
Uvod u dinamički veb sadržaj

World wide web je mreža koja se stalno razvija i koja je već daleko napustila svoju koncepciju ranih 1990-ih, kada je stvorena za rešavanje određenog problema. Najsavremeniji eksperimenti u CERN-u (Evropska laboratorija za fiziku čestica, sada najpoznatiji kao operator velikog hadronskog sudarača) proizvodili su neverovatne količine podataka – toliko da su se podaci pokazali nezgrapnima za distribuciju naučnicima koji su učestvovali, a koji su bili rasprostranjeni po celom svetu.

U to vreme, internet je već bio uspostavljen i povezivao je nekoliko stotina hiljada računara, pa je Tim Berners-Lee (član CERN-a) smislio metod navigacije između njih pomoću hiperveza, što je postao poznat kao Hipertext Transfer Protocol ili HTTP. Takođe je stvorio jezik za označavanje pod nazivom Hipertext Markup Language ili HTML. Da bi ih spojio, napisao je prvi veb čitač i veb server.

Danas ove alate uzimamo zdravo za gotovo, ali tada je koncept bio revolucionaran. Korisnici modema su tada imali mogućnosti telefonskog povezivanja od kuće na oglasnu tablu (eng. *bulletin board*) na kojoj su mogli komunicirati i menjati podatke samo s drugim korisnicima te usluge. Shodno tome, morali ste da budete član mnogih sistema oglasnih tabli da biste efikasno elektronski komunicirali sa svojim kolegama i prijateljima.

Ali Berners-Lee je sve to promenio jednim potezom, pa su do sredine 1990-ih postojala tri glavna grafička veb čitača (eng. *browser*) koja su se takmičila za pažnju pet miliona korisnika. Ubrzo je, međutim, postalo očigledno da nešto nedostaje. Da, stranice teksta i grafike sa hipervezama koje vas vode na druge stranice bile su briljantan koncept, ali rezultati nisu odražavali trenutni potencijal računara i interneta da zadovolje posebne potrebe svakog korisnika sa dinamički promenljivim sadržajem. Korišćenje veba je bilo veoma prosto i jednostavno iskustvo, čak iako je postojalo pomeranje teksta i animirani GIF-ovi!

Korpa za kupovinu, pretraživači i društvene mreže jasno su promenile način na koji koristimo veb. U ovom poglavlju ćemo ukratko pogledati različite komponente koje čine veb i softver koji pomaže da njegovo korišćenje postane bogato i dinamično iskustvo.



Potrebno je manje više odmah početi koristiti neke skraćenice. Pokušao sam da ih jasno objasnim pre nego što nastavim, ali ne brinite previše o tome šta predstavljaju ili šta znače ova imena, jer će detalji postati jasni dok čitate.

HTTP i HTML: Osnove Berners-Leea

HTTP je komunikacioni standard koji upravlja zahtevima i odgovorima koji se šalju između čitača koji radi na računaru krajnjeg korisnika i veb servera. Zadatak servera je da prihvati zahtev klijenta i pokuša da na njega odgovori na smislen način, obično posluživanjem tražene veb stranice – zbog čega se koristi izraz *server* (poslužitelj). Prirodni pandan serveru je *klijent* (korisnik), pa se taj izraz primenjuje i na veb čitač i na računar na kojem je čitač.

Između klijenta i servera može postojati nekoliko drugih uređaja, poput rutera, proksija, mrežnih prolaza itd. Oni imaju različite uloge u obezbeđivanju ispravnog prenosa zahteva i odgovora između klijenta i servera. Obično koriste internet za slanje ovih informacija. Neki od ovih uređaja između uređaja mogu takođe pomoći ubrzanju Interneta tako što lokalno skladište stranice ili informacije u pomoćnoj memoriji, takozvanom *kešu* (eng. *cache*), a zatim poslužuju ovaj sadržaj klijentima direktno iz keša, umesto da preuzimaju sve sa izvornog servera.

Veb server obično može da obrađuje više istovremenih veza, a kada ne komunicira sa klijentom, provodi vreme slušajući dolaznu vezu. Kada nešto stigne, server šalje odgovor da potvrdi prijem.

Procedura zahtev/odgovor

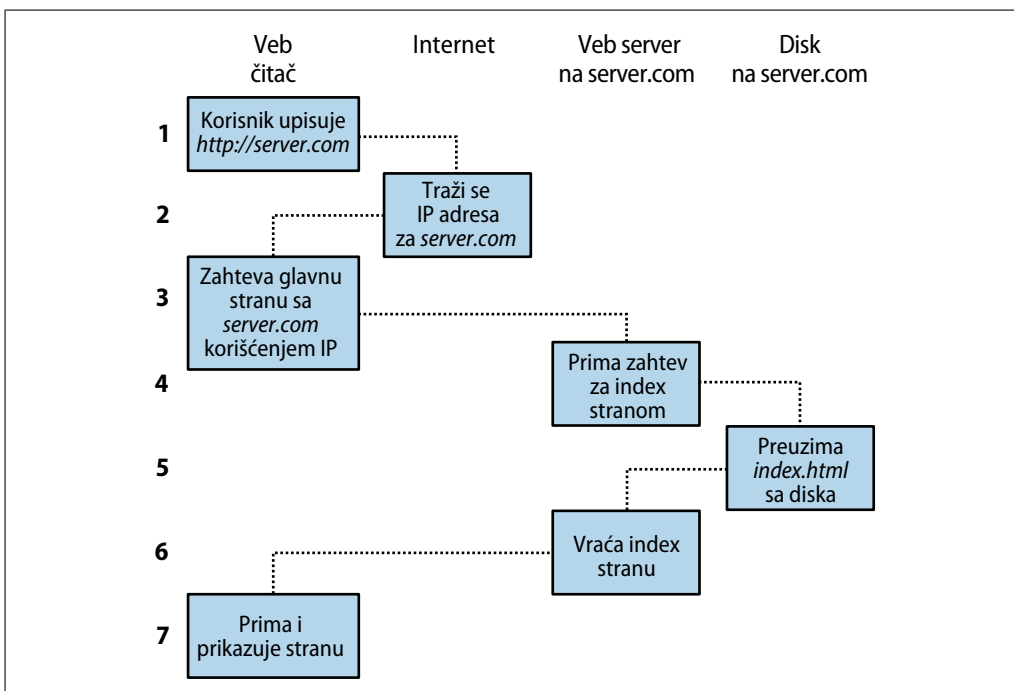
Na svom najosnovnijem nivou, proces zahtev/odgovor (request/response) sastoji se od veb čitača ili drugog klijenta koji traži od veb servera da mu pošalje veb stranicu, a server šalje nazad stranu. Pretraživač se tada brine o prikazivanju ili iscrtavanju stranice (slika 1-1).

Koraci u nizu zahteva i odgovora su sledeći:

1. Unosite *http://server.com* u polje za adresu čitača.
2. Vaš čitača traži adresu Internet protokola (IP) za *server.com*.
3. Vaš čitač šalje zahtev za početnu stranicu na *server.com*.
4. Zahtev prolazi internetom i stiže na *server.com* veb server.
5. Veb server, nakon što je primio zahtev, traži veb stranicu na svom disku.
6. Veb server preuzima stranicu i vraća je čitaču.
7. Vaš čitač prikazuje veb stranicu.

Za prosečnu veb stranicu, ovaj proces se takođe odvija jednom za svaki objekat na stranici: grafiku, ugrađeni video ili Flash datoteku, pa čak i CSS šablon.

U koraku 2, vidimo da čitač traži IP adresu *server.com*. Svaka mašina povezana sa internetom ima IP adresu – i vaš računar – ali generalno pristupamo veb serverima po imenu, kao što je *google.com*. Pretraživač konsultuje dodatnu internetsku uslugu pod nazivom Sistem imena domena (Domain Name System, DNS) kako bi pronašao IP adresu servera, a zatim je koristi za komunikaciju sa računarom.



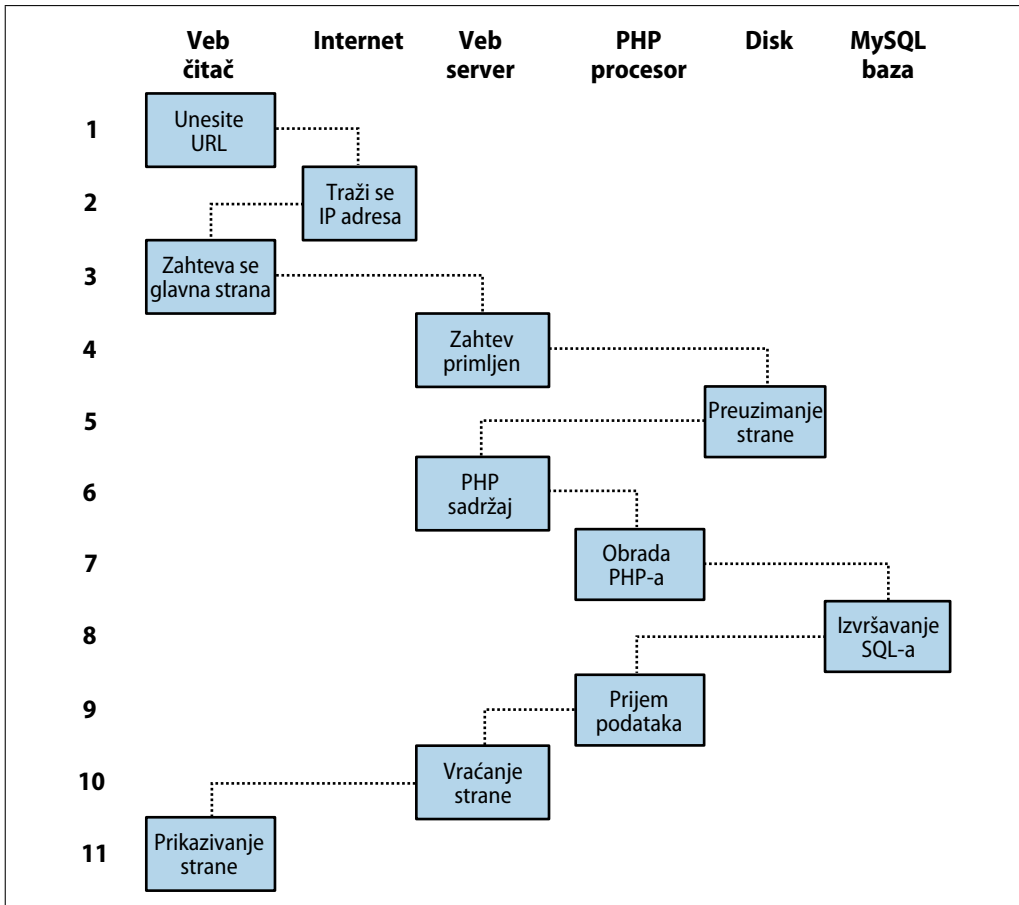
Slika 1-1. Osnovni redosled zahteva/odgovora klijent/servera

Za dinamičke veb stranice, postupak je malo složeniji, jer može uključiti i PHP i MySQL. Na primer, možete kliknuti na sliku kabanice. Zatim će PHP sastaviti zahtev koristeći standardni jezik baze podataka, SQL – čije ćete mnoge komande naučiti u ovoj knjizi – i poslati zahtev na MySQL server. MySQL server će vratiti informacije o kabanici koju ste izabrali, a PHP kôd će sve to umotati u neki HTML koji će server poslati vašem čitaču (slika 1-2).

Koraci su sledeći:

1. Unesite `http://server.com` u polje za adresu čitača.
2. Vaš čitač traži IP adresu za `server.com`.
3. Vaš čitač šalje zahtev na toj adresu za početnu stranicu veb servera.
4. Zahtev prolazi internetom i stiže na `server.com` veb server.
5. Veb server, nakon što je primio zahtev, preuzima početnu stranicu sa svog diska.
6. Pošto je početna stranica sada u memoriji, veb server uočava da je to datoteka koja sadrži PHP kôd i prosleđuje stranicu PHP interpreteru.
7. PHP interpreter izvršava PHP kôd.
8. Neki od PHP-a sadrže SQL izraze, koje PHP interpreteru sada prosleđuje bazi podataka MySQL.
9. MySQL baza podataka vraća rezultate izraza PHP interpreteru.

10. PHP interpreteru vraća rezultate izvršenog PHP koda, zajedno sa rezultatima iz MySQL baze podataka, na web server.
11. Web server vraća stranicu klijentu koji je zatražio, a koji je prikazuje.



Slika 1-2. Dinamički redosled zahteva/odgovora klijent/servera

Iako je korisno biti svestan ovog procesa kako biste znali kako tri elementa funkcionišu zajedno, u praksi se ne morate baviti tim detaljima, jer se svi oni dešavaju automatski.

HTML stranice vraćene čitaču u svakom primeru mogu sadržati JavaScript, koji će klijent lokalno interpretirati i koji bi mogao da pokrene drugi zahtev – na isti način kao i ugrađeni objekti, poput slika.

Prednosti PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5-a

Na početku ovog poglavlja predstavio sam svet veba 1.0, ali nije prošlo mnogo vremena pre nego što se krenulo u pravljenje veba 1.1, sa razvoj poboljšanja čitača kao što su Java, JavaScript, JScript (Microsoftova lakša varijanta JavaScripta) i ActiveX. Na strani servera napredovalo se na Common Gateway Interface (CGI) upotrebi skript jezika kao što su Perl (alternativa PHP jeziku) i *skriptama na strani servera* – ubacivanjem sadržaj jedne datoteke (ili izlaza pokretanja lokalnog programa) u drugu dinamički.

Kada se prašina slegla, tri glavne tehnologije stajale su rame uz rame iznad ostalih. Iako je Perl još uvek bio popularan skript jezik sa vernim sledbenicima, PHP-ova jednostavnost i ugrađene veze do programa MySQL baze podataka skupile su više nego dvostruko više korisnika. A JavaScript, koji je postao suštinski deo jednačine za dinamičko manipulisanje kaskadnim tablicama stila (Cascading Style Sheets, CSS) i HTML, preuzeli su još teže zadatke upravljanja klijentskom stranom asinhronone komunikacije (razmena podataka između klijenta i servera nakon što se veb stranica učitala). Koristeći asinhronu komunikaciju, veb stranice obavljaju rukovanje podacima i šalju zahteve veb serverima u pozadini – a da veb korisnik nije svestan da se to dešava.

Nema sumnje da je simbiotska priroda PHP-a i MySQL-a pomogla i jednom i drugom da napreduje, ali šta ih je uopšte privuklo programere njima? Jednostavan odgovor mora biti laškoća s kojom ih možete koristiti za brzo pravljenje dinamičkih elemenata na veb stranicama. MySQL je brz i moćan, ali jednostavan za korišćenje sistema baze podataka koji nudi gotovo sve što bi jednoj veb lokaciji bilo potrebno da pronade i servira podatke čitačima. Kada se PHP udruži sa MySQL-om radi skladištenja i preuzimanja ovih podataka, imate osnovne delove potrebne za početak Web 2.0.

A kada u mešavinu unesete i JavaScript i CSS, imate recept za izgradnju visoko dinamičnih i interaktivnih veb stranica – pogotovo jer sada postoji širok spektar sofisticiranih radnih okvira (eng. *framework*) JavaScript funkcija koje možete pozvati da biste zaista ubrzali veb razvoj. Ovo uključuje dobro poznati jQuery, koji je do nedavno bio jedan od najčešćih načina na koji programeri pristupaju asinhronim komunikacionim funkcijama i noviju biblioteku React JavaScript, koja postaje sve popularnija. To je sada jedan od najčešće preuzimanih i implementiranih radnih okvira, toliko da od 2020. godine oglasi za posao zaista sadrže dvostruko više potražnje za React nego za jQuery programerima.

MariaDB: Klon MySQL-a

Nakon što je Oracle kupio Sun Microsystems (vlasnika MySQL-a), zajednica je postala zabrinuta da MySQL možda neće ostati u potpunosti otvorenog koda, pa se MariaDB od nje izdvojila kako bi bila besplatna pod GNU GPL licencom. Razvoj MariaDB-a vode neki od originalnih programera MySQL-a tako da zadržava izuzetno blisku kompatibilnost sa MySQL-om. Stoga ćete možda naići na MariaDB na nekim serverima umesto MySQL-a – ali ne brinite, sve u ovoj knjizi radi podjednako dobro i na MySQL-u i na MariaDB-u. Za sve namene, možete zamijeniti jedno s drugim i nećete primetiti nikakvu razliku.

U svakom slučaju, ispostavilo se da se čini da su mnogi početni strahovi ublaženi jer MySQL ostaje otvorenog koda, pri čemu Oracle jednostavno naplaćuje podršku i izdanja koja pružaju dodatne funkcije, poput geo replikacije i automatskog skaliranja. Međutim, za razliku od MariaDB-a, MySQL više ne vodi zajednica, pa činjenica da će MariaDB uvek biti tu, ako ikada zatreba, pruža programerima miran san i verovatno obezbeđuje da će i sam MySQL ostati otvorenog koda.

Korišćenje PHP-a

Uz PHP, jednostavno je ugraditi dinamičke aktivnosti u veb stranice. Kada stranicama date proširenje *.php*, oni imaju trenutni pristup skrip jeziku. Sa tačke gledišta programera, sve što treba da uradite je da napišete kôd, poput sledećeg:

```
<?php
echo " Danas je " . date("l") . ". ";
?>
Evo najnovije vesti.
```

Početni `<?php` govori veb serveru da dozvoli PHP programu da interpretira sve sledeće kodove do oznake `?>`. Van njih, sve se šalje klijentu kao direktan HTML. Dakle, tekst *Evo najnovije vesti.* se jednostavno šalje u čitač; unutar PHP oznaka, ugrađena funkcija `date` prikazuje tekući dan u nedelji prema sistemskom vremenu servera.

Danas je Wednesday. Evo najnovije vesti.

PHP je fleksibilan jezik i neki ljudi radije postavljaju PHP graničnike direktno uz PHP kôd, na sledeći način:

```
Danas je <?php echo date("l"); ?>. Evo najnovije vesti.
```

Postoji više načina formatiranja i prikazivanja informacija, koje ću objasniti u poglavljima o PHP-u. Poenta je u tome što sa PHP-om veb programeri imaju skript jezik koji je, iako ne tako brz kao kada pišete vaš kod na C-u ili sličnom jeziku, neverovatno je brz i takođe se besprekorno integriše sa HTML oznakama.



Ako nameravate da unesete primere PHP-a iz ove knjige u vaš editor programa da radite sa mnom, morate zapamtiti da dodate `<?php` ispred `?>` nakon unetog koda kako bi se obezbedilo da ih tumač PHP obradi. Da biste to olakšali, možda ćete želeći da pripremite datoteku pod imenom *example.php* sa tim oznakama.

Koristeći PHP, imate neograničenu kontrolu nad svojim veb serverom. Bilo da treba da menjate HTML u hodu, obrađujete kreditnu karticu, dodajete podatke o korisniku u bazu podataka ili preuzimate informacije sa veb lokacije treće strane, sve to možete učiniti iz istih PHP datoteka u kojima se nalazi i sam HTML.

Korišćenje MySQL-a

Naravno, nema mnogo smisla dinamički menjati HTML izlaz, a da nemate način da pratite informacije koje korisnici pružaju vašoj veb lokaciji dok je koriste. U ranim danima veba,

mnoge veb lokacije su koristile „ravne” tekstualne datoteke za skladištenje podataka kao što su korisnička imena i lozinke. Ali ovaj pristup može da izazove probleme ako datoteka nije pravilno zaključana, jer može doći do njenog oštećenja iz više istovremenih pristupa. Takođe, ravna datoteka može postati toliko velika pre nego što postane nezgrapna za korišćenje – da ne govorimo o teškoćama spajanja datoteka i obavljanja složenih pretraživanja u bilo kom razumnom roku.

Na tom mestu relacije baze podataka sa strukturisanim upitima su postale bitne. MySQL, besplatan za upotrebu i instaliran na velikom broju internet veb servera, odlično je rastao. To je robustan i izuzetno brz sistem za upravljanje bazom podataka koji koristi komande nalik engleskom.

Najviši nivo MySQL strukture je baza podataka, u okviru koje možete imati jednu ili više tabela koje sadrže vaše podatke. Na primer, pretpostavimo da radite na tabeli pod nazivom *korisnici*, u okviru koje ste napravili kolone za *prezime*, *ime* i *e-poštu*, i sada želite da dodate još jednog korisnika. Jedna od naredbi koju možete koristiti za ovo je sledeća:

```
INSERT INTO korisnici VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Ranije ste izdali druge komande za pravljenje baze podataka i tabele i za postavljanje svih ispravnih polja, ali SQL INSERT naredba ovde pokazuje koliko jednostavno može biti dodavanje novih podataka u bazu podataka. SQL je jezik dizajniran ranih 1970-ih koji podseća na jedan od najstarijih programskih jezika, COBOL. Međutim, dobro je prilagođen za upite u bazu podataka, zbog čega se i dalje koristi nakon toliko vremena.

Podjednako je lako tražiti podatke. Pretpostavimo da imate adresu e-pošte korisnika i da morate potražiti ime te osobe. Da biste to uradili, mogli biste da postavite MySQL upit, kao što je sledeći:

```
SELECT prezime,ime FROM korisnici WHERE email='jsmith@mysite.com';
```

MySQL će tada vratiti `Smith John` i bilo koji drugi par imena koji može biti povezan sa tom adresom e-pošte u bazi podataka.

Kao što ste očekivali, postoji mnogo više toga što možete da uradite sa MySQL-om od običnog INSERT (umetni) i SELECT (izbaeri) komande. Na primer, možete kombinovati povezane skupove podataka da biste spojili povezane informacije, tražiti rezultate u raznim narudžbama, napraviti delimična podudaranja kada znate samo deo niza koji tražite, vratiti samo *n* rezultata, i još mnogo toga.

Koristeći PHP, sve ove pozive možete da upućujete direktno u MySQL, a da ne morate sami da pristupate interfejsu komandne linije MySQL. To znači da možete sačuvati rezultate u nizovima (eng. *arrays*) za obradu i izvršiti više pretraživanja, od kojih svako zavisi od rezultata vraćenih iz prethodnih pretraživanja, da biste „iskopali” podatak koja vam je potreban.

Za još veću snagu, kao što ćete videti kasnije, postoje dodatne funkcije ugrađene direktno u MySQL koje možete pozvati da biste efikasno izvršavali uobičajene operacije unutar samog MySQL-a, umesto da ih pravite sa više PHP poziva u MySQL.

Korišćenje JavaScript-a

JavaScript je napravljen da omogući skript pristup svim elementima HTML dokumenta. Drugim rečima, pruža način za dinamičku interakciju korisnika, poput provere valjanosti adrese e-pošte u obrascima za unos i prikazivanja upita poput „Da li ste to zaista mislili?” (iako se na to ne može osloniti zbog bezbednosti, i uvek treba izvoditi na veb serveru).

U kombinaciji sa CSS-om (pogledajte sledeći odeljak), JavaScript je moć koja se nalazi iza dinamičkih veb stranica koje se menjaju pred vašim očima, a ne samo kada server vrati novu stranicu.

Međutim, JavaScript je nekada bio nezgodan za upotrebu, zbog nekih velikih razlika u načinima na koje su različiti dizajneri veb čitača izabrali njegovu implementaciju. Do toga je uglavnom došlo kada su neki proizvođači pokušali da dodaju sopstvene funkcionalnosti svojim čitačima na račun kompatibilnosti sa svojim rivalima.

Na sreću, programeri su većinom došli k sebi i shvatili potrebu za potpunom međusobnom kompatibilnošću, pa je ovh dana manje potrebno optimizovati vaš kôd za različite čitače. Međutim, milioni korisnika i dalje koriste stare čitače, a to će verovatno biti slučaj još mnogo godina. Srećom, postoje rešenja za probleme nekompatibilnosti, a kasnije ćemo u ovoj knjizi pogledati biblioteke i tehnike koje vam omogućavaju da bezbedno zanemarite ove razlike.

Za sada, pogledajmo kako se koristi osnovni JavaScript, prihvaćen od svih čitača:

```
<script type="text/javascript">
  document.write("Danas je " + Date() );
</script>
```

Ovaj deo koda govori veb čitaču da interpretira sve unutar `<script>` oznaka kao JavaScript, ono što čitač radi je ispisivanje teksta `Danas je` u tekući dokument, zajedno sa datumom, koristeći JavaScript funkciju `Date`. Rezultat će izgledati otprilike ovako:

Danas je Wed Jan 01 2025 01:23:45



Osim ako ne morate da navedete tačnu verziju JavaScripta, možete normalno izostaviti `type="text/javascript"` i koristite samo `<script>` da bi započelo interpretiranje JavaScripta.

Kao što je ranije pomenuto, JavaScript je prvobitno razvijen da nudi dinamičku kontrolu nad različitim elementima u HTML dokumentu, i to je i dalje njegova glavna upotreba. Ali sve više, JavaScript se koristi za *Ajax*, proces pristupa veb serveru u pozadini.

Asinhrona komunikacija je ono što omogućava veb stranicama da počnu da liče na samostalne programe, jer se ne moraju ponovo učitavati u celini da bi se prikazao novi sadržaj. Umesto toga, asinhroni poziv može ažurirati jedan element na veb stranici, poput promene fotografije na društvenoj mreži ili zamene dugmeta na koje kliknete odgovorom na pitanje. Ova tema je u potpunosti obrađena u poglavlju 18.

U poglavlju 22 detaljno obrađujemo jQuery, koji možete koristiti za uštedu ponovnog otkrivanja točka kada vam je potreban brz, kôd za prikaz i manipulaciju vašim veb stranicama. Naravno, dostupni su i drugi paketi, pa smo u poglavlju 24 obradili i React, jedan od najpopularnijih izbora današnjice. Oba su izuzetno pouzdani i glavni alati u kompletu alata mnogih iskusnih veb programera.

Korišćenje CSS-a

CSS je ključni pratilac HTML-a, koji obezbeđuje da su HTML tekst i umetnute slike postavljene dosledno i na način koji odgovara ekranu korisnika. Sa pojavom standarda CSS3 poslednjih godina, CSS sada nudi nivo dinamičke interaktivnosti koji je ranije podržavao samo JavaScript. Na primer, ne samo da možete stilizovati bilo koji HTML element, da promenite njegove dimenzije, boje, okvir, razmake itd, već sada možete na veb stranici dodati i animirane prelaze (eng. *transitions*) i transformacije, koristeći samo nekoliko redova CSS-a.

Korišćenje CSS-a može biti jednostavno kao umetanje nekoliko pravila (komandi) između `<style>` i `</style>` oznaka u zaglavlju veb stranice, na primer:

```
<style>
  p {
    text-align: justify;
    font-family: Helvetica;
  }
</style>
```

Ova pravila menjaju osnovno poravnanje teksta, tako da pasusi koji sadrže `<p>` postaju obostrano poravnati (eng. *justify*) i koriste font Helvetica.

Kao što ćete saznati u poglavlju 19, postoji mnogo različitih načina na koje možete upotrebiti CSS pravila, a takođe ih možete uključiti direktno u HTML oznake ili sačuvati skup pravila u spoljnu datoteku i da ih učitavate. Ova fleksibilnost ne samo da vam omogućava da precizno oblikujete svoj HTML, već može (na primer) da obezbedi ugrađenu funkciju lebdenja (eng. *hover*) za animaciju objekata dok mišem prelazite preko njih. Takođe ćete saznati kako da pristupite svim CSS svojstvima elementa iz JavaScripta, kao i iz HTML-a.

I evo ga HTML5

Koliko god da su bili korisni svi ovi dodaci veb standardima, nisu bili dovoljni za rastući broj ambicioznih programera. Na primer, još uvek nije postojao jednostavan način za manipulisanje grafikom u veb čitaču bez pribegavanja dodacima kao što je Flash. Isto se odnosi i na umetanje audio i video zapisa na veb stranice. Osim toga, nekoliko dosadnih nedoslednosti uvuklo se u HTML tokom njegove evolucije.

Dakle, da bi se sve ovo račistilo i internet otišao dalje od Weba 2.0 i u njegovu sledeću iteraciju, stvoren je novi standard za HTML koji će otkloniti sve ove nedostatke: *HTML5*. Njegov razvoj započeo je 2004. godine, kada su prvi nacrt sastavile Mozilla Foundation i Opera

Softvare (programeri dva popularna veb čitača), ali je tek 2013. konačni nacrt dostavljen u World Wide Web Consortium (W3C), međunarodno upravljачko telo za veb standarde.

Bilo je potrebno nekoliko godina da se HTML5 razvije, ali sada smo na vrlo čvrstoj i stabilnoj verziji 5.1 (od 2016.). To je, međutim, beskonačan razvojni ciklus, a vremenom će se u njega ugraditi više funkcija, s verzijom 5.2 koja je objavljena kao preporuka W3C 2017., HTML 5.3 (sa predloženim funkcijama, kao što je automatsko pisanje velikih slova) koje se planira još od 2020. itd. Neke od najboljih funkcija u HTML5 za rukovanje i prikazivanje medija uključuju `<audio>`, `<video>` i `<canvas>` elemente koji dodaju zvuk, video i naprednu grafiku. Sve što treba da znate o ovim i svim drugim aspektima HTML5 detaljno je obrađeno počevši od poglavlja 25.



Jedna od sitnica koje mi se sviđa u HTML5 je specifikacija da XHTML sintaksa više nije potrebna za elemente koji se sami zatvaraju. Ranije ste mogli da prikazete prelom redova koristeći `
` element. Zatim, kako bi se osigurala buduća kompatibilnost s XHTML-om (planirana zamena za HTML koja se nikada nije dogodila), ovo je promijenjeno u `
`, u kojem je završni `/` znak dodat (pošto se očekivalo da svi elementi sadrže završnu oznaku sa ovim znakom). Ali sada su stvari rešene i možete koristiti bilo koju verziju ovih vrsta elemenata. Dakle, radi sažetosti i manje pritiska na tastere, u ovoj knjizi sam se vratio na prethodni stil `
`, `<hr>` itd.

Apache veb server

Pored PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5, postoji i šesti junak na dinamičkom vebu: veb server. U slučaju ove knjige, to znači Apache veb server. Malo smo objasnili šta veb server radi tokom razmene HTTP server/klijent, ali iza scene radi mnogo više.

Na primer, Apache ne posluđuje (servira) samo HTML datoteke – već rukuje širokim spektrom datoteka, od slika i Flash datoteka do MP3 audio datoteka, RSS (Really Simple Syndication) izvorima itd. I ti objekti ne moraju biti statičke datoteke poput GIF slika. Svi oni mogu biti generisani programima poput PHP skripti. Tako je: PHP može čak i da stvara slike i druge datoteke za vas, bilo u hodu ili unapred da bi se kasnije upotrebile.

Da biste to uradili, obično imate module koji su unapred kompajlirani u Apacheu ili PHP-u ili se pozivaju tokom izvršavanja. Jedan takav modul je biblioteka GD (Graphics Draw), koju PHP koristi za izradu i rukovanja grafikom.

Apache takođe podržava veliki broj sopstvenih modula. Pored PHP modula, najvažniji za vaše potrebe veb programera su moduli koji se bave bezbednošću. Drugi primeri su modul Rewrite (Prepisivanje), koji veb serveru omogućava rukovanje nizom URL adresa i njihovo prepisivanje prema sopstvenim internim zahtevima, i Proxy modul, koji možete koristiti za posluživanje često zahtevanih stranica iz keša kako biste smanjili opterećenje na serveru.

Kasnije u knjizi videćete kako da koristite neke od ovih modula za poboljšanje funkcija koje pružaju tri osnovne tehnologije.

Rukovanje mobilnim uređajima

Sada smo čvrsto u svetu međusobno povezanih mobilnih računarskih uređaja, a koncept razvoja veb stranica samo za desktop računare je zastareo. Umesto toga, programeri sada imaju za cilj da razviju responzivne veb lokacije i veb aplikacije koje se prilagođavaju okruženju u kojem se nalaze.

Novo u ovom izdanju je da pokazujem kako možete lako da stvarate ove vrste proizvoda koristeći samo tehnologije opisane u ovoj knjizi, zajedno sa moćnom jQuery Mobile bibliotekom odgovarajućih JavaScript funkcija. Pomoću nje ćete moći da se fokusirate na sadržaj i upotrebljivost vaših veb lokacija i veb aplikacija, znajući da će način na koji se prikazuju biti automatski optimizovan za niz različitih računarskih uređaja – jedna stvar manje o kojoj morate brinuti.

Da bi demonstrirao kako u potpunosti iskoristiti raspoloživu moć, u poslednjem poglavlju ove knjige pravimo, kao primer, jednostavnu veb lokaciju društvene mreže, koristeći jQuery kako bi bila potpuno responzivna (odzivna) i obezbedila da se dobro prikazuje na bilo čemu, od malog ekrana mobilnog telefona do tablet ili desktop računar. Mogli smo podjednako da koristimo React (ili druge JavaScript biblioteke ili radne okvire), ali možda je to vežba koju ćete želeti da sami prođete kada završite izučavanje ove knjige.

0 otvorenom kodu

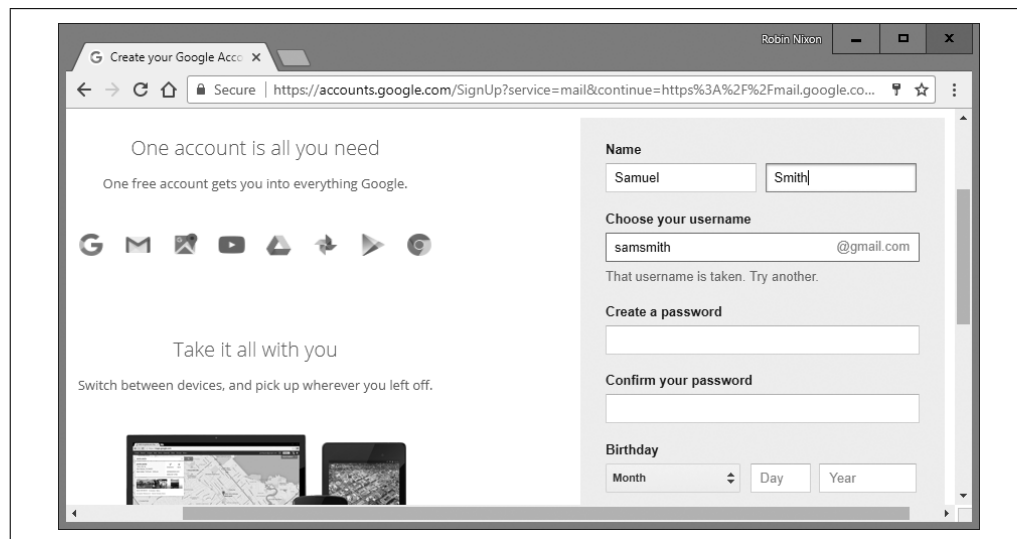
Tehnologije u ovoj knjizi su otvorenog koda: svakome je dozvoljeno da čita i menja kôd. Često se raspravljalo o tome da li je ovaj status razlog zašto su ove tehnologije toliko popularne, ali PHP, MySQL i Apache *jesu* tri najčešće korišćena alata u svojim kategorijama. Ono što se definitivno može reći je da njihov otvoreni kôd znači da ih je razvila zajednica programera koji su napisali funkcije koje su im potrebne, sa originalnim kodom koji je dostupan svima, da svako može da vidi i promeni kôd. Greške se mogu brzo pronaći, a bezbednosni propusti mogu se sprečiti pre nego što se dogode.

Postoji još jedna prednost: svi ovi programi su besplatni za upotrebu. Nema brige oko kupovine dodatnih licenci ako morate da povećate svoju veb lokaciju i dodate više servera, a ne morate da proveravate budžet pre nego što odlučite da li ćete se nadograditi na najnovije verzije ovih proizvoda.

Sve to zajedno

Prava lepota PHP-a, MySQL-a, JavaScripta (ponekad uz pomoć Reacta ili drugih biblioteka), CSS-a i HTML5 je divan način na koji svi zajedno rade na stvaranju dinamičkog veb sadržaja: PHP obavlja sav glavni posao na veb serveru, MySQL upravlja svim podacima, a kombinacija CSS-a i JavaScripta brine o prezentaciji veb stranice. JavaScript takođe može komunicirati sa vašim PHP kodom na veb serveru kad god treba nešto ažurirati (bilo na serveru ili na veb stranici). A pomoću moćnih funkcija u HTML5, kao što su canvas (platno), audio i video i geolokacija, možete učiniti svoje veb stranice visoko dinamičnim, interaktivnim i multimedijjskim.

Bez korišćenja programskog koda, rezimirajmo sadržaj ovog poglavlja posmatrajući proces kombinovanja nekih od ovih tehnologija u svakodnevnu funkciju asinhronne komunikacije koju koriste mnoge veb lokacije: provera da li željeno korisničko ime već postoji na veb lokaciji kada se korisnik prijavljuje da otvori novi nalog. Dobar primer ovoga može se videti sa Gmailom (slika 1-3).



Slika 1-3. Gmail koristi asinhronu komunikaciju da proverí da li je slobodno korisničko ime

Koraci koji čine ovaj asinhroni proces biće slični sledećim:

1. Server šalje HTML za pravljenje veb obrasca, a obrazac traži potrebne detalje, kao što su korisničko ime, ime, prezime i adresa e-pošte.
2. Istovremeno, server dodaje JavaScript HTML-u za nadgledanje polja za unos korisničkog imena i proveru dve stvari: da li je unesen tekst u polje i da li je korisnik napustio polje za unos jer je prešao na drugo polje za unos.
3. Nakon što je tekst unet i polje napušteno, JavaScript kôd u pozadini prosleđuje korisničko ime koje je uneto, nazad u PHP skriptu na veb serveru i čeka odgovor.
4. Veb server traži korisničko ime i odgovara JavaScriptu da li je to ime već zauzeto.
5. JavaScript zatim postavlja oznaku pored polja za unos korisničkog imena kako bi pokazao da li je ime dostupno korisniku – možda znak zelene štikle ili crvenog krstića, zajedno sa nekim tekstom.
6. Ako korisničko ime nije dostupno, a korisnik pošalje obrazac, JavaScript prekida slanje i ponovo naglašava (možda sa jačom grafikom i/ili okvirom upozorenja) da korisnik mora izabrati drugo korisničko ime.
7. Opciono, poboljšana verzija ovog procesa bi čak mogla na osnovu korisničkog imena koje je zatražio korisnik, da korisniku predloži alternativna imena koja su trenutno dostupna.

Sve ovo se odvija tiho u pozadini i omogućava ugodno i besprekorno korisničko iskustvo. Bez asinhronne komunikacije, ceo obrazac bi morao da se dostavi serveru, koji bi zatim vratio HTML, kojim bi se potom iscrtala cela stranica sa naznačenim delovima. To bi bilo upotrebljivo rešenje, ali ni izbliza tako uredno i prijatno kao obrada polja u letu (eng. *on-the-fly*).

Asinhrona komunikacija se može koristiti za mnogo više od jednostavne provere i obrade unosa; kasnije u ovoj knjizi ćemo istražiti mnoge dodatne stvari koje možete učiniti sa njom.

U ovom poglavlju ste pročitali solidan uvod u osnovne tehnologije PHP-a, MySQL-a, JavaScripta, CSS-a i HTML5 (kao i Apachea) i naučili kako zajedno rade. U poglavlju 2, pogledaćemo kako možete da instalirate sopstveni server za veb razvoj na kome ćete vežbati sve ono što ćete učiti.

Pitanja

1. Koje četiri komponente (minimum) su potrebne za izradu potpuno dinamičke veb stranice?
2. Šta znači *HTML*?
3. Zašto naziv *MySQL* sadrži slova *SQL*?
4. *PHP* i *JavaScript* su dva programska jezika koja generišu dinamičke rezultate za veb stranice. Koja je njihova glavna razlika i zašto biste koristili oba?
5. Šta znači *CSS*?
6. Navedite tri glavna nova elementa uvedena u *HTML5*.
7. Ako naidete na grešku (što je retko) u nekom od alata otvorenog koda, šta mislite kako biste mogli da je rešite?
8. Zašto su radni okviri kao što su *jQuery* ili *React* toliko važni za razvoj savremenih veb lokacija i veb aplikacija?

Za odgovore na ova pitanja pogledajte „Poglavlje 1: odgovori” u dodatku A.

