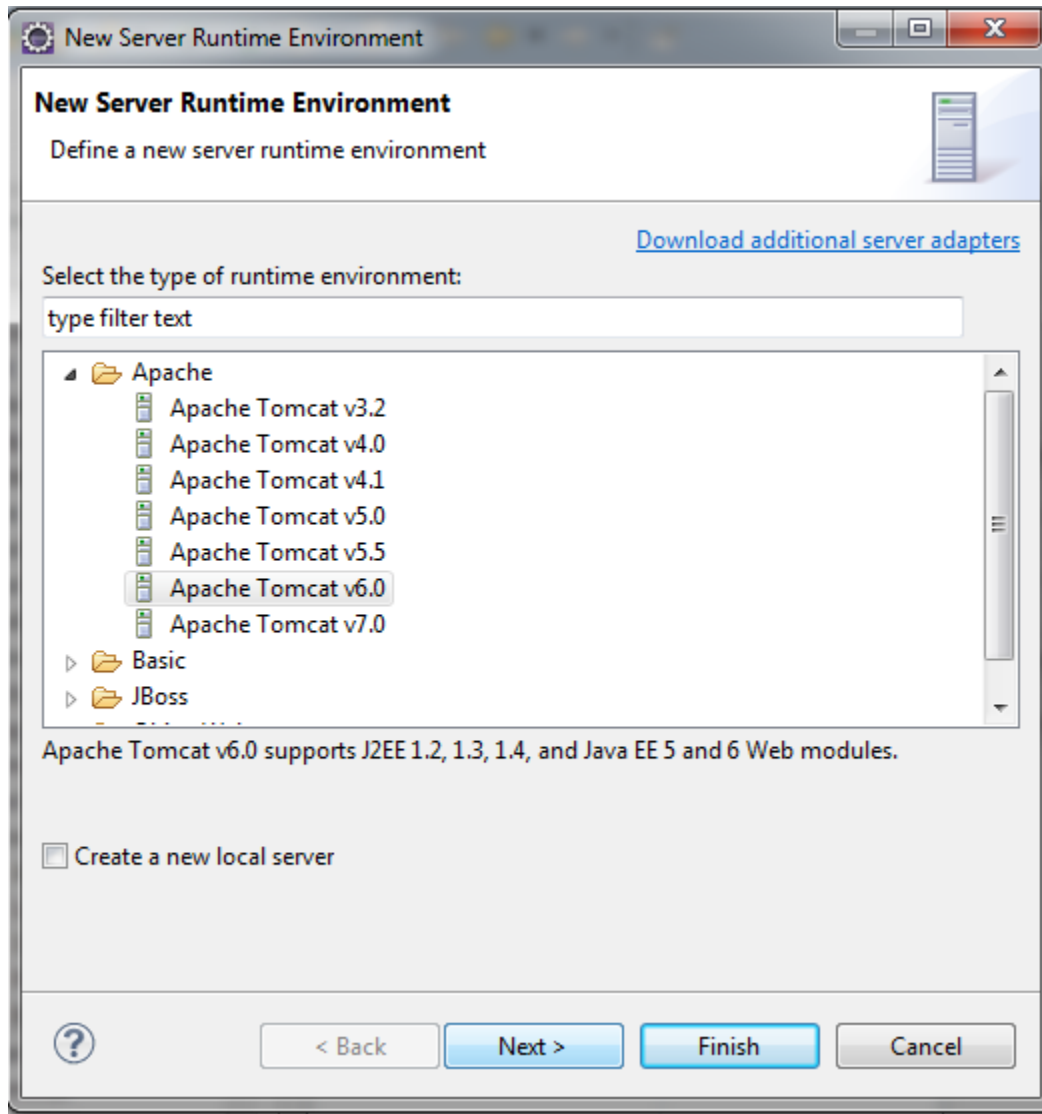


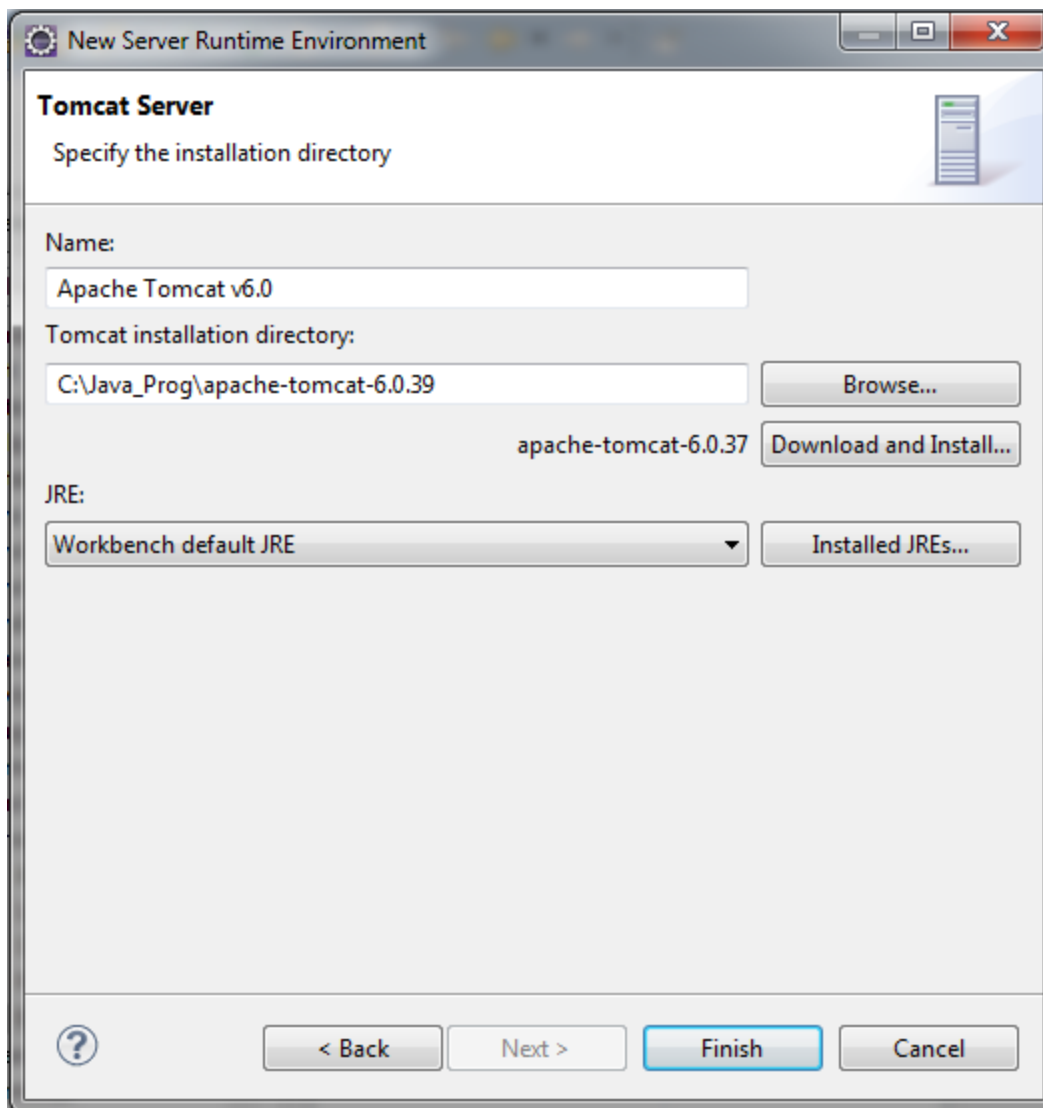
## Razvoj Web aplikacija(prejekata) u Eclipse okruženju

Pre nego što napravite bilo koji Dinamički web projekat uradite sledeće u Eclipsi:  
Odaberite Window->Preferences->Server->RuntimeEnvironments, a zatim kliknite na dugme Add. Iz liste izaberite Apache Tomcat v6.0.



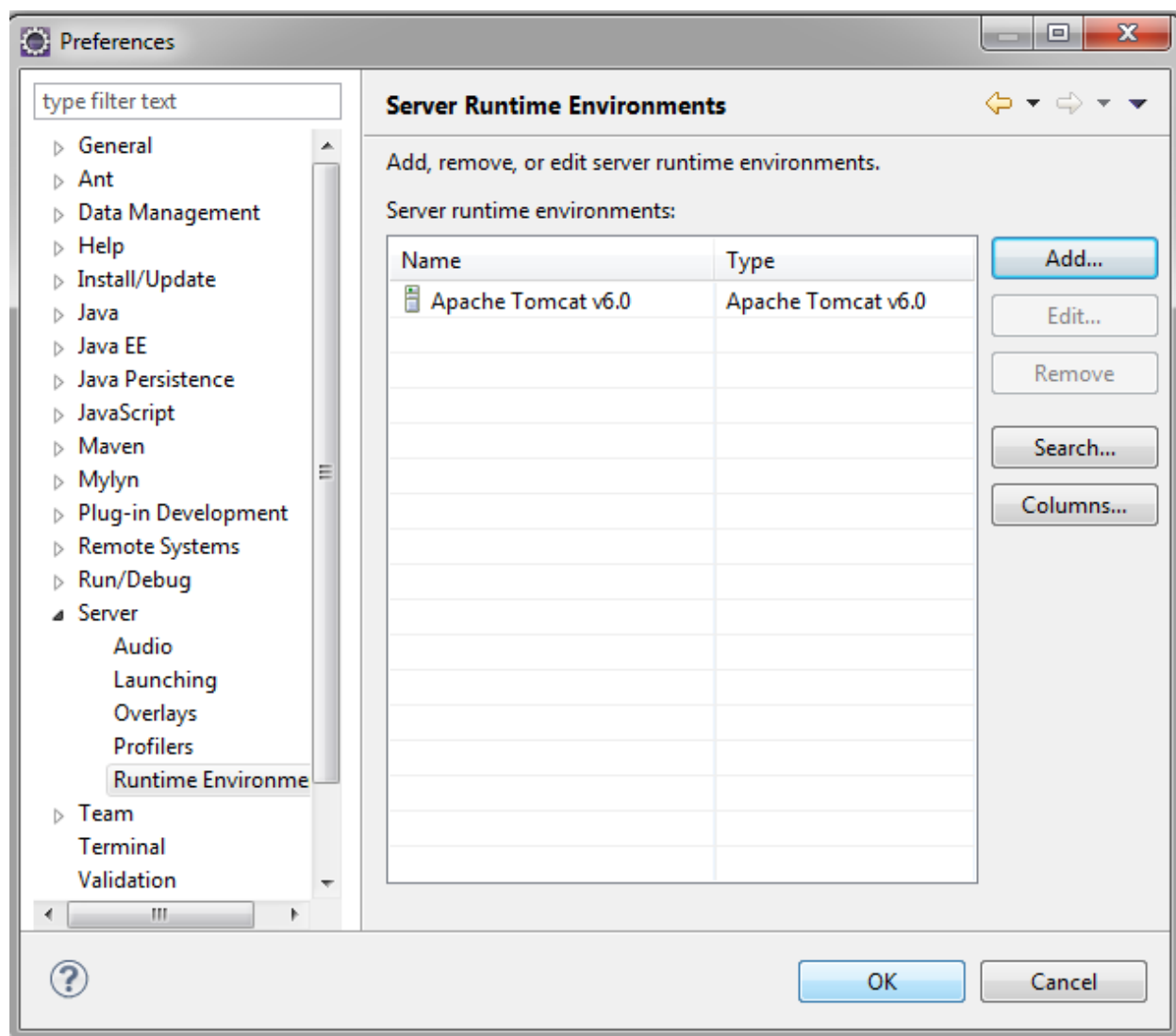
Slika 1 - Runtime Environment

Na sledećem dijalogu preko dugmeta Browse podesite putanju na vašem računaru do raspakovanog Tomcat web servera (Na slici 2 je primer putanje na mom hard disku).



Slika 2 - Dijalog za postavljanje putanje Servera

Kada odaberete dugme Finish imaćete sledeću situaciju:

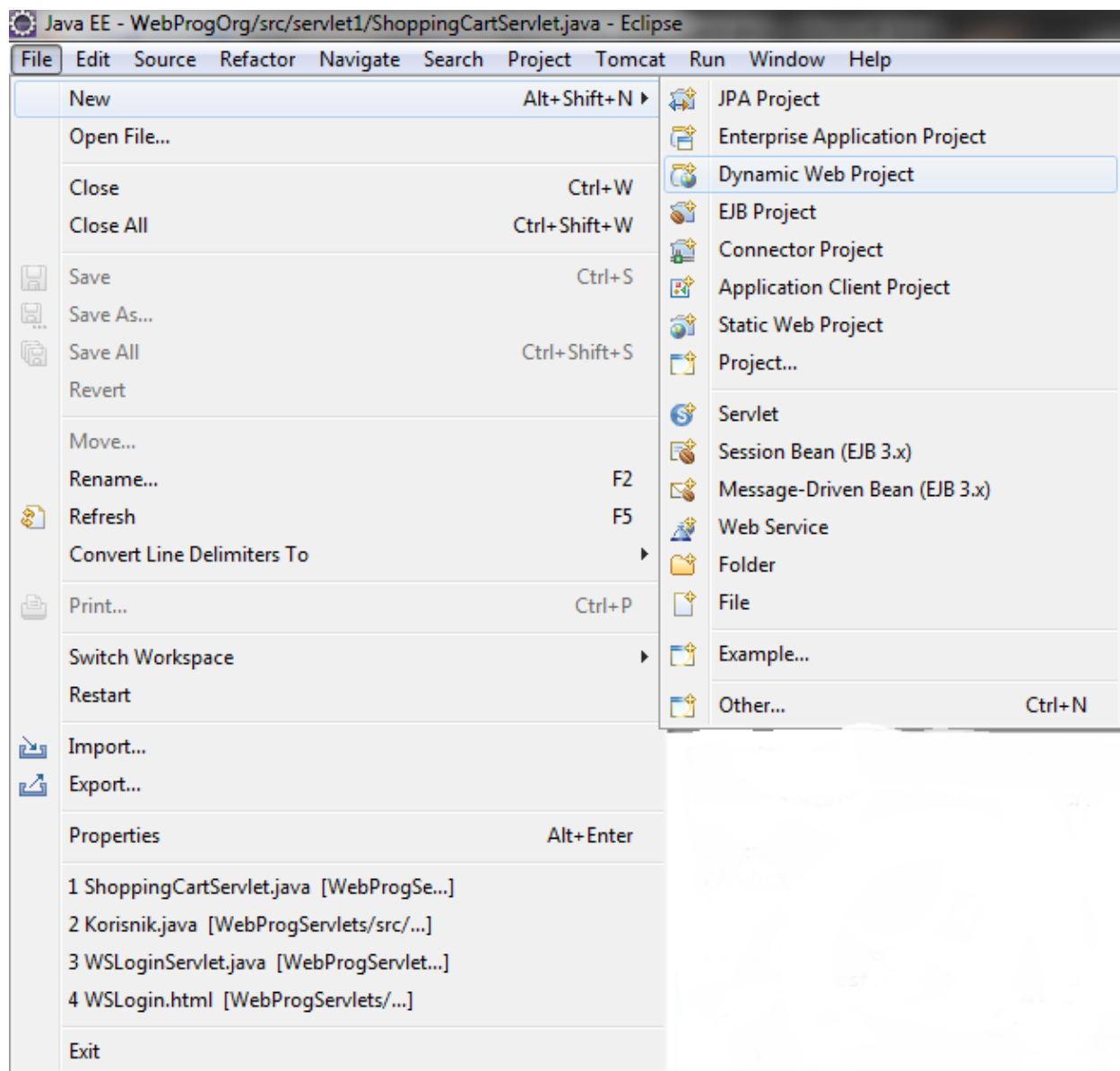


Slika 4 - Dodata veza između Tomcat i Eclipse

Ovime smo obezbedili da Eclipse projekat bude svestan `servelt-api.jar` biblioteke iz Tomcat. Inače bi smo morali da stavimo ovu biblioteku u `WEB-INF/lib` folder u svaki novi dinamički web projekat. **Ovo je mnogo bolja varijanta nego da kopiramo `servelt-api.jar` biblioteku.**

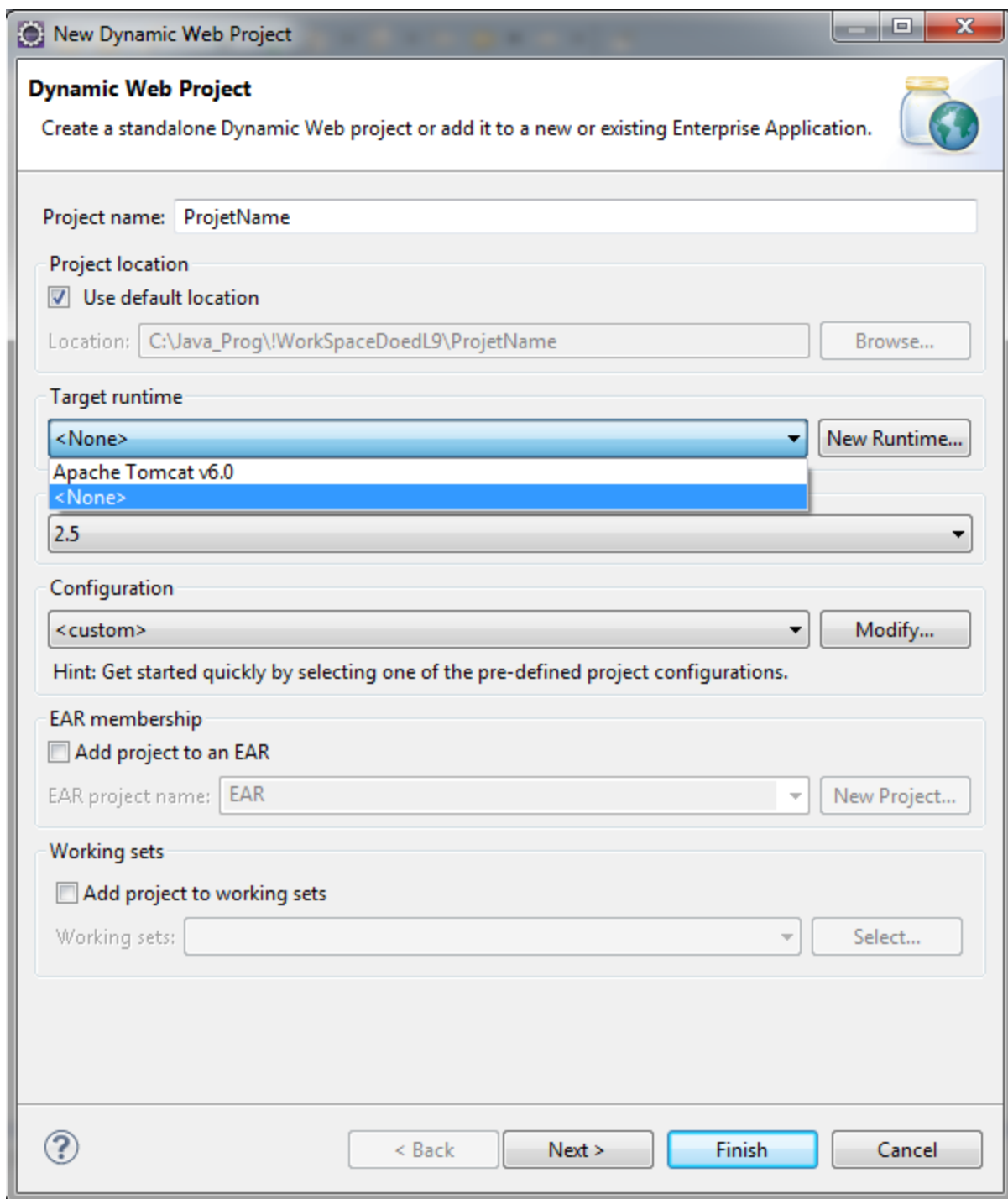
Postoje dva tipa web projekata: dinamički i statički. Dinamički web projekti mogu da sadrže servlete, jsp stranice i slične "dinamičke" sadržaje koje pozivaju korisnici slanjem odgovarajućih zahteva. Dok statički projekti sadrže samo HTML stranice, slike i druge "statičke" resurse koji se isporučuju klijentu u istom (nepromenjom) obliku u kakvom se nalaze i na serveru.

Novi dinamički projekat se kreira u Eclipsu kao na slici 5.



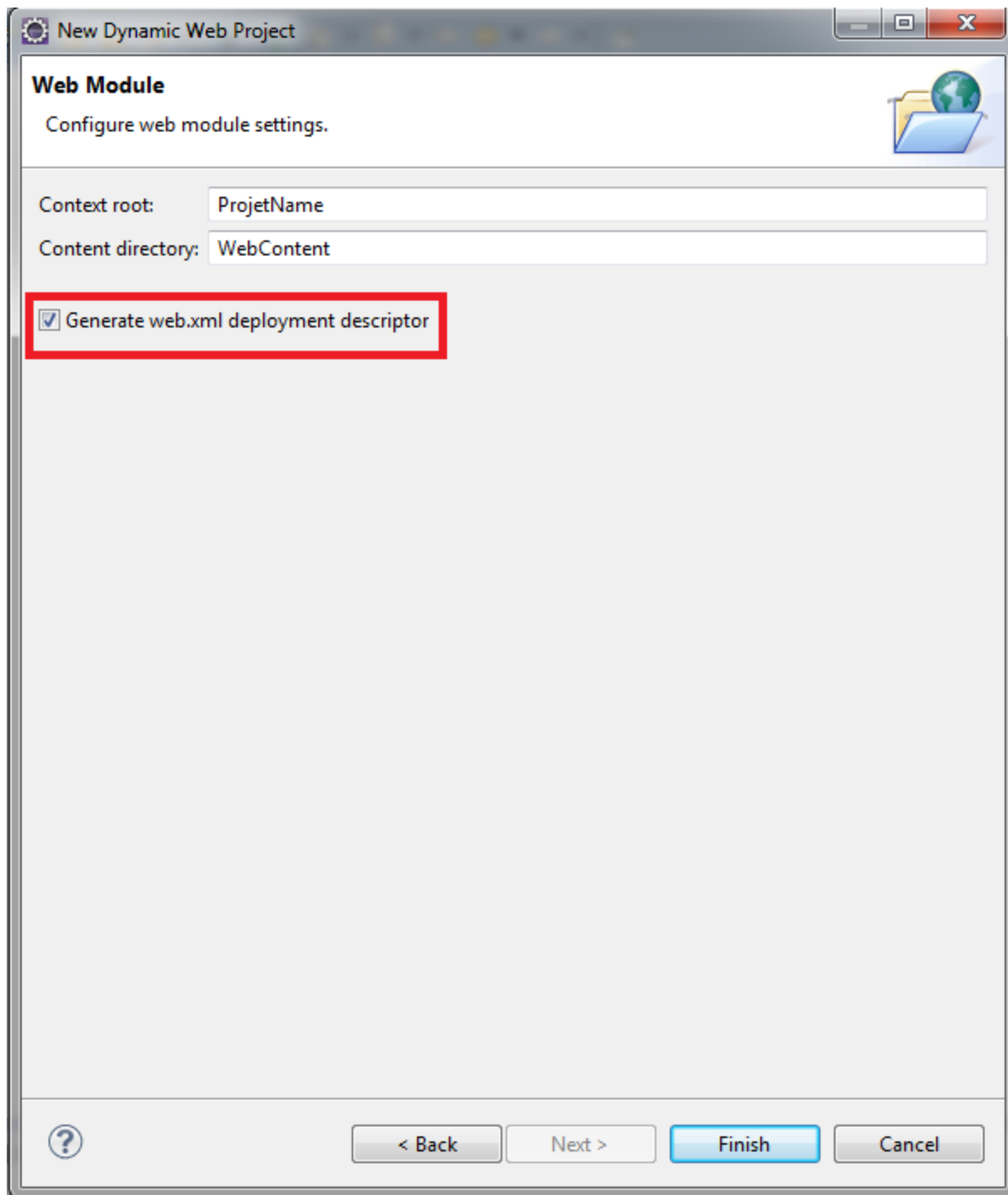
Slika 5 - Dinamički Web projekat

Na sledećem dijalogu odabiremo ime projekta i Runtime Environment Apache Tomcat v6.0 (to je ono što smo prethodno podesili) verzija Dynamic Web Module je 2.5. To je verzija Servleta sa kojom Tomcat 6 adekvatno funkcioniše.



Slika 6 - Podešavanje Dynamic Web project

Kliknemo na Next dugme dok ne dođemo do sledećeg dijaloga:



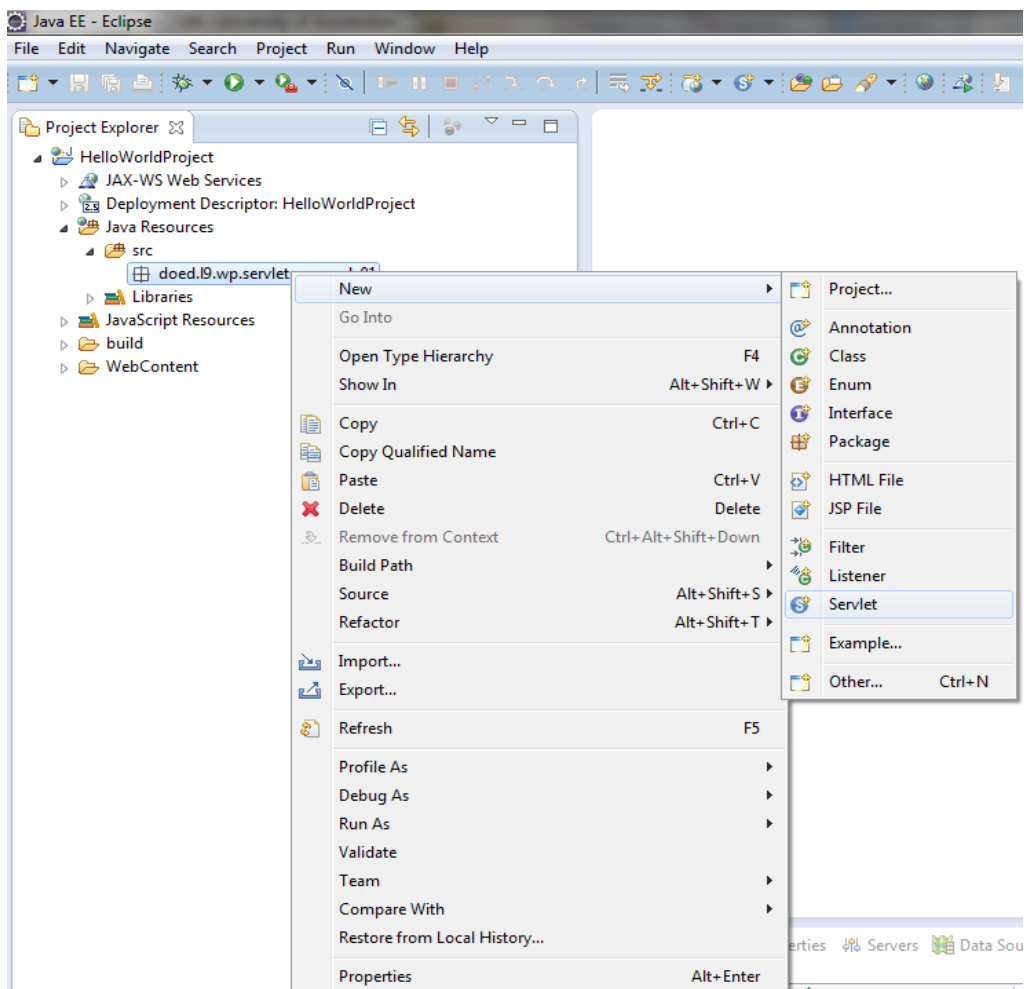
Slika 7 - Automatsko generisanje Deployment descriptora

Na njemu obavezno naznačiti da se prilikom formiranja projekta automatski generiše i web.xml (fajl u kome mapiramo Servlete).

Kreirani Dinamički Web projekat treba da ima sledeću strukturu foldera:

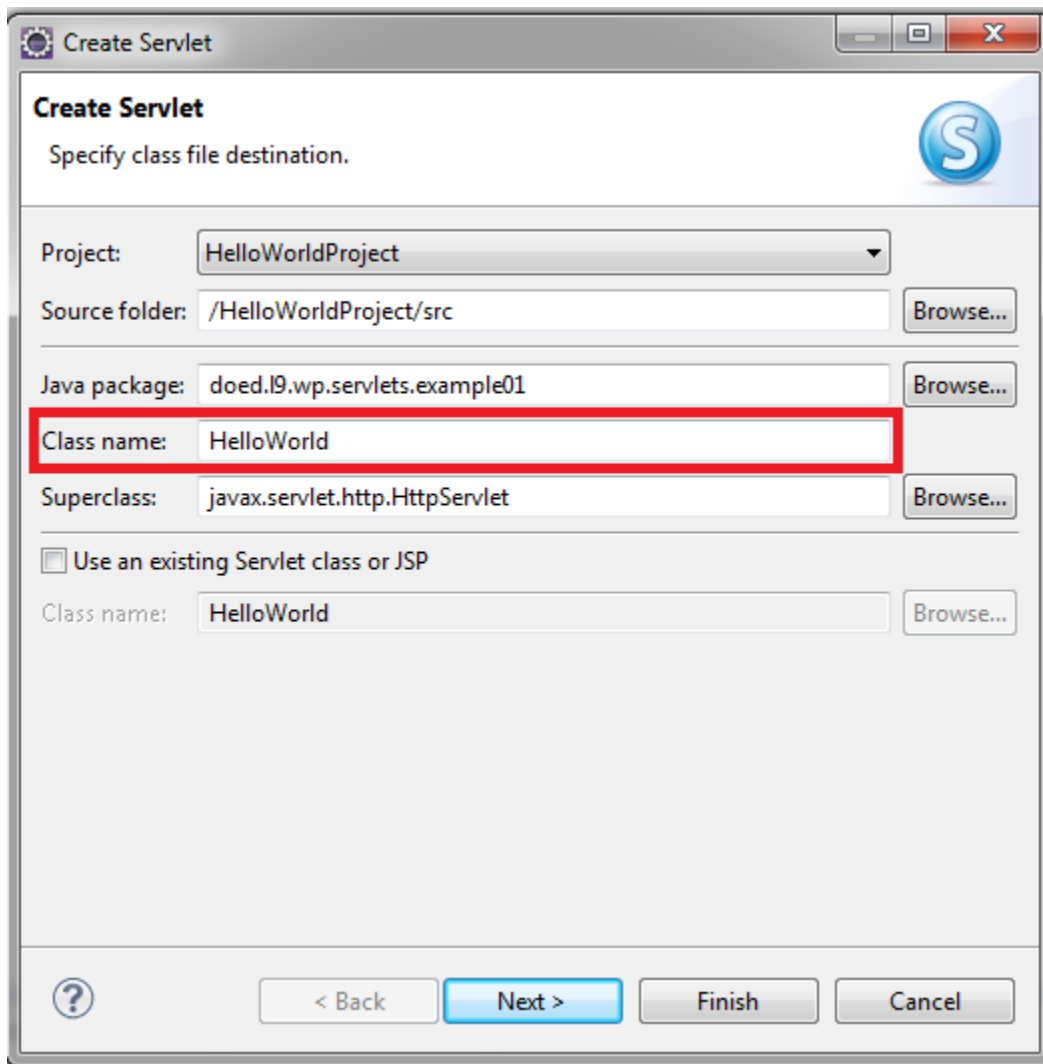
- Java ReSources - U njega ćemo smeštati java izvorne datoteke (.java), binove i servlete. Oni se u Eclipsi automatski kompajliraju i njihovi class fajlovi su smešteni u WEB-INF/classes direktoriju ili u build/classes.
- WebContent - Folder u kome lokaciju nalaze Web resursi. Tako se u njemu mogu naći dinamički resursi (JSP stranice) ili statički (slike, fajlovi, HTML stranice, CSS stilovi ...). Kada se aplikacija prebaci na server, on neće moći da pristupi pomenutim web resursima ako se oni ne nalaze u ovom folderu ili nekom od njegovih podfoldera.
  - WEB-INF - Podfolder u WebContent. sadrži relevantne web resurse za funkcionisanje web aplikacije. Sadrži i fajl web.xml o kome će biti reči kasnije.
    - classes - Podfolder WEB-INF je output direktorijum prekomajliranih Java klasa (one iz JavaResources). Podfolderi u ovom folderu čuvaju strukuru paketa iz Java ReSources
    - lib - Podfolder u WebContent. Nalaze se sve eksterne biblioteke (jar fajlovi) koji su neophodni za funkcionisanje Web aplikacije.

Novi Servlet kreiramo tako što u okviru Java Resources folder src napravimo paket i u njemu odaberemo kreiranje novog servleta kao na slici 8.



Slika 8 - Kreiranje novog Servleta

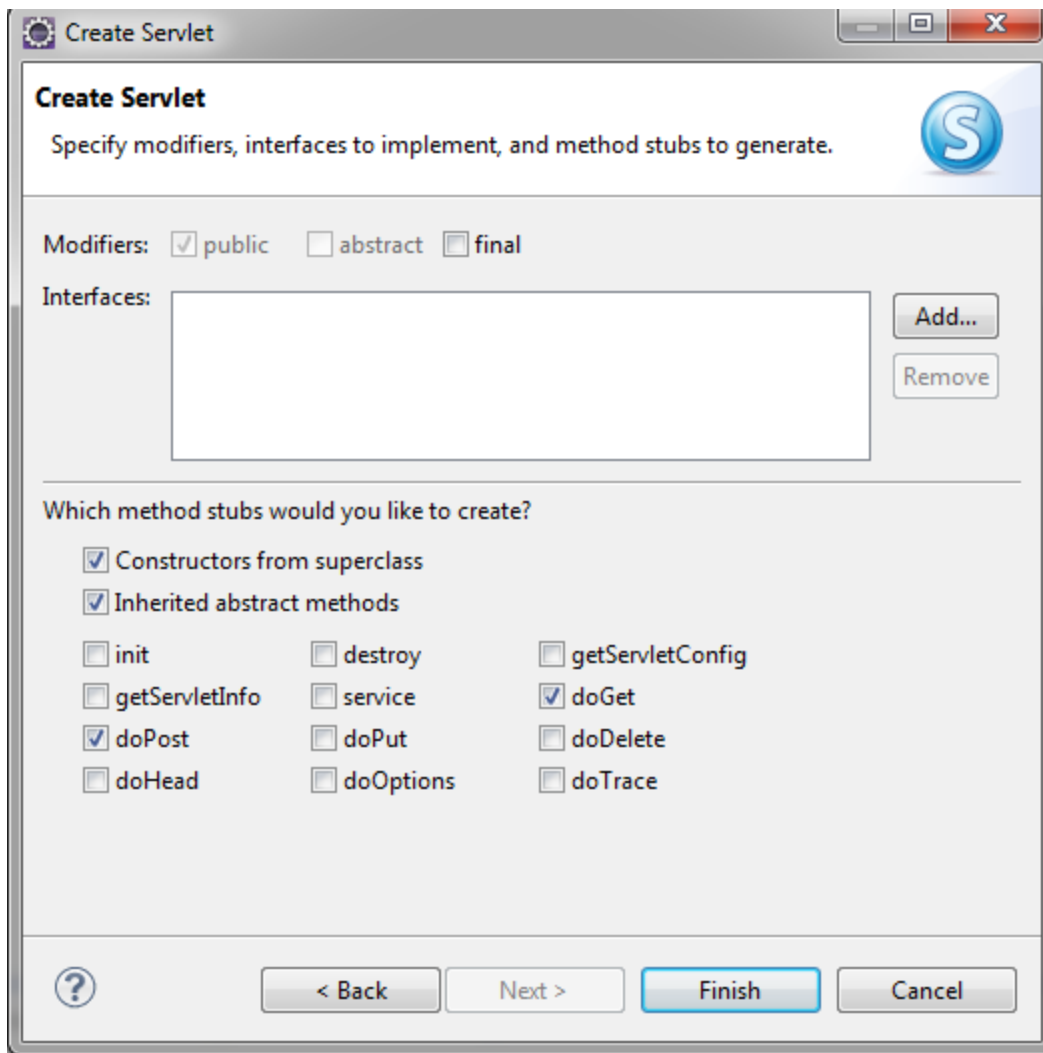
Na sledećem dijalogu odabiremo ime klase



Slika 8 - Definisanje imena klase Servlet

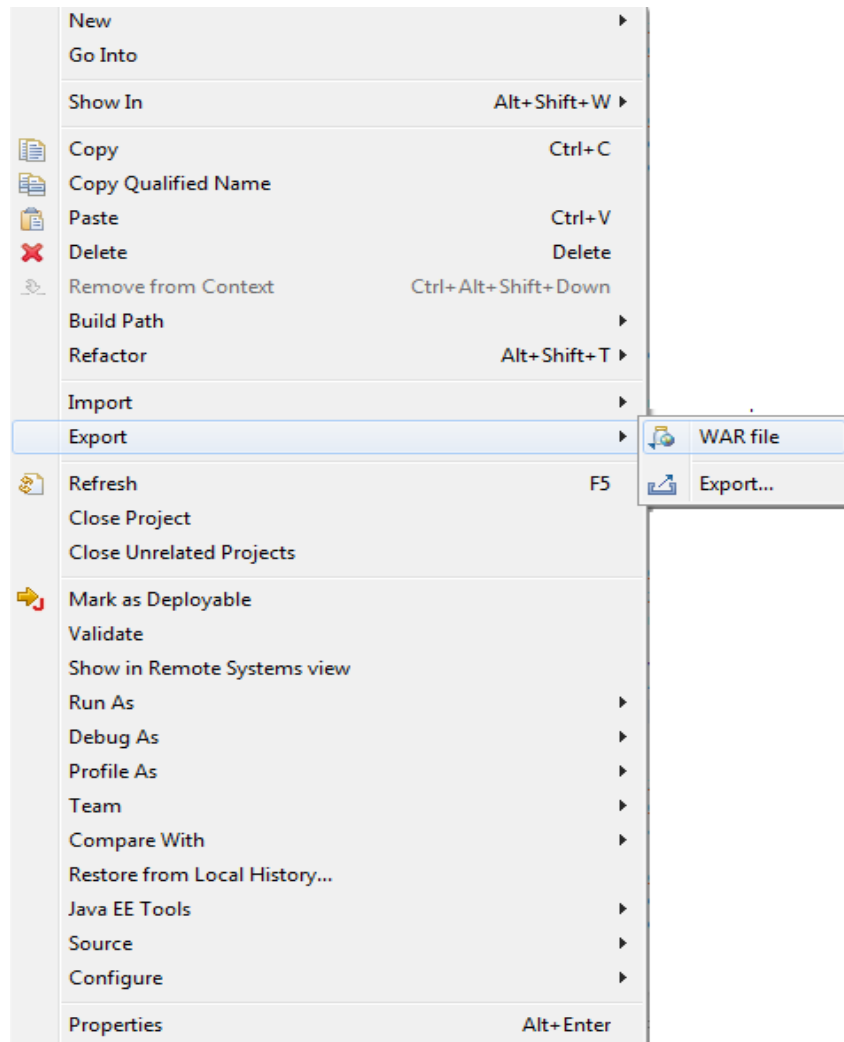
kada stignemo do poslednjeg dijaloga neophodno je naznačiti koje metode će Servlet incijlano generisati. Nama su potrebne doGet,doPost i init metoda,





Slika 9 - Metode Servleta

Da bi se dinamička aplikacija pokrenula na Web serveru neophodno je da se aplikacija prebaci (eksportuje) u oblik čitljiv za Web server. Kako ćemo mi našu dinamičku web aplikaciju pokretati pod Apache Tomcat web serverom neophodno je da od aplikacije napravimo war arhivu (slika 10). War arhiva je packed arhiva slična zip ili jar arhivi.



Slika 10 - Eksport u war

U pomenutu war arhivu ide sadržaj WebContent foldera i podfoldera, iliti sve prekompajlirane java klase i web resursi. War arhiva se mora smestiti u webapps folder Tomcata.

## Web Deployment Descriptor - web.xml

Fajl u kome se opisuje naša web aplikacija za potrebe servera. U web terminologiji naziva se “web application's deployment descriptor”. U ovom fajlu se opisuju klase, resursi i konfiguracije, kao i način kako će ih web server koristiti kada odgovara na zahteve klijenata. Ovaj fajl se mora nalaziti u WEB-INF folderu. Prilikom formiranja war fajla i ovaj fajl će se naći u arhivi u prethodno navedenom folderu. U nastavku će biti dat primer web.xml fajla i biće objašnjeni njegovi najznačajniji (najkorišćeniji) elementi.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
```

```

<servlet>
  <servlet-name>comingssoon</servlet-name>
  <servlet-class>mysite.server.ComingSoonServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>comingssoon</servlet-name>
  <url-pattern>/comming</url-pattern>
</servlet-mapping>
<servlet>
  <display-name>HelloWorld</display-name>
  <servlet-name>HelloWorld</servlet-name>
  <servlet-class>rs.ac.uns.ftn.informatika.dosk.java.vezbe03.primero1.HelloWorld</servlet-
class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloWorld</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
</web-app>

```

Element <servlet> definiše java web servlet klasu, pod element <servlet-name> predstavlja ime servleta, <servlet-class> predstavlja putanju (hijerarhiju paketa-direktorijuma) do konkretne klase u našem projektu. <servlet-mapping> ide u paru sa elementom <servlet>. Za svaki servlet moramo imati mapiranje koje koristi sam web server prilikom odgovora na zahteve korisnika. Pod element <servlet-name> mora da se poklapa sa istim podlementom iz <servlet> elementa. <url-pattern> je naziv koji će vam omogućiti da pristupite konkretnom resursu servletu. Na primer, ako se projekat zove **Examples** servlet HelloWorld a elementu <url-pattern> piše **/hello**, mi ćemo pristupiti tom servletu kroz web čitač ako unesemo:

localhost:8080/Examples/hello.

U slučaju da unesemo samo localhost:8080/Examples/

web server će pogledati da li postoji element <welcome-file> i odgovoriće sa prvim resursom navedenim u ovoj listi (index.html).

## Apache Tomcat

Apache Tomcat je web server otvorenog koda razvijen od strane Apache Software Foundation. Tomcat predstavlja web server koji u potpunosti podržava java web bazirane aplikacije. Kako bi mogli da upravljamo Tomcat serverom neophodno je da poznamo i strukturu foldera istog. Tomcat sadrži sledeće foldere:

- bin - sadrži fajlove koji omogućavaju pokretanje i zaustavljanje samog servera.
- conf - čuva konfiguracione fajlove kojima se može menjati ponašanje servera.
- lib - ugrađene biblioteke koje koristi prilikom rada sa svojim aplikacijama. biblioteka koja se nalazi u ovom folderu dostupna je za sve aplikacije koje su podignute pod Tomcat serverom.

- webapps - najznačajniji folder. U njemu kopiramo war arhivu koja se raspakuje prilikom pokretanja tomcat servera. Nakon raspakovanja dobijamo folder sa istim imenom kao i war arhiva u kome se nalaze svi web resursi (sve iz WebContent iz Eclipse) koje je programer definisao prilikom konstruisanja web aplikacije.
- temp, work - tomcat koristi ove foldere za smeštanje privremenih resursa(fajlova) tokom svog rada.

## **Pokretanje Apache Tomcat**

### **Iz komandnog prozora (konzole)**

Tomcat se startuje pokretanjem određenog fajla (skripte) iz bin foldera:

Linux: sh catalina.sh run

Windows: startup.bat

Da bi ste proverili da li radi Tomcat u web browseru ćete ukućati localhost:8080

Pokretanje konkretne aplikacije koja radi pod Tomcat web serverom obavljamo tako što u browseru kucate sledeću adresu:

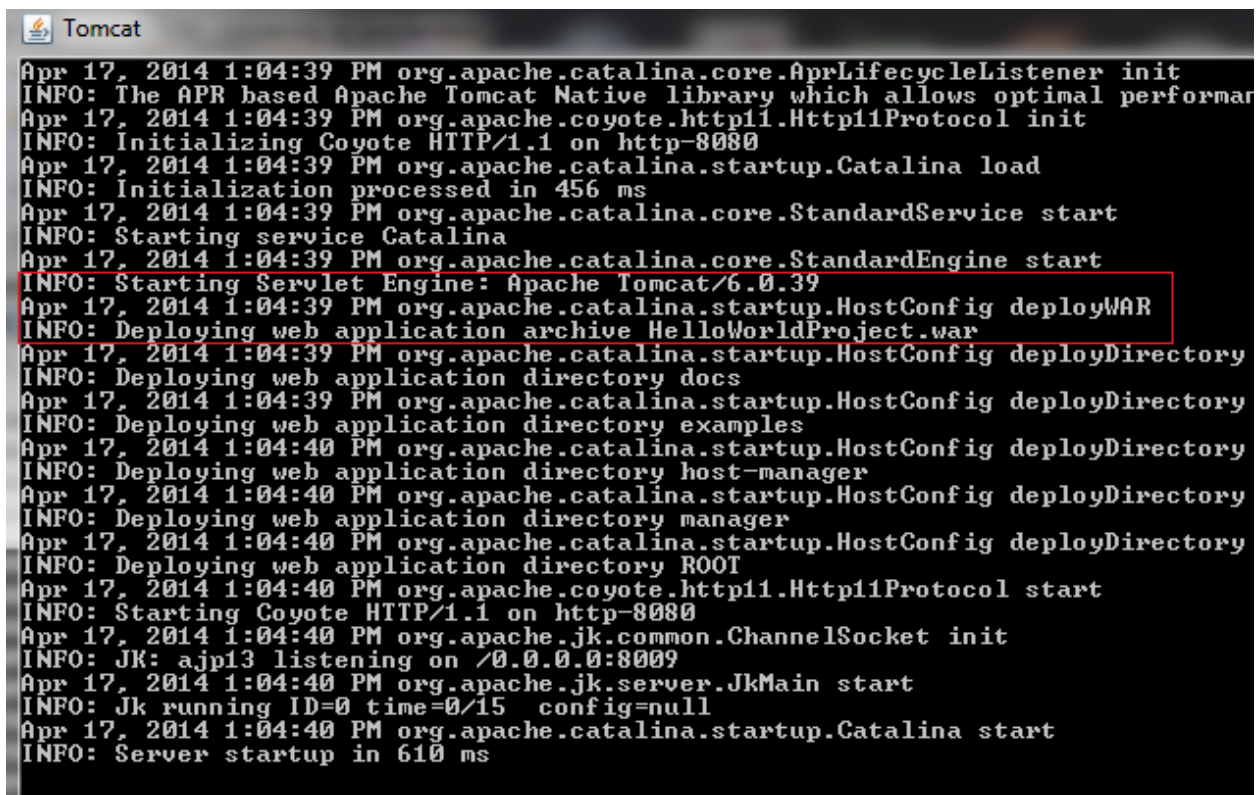
localhost:8080/Ime\_Aplikacije

localhost – je simboličko ime vaše lokalne mašine. Umesto ovoga može da stoji i IP adresa.

8080 – predefinisani port na kome je pokrenut Tomcat web server

Ime\_Aplikacije – Najčešće se poklapa sa imenom projekta(aplikacije) koju ste razvijali.

Indikator da je vaša aplikacija (war) smeštana i adekvatno pokrenuta pod Tomcat je natpis na slici:

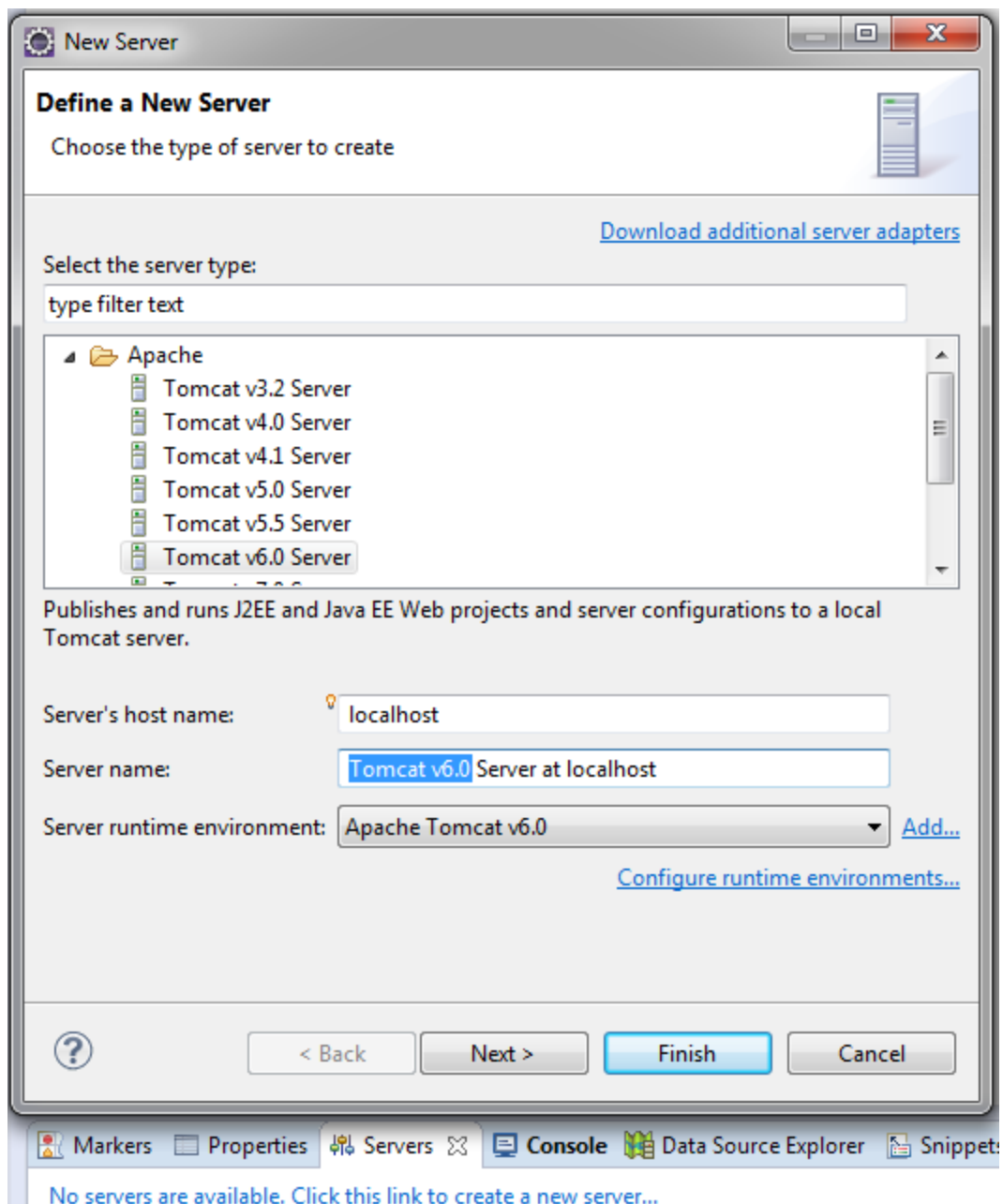
A screenshot of a Tomcat console window. The title bar says 'Tomcat'. The console displays a series of log messages from April 17, 2014, at 1:04:39 PM. The messages show the initialization of the APR-based Apache Tomcat Native library, the Coyote HTTP/1.1 protocol, and the startup of the Catalina service. A red rectangle highlights the line 'INFO: Starting Servlet Engine: Apache Tomcat/6.0.39' and the subsequent line 'INFO: Deploying web application archive HelloWorldProject.war'. Other messages include 'INFO: Initialization processed in 456 ms', 'INFO: Starting service Catalina', 'INFO: Starting StandardEngine start', 'INFO: Deploying web application directory docs', 'INFO: Deploying web application directory examples', 'INFO: Deploying web application directory host-manager', 'INFO: Deploying web application directory manager', 'INFO: Deploying web application directory ROOT', 'INFO: Starting Coyote HTTP/1.1 on http-8080', 'INFO: JK: ajp13 listening on /0.0.0.0:8009', 'INFO: Jk running ID=0 time=0/15 config=null', and 'INFO: Server startup in 610 ms'.

Slika 11 - Izgled konzole kada startovan Tomcat

Da bi ste zaustavili (ugasili) Tomcat morate da zatvorite konzolu u kojoj je on startovan ili istovremeno pritisnete tastere Ctr i c dok se nalazite u konzoli.

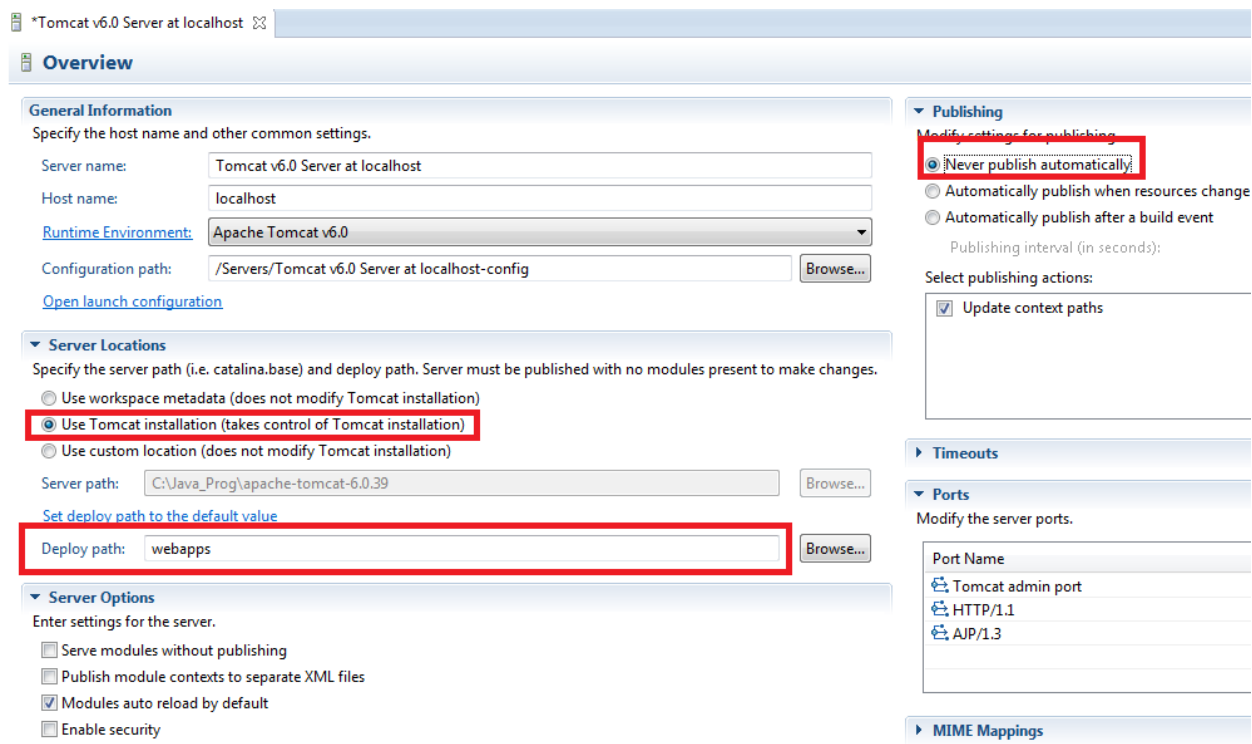
### **Pokretanje Apache Tomcat iz Eclipse okruženja**

Ukoliko je nekome zgodnije može pozivati Tomcat direktno iz Eclipse. Za to je neophdno da povežemo na specifičan način raspakovani Tomacat server sa našom Eclipse. U Eclipse treba da se odabere Window->ShowView->Server. U Donjem delu Eclipse (tamo gde je i konzola) otvoriće se pogled za upravljanje serverima. Nakon toga treba kliknuti na link sa slike 12 i dijaloga odabrati **Tomcat v6.0**.



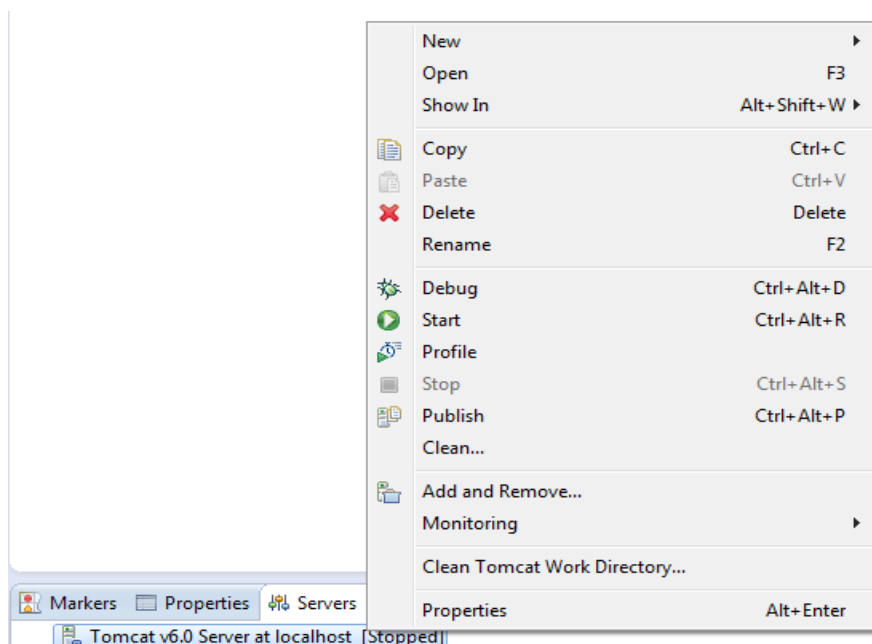
Slika 12 - Kreiranje Tomcat servera u Eclipsi

Nakon toga kliknuti na dugme Finish. Posle uradite dvoklik na novonastali server pa će se pojaviti dijalog kao na slici 13



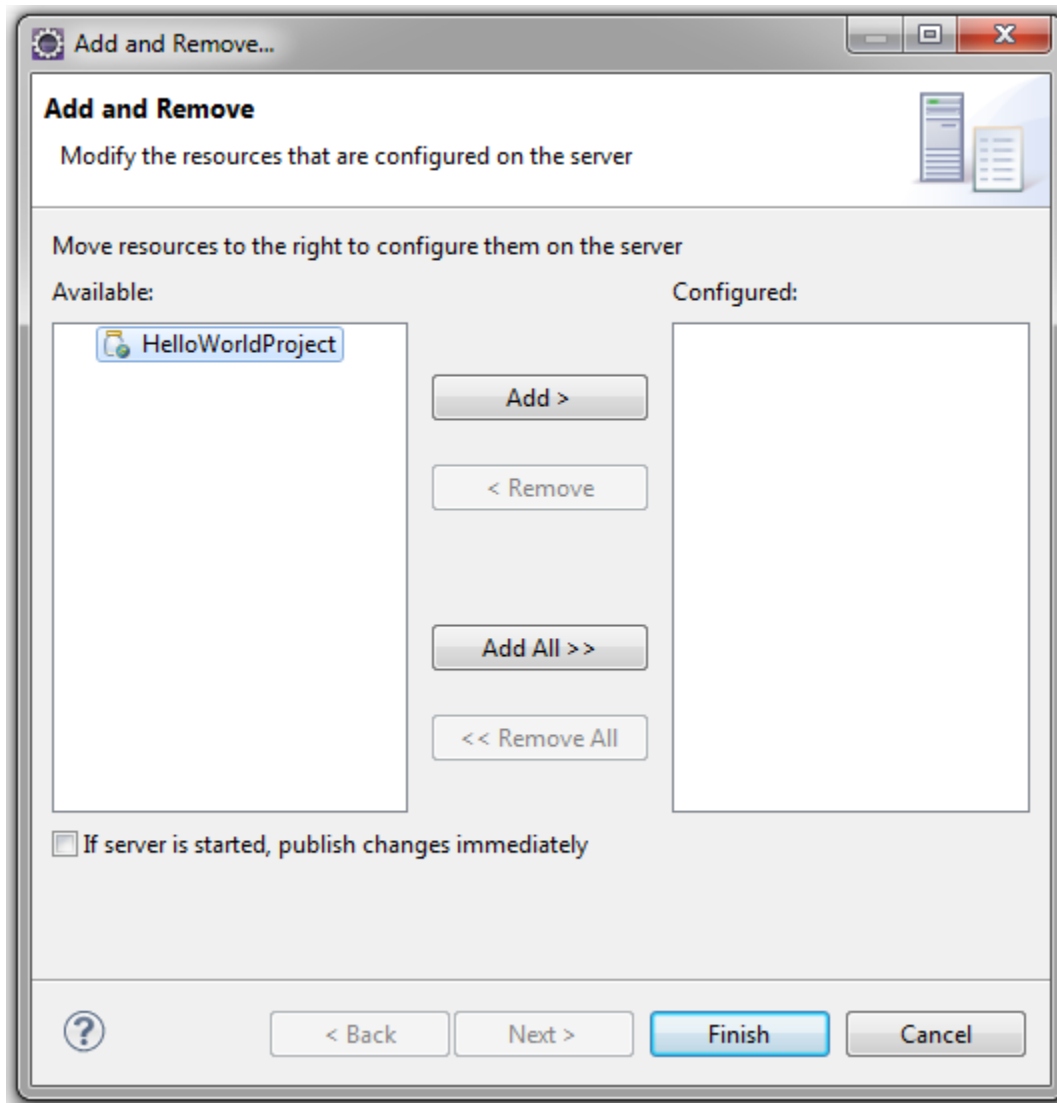
Slika 13 - Podešavanje Tomcat servera u Eclipsi

U Server Location sekciji postavite drugu opciju i kao deploy path upišite webapps, dok u sekciji Publishing odabariate prvu opciju kao na slici 14. Na kraju sa File->Save ili Ctrl+s snimite ove izmene. Kada završite sa podešavanjima kliknite desnim tasterom miša na server i dobićete dijalog kao na slici 14.



Slika 14 - Upravljanje Tomcat serverom u Eclipsi

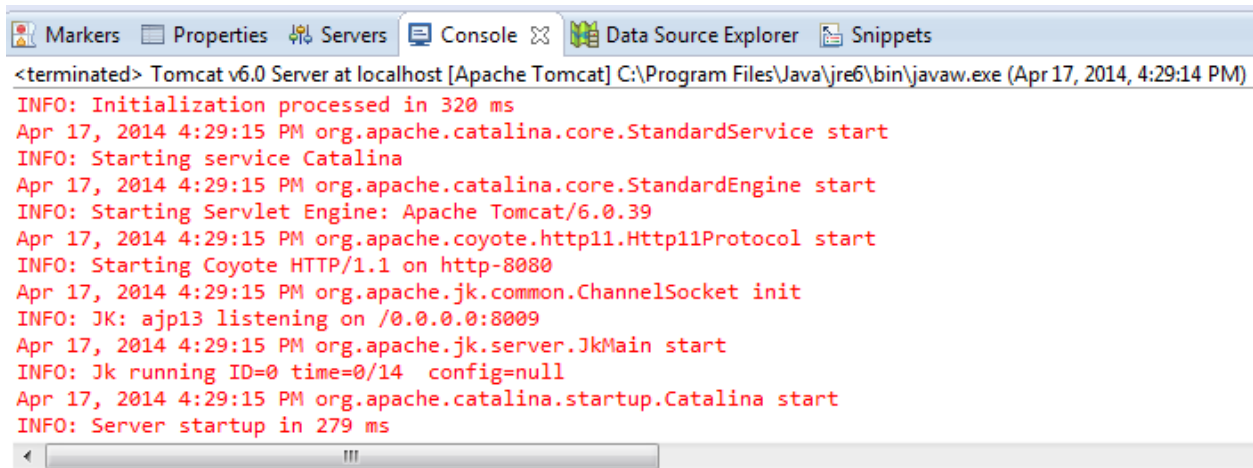
Neophodne opcije su vam Start i Stop. Međutim pre pokretanja servera neophodno je da **dodate aplikaciju na Server** klikom na Add and Remove. Nakon toga kao na slici 15 odabir željenu aplikaciju i dugmetom Add je dodajem na Server.



Slika 15 - Dodavanje aplikacije na Tomcat server u Eclipse

Kada završim sa dodavanjem neophodno je samo da pokrenemo server. Informacije o radu Servera možemo pratiti u konzoli (slika 16)





Slika 16 - Izgled Tomcat konzole u Eclipse

Redosled akcija koje trebate da radite kada koristite ugrađeni Tomcat su sledeće

1. Programirajte u Eclipsi dok je server zaustavljen (stop)
2. Samo prvi put je neophodno da dodate aplikaciju, nakon toga Tomcat će biti svestan promena u vašoj aplikaciji samo njegovim ponovnim startovanjem
3. Bilo bi dobro da pre svakog pokretanja (start) pozovete Clean opciju

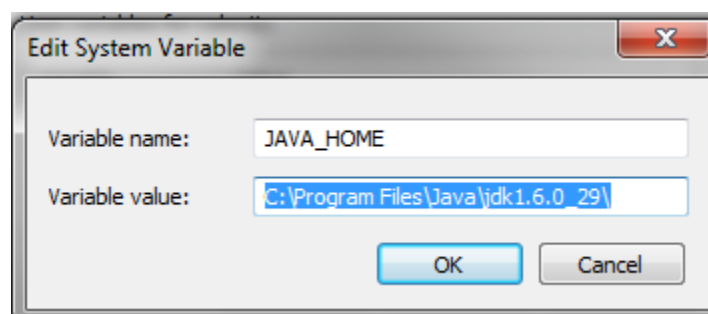
### Dodatno

**Napomena:** Da bi ste pokrenuli Tomcat neophodno je da imate podešenu sistemsku promenljivu JAVA\_HOME !!! Ona se podešava na nivou operativnog sistema tako je za:

Windows

Desno dugme na Computer->Properties->AdvanceSystemSetings->EnviromentVariable

U dijalogu treba da se napravi nova promenljiva pod imenom JAVA\_HOME čija vrednost mora biti putanja do instalacije Java (SDK)



Slika 3 -Sistemska varijabla

## Opšta preporuka prilikom razvoja web aplikacije

1. Dok razvijate aplikaciju bilo bi dobro da Tomcat bude ugašen.

2. Kada završite sa određenim delom i želite da to i proverite kako izgleda pod Tomcat uradite *Deploy To War* pa onda startujte Tomcat.
3. Kada završite sa proverom onoga što ste uradili ugasite Tomcat i vratite se u Eclipse kako bi ste nastavili razvoj svoje aplikacije