



Distance Learning System

# Java Web development

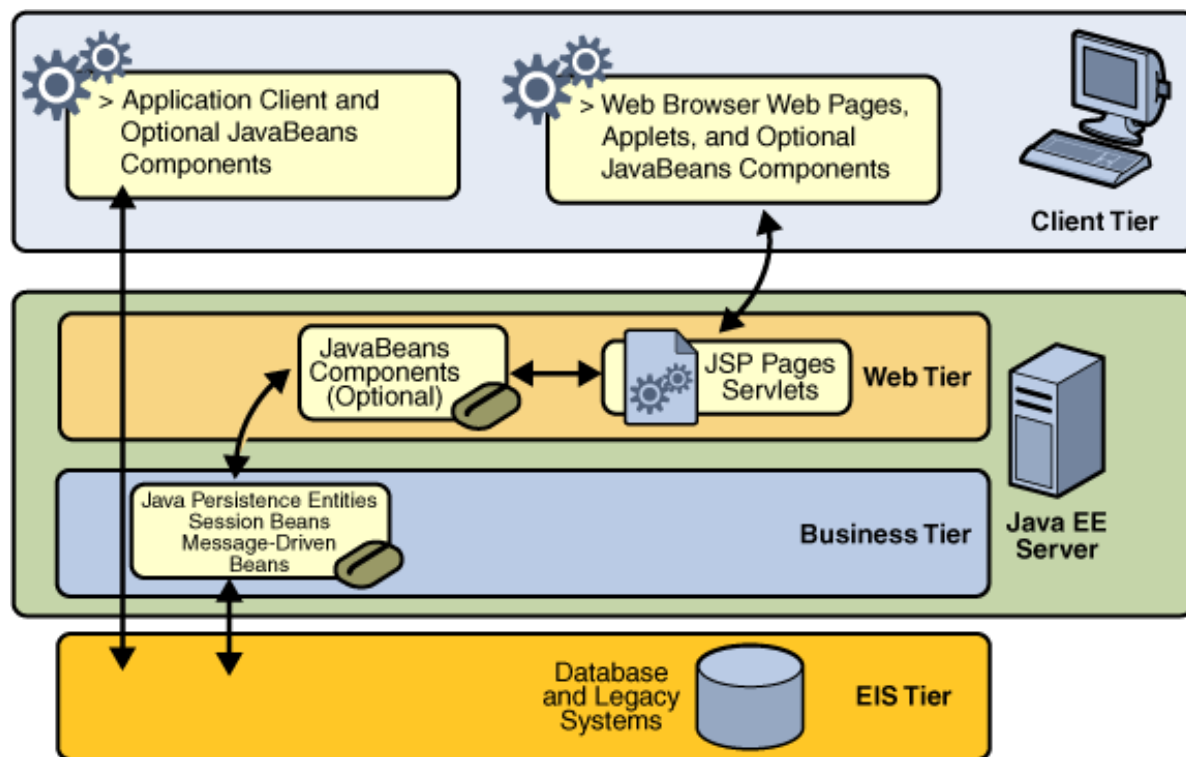
Uvod u Java Web

# Šta je Java EE

---

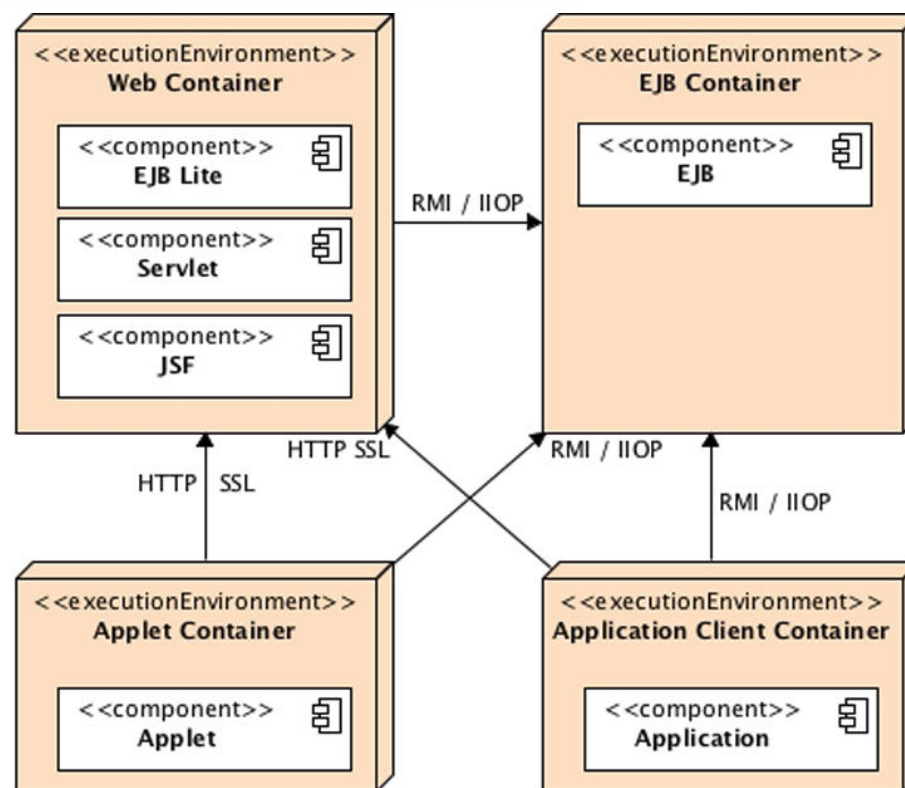
- Java EE (Enterprise Edition) je set biblioteka kojima se proširuje standardna Java funkcionalnost
- Java EE dodaje javi mogućnost korišćenja distribuiranih i nedistribuiranih transakcija (**JTA**), perzistentnih objekata (**JPA**), sistema poruka (**JMS**), web servisa (JAX-WS)
- Glavni fokus Java EE tehnologije su web i enterprise aplikacije

# Java EE arhitektura



# Java EE arhitektura

- Ultimativni proizvod u Java EE je enterprise aplikacija
- Enterprise aplikacija se sastoji od jednog ili više kontejnera
  - Kontejneri su logičke celine koje sadrže skup komponenti koji obezbeđuju funkcionalnost jednog aplikativnog sloja
- Kontejneri se mogu smatrati nezavisnim aplikacijama koje uzajamno mogu a ne moraju interagovati



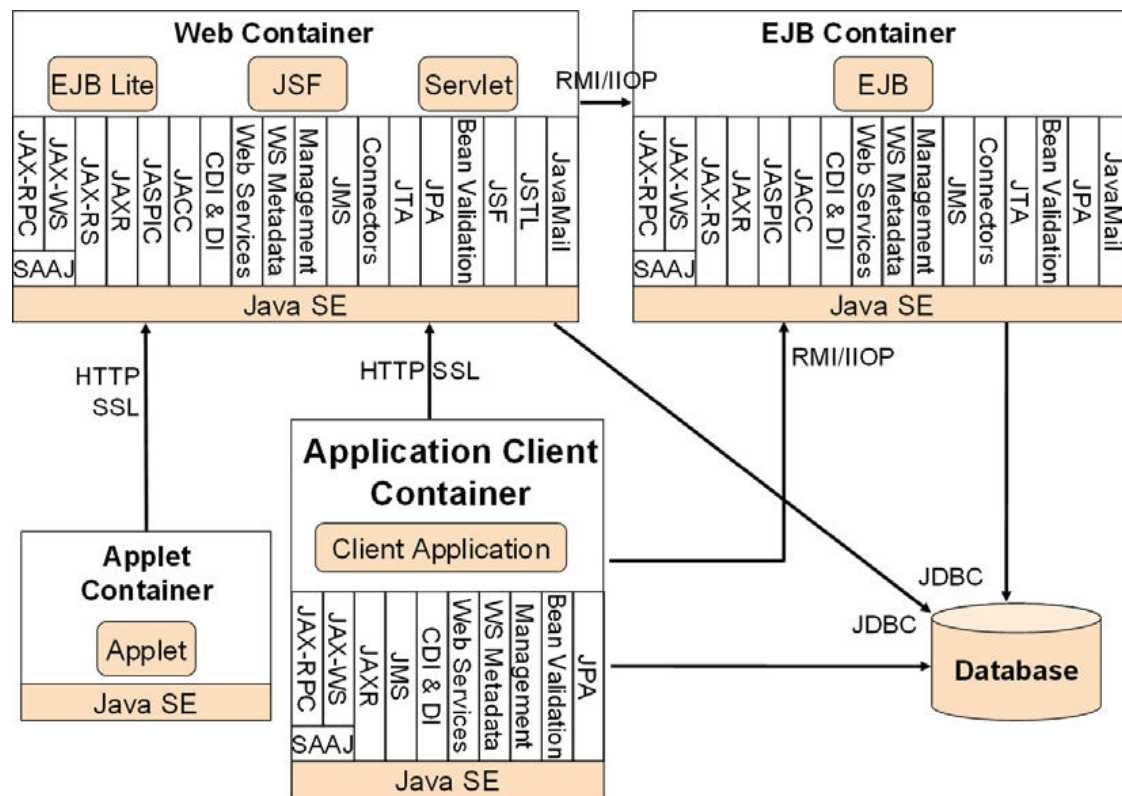
# Java EE kontejneri

---

- Java EE razlikuje četiri vrste kontejnera
  - Applet kontejner
    - Klijentski deo aplikacije (RIA)
  - Application client container (ACC)
    - Stand alone klijentske aplikacije koje imaju jaku vezu sa serverom
  - Web kontejner (standardne web aplikacije)
  - EJB kontejner
    - EJB aplikacije

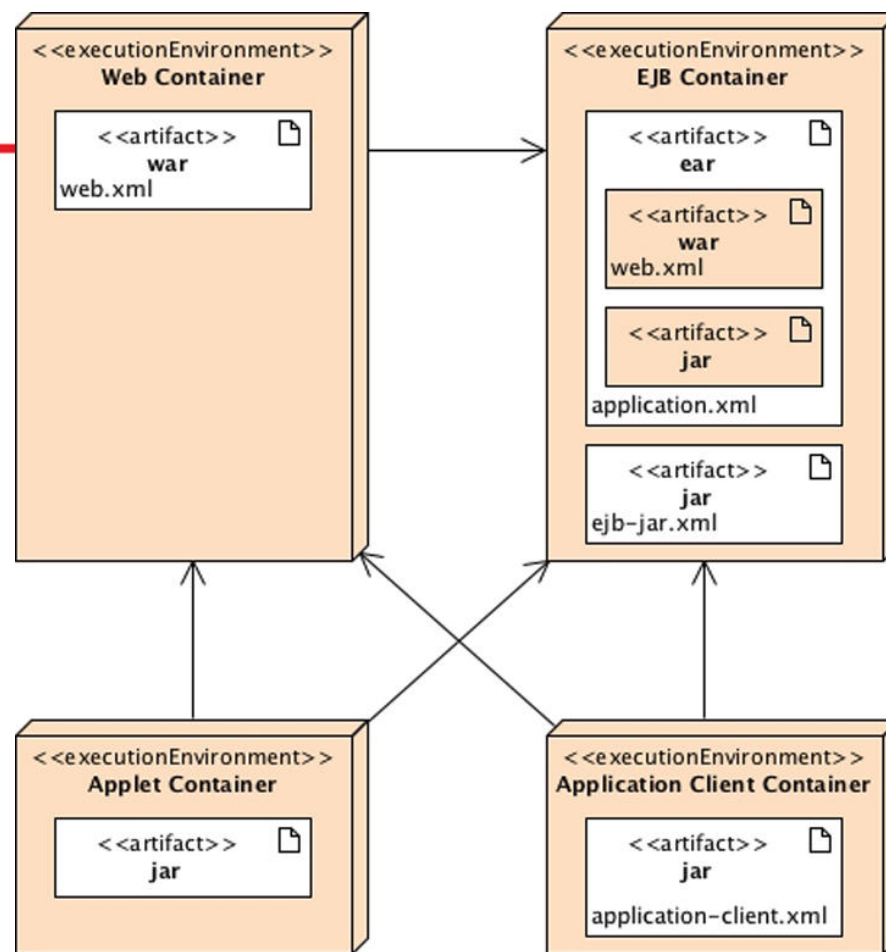
# Servisi

- U zavisnosti od mogućnosti kontejnera, na raspolaganju su neki od odgovarajućih servisa



# Java EE deploy

- Java EE aplikacije se pakuju na različite načine u zavisnosti od kontejnera na koji se vrši deploy
- Web aplikacije se pakuju u war pakete
  - EJB aplikacije u ear
- AAC i Applet aplikacije se pakuju u jar
  - Svaka vrsta pakovanja podrazumeva odgovarajući **deployment descriptor**



# Deployment descriptori

---

- Deployment descriptori su fajlovi koji sadrže konfiguracione direktive projekta
- Ovi descriptori mogu biti minimalni, ali takođe mogu sadržati kompletan deklarativni deo aplikacije



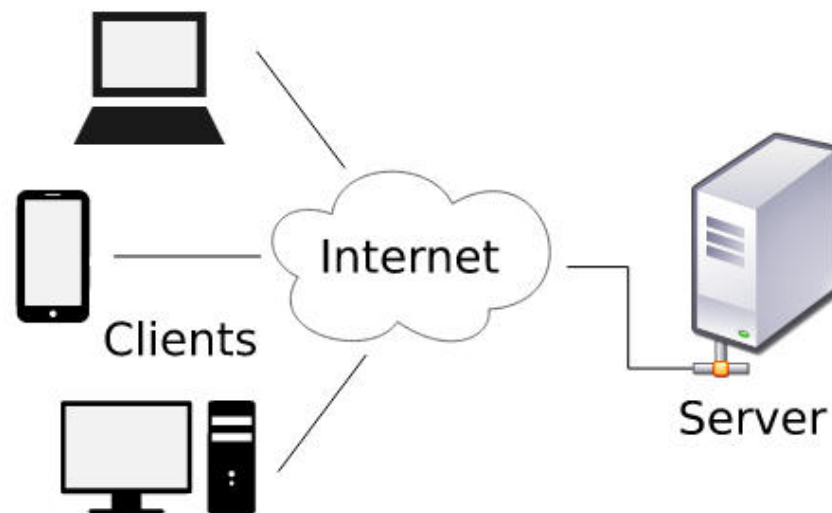
# Java EE deployment descriptor-i

---

File	Specification	Paths
application.xml	Java EE	META-INF
application-client.xml	Java	EE META-INF
beans.xml	CDI	META-INF or WEB-INF
ra.xml	JCA	META-INF
ejb-jar.xml	EJB	META-INF or WEB-INF
faces-config.xml	JSF	WEB-INF
persistence.xml	JPA	META-INF
validation.xml	Bean Validation	META-INF or WEB-INF
web.xml	Servlet	WEB-INF
web-fragment.xml	Servlet	WEB-INF
webservices.xml	SOAP Web Services	META-INF or WEB-INF

# Uvod u (Java) web programiranje

- Web aplikacije na Java platformi u mnogome funkcionišu kao i web aplikacije na ostalim programskim platformama, ali se njihova funkcionalnost, sa druge strane, drastično razlikuje u odnosu na java desktop aplikacije. Zbog toga je, pre kreiranja java web aplikacija, potrebno dobro se upoznati sa konceptom web programiranja uopšte, kao i implementacijom web programiranja na java platformi. Za razliku od desktop aplikacija, web aplikacije razlikuju dve celine u toku svog izvršavanja. Jednu celinu čini serverska logika, a drugu klijentska. Zato ćemo u opisu web aplikacije, često čuti pojmove **server** i **klijent**.



# Web server

---

- Server je program koji preuzima i obrađuje klijentske zahteve, a zatim, na osnovu parametara iz tih zahteva, kreira odgovarajuće odgovore. Svaka Internet aplikacija, tačnije, svaki sajt koji otvorimo, funkcioniše po ovom principu, ali se tehnologije, koje izvršavaju ove procese, (preuzimanje zahteva i slanje odgovora) razlikuju.
- Ova razlika nije konceptualna, jer svaka tehnologija poštuje isti princip preuzimanja i vraćanja odgovora, ali, obzirom da se same pozadinske tehnologije razlikuju, neophodno je i poznavanje svake od njih da bi se njome moglo adekvatno rukovati.
- Postoji nekoliko dominantnih web server aplikacija.
  - **Microsoft Internet Information Services** (IIS).
  - **Apache** web server
  - **Tomcat, JBoss, Geronimo, GlassFish...**

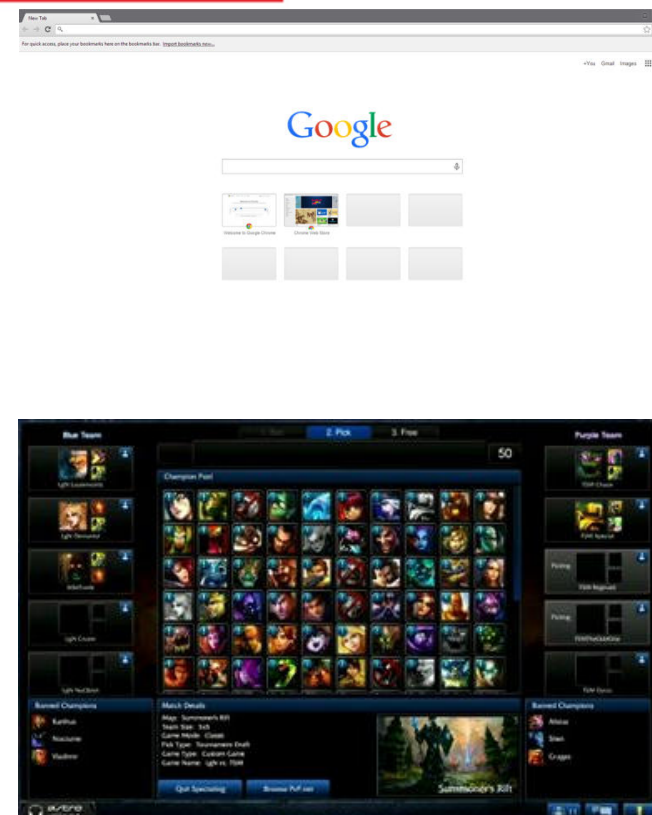
# Izvršavanje web aplikacija

---

- Često se web serveri nazivaju i application serveri, zato što su u stanju da startuju i određene aplikacije prilikom obrade klijentskog zahteva.
- Onog trenutka kada određena web prezentacija (sajt) u svom postojanju upotrebi neku od aplikacija (funkcionalnosti) na serveru, ona, zapravo, postaje web aplikacija. Programski kod (program) koji se izvršava na serveru prilikom aktivacije klijentskog zahteva, naziva se **serverski kod**.
- Razlikuje se više načina izvršavanja serverskog koda, odnosno, više mogućnosti njegove implementacije. Kada web server primi klijentski zahtev, čita njegove karakteristike (sadržane u zaglavlju zahteva) i na osnovu njih izvršava određenu akciju. Ukoliko zahtev ne sadrži zahtev za aktivaciju web aplikacije, već samo potražuje neki dokument sa servera, server će pronaći dokument na fajl sistemu, a zatim ga proslediti klijentu kroz odgovarajući odgovor. Ovakav scenario podrazumeva preuzimanje statičkih sadržaja sa web servera (html dokumenata, slika i sl.).
- Ukoliko zahtev zahteva angažovanje serverskog koda, procedura je drugačija. Server, takođe, pronalazi dokument, ali ga prosleđuje **java virtualnoj mašini**, koja ga izvršava i prosleđuje klijentu. Svi elementi web aplikacije, koji se u javi izvršavaju na ovaj način, posredstvom serverskog koda, nazivaju se **java web components**.

# Web klijent

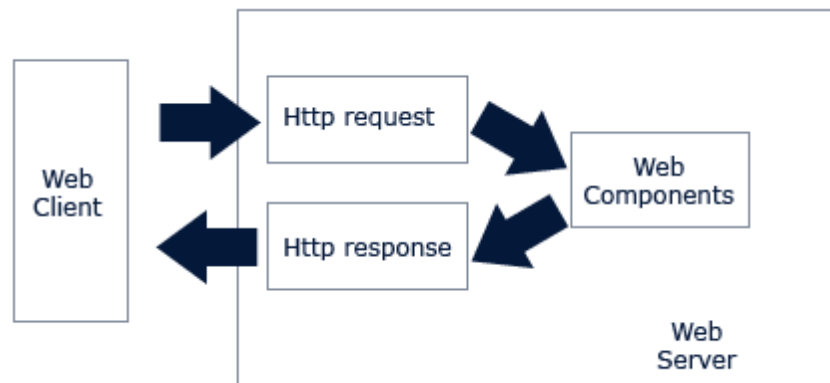
- Svaka web aplikacija ima dve strane u toku svog izvršavanja. Jedna strana je serverska, a druga klijentska. Klijentska strana je krajnja tačka izvršavanja web aplikacije i ona, obično, takođe, sadrži logiku koja je u stanju da pošalje zahtev serveru, kao i da preuzme i adekvatno pročita odgovor. Web pretraživač (mozilla, internet explorer, opera, safari...) je najčešći oblik klijentske web aplikacije. Ali konzument web aplikacije ne mora biti obavezno web pretraživač, već može biti bilo koja aplikacija, sve dok u svom izvršavanju podrazumeva i komunikaciju sa web serverom.



# Problemi u web programiranju

---

- Koncept klijent/server je izuzetno problematičan sa stanovišta
  - Bezbednosti
  - Brzine transporta informacija
- Postoje razne tehnike kojima se ovi problemi zaobilaze, ali je i dalje, baš zbog njih, kreiranje web aplikacija jedan od najvećih programerskih izazova danas



# Web programiranje u Javi

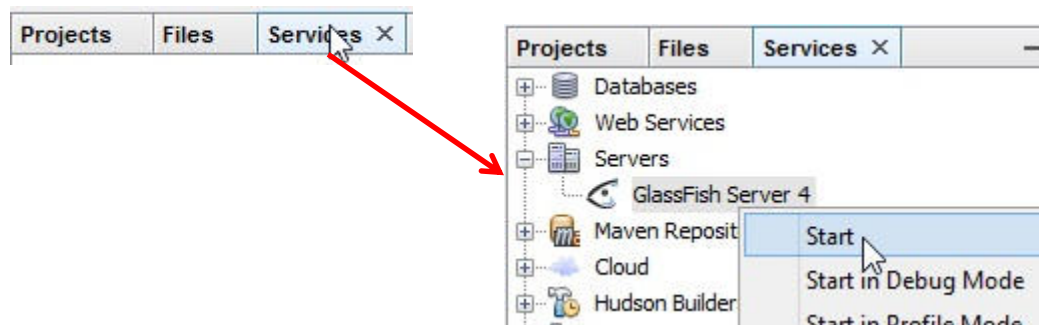
---

- Tehnologije za kreiranje web aplikacija u Javi nazivaju web components
- Web components su podrazumevani element Java EE edicije i podrazumevaju dve ključne stavke **Java Server Pages** (JSP) i **Java Servlets**.
- Java Server Pages je tehnologija koja podrazumeva implementaciju java koda u markup kod (html, xml i slično), omogućavajući da se inače nefunkcionalan HTML kod obogati nekom funkcionalnom logikom (kroz Javu).
- Servleti funkcionišu kao Java komponente iz kojih je neophodno eksplicitno emitovati HTML (ili neki drugi) kod, zbog čega se najčešće koriste kao pozadinska logika unutar web aplikacije, kao i sistemi za dobavljanje podataka u procesu zahteva i odgovora. Web komponenta egzistira u okruženju koje se naziva **web kontejner**
- Da bi bila izvršena u web kontejneru (na app serveru) Web komponenta (java web aplikacija) mora biti instalirana/pripremljena za server. Odnosno, mora biti izvršen njen **deploy**. Prilikom procesa pripreme (deploy) aplikacije, poziva se alat deploytool, koji gradi aplikaciju i generiše **war** (web application archive) fajl, odnosno, arhivu u kojoj se nalazi web aplikacija.
- War u sebi sadrži strukturu web aplikacije, uključujući i sve njene resurse (slike, html dokumente i slično).

# Kreiranje prve web aplikacije

---

- GlassFish aplikativni server je sastavni deo Java EE developers kit-a
- Ukoliko je NetBeans instaliran sa podrškom za Java web, zajedno sa njim je automatski instaliran i glassfish server
- Glassfish server se može pokrenuti putem NetBeans okruženja



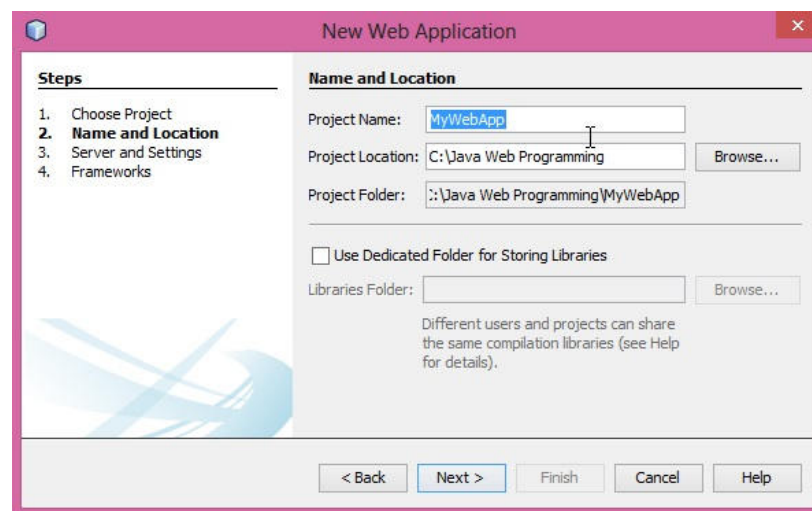
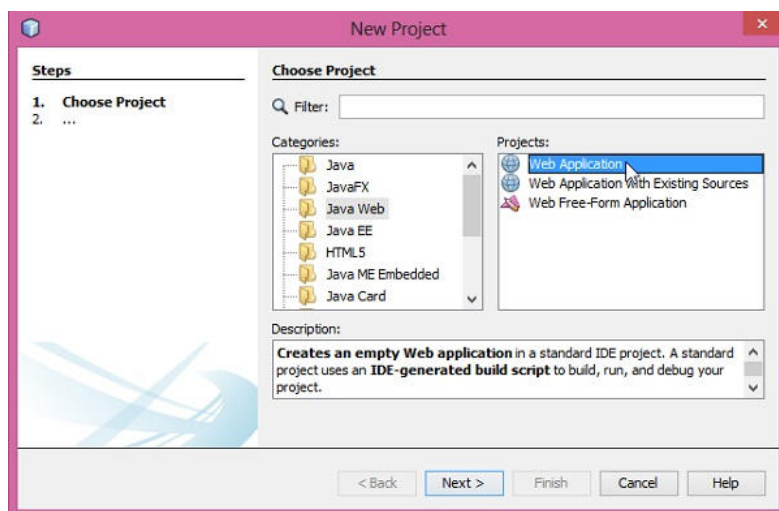
**Kreiranje web aplikacije pomoću NetBeans-a**

**LINKgroup**



# Kreiranje web projekta

- Da bi se kreirala jedna Java veb-aplikacija, potrebno je uraditi sledeće:
- **1. File -> New Project...**
- **2. za kategoriju odabrati Java Web, a za projekat Web Application**

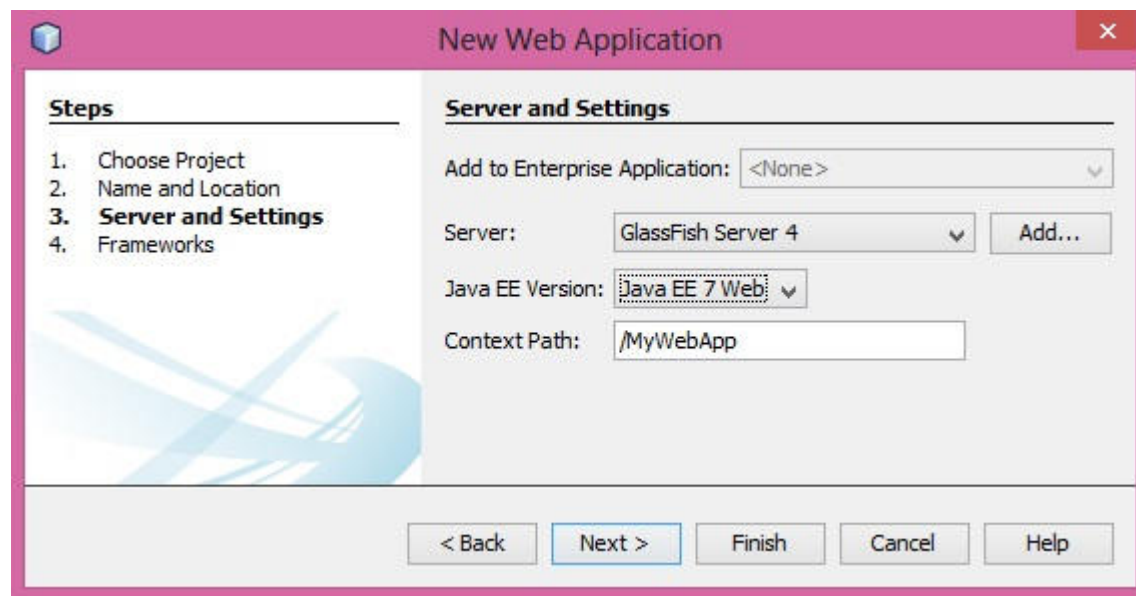


Kreiranje web aplikacije pomoću NetBeans-a

**LINKgroup**

# Kreiranje web projekta

- U dijalogu Server and Settings moguće je postaviti podešavanja koja se tiču servera; da server nismo pokrenuli pre kreiranja projekta, ovde bismo imali opciju da to uradimo: Context Path, ukazuje na deo putanje pod kojom će naša aplikacija biti dostupna, na primer iz nekog browsera;

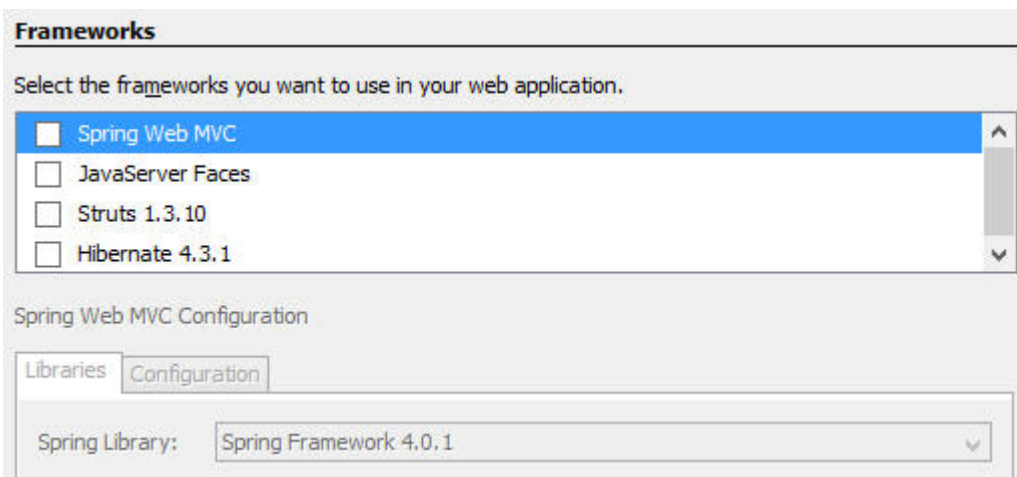


**Kreiranje web aplikacije pomoću NetBeans-a**

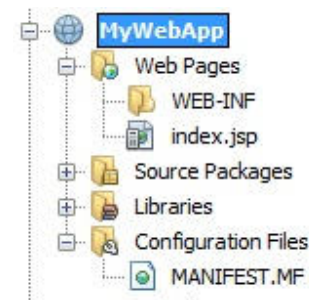
**LINKgroup**

# Kreiranje web projekta

- U dijalogu Frameworks moguće je odabrati neki od frameworka, koji se mogu koristiti u projektu



Pritiskom na taster FINISH, projekat biva uspostavljen.



**Kreiranje web aplikacije pomoću NetBeans-a**

**LINKgroup**

# Web (pages) direktorijum

---

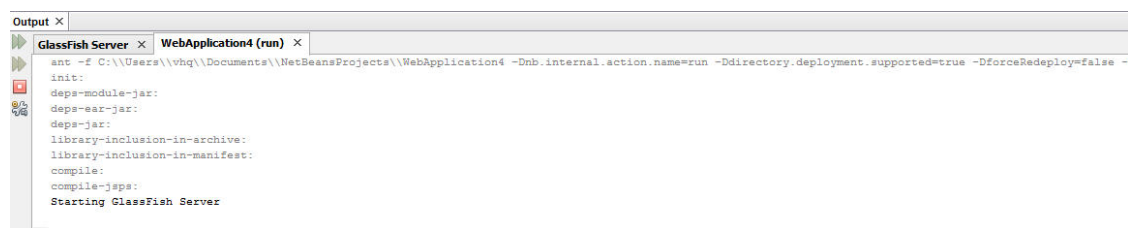
- Ovaj direktorijum sadrži javni sadržaj web aplikacije
- Sve ono što se nalazi u ovom direktorijumu biće izloženo, osim direktorijuma WEB-INF. U ovom direktorijumu se najčešće nalaze fajlovi koji ne treba da budu dostupni javnosti
- Prvi stanovnik ovog direktorijuma često je fajl **web.xml** – takozvani **web deployment descriptor**
- U ovom fajlu se nalaze informacije koje objašnjavaju serveru na koji način će tretirati određene delove aplikacije

# Startovanje aplikacije

- Startovanje web aplikacije iz NetBeans-a se vrši kao i startovanje bilo koje druge aplikacije – aktivacijom tastera F6 ili zelene strelice



- Ipak, koraci koje u trenutku startovanja NetBeans (ant) mora da izvrši kako bi startovanje bilo uspešno, razlikuju se od startovanja obične java aplikacije, jer se pre samog startovanja mora aktivirati web / aplikativni server



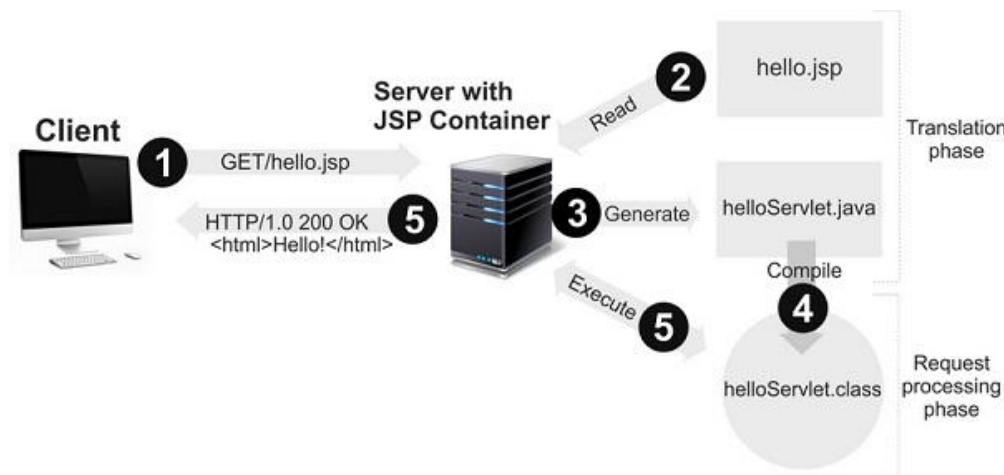
# Ubacivanje korisnički definisanog sadržaja na stranu

- Osnova svrha korišćenja jave za veb-programiranje jeste mogućnost izvršenja različite logike na serveru.
- Ovo se postiže ili ugradnjom Java koda u html ili kreiranjem zasebnih fajlova sa isključivo Java kodom. U primeru nemamo nikakav Java kod integrisan u html. Radi boljeg razumevanja, kao demonstraciju ćemo promeniti liniju sa h1 tagom. Umesto postojeće linije postavite sledeću:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1><% out.println("Hey, I am created dynamically!"); %></h1>
  </body>
</html>
```

# Ubacivanje korisnički definisanog sadržaja na stranu

- Kada zatražimo neku JSP stranicu putem pretraživača, aplikacioni server proveri da li ta strana postoji u prevedenom obliku. Ukoliko ne postoji, izvršava se prevođenje strane i emitovanje korisniku; u suprotnom, odmah se preuzima prevedena verzija.
- U trenutku prevođenja JSP strane, prevodilac kompletan HTML, XML ili neki drugi ne-Java kod prosleđuje u njegovom izvornom obliku, da bi, u trenutku kada naiđe na tag sa oznakom %, izvršio prevođenje sadržaja taga, a zatim nastavio sa prosleđivanjem ne-Java koda. Nakon prevođenja, JSP strana dobija formu servleta i više se ne prevodi sve do promene sadržaja dokumenta



**Uvođenje dinamičkog sadržaja na stranu**

**LINKgroup**

# Vežba (jwex012014 ShowTime)

---

- Potrebno je napraviti jednostavnu veb-aplikaciju koja prikazuje trenutno vreme na stranici.

