

# JavaScript i Ajax

## 1. Sta je JavaScript?

-jezik na strani klijenta koji omogućava autorima web stranica da razvijaju interaktivne web stranice i sajtove.

-**Skriptovanje na strani klijenta** se odnosi na programski jezik koji se izvršava u okviru web pretraživača umesto na web serveru.

-**Jezik za skriptovanje** se odnosi na bilo koju vrstu jezika koji je u stanju da programski kontroliše web stranice.

-Deo pretraživača koji izvršava kod script jezika naziva se script mašina. To je samo jedna vrsta prevodioca. **Prevodilac** se odnosi na bilo koji program koji izvršava kod script jezika. Web pretraživač koji sadrži script mašinu zove se **script domaćin**.

-Java programski jezik ne može da se koristi van web pretraživača iz bezbednosnih razloga.

## 2. JavaScript aplikacije

-Java script omogućava web programerima da obavljaju zadatke kao što su:

- \*čitanje tagova iz dokumenata i pisanje teksta u dokumentima,
- \*premeštanje teksta,
- \*obavljanje matematičkih proračuna nad podacima,
- \*reagovanje na događaje,
- \*primanje tekućih podataka i vremena od korisnikovog računara ili kada je poslednji put document bio modifikovan,
- \*određivanje veličine ekrana korisnika, verzije pretraživača, ili rezolucije ekrana
- \*preduzimanje radnje u skladu sa uslovima kao što su upozoravanje korisnika ako unese pogresnu informaciju u formu.

## 3. JavaScript alati

- \*Adobe Dreamweaver,
- \*MS Expression Web,
- \*NetBeans,
- \*Xcode,
- \*MS Visual Studio,
- \*Aptana,
- \*Komodo,
- \*Notepad++...

## 4. Web čitači i JavaScript alati

- \*Internet Explorer-developer tools,
- \*Firebug-Firefox,
- \*Firebug Lite-IE6+, Firefox, Opera, Safari, Chrome.

## 5. Da li bi trebalo koristiti Client-Side ili Server-Side Scripting?

-Opšte pravilo je da se omogući klijentu da rukuje obradom korisničkog interfejsa i obradom osvetljenosti, kao što je validacija podataka, a da Web server obavlja intenzivna računanja i skladištenje podataka.

-Na primer, ako želite da kontrolisete Web pretraživač, morate koristiti JavaScript. Ukoliko želite da pristupite bazi podataka na Web serveru morate da koristite serverski script. Postoje i zadaci koje oba jezika mogu postići, kao što su validacija formi i upravljanje kolacicima.

## 6. Kako se dodaje skript na web stranicu?

-JavaScript može biti ugrađen u stranicu ili stavljen u eksternu script datoteku.

-Da bi JavaScript funkcionisao, JavaScript podrška mora da bude omogućena na samom pretraživaču.

-Skriptovi se dodaju na web stranicu unutar `<script>` elemenata.

-Tip atributa na početku taga `<script>` ukazuje koji skript programski jezik se nalazi unutar elementa, tako da za JavaScript koristite vrednost `text/javascript`.

```
<html>
<body>
  <p>
    <script type="text/javascript">
      document.write("My first JavaScript")
    </script>
  </p>
</body>
</html>
```

-Postoje tri mesta gde možete staviti vaše Java skriptove, a jedan XHTML dokument može da koristi sva tri:

- \*U okviru **<head>** tag-a: pozivaju se kada ih aktivira događaj
- \*U **<body>** tag-u: rade kada se stranica učitava
- \*U **eksternoj datoteci**.

## 7. Eksterna JavaScript datoteka

-JavaScript kod se može smestiti u eksternom dokumentu koji ima ekstenziju **.js**. Ovo je dobra opcija jer:

- \*ako script koristi više stranica ne mora se ponoviti script u svakoj stranici koja ga koristi
- \*ako je želite ažurirati script, potrebno je promeniti ga samo ne jednom mestu
- \*to čini XHTML stranicu čistijom i lakšom za čitanje.

-Kada postavite JavaScript u eksternu datoteku, potrebno je da koristite atribut `src` u `<script>` elementu.

```
<script type="JavaScript" src="scripts/validation.js"></script>
```

## 8. Koriscenje <script> elementa

```
<script type="text/javascript">
document.write("<p>Can you "); document.write ("hear me
now?</p>");
</script>

<script type="text/javascript">
document.write("<p>Can you ")
document.write("hear me now?</p>")
</script>
```

## 9. <noscript> element

nudi alternativni sadržaj za korisnike kod čijih web pretraživača nije omogućen JavaScript.

```
<html>
<body>
  <script src="external.js" type="text/JavaScript">
  </script>
  <noscript>This only shows if the browser has JavaScript turned off.
  </noscript>
</body>
</html>
```

## 10.Document Object Model (DOM)

-Interfejs koji radi sa Web stranicom se zove Document Object Model-DOM.

Ovaj model opisuje kako program može da predstavlja entitete iz realnog života korišćenjem skupa objekata, a takodje specificira skup metoda, svojstava i događaja koje jedan objekat može da ima.

-U ovom modelu stranica je predstavljena korišćenjem objekta – *document*; veze na stranici su predstavljene korišćenjem *link* objekta; forme su predstavljene *form* objektima; slike su predstavljene *image* objektima itd.

-DOM opisuje kako možete da:

- \*preuzmete(*get*) i podesite(*set*) svojstva Web stranice kao sto je na primer boja pozadine;
- \*pozivate metode(*methods*) koje vrse radnje kao sto su ispisivanje nove linije na stranici;
- \* reagujete na događaje(*events*) kao sto je korisnikov pritisak na dugme *Submit* na obrascu.

## 11.Pristupanje vrednostima koriscenjem Dot (.) notacije

-Da bi pristupili različitim objektima u DOM-u, izlistavate objekte u vidu strukture stable, počevši sa objektom dokumenta pa sve do objekta koji vas zanima.

-Svaki objekat je odvojen sa tačkom(dot), zato je ovo i poznato kao dot notacija.

```
document.links[0].href
```

## 12.Svojstva Document objekta

alinkColor, bgColor, fgcolor, lastModified, linkColor, referrer, title, vlinkColor

## 13.Metodi Document objekta

write(string), writeln(string)

```
document.write('Page last modified on ' + document.lastModified);
```

## 14.Svojstva i metodi Form objekata

-SVOJSTVA: action, length, method, name, target,

-METODE: reset (), submit ().

```
document.forms[0].action
```

```
document.frmLogin.action
```

## 15.Svojstva i metodi Form elemenata

-SVOJSTVA: checked, disabled, form, length, name, selectedIndex, type, value,

-METODE: blur(), click(), focus(), select().

Primer:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
  <title>Accessing form data</title>
</head>
<body>
  <h1>Accessing Form Data</h1>
</body>
</html>
```



```
<body>
<h1>Accessing Form Data</h1>
<form name="frmLogin">
  Username <input type="text" name="txtUsername" size="12" /><br /><br />
  <input type="submit" value="Click here" />
</form>
</body>
```

```
<form name="frmLogin" onsubmit="alert(document.frmLogin.txtUsername.value)">
  Username <input type="text" name="txtUsername" size="12" /> <br />
  <input type="submit" value="Click here" />
</form>
```

## 16.Images objekti

border, complete, height, hspace, lowsrc, name, src, vspace, width

```
document.images[0].src
```

```
document.imgHome.src
```

## 17.Dodavanje komentara u JavaScript kodu

```
<script type="text/javascript">
/*
This line is part of the block comment.
This line is also part of the block comment.
*/
document.writeln("<h1>Comments Example</h1>"); // Line
comments can follow code statements
// This line comment takes up an entire line.
/* This is another way of creating a block comment. */
</script>
```

## 18.Osnove JavaScript-a

- \*Promenljiva(*variable*) koristi se za skladištenje informacija, predstavlja delić memorije u kome možete pamtiti brojevi, stringovi ili reference na objekte
- \*Operatori(*operators*) izvršavaju funkcije nad promenljivim
- \*Funkcije(*function*) delovi skripta koji su grupisani zajedno u cilju realizacije određenog zadatka
- \*Uslovne naredbe(*conditional statements*) omogućavaju izvršavanje različitih akcija koje se baziraju na uslovu
- \*Petlje(*loops*) omogućavaju da se blok koda izvršava određeni broj puta ili dok je uslov zadovoljen.

## 19.Promenljive(*variable*)

- \*ime koje se dodeljuje promenljivoj se naziva identifikator
- \* identifikator mora početi slovom ASCII kodnog rasporeda, znakom \$ ili donjom crtom)
- \*mogu se koristiti brojevi u identifikatoru ali ne kao prvi karakter
- \*nema praznina
- \*nema rezervisanih reči.
- Promenljive(varijable) imaju sledeće karakteristike:

- \***var** naredba,
- \*naziv,
- \*početna vrednost,
- \*tip podataka (**Integer**-celi brojevi pozitivni I negativni, **Float**-brojevi sa pokretnim zarezom, decimalni pozitivni I negativni brojevi, **Boolean**-tacno/netacno, **Text**-odnosi se na string podatke, **nizovi I objekti**-slozeni tipovi podataka)

## 20. Operatori

-Sam operator je ključna rec ili simbol koji radi nešto sa vrednostima kada se koriste u izrazu.

-Vrste operatora:

- \*aritmetički operatori,
- \*operatori dodeljivanja,
- \*operatori poređenja,
- \*logički operatori,
- \*string operatori

## 21. JavaScript I objekti

-JavaScript podržava OOP.

-JS objekti imaju sledeće karakteristike:

- \*svojstva,
- \*metode,
- \*događaje.

-Osnova svih objekata jeste *Object* klasa.

-Primeri objekata: String, Math, Date ...

-Postoje dve vrste JS metoda:

- \*statički - može da se koristi u bilo kom trenutku bez kreiranja instance;
- \*instance - rade sa podacima koji se nalaze u instanci klase.

## 22. Funkcije

Funkcija se sastoji od koda koji obavlja određeni zadatak.

```
function name_of_function(parameters) {  
    statements;  
}
```

```
function calculateArea(width, height) {  
    area = width * height  
    return area  
}
```

```
<form name="frmArea" action="">  
Enter the width and height of your rectangle to calculate the size:<br />  
Width: <input type="text" name="txtWidth" size="5" /><br />  
Height: <input type="text" name="txtHeight" size="5" /><br />  
<input type="button" value="Calculate area"  
onclick="alert(calculateArea(document.frmArea.txtWidth.value,  
document.frmArea.txtHeight.value))" />  
</form>
```

## 23. Uslovne naredbe

-*if* naredbe - koriste se kada zelimo da se script izvrši ako je uslov tačan

-*if...else* naredbe – koriste se kada zelite da izvršite jedan set koda ako je uslov tacan a drugi ako je netacan

-*switch* naredbe – koriste se kada se bira jedan blok koda od mnogih u zavisnosti od situacije.

```
<script type="text/JavaScript">
  date = new Date();
  time = date.getHours();
  if (time < 12) {
    document.write('Good Morning');
  }
</script>
```

```
<script type="text/JavaScript">
  date = new Date();
  time = date.getHours();
  if (time < 12) {
    document.write('Good Morning');
  }
  else {
    document.write('Good Afternoon');
  }
</script>
```

```
<p>Enter the name of your favorite type of animal that stars in a
cartoon:</p>
<form name="frmAnimal">
  <input type="text" name="txtAnimal" /><br />
  <input type="button" value="Check animal" onclick="checkAnimal()" />
</form>
```

```
function checkAnimal() {
  switch (document.frmAnimal.txtAnimal.value){
    case "rabbit":
      alert("Watch out, it's Elmer Fudd!")
      break;
    case "coyote":
      alert("No match for the road runner - meep meep!")
      break;
    case "mouse":
      alert("Watch out Jerry, here comes Tom!")
      break;
    default : alert("Are you sure you picked an animal from a cartoon?");
  }
}
```

## 24. Petlje

```
<script type="text/JavaScript">
i = 1
while (i < 11) {
  document.write(i + " x 3 = " + (i * 3) + "<br />" );
  i ++
}
</script>
```

```
<script type="text/JavaScript">
i = 12
do {
  document.write(i + " x 3 = " + (i * 3) + "<br />" );
  i ++
}
while (i < 11)
</script>
```

```
for (i=0; i<11; i++) {
  document.write(i + " x 3 = " + (i * 3) + "<br />" );
}
```

## 25. Ugrađeni objekti

- \*string,
- \*date,
- \*math,

\*array,  
\*window

```
MyString = new String("How many characters are in this sentence?")  
alert(myString.length)  
  
var birthDate = new Date("April 16, 1975")  
document.write(birthDate)  
  
fruit = new Array("apple", "banana", "orange", "mango", "lemon")  
document.write(fruit.length)  
  
<script type="text/javascript">  
  function statusBarText()  
  {  
    window.status = "Did you see me down here?"  
  }  
</script>  
  
<body onload="statusBarText()">
```

## 26. Validacija forme

-Postoji nekoliko nacina na koje mozete da izvrсите validaciju formulara. Obicno, kada korisnik pritisne dugme za slanje podataka(*submit*), koje aktivira *onsubmit* upravljac događajima što zauzvrat poziva funkciju za validaciju smeštenu ili u posebnom skriptu ili u zaglavlju dokumenta.

-Funkcija mora da vrati *true* kako bi se izvršilo slanje podataka sa forme, a *false* u slučaju greške- forma korisnika nije poslata.

## 27. JS I rad sa prozorima

Window objekat sadrži druge objekte koji se mogu koristiti za rad sa web brauzerom:

- \*Screen,
- \*History,
- \* Navigator,
- \*Location.

# AJAX

## 28. RIA

-Rich Internet Applications (RIA) su web aplikacije koje pružaju izgled, osećaj I upotrebljivost desktop aplikacija. RIA karakteristike: performanse, bogat grafički korisnički interfejs (GUI).

- Performanse RIA se zasnivaju na Ajax tehnologiji, koja koristi scripting na strani klijenta u cilju boljeg odziva aplikacija.

## 29. Ajax

-*Asynchronous JavaScript and XML* se odnosi na kombinaciju tehnologija koje omogućavaju Web stranicama prikazanim na klijentskom računaru da brzo komuniciraju I razmenjuju podatke sa Web serverom bez ponovnog učitavanja cele Web stranice.

-Prvenstveno se oslanja na JavaScript I HTTP zahteve za razmenu podataka između računara I klijenta Web servera.

-Druge tehnologije koje sačinjavaju Ajax su XHTML, CSS, Document Object Model ali se odnose na upravljanje prikazom I prezentacijom podataka unutar Web pretraživača, dok su HTTP I XML odgovorni za razmenu podataka. JavaScript spaja sve zajedno.

-**XMLHttpRequest object** koristi HTTP za razmenu podataka između računara klijenta I Web servera.

### 30. Načini za implementaciju Ajax funkcionalnosti

Ajax funkcionalnost implementira samostalno od strane programera koristi JavaScript za slanje zahteva asinhrono na server I zatim azurira stranicu pomocu DOM-a.

### 31. Pregled kreiranja AJAX Script-a

-Da bi se kreirao Ajax skript potrebno je izvršiti sledeće korake:

- \*kreirati instance XMLHttpRequest objekta za web pretraživač gde će se script pokrenuti,
- \*koristiti XMLHttpRequest za slanje zahteva serveru,
- \*pročitati I obraditi vraćene podatke sa servera.

### 32. JavaScript biblioteke

JS datoteke koje sadrže funkcije koje želite da koristite u više stranica se nazivaju *JavaScript biblioteke*:

- \*jQuery,
- \*Scriptaculous,
- \*MochiKit,
- \*Yahoo User Interface (YUI),
- \*Dojo,
- \*Spry,
- \*MooTools,
- \*Ext JS.

### 33. Karakteristike JQuery biblioteke

- \*moćna biblioteka,
- \*mala veličina datoteke,
- \*podržava fleksibilni mehanizam selekcije,
- \*odlična podrška za animacije,
- \*lakši rad sa AJAX-om,
- \*podrška za različite web pretraživače,
- \*sadrži podatke za rad sa korisničkim interfejsom,
- \*lakši rad sa CSS-om,
- \*podrška za mobilne uređaje,
- \*proširivost,
- \*besplatan I otvorenog koda,
- \*iskoristivost znanja...

### 34. JQuery UI ToolKit

-U ovaj okvir sadrži neke interesantne funkcionalnosti koje uključuju nove vizuelne elemente (*widgets*), skup ikonica I mehanizam za lako generisanje atraktivnih CSS stilova.

-Preuzimanje I konfigurisanje biblioteke:

- \*odabrati temu,
- \*preuzeti temu,
- \*odabrati elemente,
- \*preuzeti datoteku,
- \*instalirati je,
- \*povežite se sa CSS I JS datotekama.



-**JQuery biblioteka pruža dosta AJAX funkcionalnosti** koje su podeljene u četiri segmenta:

- \*Globalni AJAX upravljači događajima,
- \*Helper funkcije,
- \*Interfejs na nižem nivou,
- \*Skraćeni metodi.

---

## XML

### 1.Sta je XML

-XML je standard za opis sadržaja i strukture dokumenata i razmenu dokumenata na web-u, platformski je nezavisan, dobro podržan, slobodan za upotrebu.

-XML - eXtensible Markup Language:

\***Markup**-dodavanje specijalnog značenja podatku (koristi se tag za predstavljanje markup-a). Ideja jezika je da se korisni sadržaj uokviri odgovarajućim oznakama (koje treba da budu jednostavne, čitljive, razumljive). XML razdvaja strukturirani sadržaj dokumenta od njegove prezentacije. Znači, XML se sastoji iz teksta i tagova.

\***eXtensible**-proširiv jezik, dozvoljava definisanje novih tagova, meta jezik omogućava definisanje drugih markup jezika(WSDL).

### 2.Standardi zasnovani na XML-u

-SOAP-jednostavan protokol baziran na XML-u koji omogućuje aplikacijama razmenu podataka u tekstualnom obliku preko HTTP protokola;

-WSDL-jezik baziran na XML-u koji omogućuje opis web servisa;

-UDDI-Universal Description,

-Discovery and Integration Protocol.

### 3.XML vs HTML

-HTML služi za formatiranje i prezentovanje dokumenata na web-u.

-Koristi fiksni, predefinisani skup tagova.

-Nema jasnu granicu između sadržaja i forme pa je to jedna od glavnih prednosti XML-a u odnosu na HTML.

-XML nije evoluirani HTML. On je komplement HTML-u i dizajniran je ne da zameni HTML već da ponudi ono što HTML ne može.

-XML se koristi za: odvajanje podataka od prezentacije, razmenu podataka, čuvanje podataka...

### 4. Osnovni oblik XML deklaracije

```
<?xml version="1.0"?>
```

#### 1.Gde se koristi XML?

- Standardni format za razmenu i skladištenje podataka
- Bezicne komunikacije
- Aitomatizacija komunikacije
- E-Comerce
- Multimedijalni sistemi

#### 2.XML se koristi za kreiranje novih jezika (XML recenica)

- Na primer:

\*XHTML

\*WSDL - za opis Web servisa

\*WML - za mobilne uređaje

\*RSS - za preuzimanje online sadržaja

\*RDF I OWL - za opisivanje web resursa I ontologije

\*SMIL - primena kod multimedijalnih sistema

### 3.XML uredjivaci

-Notepad

-XML notepad

-Altova XMLSpy

-Stylus Studio XML

-EditX

-Xmetal

-Oxygen XML Editor

### 4.Pravila za imenovanje XML tag-ova

-Prvi karakter svakog taga mora biti slovo ili “\_”, brojevi I drugi karakteri nisu dozvoljeni.

-Nakon prvog karaktera dozvoljeni su brojevi kao I karakteri “-” i “.”

-Nazivi tagova ne mogu sadržati prazna mesta(blank)

-Nazivi tagova ne mogu da sadrže “:” jer je to rezervisani karakter koji se koristi kod XML imenovanih prostora (namespaces)

-Naziv taga ne može započeti sa “XML”

-Nije dozvoljeno prazno polje odmah nakon “<”

-Medjutim prazno polje može da prethodi “>” znaku.

### 5.XML sintaksna pravila

-svi XML elementi moraju imati završni tag

-XML tagovi su *case sensitive*

-XML elementi moraju da budu ugnjezdjeni na odgovarajući način

-XML dokumenti moraju imati glavni (root) element

-Vrednosti XML atributa moraju se staviti u navodnike

-Reference entiteta

-Komentari u XML-u (<!--Ovo je komentar-->)

-Komentari I instrukcije za obradu (PI) ne mogu da se pojave unutar tagova.

-Element ne može da ima dva atributa sa istim nazivom

- “<” I “&” karakteri ne mogu da se pojavljuju u sadržaju elemenata ili atributa.

### 6.XML DOM

\*Primer: preuzimanje tekstualne vrednosti prvog <title> elementa:

```
txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

\*Primer: preuzimanje tekstualne vrednosti “lang” atributa prvog <title> elementa:

```
txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");
```

\*Primer: azuriranje tekstualne vrednosti prvog <title> elementa:

```
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
```

```
x.nodeValue="Easy Cooking";
```

## 7.XLink I XPointer

-XLink definise standard za kreiranje hiperlinkova u XML dokumentima.

-XPointer omogućava hiperlinkovima da ciljaju specificne delove XML dokumenata.

# Web servisi

## 8.Arhitrtura web servisa

-Web servisi ukljucuju tri glavne uloge:

- \*Provajder servisa

- \*Registar servisa

- \*Korisnik servisa

-Tri glavne operacije vezane za web servise:

- \*Publikovanje

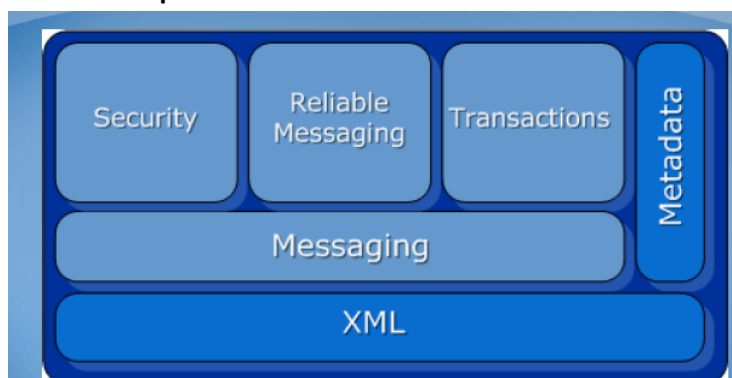
- \*Pronalazenje

- \*Povezivanje

## 9.Servisno orjentisana arhitektura



## 10.Sematski prikaz WS-\*arhitekture

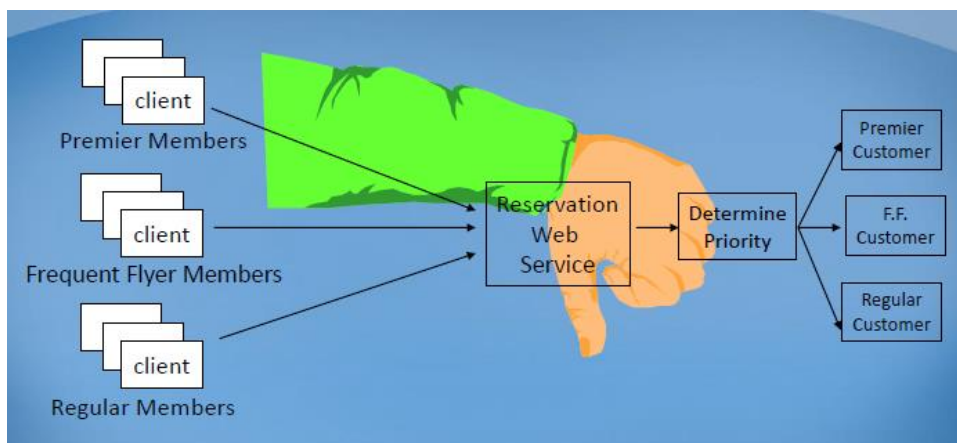


## 11.Sta je to REST?

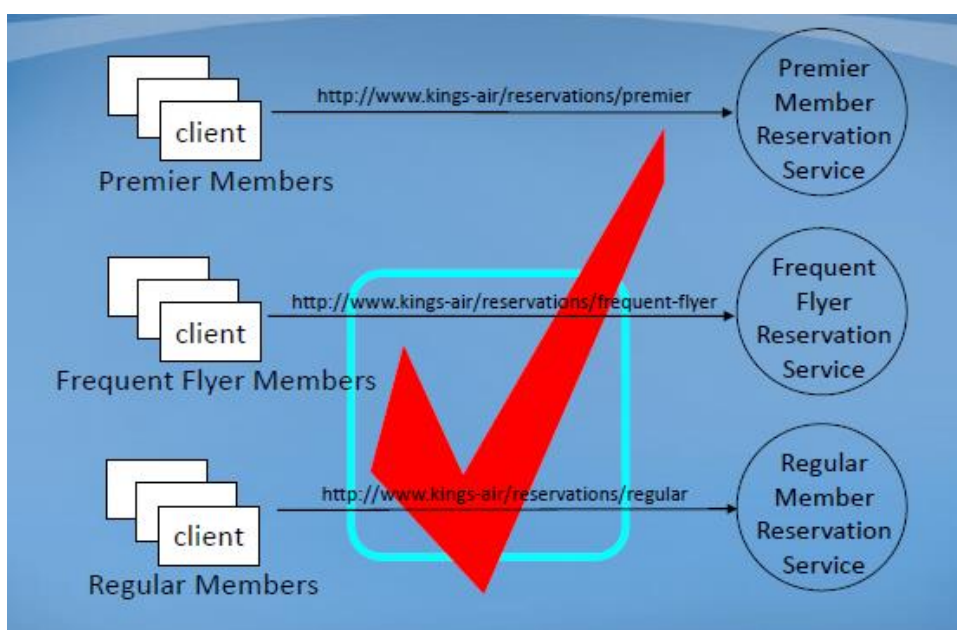
-REST je sablon za dizajn web servisa.

-Predstavlja alternativni nacin kreiranja web servisa.

Ovo nije REST



Ovo je REST



## 12. RESTfull web servisi

Konkretna implementacija REST web servisa sledi cetiri glavna dizajn principa:

- Eksplicitno koriscenje HTTP metoda
- Nezavisnost od konekcije (stateless)
- Izlaganje resursa putem URI-a
- Koristiti XML i/ili JavaScript Object Notation (JSON), za prenos podataka.

## 13. Bezbednost web servisa

-Autentifikacija – identitet

\*Ko je korisnik?

\*Kako dokazujemo da je neko stvarno to sto tvrdi?

-Autorizacija – kontrola pristupa

\*Sta korisnik moze da uradi

-Poverljivost – enkripcija

\*Kako da sprecimo uvid u podatke od neovlascenih lica?

-Integritet – onemogucavanje izmena

\*Kako da sprecimo izmenu poruka na putu od posiljaoca do primaoca?

- Autentifikacija – identitet
  - \*HTTP autentifikacija
  - \*SSL digitalni sertifikati
- Autorizacija – kontrola pristupa
  - \*Single Sign-On (SSO)
  - \*SAML(Security Assertion Markup Language)
- Poverljivost – enkripcija
  - \*SSL enkripcija
  - \*XML Encryption
- Integritet –onemogućavanje izmena
  - \*XML Digital Signatures