható, hogy a változás automatikusan frissült.

Form feldolgozása JSP-vel

Módosítsuk az előző projektünket a következő képen: hozzunk létre egy html oldalt rajta egy formmal majd egy jsp oldalt, amely feldolgozza a formot.

Válaszunk a Menüsávban a File → New → Other..., ezen belül pedig Web → HTML File pontot.



Adjuk meg a html fájlunk nevét és válasszunk egy html szabványt.
A html fájl body részébe írjuk a következő kódrészletet:

```
<form action="example.jsp">  
  Hány sor legyen?:<input name="sorokSzama" type="text" />  
  <br>  
  <input type="submit" value="Elküld" />  
</form>
```

Majd hozzuk létre a az example.jsp-t, amely feldolgozza ezt a formot.

Válasszuk a menüben a File → New → Other... majd ezen belül Web → JSP File pontot.

Majd mint ahogyan az előzőekben tettük, adjuk meg a nevét, amely most example.jsp, majd válasszuk a JSP File (html) templétet.

A body részhez adjuk hozzá a táblázat létrehozását megvalósító kódot.

```

<table>
  <%
    String sorok = request.getParameter("sorokSzama");

    if(sorok == null || "".equalsIgnoreCase(sorok.trim())) {
      out.print("Nem adott meg sort!!");
    } else {
      int sorokSzama = Integer.parseInt(sorok);
      for(int i=0; i < sorokSzama; i++) {
        <%
          <tr>
            <td>Number</td>
            <td><%= i+1 %></td>
          </tr>
        <%
      }
    }
  <%
</table>

```

A html fájlok a következő oldalt eredményezi:

The screenshot shows a web browser window with the title "Example form". The address bar displays "localhost:8080/JSPExample/example.html". The main content area of the browser contains the text "Hány sor legyen?:" followed by a text input field containing the number "6". Below the input field is a button labeled "Elküld".

Példaként adjuk meg 6-ot és nézzük meg a feldolgozás eredményét.

