

# Moderni front-end

Web je razmazio korisnike, utjecao na korisnička očekivanja i izgradio sliku modernog front-enda. Upravo zbog toga danas imamo beskompromisne korisnike kojima je često kriterij hoće li neku aplikaciju prihvatiti ili odbaciti upravo korisničko sučelje, bez obzira na stvarnu korisnost same aplikacije

■ TOMISLAV MESIĆ

U kontekstu interakcije čovjeka i stroja, a iz pozicije softverskih rješenja u *enterprise* i *consumer* svijetu (svejedno), front-endom se smatra vizualno korisničko sučelje kojim čovjek vrši interakciju s nekim uređajem, mobitelom, računalom ili jednostavno strojem.

Iako se već pri spomenu "front-enda" većini onih koji prepoznaju o čemu bi se tu moglo raditi zaiskre oči pri pomisli na razvojnu platformu, programske jezike, IDE, razvojne alate, IOS, Android, Firefox OS... ne, ne, ovdje se ne radi o tehnologiji. Moderni front-end nije samo tehnologija niti je samo dobar dizajn, ni visok stupanj upotrebljivosti nekog softverskog rješenja, ni dobro posloženi sadržaj, pravilan raspored kontrola, fluidno i rezpozivno sučelje... Moderni front-end čak nije ni sve to zajedno.

Budući da se radi o prilično mladoj grani IT-a (iako je to tako trebalo biti oduvijek, danas sa sigurnošću možemo reći da se radi o potpuno zasebnoj niši), ne tako dobro definiranoj i s još nejasno složenim procesima, teško je dati jasan nedvosmislen odgovor na pitanje "Što bi bilo dobro korisničko sučelje?" Zasad neka odgovor bude "moderni front-end puno je više od svega prethodno navedenog."

## It's not about technology

Iako svaki uređaj podrazumijeva nekakvo korisničko sučelje kojim komunicira s okolinom, ovdje ćemo se ograničiti samo na uređaje i korisnička sučelja koja spadaju i domenu desktop i mobilnih računala, smartfona i tableta. Također, važno je napomenuti da kada se govori o korisničkom sučelju, postoji bitna razlika među *enterprise* i *consumer* svjetovima.

Iako je za očekivati da će se u budućnosti takva slika promijeniti, u *enterprise* svijetu (javna uprava, svemirska stanica, naftne platforme, nuklearne elektrane) još uvijek vrijedi da je važno da nešto radi, ne i kako izgleda, dok kada govorimo o *consumer* svijetu (zabava, glazba, igranje, javni web, mobilne aplikacije), tu više nema kompromisa; sustav mora savršeno raditi i savršeno izgledati iznutra i izvana. Sve skupa zapravo je simbioza softvera i hardvera, što je također vrlo zanimljiva tema, no o tome možda nekom drugom prilikom.



Okvir za izgradnju modernog front-enda

## Osiguranje dobrog front-enda

Ako isključimo računalne igre, nekada su korisnička sučelja projektirali, dizajnirali, razvijali i programirali sami programeri jednostavnim primjenom kontrola korisničkog sučelja koje nudi alat koji bi koristili za razvoj. Danas je kod razvoja modernih aplikacija takav scenarij nezamisliv.

Ako se ne radi o pukoj sreći, teško da ćete postići dobar front-end bez prolaska kroz sve faze (ili barem većinu) razvojnog/životnog ciklusa modernog front-enda: od dobre analize i planiranja, preko dizajna, prijeloma i razvoja, do povezivanja UI logike s poslovnim logikom u pozadini.

Osim u već spomenutim igrama ili u aplikacijama iz *enterprise* svijeta, iz današnje je perspektive ta prošlost toliko daleka da nove generacije dionika koji sudjeluju u procesu razvoja modernog front-enda i ne znaju da je to nekada stvarno tako bilo. Danas moderni front-end teško možemo zamisliti bez front-end arhitekata, Interaction Designera, UX i grafičkih dizajnera, front-end

implementatora, razvojnih inženjera i programera te informacijskih arhitekata, a potrebna su znanja u rasponu od čisto matematičkih i inženjerskih, preko socioloških i psiholoških, do ekonomsko-marketingških. Dakle, vrlo je lako zaključiti da je moderni front-end puno više od sume elemenata razvojnog procesa i jasno je zašto članove procesa zovemo dionicima, a ne jednostavno inženjerima (developerima). Raslojavajući proces izgradnje modernog front-enda na potprocese i pokušavajući ih grupirati na neki smislen način, vrlo se brzo dolazi do zaključka da se okvir koji prati izgradnju front-enda sa svojim fazama podudara s popularnim okvirima poput ITIL-a ili MOF-a. Dobar dio stručnjaka - među koje spadaju projektanti, arhitekti, dizajneri, razvojni inženjeri te ljudi koji vode računa o projektu, kvaliteti i inovacijama - složiti će se da se izgradnja modernog front-enda može pratiti kroz tri glavne faze razvojnog ciklusa: faze dizajna, izgradnje i tranzicije. U fazi dizajna analiziraju se poslovni slučajevi, korisničke potrebe; gradi se informacijska arhitektura; dizajnira se tok aplikacije sa stajališta korisnika i njegove interakcije s uređajem i aplikacijom. Faza izgradnje odnosi se na razvoj fizičkog grafičkog dizajna, njegova prijeloma u nekakav meta jezik razumljiv računalu, kao i razvoj same UI logike. Faza tranzicije podrazumijeva spajanje našeg front-enda s

## Front-end na web principima



Platforma za razvoj modernog front-enda baziranog na webu

nekom pozadinskom logikom, ostalim slojevima aplikacije ili servisima nekih trećih strana. Sve faze kontinuirano se moraju nadgledati, mjeriti i kontrolirati, a sve skupa nadziru uloge koje upravljaju projektom, inovacijom i kvalitetom.

### Kralj je mrtav

Kada se govori o modernom front-end sučelju, odmah na pamet padaju imena iz mobilnog svijeta poput Rdia, Papera, Ultravisuala, Listena, NG City Guidesa, NTC-a, Checkmarka, Circlea ili nekih koji nas tek trebaju osvojiti, poput Shoutemova Loyalty Mobile Appsa s vrlo inovativnim korisničkim sučeljem, no treba imati na umu da je moderni front-end zapravo začeo puno prije pojave mobilnih uređaja kakve imamo danas. Svi koji se sjećaju vremena Shockwavea znaju da je od ranih verzija HTML-a, preko Flasha i danas HTML-a 5, moderno i bogato korisničko sučelje stiglo upravo s weba.

Web je vrlo vjerojatno bio upravo taj ključni čimbenik u razvoju modernog front-enda. Web se, osim kao korisničko sučelje, razvijao i kao medij i kao platforma i kao tehnologija (bez obzira na to radi li se o poslužiteljskoj ili klijentskoj strani), a kroz web su se razvijali i korisnici weba i njihova očekivanja.

Web je razmazio korisnike, utjecao na korisnička očekivanja i izgradio sliku modernog front-enda te upravo zbog toga danas imamo beskompromisne korisnike kojima je često kriterij hoće li neku aplikaciju prihvatiti ili odbaciti upravo korisničko sučelje, bez obzira na stvarnu korisnost same aplikacije.

Sve ovo samo govori da poznata mantra "*content is king*" danas više ne vrijedi. Danas se iza toga da je sadržaj kralj mogu kriti samo loši dizajneri ili ljudi iz prodaje i marketinga koji nisu u doticaju sa stvarnošću - ili su jednostavno zapeli u *enterprise* svijetu, gdje ova fraza još uvijek prolazi.

Za sadržaj (kao i za svakog kralja) vrijedi da njegova snaga ne dolazi iz same činjenice da je on rođen kao kralj, već iz snage njegove vojske, a vojska u slučaju sadržaja na webu upravo je njegova arhitektura, dizajn, izgled i ponašanje.

### Što možemo naučiti od weba?

Web se dugo godina tražio, prilagodio i profilirao (i dalje to čini), no s vremenom se uspješno prilagodio i stvorio konvergenciju između čistog sadržaja i njegova konzumenta, i to upravo kroz navedene čimbenike arhitekture, dizajna i ponašanja sadržaja.

Osim čiste prezentacije, upotrebljivosti i korisničkih očekivanja web je bitno utjecao i na definiranje procesa razvoja modernih korisničkih sučelja te se od weba može mnogo naučiti. Igrači poput Microsofta, Mozille i Googlea to su prepoznali i danas na web principima grade moderna korisnička sučelja za širok spektar uređaja i platformi poput Windows Store Appsa, FirefoxOS-a i ChromeCast aplikacija.

Osnovna je zadaća svakog weba da kao servis služi potencijalnom korisniku. Koliko će taj korisnik biti potencijalan, ovisi i o webu, kvaliteti sadržaja, izgledu, UX-u i pristupačnosti.

Pristupačnost je vrlo vjerojatno jedini pravi (ili barem najveći) razlog tolike webove popularnosti i zašto je internetski preglednik najkorištenija aplikacija bez obzira na platformu i vjerojatno jedina aplikacija koja je instalirana na apsolutno svaki pametni uređaj. Pristupačnost weba ne znači da nam web mora biti raspoloživ i dostupan bilo u kom trenutku bilo s koje lokacije; to je ipak jedna niža tehnička razina koja se danas podrazumijeva.

Pristupačnost podrazumijeva mogućnost konzumacije weba na jednako kvalitetan način bez obzira na platformu i hardversko/softverske karakteristike klijenta te bez obzira na psihofizičke sposobnosti korisnika. U praksi to znači da bi putem našeg weba neki tinejdžer trebao moći kupiti pjesmu putem mobitela, kućanica preko svog osobnog računala, a slijepa osoba putem svog web preglednika specifičnog za slabovidne osobe. To je pristupačnost weba, a pristupačnost bi trebala biti prvi i osnovni postulat koji se može primijeniti bilo na koju platformu, kategoriju aplikacije ili uređaja bez obzira na način interakcije i korištenja aplikacije.

Prije svega, da bi se osigurala ovakva jednostavnost na najnižoj razini, ponekad se treba nečega i odreći. Kod weba je to slučaj s jednostavnom činjenicom da je web medij koji ne pamti stanje (*stateless*), što pak znači da web stranice i komponente web stranica nisu svjesne jedni drugih ni stanja među njima. Iako se to stanje može simulirati i mnogi su proizvođači više ili manje uspješno pronašli načine kako da to postignu, to ipak nije prirodno ponašanje weba. No upravo zahvaljujući ovom, kao i nekim drugim ovakvim jednostavnim KISS (Keep It Simple, Stupid) principima, web i danas ostvaruje osnovni koncept pristupačnosti. Naravno, pristupačnost na ovoj razini u desktop ili mobilnom svijetu često je tehnički neizvediva, no ipak tome treba težiti jer ako je upalilo na webu, u budućnosti bi moglo upaliti i bilo gdje drugdje.

### Separation of Concerns

Moderni principi razvoja softvera, a i inženjerski pristup općenito, nalažu da bi se svaka razvojna platforma trebala raslojiti na nekoliko slojeva i za tako nešto sigurno postoje dobri razlozi, što razvojni inženjeri i arhitekti jako dobro znaju. Isto vrijedi i za razvoj front-enda. Prema web principima, za svaku web stranicu vrijedi da s jedne strane imamo sadržaj i strukturu sadržaja (logički dizajn sadržaja), za što je zadužen HTML, s druge strane imamo fizički dizajn sadržaja koji je određen CSS-om, a sve to povezuje ponašanje našeg sadržaja, što nam osigurava JavaScript. Time su slojevi web stranice jasno određeni i tehnološki jer su HTML, CSS i JavaScript tehnologije koje su potpuno različite, ali međusobno komplementarne; jedna bez druge ne vrijede mnogo, no zajedno čine vrlo moćno oruđe.

Kod bilo koje druge mobilne ili desktop platforme i skupine tehnologija ovakvo jasno raslojavanje korisničkog sučelja trenutno nije moguće i međusobno su sva ova tri sloja vrlo isprepletena te je čest slučaj da razvojni inženjer koji radi na

## MICROSOFT I YAHOO!

Prava je šteta što Microsoft u prošlom desetljeću nije uspio kupiti Yahoo. Ne toliko zbog potencijala ogromne baze korisnika, potencijala elektroničkog poslovanja, proizvoda ili drugih tvrtki koje je Yahoo! akvirirao tijekom godina, koliko zbog činjenice da je u to doba YUI bio vrlo vjerojatno najnaprednija front-end biblioteka za razvoj korisničkih sučelja na webu. Tada je iza YUI-a stajao Yahooov tim na čijem je čelu tada bio Eric Miraglia, koji je YUI doveo na sam vrh i stekao reputaciju najbolje, najdokumentiranije (što je za biblioteku ovih razmjera od neprocjenjive važnosti) i jednostavno najprofesionalnije razvijene biblioteke za razvoj web korisničkih sučelja. Da je kojim slučajem njega i ostatak tima Microsoft tih godina uspio dobiti na svoju stranu, vrlo vjerojatno ne bi potrošio tolike godine na otkrivanje vlastitih putova za razvoj web korisničkih sučelja i klijentskih tehnologija. Treba se samo prisjetiti .NET Ajaxa (kodno mu je ime bilo Atlas), koji je konceptualno bio dobro zamišljen, razvojnim inženjerima donosio je puno slobode



Microsoftov zaokret prema ozbiljnim tehnologijama kao što je HTML 5

i poznati pristup razvoju, no zapravo je bio sve samo ne Ajax. No Microsoft je na kraju ipak uspio; bitno se okrenuo Open Source zajednici, od njih preuzeo neke dobre koncepte, shvatio da ne mora sve sâm i sada imamo odlična korisnička sučelja na Windows Azure portalu, na Office365 seriji alata i naravno na hrpi Windows Store aplikacija baziranih na HTML 5 tehnologiji. Činjenica koliko je lako odustao od Silverlighta na webu također govori o tome koliki je zaokret Microsoft napravio prema ozbiljnim tehnologijama kao što je HTML 5. Također, IE je u najnovijoj inačici u mnogim kategorijama ponovno preglednik broj 1 i realno je očekivati da će tako i ostati jer razvoj, stabilnosti i pouzdanost Windows Store aplikacija ispod haube ovisi upravo o IE-u. Koliko je to bio dobar potez, pokazuje i činjenica da je time Microsoft ubio dvije muhe jednim udarcem i za sada u odnosu na sve ostale velike igrače na najbolji način doveo primjenu web tehnologija u desktop svijet, a za očekivati je da će se uskoro isto dogoditi i s mobilnim svijetom.

razvoju npr. iOS aplikacije gubi 80% vremena na implementaciju i poliranje korisničkog sučelja. Kod weba to nije slučaj; dedikirani web front-end inženjer ne gubi vrijeme na implementaciju i poliranje korisničkog sučelja jer mu je upravo to posao. Problem je zapravo u tome što trenutno tehnološki nije omogućeno da imamo dedikirane front-end inženjere za npr. Android (ili bilo koju drugu platformu osim weba). Najdalje po ovom pitanju došao je Microsoft sa svojim Silverlightom i XAML-om kao metajezikom za definiranje strukture, sadržaja i dizajna korisničkog sučelja. No čini se da to nije dovoljno te su veliki igrači dobrim dijelom zagrizli u front-end svijet primjenjujući web principe i web tehnologije za izgradnju svojih korisničkih sučelja. Najdalje je tu opet otišao Microsoft i danas je, osim na XAML-u i C#-u, Windows Store aplikacije moguće potpuno ravnopravno razvijati na HTML-u 5, što podrazumijeva HTML, CSS i JavaScript, primjenjujući naravno Web SOC principe.

### Semantika

Semantika je svima koji dolaze iz svijeta weba vrlo dobro poznata stvar. Semantika nam na ontološkoj razini omogućava da se razumijemo ili, još bolje, da stroj/računalo razumije što mu želimo reći kako bi shvatio kako treba interpretirati hrpu podataka koje je dobio u obliku HTML koda. HTML 5 je, iako se još doraduje i još će se doradivati, tu vrlo dobro definiran te za svaki element koji možemo koristiti postoji neka semantička oznaka poput odlomaka, strukturalnih elemenata i sl. Iako i na drugim platformama poput iOS-a ili Androida postoje neke naznake semantike za definiranje sadržaja, strukture i sučelja, to nije ni blizu onome što nudi HTML.

Veći je problem što inženjerima koji razvijaju na navedenim platformama nije ni važno koji će konkretni element sučelja koristiti za neki konkretni slučaj, pa se često događa da dva inženjera imaju potpuno različita razmišljanja o tome kako bi strukturalno korisničko sučelje trebalo izgledati i koje elemente koristiti za njegovu izgradnju. Ovdje opet imamo problem što na drugim platformama trenutno nemamo tako izraženu potrebu za dedikiranim front-end razvojnim inženjerima. Slična je situacija i s razvojnim inženjerima koji korisničko sučelje grade na web tehnologijama, a sami ne dolaze iz web svijeta i samim tim ne vide dobre strane semantičkog pristupa strukturiranju sadržaja. Jednom kada se ta slika promijeni, semantika će postati vrlo važan čimbenik jer će, ako ništa drugo, barem osigurati razumijevanje među razvojnim inženjerima.

### Microsoftova uloga

Microsoft je shvatio važnost front-enda pa ne samo da nudi dobre principe razvoja već i sâm razvija dobar front-end, što se može vidjeti u dosljednosti principima primijenjenim na velikom broju ključnih proizvoda od Windowsa 8 na platformi server-desktop, preko Windowsa Phone na mobilnoj platformi, korporativnog weba i Xboxa na platformi *business-entertainment*, do Azure portala kao čisto poslovnog okruženja. Bez obzira na to sviđa li se nekome Microsoftova front-end filozofija, sigurno je da ništa nije prepušteno slučaju.

Na kraju krajeva, ovaj razvojni zaokret nije izveden zbog razvojnih inženjera, već zbog krajnjih korisnika, koji su se na mobilnim platformama dosada naučili na jednostavnost, pristupačnost i vrlo ugodno korisničko iskustvo. 