

## Apache Tomcat - Постављање прве веб апликације

Веб апликације су колекције које садрже сервлете, JSP странице, библиотеке са таговима, HTML документе, слике, CSS стилове и остале садржаје који обезбеђују захтеване функционалности или сервисе. Веб апликације које садрже сервлете и JSP странице се могу извршавати само на веб серверима који имају подршку за сервлете. Ако се веб апликација дизајнира уз поштовање одговарајућих правила, она се може поставити и на другом серверу.

Свака веб апликација има свој јединствени локатор (Uniform Resource Locator, URL) помоћу којег јој се приступа, због чега се може више веб апликација поставити на исти сервер. Стандардна структура и независност веб апликација на серверу поједностављују развој.

### ПРИМЕР (testweb)

*Подешавање контекста веб апликације*

Да би се веб апликација могла извршити на серверу потребно је извршити њену пријаву, тј. навести њен контекст (context) у фајл **server.xml**. Више детаља о конфигурисању контекста веб апликације се може пронаћи на адреси:

<https://tomcat.apache.org/tomcat-9.0-doc/config/context.html>

Може извршити подешавање да ли ће се користити **cookies** **URL rewriting** ( **debugging-** ),

- **path** - веб адреса приликом позива апликације (the URL prefix)
- **docBase** - директоријум где је веб апликација инсталирана у односу на **tomcat\_dir/webapps** директоријум.

За веб апликацију **testweb** су ови параметри:

```
<Context path="/testweb" docBase="testweb" debug="0" reloadable="true"/>
```

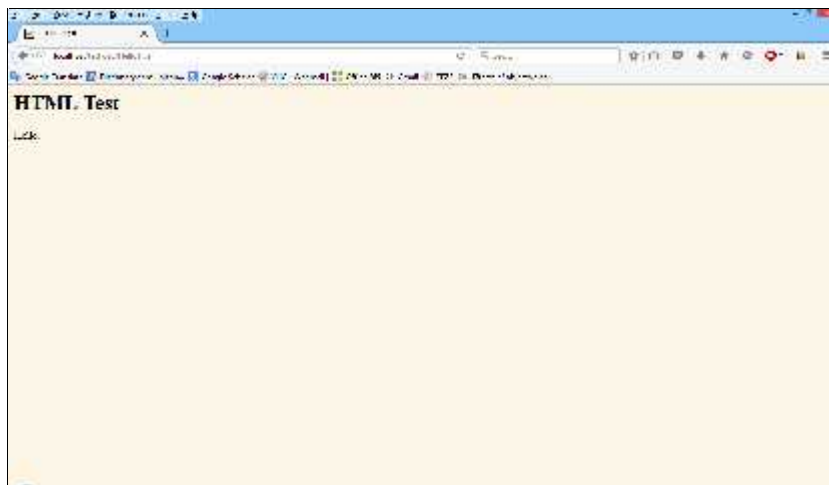
Ова линија се додаје у **server.xml** датотеку и то непосредно након линије **</Host>**, као што је приказано у следећем исечку кода:

```
</Host>
  <Context path="/testweb" docBase="testweb" debug="0" reloadable="true"/>
</Engine>
</Service>
</Server>
```

*Локација и приступ јавним ресурсима стандардне веб апликације*

У основном фолдеру, који је истовремено и назив веб апликације, се смештају јавно доступни објекти као што су JSP странице, библиотеке са таговима, HTML странице, слике, CSS стилови итд. приступ овим објектима је помоћу назива апликације и назива ресурса (JSP или HTML странице). У примеру који је део ове лекције се јавно доступним страницама приступа на следећи начин:

<http://localhost/testweb/Hello.html>



<http://localhost/testweb/Hello.jsp>

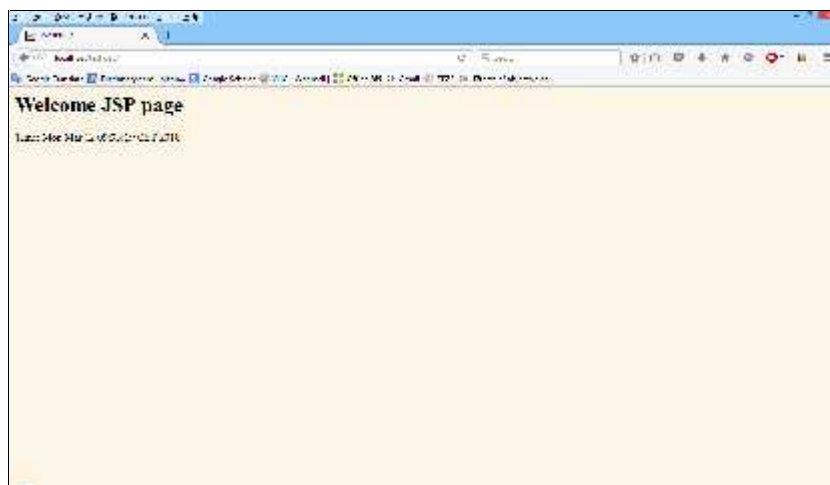


Ако се жели обезбедити **листа подразумеваних страница** које се појављују када се у веб претраживачу откуца назив апликације, тада се у конфигурациони фајл апликације **web.xml** додаје ставка **welcome-file-list** у коју се уписују странице у редоследу којим ће бити позиване приликом позива апликације. Ако се не пронађе ни једна од наведених страница веб сервер ће реализовати подразумевано понашање (нпр. порука о грешци ил листинг директоријума). Пример листе страница које се стартују приликом позива апликације је:

```
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>index.html</welcome-file>
</welcome-file-list>
```

Приликом позива веб апликације само помоћу назива позива се прво **index.jsp** страница, па ако она не постоји позива се **index.html** страница.

<http://localhost/testweb>



*Локација и приступ сервлетима*

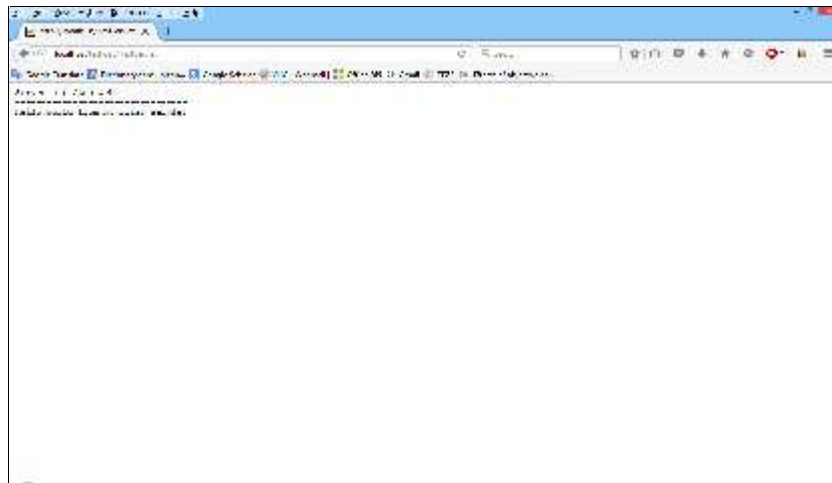
Да би се извршио сервлет се мора декларисати и мапирати адреси која га лоцира (URL) тако што се то додаје у фајл **web.xml**. Серверу се мора рећи да сервлет постоји тако што се упише пуно име класе сервлета. Исто тако се мора јасно назначити која адреса (URL) треба да се специфицира приликом позива сервлета. Пример:

```
<servlet>
  <servlet-name>TextServlet</servlet-name>
  <servlet-class>TextServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>TextServlet</servlet-name>
  <url-pattern>/TextServlet</url-pattern>
</servlet-mapping>
```

Елемент **servlet** и његови поделементи дају информацију серверу о имену које је додељено сервлету и име класе која представља сервлет. Елемент **servlet-mapping** и његови поделементи указују серверу који сервлет ће бити позван када се захтева наведена URL адреса (**url-pattern**). На пример ако се позове следећа адреса

**<http://localhost/testweb/TextServlet>**

биће позван сервлет **TextServlet** који је у основном фолдеру **class** веб апликације **testweb**.

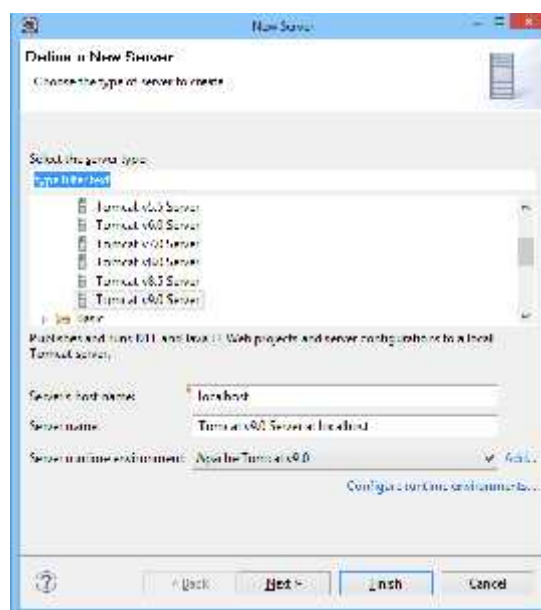


### Креирање пројекта у развојном окружењу Eclipse

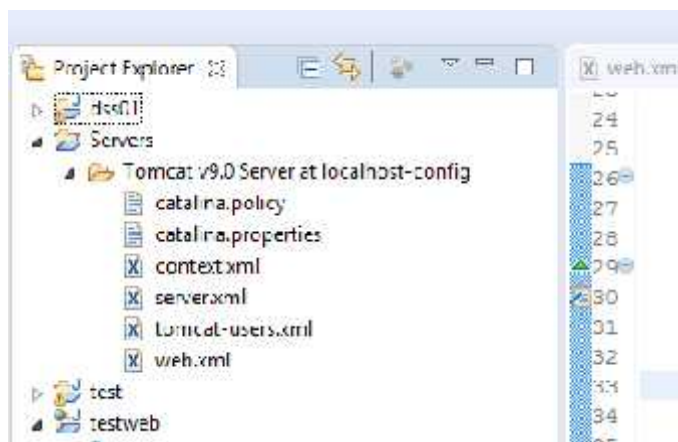
Да би се креирао пројекат динамичке веб апликације треба у развојном окружењу променити перспективу у JEE. Потом треба додати инсталирани сервер Tomcat 9, да би се могла правити апликација са структуром коју подржава Tomcat. То се ради тако што се у картици Servers покрене линк који указује да нема тренутно инсталираних сервера и да треба додати нови сервер



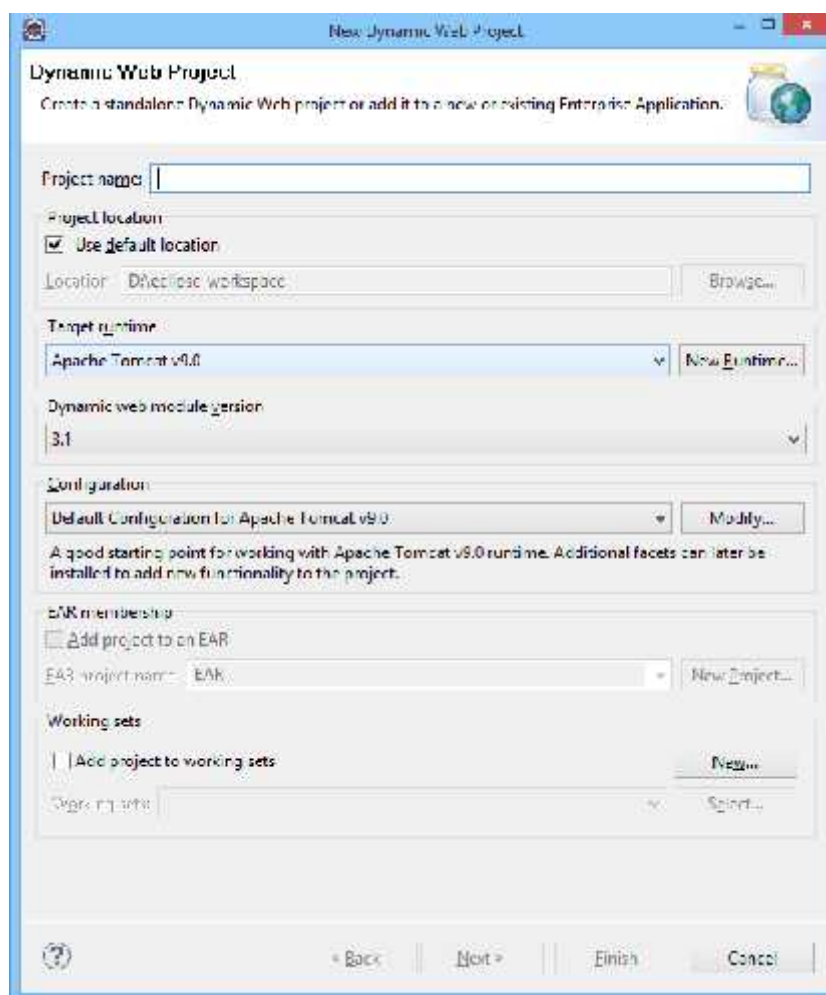
при чему треба одабрати инсталирани сервер:



Након тога ће инсталирани сервер бити доступан у развојном окружењу из којег се може вршити и конфигурација сервера. **Препорука је да се сервер конфигурише директно помоћу његових алата а не из развојног окружења Eclipse.**

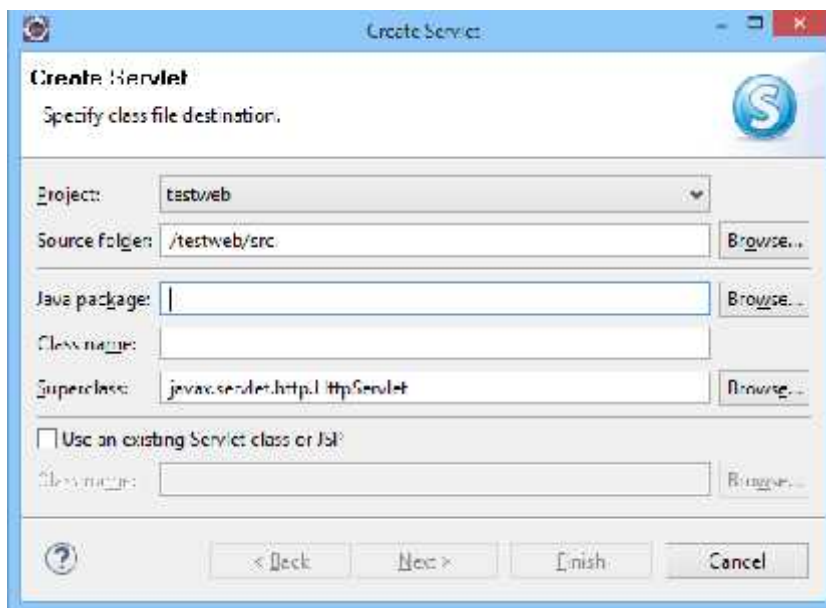


Приликом креирања пројекта треба одабрати опцију **Dynamic Web Project**, где треба одабрати инсталирани сервер:



У тако креирану апликацију се могу додавати сервлети и остали елементи и позиционирати у складу са дефинисаном структуром веб апликације.

Приликом креирања новог сервлета, треба навести пакет где ће сервлет бити лоциран и име сервлета, а подразумева се да наслеђује класу `javax.servlet.http.HttpServlet`.



Компајлирање сервлета се аутоматски ради, па се на сервер пребацују само компајлирани `*.class` фајлови.

#### Literature and Links

- [1] **The Apache Tomcat.** <http://tomcat.apache.org/>
- [2] **Apache Tomcat 9 Documentation.** <https://tomcat.apache.org/tomcat-9.0-doc/index.html>
- [3] **JavaServer Pages (JSP) Specification, Version 2.3.**  
<http://jcp.org/aboutJava/communityprocess/mrel/jsr245/index2.html>
- [4] **Servlet API Specification, Version 4.0.**  
<https://jcp.org/aboutJava/communityprocess/final/jsr369/index.html>
- [5] **Java(TM) EE 8 Specification APIs.** <https://javaee.github.io/javaee-spec/javadocs/>
- [6] **Apache Tomcat 9.0.5 API.** <https://tomcat.apache.org/tomcat-9.0-doc/api/index.html>
- [7] **Servlet 4.0 API - Apache Tomcat.** <https://tomcat.apache.org/tomcat-9.0-doc/servletapi/index.html>
- [8] **JSP 2.3 API - Apache Tomcat.** <https://tomcat.apache.org/tomcat-9.0-doc/jspapi/index.html>