

# Serverske veb tehnologije - komunikacija distribuiranih objekata -

Dragan Ivanović

Katedra za informatiku, Fakultet Tehničkih Nauka, Novi Sad

2022.

# Veb servisi

- Komunikacija putem SOAP protokola ugrađena u HTTP
- XML - podaci
- wsdl - opis servisa

# REST

- REST je skup principa koji definišu kako da se koriste standardi (HTTP, URI, ...)
- Pretpostavka je da će aplikacija koju razvijamo dobro koristiti arhitekturu veba ako se pridržavamo ovih principa
- RESTful je implementacija servisa poštujući REST principe
- REST je arhitektura, RESTful je pridev koji ide uz reč servis

# REST principi

- Svaki resurs treba da ima svoj ID
- Resursi treba da budu povezani
- Treba koristiti standardne HTTP metode
- Resursi treba da imaju višestruke reprezentacije
- Komunikacija treba da bude stateless

# JSON

- Laki (lightweight) format za razmenu podataka
- Jednostavan
  - Za ljude koji ga pišu
  - Za mašine koje ga procesiraju
- JSON je tekstualni format koji je nezavistan od programskog jezika

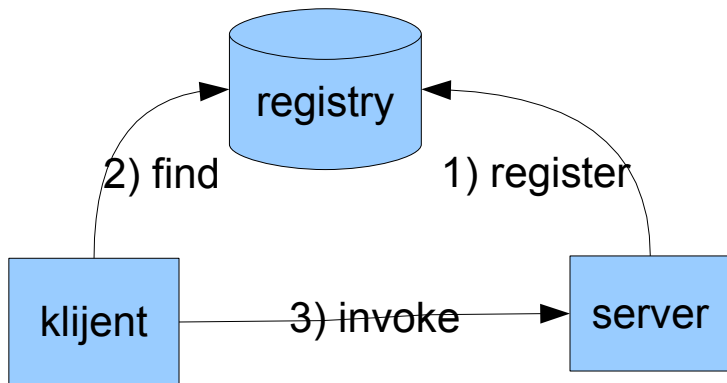
# REST, JSON i Spring

- Kasnije o ovome!!!

# Remote Method Invocation (RMI)

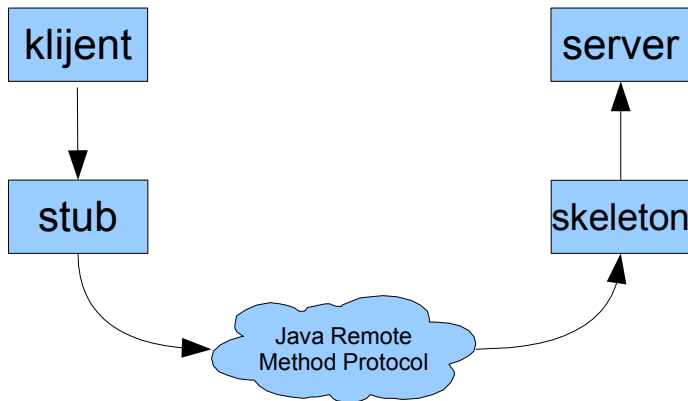
- Osnovna tehnologija za rad sa binarnu komunikaciju distribuiranih objekata u Javi
- Objekti na udaljenim JVM su dostupni preko mreže
- Pozivamo njihove metode na isti način kao i za lokalne objekte
- Objekti se registruju u katalozima (registry)
- Klijenti ih tamo pronalaze

# RMI registry, server i klijent





# RMI klijent i server komuniciraju preko posrednika



# Primer elementarnog RMI klijenta i servera

- Primer 1

# RMI i prenos programskog koda

- Prilikom poziva metode RMI objekta...
- ...kao parametar možemo proslediti instancu klase koja nasleđuje tip parametra
- Tada će JVM preneti i programski kod potrebne klase!
- Primer 2

# RMI i Spring

- Moguće je i Spring Bean-ove učiniti dostupnim putem RMI
- <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/remoting.html>

# JNDI

- Java Naming and Directory Interface
- Standardni API za pristup različitim servisima imena
- I različitim direktorijumskim servisima
- Servis imena: mapira ime (string) ↔ objekat
  - Fajl-sistem
  - RMI registry
- Direktorijumski servis: objekti se dodatno opisuju pomoću atributa
  - DNS
  - LDAP / Active Directory / NDS...
- Pristup različitim servisima obavlja se kroz isti API ali preko različitog "provajdera" (tj. drajvera)
  - analogno sa JDBC
- *JndiObjectFactoryBean* za upotrebu JNDI u Spring aplikacijama

# JNDI

- Primer 3

# CORBA

- Tehnologija distribuiranih objekata koja je platform- i language-neutral
- Interfejs serverskog objekta opisuje se IDL jezikom
- Na osnovu IDL fajla generiše se kod u odredišnom programskom jeziku (C++, Java, COBOL, ...)
- Dalja implementacija u programskom jeziku
- Isti osnovni koncepti kao i RMI, različita terminologija
  - stub  $\leftrightarrow$  stub
  - skeleton  $\leftrightarrow$  tie
  - JRMP  $\leftrightarrow$  IIOP
  - RMI registry  $\leftrightarrow$  Naming Service
- Primer 4

# RMI vs CORBA

- RMI: jednostavniji, ali vezan za Javu
- CORBA: složenija, ali nezavisna od jezika
- RMI-CORBA interoperabilnost?
- RMI-over-IIOP
  - pišu se RMI klijent i server
  - ali se u komunikaciji koristi IIOP umesto JRMP
- Neznatne izmene u kodu
- Primer 5



# RMI + JNDI

- Umesto Naming.lookup() možemo da koristimo JNDI API za pronalaženje RMI objekta
- Primer 6