

Metodologija stručnog i naučnog rada

Veb u realnom vremenu

Stefan Isidorović

Univerzitet u Beogradu, Matematički fakultet
Studentski trg 16
isidorovicstefan@gmail.com

Abstract. Ovaj rad služi da bliže objasni pojam Veba u realnom vremenu, svrhu i namenu, kao i nekoliko metoda za realizaciju.

Keywords: web, real time

1. Uvod

U Uvodu će biti par reči o samom Vebu. Njegovoj arhitekturi, protokolima na kojima se zasniva i ciljevima koje nastoji da ispuni. Nakon Uvoda i bližeg upoznavanja sa Vebom i protokolima na kojim počiva uvešćemo pojam "Veb u realnom vremenu", potrebe da tako nešto postoji, prednosti i mane, kao i osnovne probleme u realizaciji. Takođe će biti reči o nekoliko osnovnih metoda za realizaciju osobina kojih rade u realnom vremenu. Rad će se fokusirati na samoj komunikaciji između klijenta i Veb servera, koja predstavlja možda najveći problem u realizaciji Veba u realnom vremenu. Biće pomenuti i problemi na serverskoj strani, odnosno realizovanju sprege između servera za upravljanje bazama podataka i samog veb servera, ali se neće dublje ulaziti u taj deo problema.

1.1. Veb

"Veb (WWW - World Wide Web) je mreža servera koja sadrži programe i datoteke. Mnoge od tih datoteka sadrže hipertekstualne veze do drugih dokumenata dostupnih kroz tu mrežu."

Definicija Veba - IBM Dictionary of Computing, 1999

Veb kao pojam je veoma teško definisati jer se neprekidno razvija i širi iz dana u dan. Navedena definicija je prilično dobra i liči dosta na većinu definicija Veba, međutim nedostaje konkretizacija, jer je Veb tačno jedan. Ideja za Veb je nastala 1989. godine na CERN institutu, a prvi put zvanično predstavljena 6. avgusta 1991. godine od strane njenog idejnog tvorca Tima Berners-Lija. Treba praviti razliku između interneta i Veba. Internet je gradivni materijal i sredstvo u izgradnji Veba (Veb je jedan od servisa koje pruža internet). Osnovna zamisao Veba je da postoji određen broj resursa kojima se može pristupiti preko mreže. Taj jedinstveni identifikator se naziva URI (Uniform Resource Identifier) i služi da jedinstveno opiše svaki resurs na Veba (više informacija na http://en.wikipedia.org/wiki/Uniform_resource_identifier), resursi (mediji) predstavljaju pomoću HTML (Hypertext Markup Language) jezika. Protokol koji Veb koristi je HTTP (Hypertext Transfer Protocol) razvijan baš za potrebe interneta. HTTP je protokol bez stanja koji se sastoji od niza zahteva i odgovora. Gde zahtev podnosi korisnik, a odgovor dobija od strane servera. Kako zahtev treba da bude formulisan, kao i kakav odgovor treba da očekuje je strogo definisano pomoću HTTP protokola, koji se usput rečeno može dodatno modifikovati za potrebe Veb aplikacije, naravno toga treba da budu svesni klijentska kao i serverska strana koji koriste modifikovanu verziju tog protokola.

Razvijanjem Veb i njegovom eksplozijom krajem 90ih godina, dolazi do naglog razvijanja tehnologija za Veb, kao i samim seljenjem postojećih sistema i aplikacija u Veb, gde mu može pristupiti veliki broj ljudi. Kao i razvijanjem hardverske tehnologije (danas računari i "ugrađeni" (embedded) uređaji koji se koriste za pristup Vebu imaju daleko veću procesnu moć nego nekada). Pojava takvih uslova dovodi do drugačijeg pristupa u korišćenju Veba. Nekada je Veb strana predstavljala skup statičkih podataka, danas su Veb strane uglavnom ozbiljne Veb aplikacije, koje sem što nude tražene informacije, mogu i da interaguju sa korisnikom. Ovakav pristup se naziva "Arhitektura teških klijenata" gde korisnik kada pristupa Veb strani zapravo skida neku vrstu aplikacije koja se izvršava na

njegovom računaru i komunicira sa serverom. Ovo se popularno naziva "Veb 2.0", koji označava novu eru Veba. Većina Veb aplikacija danas je zapravo baš ovakav vid aplikacije, gde korisnik nesvesno izvršava dinamički "program" na svom računaru, a Veb koristi samo da bi taj "program" interagovao sa serverom i time dobili privid visoko dinamične Veb strane. Sve ovo i dalje funkcioniše na istom HTTP protokolu na kome je funkcionisao Veb na samom početku, naravno i sam HTTP protokol je doživeo izmene i unapređivanja (danas aktuelna HTTP/1.1 verzija).

2. Veb u realnom vremenu

Pojavom prethodno pomenutih Veb aplikacija, dolazi do nove dimenzije razvijanja Veba. Dinamičko učitavanje sadržaja na zahtev klijenta, učitavanje sadržaja sa više servera odjednom na istu stranu, slanje podataka, obrada i slično. Kako je cilj da korisnik dobije traženu informaciju, to se ispunjava. Ali šta ako se informacija promeni u međuvremenu? Ili pristignu nove informacije koje su vezane za korisnika ili bitne za ono što korisnik traži ili radi? Da bi se informacija učitala na strani klijenta potrebno je da postoji zahtev koji klijent podnosi serveru za učitavanje informacija. Sposobnost da klijent dobije informaciju u istom ili gotovo istom trenutku (razlika reda veličina par sekundi) se naziva "Veb u realnom vremenu". Danas se ovo koristi masovno, a najveću primenu doživeo je na socijalnim mrežama (Facebook, Twitter) gde se može videti na automatskom učitavanju novosti, pristizanju obaveštenja, poruka itd. Istorijski gledano Veb u realnom vremenu je nastao 2000 godine u mc2labs.com od strane nezavisnog italijanskog istraživača. Veb u realnom vremenu u mnogome širi upotrebnu vrednost Veba. Sada se mogu razvijati kolaboracioni alati kojima može pristupiti svako širom sveta, možemo razmenjivati poruke koje će stizati momentalno i još mnogo toga. Danas ukoliko koristite internet (kao što koristite) sigurno koristite i prednosti Veba u realnom vremenu, pošto je danas široko primenjen gotovo svuda. Preko već pomenutih socijalnih mreža, preko internet pretraga (Google, Yahoo, Bing), raznih blogova, gotovo u svim modernijim Veb aplikacijama danas postoji komponenta koja radi u realnom vremenu. 2009 godine Google je uveo u svoju pretragu komponentu koja radi u realnom vremenu i pretražuje milijarde i milijarde resursa širom Veba. Ovo je ujedno i najveći projekat koji efektivno koristi Veb u realnom vremenu.

Razlike Veba u pravom vremenu i Veba u realnom vremenu Danas je veoma popularan termin "Veb u pravom vremenu" (Right Time Web) i mnogi ga često mešaju sa pojmom Veba u realnom vremenu. Veb u pravom vremenu propagira sposobnost da informacija stigne do korisnika u najpovoljnije vreme, odnosno baš kad mu zatreba, nezavisno od toga da li je korisnik svestan da ta informacija postoji ili ne. U velikom broju slučajeva pravo vreme za korisnika je ono kada ta informacija nastane, što dovodi do zabune. Primer koji ilustruje tako nešto je da kada vam tražite da kupite auto i na osnovu toga vam se prikazuju reklame za automobile, dok će se nekoj domaćici prikazivati sniženja šećera, brašna i slično (Google AdWords funkcioniše na sličnom principu). Informacije o automobilima (ili sastojcima) nisu nastale baš u tom trenutku ali se dinamički učitavaju na osnovu ponašanja korisnika. Veb u realnom vremenu može biti jedan preduslova za realizaciju Veba u pravom vremenu.

2.1. Problemi u realizaciji

Veb u realnom vremenu zahteva intenzivnu komunikaciju u smeru od servera ka klijentu, bez eksplicitnog zahteva klijenta za baš tu informaciju. Dinamičko učitavanje informacija nije problem, dok god klijent zahteva te informacije. Sve se zasniva na matičnom protokolu Veba, HTTPu, koji nema stanja. Veb funkcioniše preko serije zahteva i odgovora, kao što je već pomenuto u Uvodu. Problem nastaje kako obavestiti klijenta da je došlo do novih informacija za koje je on zainteresovan? Pošto da bi se informacija prikazala klijentu, on je mora zatražiti, server ne može poslati klijentu zahtev.

2.2. Tehnologija "guranja"

Tehnologija "guranja" (Push technology) je tehnologija gde server sprema podatke i emituje ih klijentu. Gde se omogućuje komunikacija servera sa klijentom, bez prethodnog zahteva za informaciju. Ovo se ostvaruje različitim tehnikama gde ovakva komunikacija zapravo postoji ili se veštačkim putem omogućava. "*Gurana obaveštenja*" (*Push notifications*) je termin koji se vezuje za tehnologiju guranja, ali se svodi na to da na jednom kanalu server emituje informacije koje su vezane za sve korisnike aplikacije a klijentski delovi sa vremena na vreme proveravaju da li na kanalu postoje nove informacije. Koristi se za dostavljanje informacija velikom broju korisnika. Tehnologija guranja je najzastupljeniji vid realizacije komponenti Veba u realnom vremenu. Da bi ovakav pristup bio moguć koriste se tehnike "trajne" HTTP konekcije ili otvaranje neke vrste socketa.

Trajne HTTP konekcije Svaki HTTP zahtev u zaglavlju ima vreme za koje je taj zahtev aktuelan, odnosno rok trajanja. Ukoliko klijent ne dobije odgovor za to vreme, smatra se da je taj HTTP zahtev propao odnosno smatra se kao da nije ni stigao do servera. Trajna HTTP konekcija se ostvaruje slanjem zahteva koji ima neograničen rok trajanja, odnosno aktuelan je dok god se na njega ne odgovori. Ovakvim pristupom se prevazilazi problem da server odgovori u bilo kom trenutku klijentu i prosledi mu informacije u momentu kada pristignu. Ali otvara probleme koji mogu nastati ukoliko server nije dostupan ili se informacija ne može dobiti ili čak i u određenim situacijama stvoriti i bezbednosne probleme u Veb aplikaciji.

Veb soketi Veb soketi predstavljaju otvaranje direktnog kanala između servera i klijenta. Ovakav pristup je najmlađi. Podrška za Veb sokete je došla zaključno sa verzijom 5 HTMLa, dok je ovakva tehnologija još uvek u eksperimentalnoj fazi u PHPu. Pored "native" podrške za sokete postoje i razne druge biblioteke koje na jedan ili drugi način imitiraju ovakav vid komunikacije. Za ovakve potrebe su razvijeni i posebne tehnologije za implementaciju servera a najpoznatiji je NodeJS, koji predstavlja implementaciju serverske strane u JavaScript jeziku. Ovakav pristup je najbolji za realizaciju Veba u realnom vremenu jer dopušta neometanu komunikaciju u oba pravca nezavisno od toga da li postoji zahtev ili ne.

2.3. Polling

Polling je tehnologija koja se zasniva na drugačijim principima. Istorijski gledano Polling je tehnologija starija od tehnologije guranja. Ideja je manje više ista, stim što je realizacija

drugačija, jer je razvijana da prati principe i tehnologije koje su već postojale i predstavljaju osnovu Veba. Danas se koristi kada tehnologiju "guranja" nije moguće implementirati ili realizovati iz raznih bezbednosnih ili nekih drugih razloga, što je mnogo češće nego što se čini. Polling predstavlja suprotno od tehnologije guranja. Kod tehnologije guranja server emituje podatke klijentu i klijent ih prima u isto vreme obrađuje ih i prikazuje, kod pollinga je stvar suštinski drugačija jer klijent zahteva podatke od servera nakon čega ih dobija. Trajne HTTP konekcije liče dosta na Polling ali nisu suštinski deo Pollinga jer se oslanjaju na deo HTTP protokola koji je kasnije dodat (ili veštački omogućen, odnosno nije bio deo standarda), kao i većina tehnika koje spadaju pod tehnologiju guranja. Polling tehnologija se zasniva na kontinualnom slanju zahteva serveru za određenu informaciju (nagađamo da li je ta informacija pristigla, jednom ćemo biti u pravu). Server odgovara na te zahteve praznim odgovorima (ili predefinisanim porukama), nakon čega se šalje sledeći zahtev. Polling tehnike se mogu iskoristiti da se emulira tehnologija guranja. Jedan "push" zahtev se može realizovati pomoću konačnog broja "pull" zahteva. Mana koja je najuočljivija u ovakvom pristupu jeste ogroman broj zahteva, pošto se polling zasniva u kontinualnom slanju zahteva, to samim tim povlači veliki broj zahteva i povećanje "workflow-a" servera a i klijenta. Pa je ovakav pristup evoluirao i unepređen je u Long-Polling tehniku.

LongPolling Suštinska razlika između LongPollinga i običnog Pollinga je u broju zahteva koji se šalje. Kod običnog pollinga se šalju zahtevi jedan za drugim, odnosno drugi se šalje onog trenutka kada stigne odgovor od servera. Odgovori koji stižu od servera kod običnog Pollinga su u velikoj većini slučajeva prazni i stižu vrlo brzo, što povlači veliki broj zahteva u kratkom vremenskom periodu. LongPolling propagira veći vremenski period između slanja zahteva i zahtevi ostaju otvoreni taj vremenski period. LongPolling predstavlja hibrid trajne HTTP konekcije i pollinga. Klijent šalje zahtev koji je ograničen na neki vremenski period t , kada server primi taj zahtev, on ostaje otvoren do trenutka kada pristigne informacija za koju je klijent zainteresovan, ili dok ne istekne period t , nakon čega server šalje odgovor klijentu. Klijent nakon primanja i obrade odgovora šalje nov zahtev serveru, nakon čega se proces ponavlja. Iskustvo govori da se ova tehnika danas najčešće koristi za realizaciju komponenti Veba u realnom vremenu, uz naravno Veb sokete. Razlog zašto je ova tehnika toliko zastupljena je što najviše odgovara i prati osnovnu ideju Veba.

3. Najzastupljenija rešenja

Ovde će biti navedeno par najzastupljenijih rešenja za realizaciju Veba u realnom vremenu, kao i par reči o svakom od njih. Neće se dublje zalaziti u svaku od njih.

3.1. NodeJS i SocketIO

NodeJS O NodeJSu je bilo reči u radu. Ovo je JavaScript na serverskoj strani. NodeJS je razvijen 2009 godine od strane Ryan Dahl-a. NodeJS je softverska platforma za izradu Veb servera čija je osnovna ideja bila da poveća efikasnost i obradu velikog broja manjih zahteva. Priroda NodeJS je da bude baziran na događajima (event-based programming). Izgrađen je na Google V8 Javascript runtime-u. Veoma je skalabilan i lagan. NodeJS

server pokreće samo jedna nit (iako se poslovi sa mrežom i fajlovima vrše u više niti). Zbog svoje asinhronone prirode je odličan za Veb u realnom vremenu.

SocketIO je Javascript na klijentskoj strani koja omogućuje komunikaciju kroz sokete sa serverom. Primarno je implementirana za potrebe komunikacije sa NodeJS serverima. Kasnije je razvijen da može da komunicira i sa Veb serverima implementiranim u drugim jezicima.

3.2. Comet i Bayeux protokol

CometD je arhitekturni model Veb aplikacije koji implementira Bayeux protokola za realizaciju Veba u realnom vremenu. Bayeux protokol se zasniva na LongPollingu i HTTP Streaming tehnologiji (osnova Push Notificaiion tehnologije), pomoću AJAX poziva. Comet je poznati i pod imenima AJAX push, Reverse AJAX, Two-Way-Web itd. Predstavlja skup tehnika. Jedna od njih je LongPolling sa serverom, gde sa na klijentskoj strani implementira skriveni frejm (iFrame) u koji se smeštaju podaci i izvršavaju (skript tagovi i slično).

3.3. Ratchet - PHP

Ratchet je biblioteka namenjena PHP baziranim Veb serverima koja omogućava kreiranje dvosmernih komunikacionih Veb aplikacija, pomoću Veb soketa. Biblioteka je još uvek mlada i pokazuje prilično dobre rezultate. Predstavlja dobar omotač oko "native" podrške PHP Veb soketa koja je sama po sebi prilično nestabilna, ali Ratchet prilično ulepšava stvari. Komunikacija na klijentskoj strani je rešena pomoću SocketIO biblioteke ili novouvedenih Veb soketa u HTML5.

3.4. SignalR - ASP.NET

Predstavlja alternativu NodeJS serveru za .NET platformu. Razvijen je pre samo godinu dana. SignalR je serverska platforma za pisanje asinhronih, baziranih na događajima, skalabilnih Veb aplikacija. Bazira se na Veb socketima i koristi ih za dvosmernu komunikaciju i time omogućava komponente koje rade u realnom vremenu. Serverski kod se kodira u C#-u. 24 januara 2014. godine je izašla zvanična stablina verzija koja nosi v2.0.2

3.5. Eksterna rešenja

Eksterna rešenja predstavljaju gotova rešenja kroz već implementirane servere, koje mi koristimo kroz dat API. Tim serverima imate ograničen pristup i u većini slučajeva ne dozvoljavaju da se čuva trag (bar ne u besplatnim rešenjima) o razmeni informacija, ili je taj trag veoma kratkog veka. Predstavlja brzo rešenje za neke manje komponente u realnom vremenu. Kao na primer manje aplikacije za razmenu poruka i slično. Savet autora ovog rada je da se ovakav pristup kod izrade ozbiljnijih Veb aplikacija izbegava. Iako je lako rešenje i lepo funkcioniše, ne dozvoljava toliko slobodu kao rešenje koje sami implementirate. Jedno takvo rešenje je i AngularFire tehnologija koja se zasniva na FireBase platformi i AngularJS javascript biblioteci.

1. Predavanja profesora Saša Malkova sa predmeta Veb Programiranje
2. http://en.wikipedia.org/wiki/Real-time_web
3. http://en.wikipedia.org/wiki/Push_technology
4. http://en.wikipedia.org/wiki/Pull_technology
5. <http://nodejs.org/>
6. [http://en.wikipedia.org/wiki/Comet_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))
7. <http://svn.cometd.com/trunk/bayeux/bayeux.html>
8. http://en.wikipedia.org/wiki/Uniform_resource_identifier
9. <http://www.w3.org/Protocols/>
10. <http://socketo.me/>
11. <http://signalr.net/>
12. <http://angularfire.com>

Informacije o autoru **Stefan Isidorić** je student master studija Matematičkog fakulteta, Univerziteta u Beogradu. Učestvovao je u razvoju obrazovnih aplikacija u okviru fakulteta. Slobodno vreme provodi igrajući igre i programirajući. Honorarno radi na razvoju platforme za izradu Veb aplikacija. Polja interesovanja su mu Veb tehnologije, primarno arhitekture Veb aplikacija i back-end, kao i arhitektura softverskih sistema, kao i embedded programiranje u native tehnologijama.