Screenshot_1.png VISOKA TEHNIČKA ŠKOLA

STRUKOVNIH STUDIJA

KRAGUJEVAC

**SEMINARSKI RAD**

**PREDMET: ELEKTRONSKO POSLOVANJE**

**TEMA: ANGULARJS**

**Profesor Student**

**Dr. Zoran Mirović Stefan Ivić 129/2014**

Sadržaj

[Elektronsko poslovanje](#_Toc473577718)

[1.0 Elektronsko poslovanje – osnovni pojam 1](#_Toc473577719)

[1.1 Razvoj i oblici elektronskog poslovanja 1](#_Toc473577720)

[1.2 Elektronska trgovina 2](#_Toc473577721)

[AngularJS](#_Toc473577722)

[2.0 AngularJS – osnovni pojam 4](#_Toc473577723)

[2.1 Razvojna istorija AngularJS-a 5](#_Toc473577724)

[2.2 Kako funkcioniše AngularJS 6](#_Toc473577725)

[2.3 AngularJS atributi 7](#_Toc473577726)

[2.4 Scope 9](#_Toc473577727)

[2.5 Dvosmerno povezivanje podataka 10](#_Toc473577728)

[Literatura 11](#_Toc473577729)

# **Elektronsko poslovanje**

## **Elektronsko poslovanje – osnovni pojam**

Elektronsko poslovanje jeste vođenje poslova na internetu, što ne podrazumeva samo kupovinu i prodaju, već organizaciju poslovanja firme u mrežnom okruženju, organizovanje poslovne komunikacije prema klijentima i brigu o klijentima.

Izraz elektronsko poslovanje je prvi put upotrebljen od strane američke kompanije IBM da bi se označilo poslovanje uz značajnu primenu savremene, elektronske tehnologije. Do tada se ovaj izraz odnosio na poslovanje u sektoru elektronike. Pod elektronskim poslovanjem podrazumevamo obavljanje poslovnih procesa uz primenu elektronske tehnologije. Elektronska tehnologija podrazumeva kombinovanu upotrebu informacionih tehnologija i telekomunikacija. Ova vrsta tehnologije omogućava slanje velikog broja informacija, na velike daljine u kratkom vremenskom periodu. To omogućava preduzeću, koje u svom poslovanju koristi elektronsku tehnologiju, da ostvari značajne uštede u troškovima poslovanja, efikasnije obavlja svoje zadatke i, samim tim, bude konkurentnije na tržištu.

Danas e-poslovanje ostvaruje ekspanzitivan rast. U dosadašnjem periodu e-poslovanje je doživelo posebnu ekspanziju u maloprodaji, izdavaštvu i u finansijskim uslugama. Generalno posmatrano, prednost elektronskog u odnosu na tradicionalno poslovanje su vezane za povećanje kvaliteta, agilnosti, za dodatne usluge sa jedne strane, i za sniženje prodajnih cena, smanjenje vremena izlaska na tržište, odnosno realizivanje transakcija sa druge strane. Takođe se intezivira zajednički nastup na elektronskom tržištu vezan za razvoj elektronskih partnerstva.

## **Razvoj i oblici elektronskog poslovanja**

Elektronsko poslovanje se pojavilo početkom 80-ih godina prošlog veka, ali je ubrzan razvoj doživelo poslednjih godina, pre svega zahvaljujući internetu.

Elektronsko poslovanje podrazumeva obavljanje poslovnih procesa uz primenu elektronske tehnologije. Ova vrsta tehnologije omogućava slanje velikog broja informacija na velike daljine u kratkom vremenskom periodu. To svojstvo elektronske tehnologije omogućava kompanijama koje u svom poslovanju koriste ovu vrstu tehnologije da ostvare značajne uštede u troškovima poslovanja, efikasnije obave svoje zadatke i budu konkurentnije na tržištu.

Elektronsko poslovanje ima vrlo široko područje uticaja i primene, pa bi osnovni oblici bili:

• e-trgovina (engl. e-commerce) - predstavlja poslovnu komunikaciju i prenos dobara i usluga (kupovina i prodaja)

• e-bankarstvo (engl. e-banking) - predstavlja poslovanje na relaciji banka-klijent

• e-uprava (engl. e-government) - predstavlja elektronsko poslovanje u upravi ili administraciji (države, preduzeća, itd.)

## **Elektronska trgovina**

Elektronska trgovina (engl. Electronic Commerce) ili E-trgovina (engl. e-Commerce) predstavlja poslovnu komunikaciju i prenos dobara i usluga (kupovina i prodaja), preko mreže i računara, kao i prenos kapitala, korištenjem digitalne komunikacije. Uključuje i ostale poslovne funkcije preduzeća, kao što su marketing, finansije, proizvodnja, prodaja i administrativna funkcija, koje su neophodne u procesu trgovine.

Ovakvim načinom definisanja smo izbegli čestu grešku koja podrazumeva poistovjećenje elektronske trgovine sa elektronskim poslovanjem. E-trgovina je ipak samo manji podskup e-poslovanja, zajedno sa oblastima kao što su elektronsko bankarstvo, mobilno poslovanje, virtuelne organizacije, elektronske berze, elektronsko poslovanje u javnoj upravi, itd.

Poslovanjem po principima elektronske trgovine, Internet pruža niz mogućnosti za maloprodaju:

• nisu potrebne skupe nekretnine niti aranžiranje izloga

• neophodno je minimalno prodajno osoblje

• postoji mogućnost da se prodaje kupcima na bilo kojem geografskom području

• omogućena je trenutna komunikacija

• prezentuje se interaktivni multimedijalni katalog koji može da pruži onoliko informacija koliko kupac želi (bez skupih klasičnih kataloga i poštanskih troškova)

• veoma brzo se može vršiti prilagođavanje za promene u prodajnim cenama i nivou zaliha

• velika je mogućnost adaptacije zahtevima kupaca

S druge strane javljaju se i određeni problemi u prodaji preko Interneta:

• mnogi potencijalni kupci još uvek ne koriste Internet

• mnogi nemaju brze veze

• kupovina preko Interneta zahteva veliku dozu poverenja (sa obe strane)

• ugrožena privatnost: lična (proaktivni marketing, cookies), finansijska (zloupotreba kartica)

• problem stvarnog postojanja prodavca na Internetu

• bezbednost obavljanja transakcije

Neophodno je navesti i set osnovnih stvari koje sajber trgovce čine uspešnim:

• popusti

• super selekcija: bogata ponuda

• posebne usluge: podsećanje na važne datume, postprodajne usluge, ...

• komfor: iz fotelje, non-stop, mesecima unapred

• brza isporuka

• zabava: video igre, šale

• koncept dodatne vrednosti (engl. added value)

Pored toga, možemo reći da je kupovina i prodaja informacija, proizvoda i usluga putem računarske mreže i podrška za bilo koju vrstu poslovnih transakcija putem digitalne infrastrukture - digitalno omogućene komercijalne transakcije između organizacija (engl. B2B), organizacija i pojedinaca (engl. B2C), pojedinaca (engl. C2C), organizacija i vladinih agencija(engl. B2G), vladinih agencija(engl. G2G), itd.

# AngularJS

1. **AngularJS – osnovni pojam**

AngularJS je JavaScript strukturni okvir (framework) za izradu dinamičkih veb aplikacija. Angular omogućava korišćenje HTML kao templejtnog jezika i omogućava produžavanje HTML sintakse radi boljeg iskustva sa kreiranjem komponenti. Uloga Angulara jeste da olakša pisanje koda upotrebom tehnika kao što su data binding i dependency injection, time se smanjuje količina koda koju je potrebno napisati.

Ovaj okvir je je otvorenog koda koji primarno održava kompanija Google zajedno sa velikim brojem programera koji doprinose kodu.

AngularJS je napravljen sa ciljem da se deklarativno programiranje upotrebljava uprilikom izrade korisničkih interfejsa i povezivanja komponenti, dok imperativno programiranje više odgovara za definisanje aplikacione biznis logike.

Glavni ciljevi ovog okvira su :

* Razdvajanje DOM manipulacije od aplikacione logike.
* Razdvajanje klientske strane od serverske. Ovo omogućava razvoj obe strane paralelno.
* Obezbeđivanje structure za redosled razvoja aplikacije.

AngularJS implementira MVC (Model-View-Controler) arhitekturu radi razdvajanja komponenti za prezentaciju, podatke I logičke komponente. Koristeći zavisnost paketa, Angular donosi tradicionalne serverske service, kao što su kontroleri koji zavise od prezentacionog sloja, na veb aplikacije na klijentskog strani. Ovo omogućava da se smanji veliki deo tereta na server.

## **2.1 Razvojna istorija AngularJS-a**

AngularJS je prvobitno ravijen 2009. godine od strane Miška Hevery-a u Brat Tech LLC kompaniji, kao softver iza servisa za skladištenje JSON-a koji se naplaćivao po megabajtu. 2010. godine, Hevery odlučuje da objavi Angular kao okvir otvoreng koda. 2011. godine objavljuje se AngularJS verzija 1.0.

23. Septembra 2014. Na ng-Europe konferenciji najavljena je Angular 2.0 verzija. Velike promene u 2.0 verziji izazvale su veliku polemiku među programerima. Razlog je potpuna prerada u odnosu na verziju 1.0. U aprilu, 2015. godine najavljeno je da Angular 2 prelazi iz Alpha verzije u Developer Preview veriju koja je bila mnogo stabilnija i bolje dokumentovana. U decembru, 2015. godine objavljena je prva beta a u maju 2016. godine prvi release kandidat. Finalna verzija objavljena je u septembru 2016. godine.

Velike promene u odnosu na Angular 1 su :

* Pordška za mobilni razvoj - razvoj veb aplikacija sa podrškom za pametne uređaje
* Modularnost – veliki deo osnovne funkcionalnosti je preveden u module, kreirajući bržu i manju osnovu.
* Podršku za modern pretraživače.
* Upotrebu Mikrosoftovog TypeScript jezika, koji donosi poboljšanja kao što su :
  + Objektno orijentisano programiranje bazirano na klasama.
  + Static Typing ( proces verifikovanja grešaka programa zasnovanom na analizi izvornog koda.)
  + Lambdas
  + Podršku za ECMAScript 6
* Poboljšan dependency injection.
* Dinamično učitavanje
* I mnoge druge mogućnosti.

**2.2 Kako funkcioniše AngularJS**

Ovaj okvir funkcioniše tako što pročita HTML stranicu, koja sadrži modifikovanje attribute. Angular te attribute interpretira kao upustvo da poveže ulazne (inuptu) i izlazne (output) delove na stranici sa modelom koji predstavlja standardne JavaScript promenljive. Vrednosti ovih JavaScript promenljivih mogu biti manuelno podešeni u koviru koda ili dobavljeni iz statičkih ili dinamičkih JSON resursa.

Primer :

1. <!doctype html>
2. <html ng-app>
3. <head>
4. <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.1/angular.min.js"></script>
5. </head>
6. <body>
7. <div>
8. <label>Name:</label>
9. <input type="text" ng-model="yourName" placeholder="Enter a name here">
10. <hr>
11. <h1>Hello {{yourName}}!</h1>
12. </div>
13. </body>
14. </html>

Upotrebom ng-app atributa definišemo odakle počinje naša Angular aplikacija. U ovom slučaju ona će koristi celu starnicu.

1. <html ng-app>

Nakog toga potrebno je da učitamo Angular biblioteku.

1. <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.1/angular.min.js"></script>

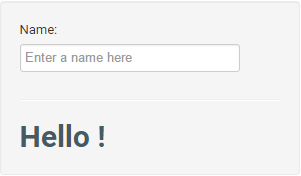
Korišćenjem imput HTML tag-a, možemo da dozvolimo korisniku da unese svoje ime, a atibutom ng-model povezujemo unos (formu) i model. Ovo znači da bilo kakva promena na kontroli promeniće podatke u modelu.

1. <input type="text" ng-model="yourName" placeholder="Enter a name here">

Nakon toga potrebno je da na deklarativan način, upotrebom vitičastih zagrada, odredimo lokaciju gde će se prikazati podaci koje unosimo u formu.

1. <h1>Hello {{yourName}}!</h1>

Konačan izgled aplikacija. Bilo kakav unos u tekstbox (yourName input) će se prikazati u h1 elementu ispod.



## **2.3 AngularJS atributi**

Zadaci koje izvršava AngularJS izvršavaju se u tri faze, nakon učitavanja objektnog modela dokumenta (engl. DOM):

1. Kreiranje novog ubrizgavača
2. Kompilacija svih direktiva
3. Povezivanje svih direktiva u scope

AngularJS direktive dozvoljavaju programerima da odrede specifične elemente i atribute nalik na HTML koji se mogu ponovo upotrebiti, a koji definišu povezivanje podataka i ponašanje komponenti korisničkog interfejsa. Neke od najčešće korišćenih direktiva su:

|  |  |
| --- | --- |
| ng-app | Deklariše koren dokumenta AngularJS aplijacije, unutar koga se mogu koristiti direktive koje će povezati podatke i definisati ponašanje. |
| ng-bind | Postavlja text objektnog modela dokumenta na vrednost izraza. Na primer <span ng-bind="name"></span> će prikazati vrednost promenjive unutar span elementa. Bilo koja promena promenjive „name“ u aplikaciji će se istovremeno promeniti i unutar objektnog modela dokumenta. |
| ng-model | Slično kao i ng-bind, ali uspostavlja dvosmernu vezu između korisničkog interfejsa i scope-a. |
| ng-model-options | Omogućava podešavanja kako se obavljaju ažuriranja na nivou modela. |
| ng-class | Omogućava dinamično dodavanje klasnih atributa. |
| ng-controller | Definiše JavaScript klasu kontrolera koji obrađuje HTML izraze |
| ng-repeat | Instancira element jednom za svaku stavku iz skupa. |
| ng-show, ng-hide | Uslovno prikazuje ili sakriva element u zavisnosti od vrednosti Bulovog izraza. Prikaz i sakrivanje je postignuto podešavanjem CSS svojstva „display“. |
| ng-switch | Uslovno instancira jedan šablon iz seta mogućih, u zavisnosti od vrednosti selekcije |
| ng-view | Osnovna direktiva odgovorna za upravljanje rutama koje rešavaju JSON pre renderovanja šablona kojima upravljaju određeni kontroleri. |
| ng-if | Osnovna direktiva „if“ petlje koja dozvoljava prikazivanje određenog elementa ako su uslovi ispunjeni. Ukoliko uslovi nisu ispunjeni, element se briše iz objektnog modela dokumenta. Kada se uslovi ispune, klon kompajliranog elementa se ponovo ubacuje u DOM |

## **2.4 Scope**

Angular koristi termin „scope“ sa nešto drugačijim značenjem nego što je to uobičajeno u kompjuterskim naukama.

Scope u kompjuterskim naukama opisuje kada je određena veza na snazi. ECMA-262 definiše scope kao leksičko okruženje koje definiše okruženje u kojem se izvršava objekat funkcije.

U Angular-u, scope je vrsta objekta koja može biti unutar ili izvan dosega u određenom delu programa, prateći uobičajena pravila dosega promenjive u jeziku JavaScript kao bilo koji drugi objekat. U nastavku, termin scope predstavlja vrstu Angular scope objekta, a ne dosega promenjive.

Primer upotrebe $scope-a kao model podataka.

Script.js

1. angular.module(‘scopeExample’, [])
2. .controller(‘MyController’, [‘$scope’, function($scope){
3. $scope.username = ‘World’;
4. $scope.sayHello = function() {
5. $scope.greeting = ‘Hello ’ + $scope.username + ‘!’;
6. };
7. }]);

Index.html

1. <div ng-controller=”MyController”>
2. Your name :
3. <input type=”text” ng-model=”username”>
4. <button ng-click=’sayHello()’>greet</button>
5. <hr>
6. {{greeting}}
7. </div>

## **2.5 Dvosmerno povezivanje podataka**

Angularovo dvosmerno povezivanje podataka je najvažnija mogućnost, i ona smanjuje količinu napisanog koda smanjujući odgovornosti servera u vezi sa prikazom šablona. Umesto toga, šabloni se renderuju u čistom HTML-u prema podacima koji se nalaze unutar scope-a definisanog u modelu. $scope servis u Angularu detektuje promene modela i menja HTML izraze na korisničkom interfejsu preko kontrolera.

Isto tako, sve promene na korisničkom interfejsu se reflektuju na model. Ovo zaobilazi potrebu da se aktivno manipuliše objektnim modelom dokumenta i podstiče samostalan rad aplikacije i brzu izradu prototipa veb aplikacija. AngularJS detektuje promene u modelu poredeći sadašnje vrednosti sa vrednostima koje su sačuvane kroz raniji proces provere, za razliku od Ember.js i Backbone.js framework-a koji se aktiviraju kada nastanu promene na modelu.

# Literatura

* 1. <https://angular.io/>
  2. <https://angularjs.org/>
  3. <https://angularzone.wordpress.com/angularjs-history/>
  4. <http://blog.jhades.org/the-java-origins-of-angular-js-angular-vs-jsf-vs-gwt/>
  5. <https://en.wikipedia.org/wiki/AngularJS>