Profesionalni JavaScript

Unapredite svoju karijeru veb developera moćnim alatima naprednog JavaScript-a

Hugo Di Francesco

Siyuan Gao

Vinicius Isola

Philip Kirkbride

CET

Računarski fakultet

Profesionalni JavaScript

Hugo Di Francesco

Siyuan Gao

Vinicius Isola

Philip Kirkbride

ISBN 978-86-7991-429-3

Autorizovan prevod sa engleskog jezika knjige Professional JavaScript

Prevod	Petar Prvulović
Recenzent	Nemanja Radosevljević
Urednik	Jovana Ristić
Lektor	Marina Đenadić
Prelom	Zoran Stanković
Izdavači	CET Computer Equipment and Trade Beograd, Skadarska 45 tel/fax: 011 3243-043, 3235-139, 3237-246 www.cet.rs
	Računarski fakultet Beograd, Knez Mihailova 6/VI tel: 011 2627-613, 2633-321 www.raf.edu.rs
Za izdavača	Dragan Stojanović, direktor
Tiraž	1000
Štampa	„Pekograf", Beograd

Nastavno-naučno veće Računarskog fakulteta na 142. elektronskoj sednici održanoj 27. 4. 2020. godine donelo je odluku da knjiga „*Profesionalni JavaScript*" od autora Hugo Di Francesco, Siyuan Gao, Vinicius Isola I Philip Kirkbride bude štampana kao univerzitetski udžbenik.

SADRŽAJ

Predgovor

O poglavlju

Ovo poglavlje ukratko predstavlja autore, kao i teme koje knjiga pokriva, kao i tehničke veštine koje su vam potrebne kako biste mogli da počnete sa učenjem i hardver-a i softver-a, koji su potrebni kako bi se kompletirale sve date aktivnosti i vežbe.

O knjizi

Duboko razumevanje JavaScript-a olakšava učenje celog spektra drugih razvojnih okruženja, uključujući React, Angular i njima bliske alate i biblioteke. Ova knjiga dizajnirana je tako da vam pomogne u savladavanju suštinskih JavaScript koncepata, potrebnih za izgranju modernih aplikacija.

Počećete sa učenjem kako da predstavite HTML dokument u Document Object Modelu (DOM). Zatim ćete kombinovati svoje poznavanje DOM-a i Node.js, kako biste kreirali veb skrejper za neke praktične situacije. Kako budete napredovali kroz poglavlja, kreiraćete Node.js RESTful API pomoću biblioteke Express za Node.js. Razumećete i kako se modularni dizajn može koristiti, kako bi se postigla bolja višestruka upotrebljivost koda i kolaboracija više programera na jednom projektu. Kasnija poglavlja će vas voditi kroz kreiranje jediničnih testova, kojima se utvrđuje da se osnove funkcionalnosti programa vremenom ne pokvare. U knizi se demonstrira i kako konstruktori, async/await i događaji mogu da postignu visoke performanse i brzinu aplikacije. Na kraju ćete dobiti koristan uvod u koncepte funkcionalnog programiranja, kao što su imutabilnost, čiste funkcije i funkcije višeg reda.

Do kraja ove knjige steći ćete veštine koje su vam potrebne da se uhvatite u koštac sa stvarnim problemima tokom JavaScript programiranja pomoću modernog pristupa JavaScript-u, kako na klijentskoj, tako i na serverskoj strani.

O autorima

Hugo Di Frančesko je softverski inženjer koji je dosta radio sa JavaScript-om. Ima zvanje mastera matematičkih izračunavanja na University College London (UCL). Koristio je JavaScript kroz celu vertikalu, kako bi kreirao skalabilne platforme visokih performansi u kompanijama kao što su Canon i Elsevier. Trenutno se bavi problemima u domenu maloprodaje sa Node.js, React i Kubernetes dok istovremeno vodi i istoimeni sajt „Kodirajte sa Hugom". Pored posla je i internacionalno priznati mačevalac, koji trenira i takmiči se po celom svetu.

Siiuan Gao je softverski inženjer u kompaniji Electronic Arts. Diplomirao je računarske nauke na Univerzitetu Purdue. JavaScript i Node.js koristi više od 4 godine, uglavnom u izradi efikasnih pozadinskih rešenja za sisteme visokog nivoa dostupnosti. Takođe je saradnik Node.js Core projekta i objavio je mnogo npm modula. U slobodno vreme uživa u učenju o dizajnu video igara i mašinskom učenju.

Vinicius Isola počeo je programirati još 1999. godine, koristeći Macromedia Flash i ActionScript. Java sertifikacijom se bavi od 2005. godine i specijalizovao se za izgradnju veb i poslovnih aplikacija. Radeći u svim delovima vertikale projektovanja, JavaScript i veb tehnologije su uvek bili prisutni u njegovim mnogobrojnim poslovima i kompanijama za koje je radio. U slobodno vreme voli da radi na projektima otvorenog koda i da se bavi mentorstvom novih programera.

Philip Kirkbrajd ima preko 5 godina iskustva sa JavaScript-om i baziran je u Montrealu. Završio je tehnički fakultet 2011. godine i od tada se bavi veb tehnologijama u različitim ulogama. Radio je sa 2Klic, IoT kompanijom, koju je Convectair, glavna kompanija za grejanje na struju, unajmila, kako bi projektovali pametne grejače koje pokreće Z-Wave tehnologija. Njegov posao sastojao se u pisanju mikro servisa u Node.js i Bash. Takođe je imao priliku da doprinese projektima otvorenog koda SteemIt (platforma za blogovanje zasnovana na blockchain-u) i DuckDuckGo (pretraživač koji se zasniva na privatnosti).

Ciljevi učenja

Do kraja ove knjige moći ćete da:

- Primenite osnovne koncepte funkcionalnog programiranja.
- Napravite Node.js projekat koji koristi biblioteku Express.js i sadrži API.
- Napravite jedinične testove za Node.js projekat, kako biste proverili njegovu ispravnost.
- Upotrebite biblioteku Cheerio sa Node.js, kako biste napravili jednostavni veb skrejper.
- Razvijete React interfejs kojim možete izraditi tokove obrade.
- Koristite funkcije povratnih poziva, kao osnovni način da vratite kontrolu.

Publika

Ako želite da napredujete od frontend programera do full-stack programera i naučite kako Node.js može da se koristi za pokretanje full-stack aplikacija, ovo je idealna knjiga za vas. Nakon čitanja ove knjige, moći ćete da napišete bolji JavaScript kôd i naučite o najnovijim trendovima jezika. Da biste lako shvatili ovde opisane koncepte, treba da znate osnovnu JavaScript sintaksu i trebalo bi da ste već radili sa popularnim bibliotekama, kao što je jQuery. Trebalo bi i da ste koristili JavaScript sa HTML i CSS, ali ne nužno i Node.js.

Pristup

Svaki odeljak ove knjige posebno je dizajniran da vas angažuje i stimuliše tako da ono što naučite možete zapamtiti i primeniti u praksi na najefektivniji mogući način. Naučićete kako da se nosite sa intelektualno stimulativnim izazovima programiranja, koji će vas kroz funkcionalno programiranje i razvoj vođen testiranjem pripremiti za realne teme. Svako poglavlje je pažljivo dizajnirano, tako da se nadograđuje na JavaScript kao osnovni jezik.

Hardverski zahtevi

Za optimalno iskustvo preporučujemo sledeću konfiguraciju

- Procesor: Intel Core i5 ili ekvivalentan;
- Memorija: 4 GB RAM
- Disk: 5 GB slobodnog prostora.

Softverski zahtevi

Preporučujemo i da imate spreman i instaliran sledeći softver:

- Git, najnovija verzija;
- Node.js 10.16.3 LTS (https://nodejs.org/en/)

Konvencije

Kodne reči u tekstu, imena tabela u bazi podataka, nazivi foldera, fajlova, ekstenzija fajlova, putanje, primeri URL-a, korisnički input i Twitter kopče su formatirani na sledeći način:

„ES6 funkcija **`import`** omogućava vam i da učitate podsekciju modula, umesto da učitavate sve. Ovo je jedna od prednosti koje ES6 import ima u odnosu na Node.js funkciju **`require`**. SUSE"

Blok koda formatiran je ovako:

```
let myString = “hello”;
console.log(myString.toUpperCase()); // vraća HELLO
console.log(myString.length); // vraća 5
```

Instalacija i podešavanje

Pre nego što počnemo da radimo neke super stvari sa podacima, treba da pripremimo produktivno okruženje. U ovom kratkom poglavlju videćete kako to da uradite.

Instaliranje Node.js i npm

Instalacije Node.js dolaze sa npm (Node.js podrazumevaniim menadžerom paketa).

Instaliranje Node.js na Windows:

1. Pronađite željenu verziju Node.js na zvaničnoj stranici sa instalacijama na https://nodejs.org/en/download/current/.
2. Utvrdite da ste izabrali Node.js 12 (tekuću verziju).

3. Utvrdite da li ste instalirali na pravu arhitekturu vašeg računara; tj. 32-bitno, ili 64-bitno. Možete pronaći ove informacije u System Properties prozoru vašeg sistema.
4. Nakon što preuzmete instaler, kliknite dva puta na fajl i pratite instrukcije na ekranu.

Instaliranje Node.js i npm na Linux:

Za instalaciju Node.js na Linux imate nekoliko dobrih opcija:

- Za instaliranje Node.js kroz Linux menadžere paketa na sistemu koji nije naveden u nastavku, pogledajte https://nodejs.org/en/download/package-manager/.
- Za instaliranje Node.js na Ubuntu pokrenite ovo (više informacija i instrukcije za ručnu instalaciju možete naći na https://github.com/nodesource/distributions/blob/master/README.md#installation-instructions):

```
– curl -sL https://deb.nodesource.com/setup_12.x | sudo -E
bash -
sudo apt-get install -y nodejs
```

- Za instaliranje Node.js na Debian distribucijama (više informacija i instrukcije za ručnu instalaciju možete naći na https://github.com/nodesource/distributions/blob/master/README.md#installation-instructions):

```
– # As root
curl -sL https://deb.nodesource.com/setup_12.x | bash -
apt-get install -y nodejs
```

- Zvanična Node.js stranica za instalaciju ima još opcija za isntalaciju na neke od Linux sistema: https://nodejs.org/en/download/current/.

Instaliranje Node.js i npm na macOS:

Slično Linux-u, postoji nekoliko načina da se Node.js i npm instaliraju na Mac.

Za instalaciju Node.js na macOS X uradite sledeće:

1. Otvorite Terminal za Max tako što ćete pritisnuti *cmd* + *Spacebar*, upisati **`terminal`** u polje za pretragu koje se otvori i pritisnuti *Enter*.
2. Instalirajte Xcode kroz komandnu liniju pokretanjem **`xcode-select --install`**.
3. Najjednostavniji način da se instalira Node.js i npm je pomoću Homebrew, koji se instalira kroz komandnu liniju pokretanjem **`ruby -e "$(curl -fsSL`** (https://raw.githubusercontent.com/Homebrew/install/master/install).

4. Konačni korak je da se instaliraju Node.js i npm. Na komandnoj liniji pokrenite **`brew install node`**.
5. Opet, možete instalirati Node.js i npm i pomoću instalera koji je dostupan na https://nodejs.org/en/download/current/.

Installing Git-a

Da biste instalirali git idite na https://git-scm.com/downloads i pratite instrukcije za vašu platformu.

Dodatni resursi

Paket koda koji prati ovu knjigu smešten je i na GitHub na https://github.com/TrainingByPackt/Professional-JavaScript. Imamo dostupne i pakete koda drugih knjiga i video-materijala iz našeg bogatog kataloga, dostupne na https://github.com/PacktPublishing/. Pogledajte čega tu ima!

1

JavaScript, HTML i DOM

Ciljevi i zadaci

Do kraja ovog poglavlja moći ćete da:

- Opišete HTML objektni model dokumenta **(DOM - Document Object Model)**
- Istražujete DOM veb stranice pomoću Chrome DevTools
- Napravite JavaScript koji čita i menja DOM
- Kreirate sopstvene komponente pomoću Shadow DOM

U ovom poglavlju naučićemo šta je DOM i kako da mu pristupimo i menjamo pomoću JavaScript-a. Naučićemo i kako da pravimo dinamičke aplikacije pomoću sopstvenih komponenti.

Uvod

HTML je zamišljen kao jezik za opisivanje statičkih dokumenata, koji bi bio jednostavan za upotrebu i mogao da se piše upotrebom bilo kog editora teksta. Kada je JavaScript postao bitan faktor u svetu interneta javila se potreba da se HTML dokumenti načine dostupnim JavaScript platformi. Tako je nastao DOM. DOM je zapravo HTML sadržaj koji je mapiran na stablo objekata, koji se mogu dohvatiti i menjati pomoću JavaScript-a.

U ovom poglavlju naučićete šta je DOM i kako da koristimo JavaScript da mu pristupimo. Naučićete kako da pronađete elemente i podatke u dokumentu, kako da menjate stanja elemenata i kako da im menjate sadržaj. Naučićete i kako da kreirate nove DOM elemente i dodajete ih u stranicu.

Kada završimo sa učenjem šta je DOM i kako se koristi, napravićete dinamičku aplikaciju na datom primeru. Konačno, naučićete kako da napravite prilagođene HTML elemente i kako da od njih formirate svoje komponente upotrebom Shadow DOM.

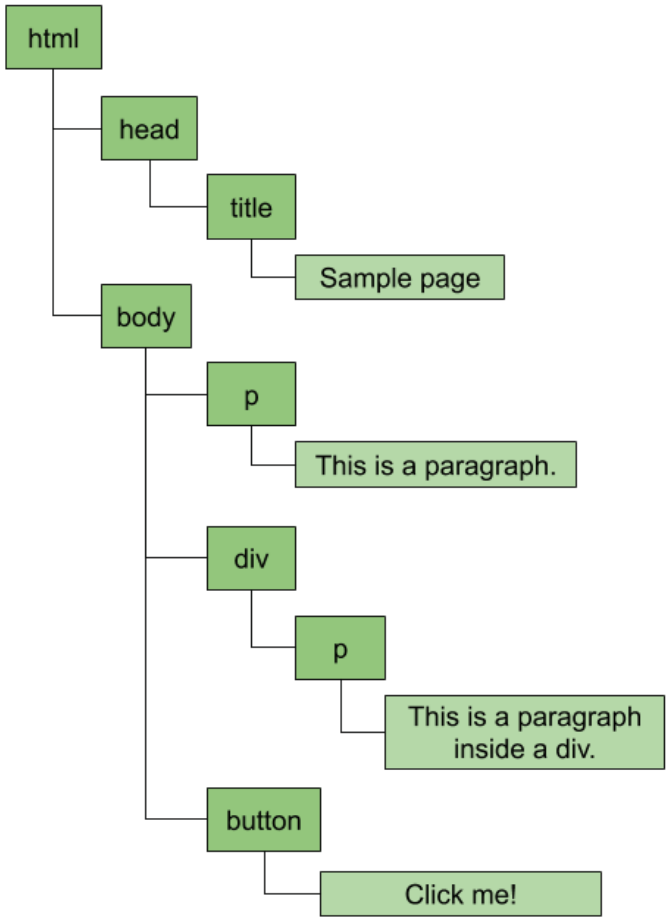
HTML i DOM

Kada veb pretraživač učita HTML stranicu kreira i strukturu stabla koje predstavlja tu stranicu. Ovo stablo bazirano je na DOM specifikaciji i koristi HTML oznake kako bi se odredilo gde koji čvor počinje i gde se završava.

Pogledajmo ovaj isečak HTML koda:

```
<html>
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <div>
      <p>This is a paragraph inside a div.</p>
    </div>
    <button>Click me!</button>
  </body>
</html>
```

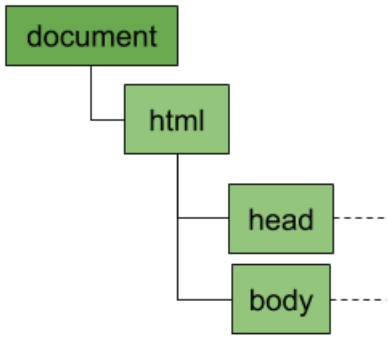
Pretraživač će kreirati ovakvu hijerarhiju čvorova:



Slika 1.1: Čvor paragrafa sadrži čvor sa tekstom

Sve postaje čvor, tekst, elementi i komentari pa sve do korenog elementa stabla. Ovakvo stablo koristi se da se elementima pridruže CSS stilovi da se formira prikaz stranice na ekranu. Pored toga, transformiše se u objekat i postaje dostupno JavaScript izvršnoj platformi.

Ali zašto se zove baš objektni model dokumenta? Zato što je HTML izvorno zamišljen za distribuciju dokumenata a ne za kreiranje dinamičkih aplikacija koje danas imamo. Otud, svaki HTML DOM počinje sa elementom dokumenta, na koji su zakačeni svi ostali elementi. Imajući to u vidu, prethodna ilustracija DOM stabla zapravo postaje ovakva:



Slika 1.2: Svako DOM stablo ima document element kao svoj koren

Šta znači kada kažem da pretraživač čini DOM dostupnim JavaScript platformi? Znači da ako pišete neki JavaScript kod u vašoj HTML stranici, možete da pristupite tom stablu i da radite neke prilično interesantne stvari sa njim. Na primer, možete da prilično lako pristupite korenom elementu dokumenta i pristupite svim čvorovima na stranici, i upravo to ćemo i uraditi u sledećoj vežbi.

Vežba 1: Prolaženje kroz čvorove u dokumentu

U ovoj vežbi napisaćemo JavaScript kod koji traži od DOM-a da pronađe i dohvati dugme i doda mu slušaoca događaja (event listener) tako da možemo da izvršimo neki kod kada korisnik klikne na to dugme. Kada se desi događaj, dohvatićemo sve elemente paragrafa, prebrojati ih i zapamtiti njihov sadržaj i na kraju prikazati obaveštenje.

Fajlove sa kodom ove vežbe možete naći na https://github.com/TrainingByPackt/Professional-JavaScript/tree/master/Lesson01/Exercise01.

Pratite sledeće korake kako biste uradili vežbu:

1. Otvorite svoj omiljeni editor teksta, kreirajte novi fajl pod imenom alert_paragraphs.html i u njega smestite primer HTML koda iz prethodnog poglavlja (možete ga naći i na GitHub-u: https://bit.ly/2maW0Sx):

```
<html>
  <head>
    <title>Sample Page</title>
```