

Spis treści

O autorze (9)

O recenzencie (11)

Przedmowa (13)

Rozdział 1. O platformie Node.js (19)

Możliwości platformy Node.js (21)

JavaScript po stronie serwera (22)

Dlaczego powinieneś używać platformy Node.js? (22)

Popularność (22)

Stosowanie JavaScriptu na wszystkich poziomach zestawu narzędzi (23)

Wykorzystanie inwestycji Google'a w rozwój silnika V8 (23)

Prostszy asynchroniczny model sterowany zdarzeniami (24)

Architektura mikrousług (24)

Platforma Node.js jest wytrzymała, ponieważ przetrwała poważny rozłam i powstanie wrogiego rozgałęzienia (24)

Wydajność i wykorzystanie zasobów (26)

Czy Node.js to rakotwórcza katastrofa w obszarze skalowania? (28)

Wykorzystanie zasobów serwera, zyski firmy i ekologiczny hosting (29)

Node.js, architektura mikrousług i systemy łatwe do testowania (30)

Node.js a model Twelve-Factor (31)

Podsumowanie (31)

Rozdział 2. Konfigurowanie platformy Node.js (33)

Wymagania systemowe (33)

Instalowanie platformy Node.js z użyciem menedżerów pakietów (34)

Instalowanie platformy Node.js w systemie Mac OS X za pomocą narzędzia MacPorts (34)

Instalowanie platformy Node.js w systemie Mac OS X za pomocą narzędzia Homebrew (35)

Instalowanie platformy Node.js w systemach Linux, *BSD i Windows z użyciem systemów zarządzania pakietami (36)

Instalowanie dystrybucji platformy Node.js z witryny nodejs.org (36)

Instalowanie z użyciem kodu źródłowego w systemach POSIX-owych (37)

Instalowanie wymaganych elementów (38)

Moduły z kodem natywnym i node-gyp (38)

Instalowanie narzędzi dla programistów w systemie Mac OS X (39)

Instalowanie platformy Node.js z użyciem kodu źródłowego we wszystkich systemach POSIX-owych (40)

Instalowanie instancji na potrzeby programowania za pomocą narzędzia nvm (41)

Polityka tworzenia wersji platformy Node.js i zalecane wersje (43)

Uruchamianie i testowanie instrukcji (44)

Narzędzia platformy Node.js uruchamiane w wierszu poleceń (44)

Uruchamianie prostego skryptu za pomocą platformy Node.js (45)

Uruchamianie serwera za pomocą platformy Node.js (46)

Npm - menedżer pakietów platformy Node.js (47)

Platforma Node.js a standard ECMAScript 6 (ES2015, ES2016 itd.) (48)

Używanie transpilatora Babel do korzystania z eksperymentalnych funkcji JavaScriptu (51)

Podsumowanie (52)

Rozdział 3. Moduły platformy Node.js (53)

Definiowanie modułu (53)

Format modułów w platformie Node.js (55)

Moduły oparte na plikach (55)

Moduły oparte na katalogach (57)

Algorytm używany w platformie Node.js dla instrukcji require(moduł) (57)

Identyfikatory modułów i ścieżki (58)

Struktura katalogów przykładowej aplikacji (59)

System zarządzania pakietami w platformie Node.js - npm (61)

Format pakietów npm (61)

Wyszukiwanie pakietów npm (63)

Inne polecenia narzędzia npm (64)

Numery wersji pakietów (69)

Krótką uwagę na temat specyfikacji CommonJS (71)

Podsumowanie (72)

Rozdział 4. Serwery i klienty HTTP - pierwsze kroki tworzenia aplikacji internetowej (73)

Przesyłanie i odbieranie zdarzeń za pomocą obiektów typu EventEmitter (74)

Teoria działania klasy EventEmitter (75)

Aplikacje w postaci serwera HTTP (76)

Wielowierszowe i szablonowe łańcuchy znaków ze standardu ES2015 (79)

Sniffer HTTP - podsłuchiwanie wymiany komunikatów przez protokół HTTP (80)

Platformy do tworzenia aplikacji internetowych (83)

Wprowadzenie do platformy Express (84)

Domyślna aplikacja z platformy Express (87)

Warstwa pośrednia w platformie Express (89)

Warstwa pośrednia i ścieżki żądań (91)

Obsługa błędów (92)

Wyznaczanie liczb ciągu Fibonacciego za pomocą aplikacji z platformy Express (93)

Kod wymagający obliczeniowo a pętla zdarzeń w platformie Node.js (97)

Zgłaszanie żądań za pomocą klienta HTTP (101)

Wywoływanie usługi w architekturze REST z zaplecza z poziomu aplikacji opartej na platformie Express (103)

Tworzenie prostego serwera w architekturze REST za pomocą platformy Express (104)

Przekształcanie aplikacji do wyznaczania liczb Fibonacciego na usługę w architekturze REST (107)

Wybrane moduły i platformy związane z architekturą REST (109)

Podsumowanie (109)

Rozdział 5. Pierwsza aplikacja oparta na platformie Express (111)

Obietnice ze standardu ES2015 i funkcje routera z platformy Express (111)

Obietnice a obsługa błędów (113)

"Spłaszczanie" asynchronicznego kodu (114)

Dodatkowe narzędzia (115)

Platforma Express i paradygmat MVC (116)

Tworzenie aplikacji Notes (117)

Pierwszy model w aplikacji Notes (118)

Strona główna aplikacji Notes (120)

Dodawanie nowej notatki - tworzenie (123)

Wyświetlanie notatek - wczytywanie (127)

Edycja istniejącej notatki - aktualizowanie (128)

Kasowanie notatek - usuwanie (129)

Motywy w aplikacjach opartych na platformie Express (131)

Skalowanie - uruchamianie kilku instancji aplikacji Notes (133)
Podsumowanie (136)

Rozdział 6. Implementacja paradygmatu Mobile-First (137)

Problem: aplikacja Notes nie jest dostosowana do urządzeń mobilnych (138)
Paradygmat Mobile-First (najpierw wersja mobilna) (139)
Zastosowanie w aplikacji Notes platformy Bootstrap firmy Twitter (141)
 Konfigurowanie platformy Bootstrap (141)
 Dodawanie platformy Bootstrap do szablonów aplikacji (143)
Projekt aplikacji Notes w modelu Mobile-First (145)
 Podstawy systemu tabelowego z platformy Bootstrap (145)
 Ulepszanie listy notatek na stronie głównej (148)
 Ścieżka powrotu w nagłówku strony (149)
 Porządkowanie formularza do dodawania i edytowania notatek (152)
Tworzenie niestandardowych arkuszy stylów platformy Bootstrap (154)
 Narzędzia do dostosowywania platformy Bootstrap (157)
Podsumowanie (158)

Rozdział 7. Przechowywanie i pobieranie danych (159)

Przechowywanie danych i kod asynchroniczny (160)
Rejestrowanie informacji w dzienniku (160)
 Rejestrowanie żądań przy użyciu pakietu Morgan (161)
 Komunikaty diagnostyczne (163)
 Przechwytywanie zawartości strumieni stdout i stderr (163)
 Nieprzechwycone wyjątki (164)
Zapisywanie notatek w systemie plików (165)
 Zapisywanie notatek w systemie LevelUP (171)
 Zapisywanie notatek za pomocą SQL-a w systemie SQLite3 (174)
 Przechowywanie notatek za pomocą podejścia ORM i modułu Sequelize (180)
 Zapisywanie notatek w systemie MongoDB (187)
Podsumowanie (193)

Rozdział 8. Uwierzelnianie wielu użytkowników za pomocą mikrousług (195)

Tworzenie mikrousług z informacjami o użytkownikach (196)
 Model z informacjami o użytkownikach (198)
 Serwer REST z informacjami o użytkownikach (202)
 Skrypty do testowania serwera uwierzelniania użytkowników i zarządzania nim (209)
 Obsługa logowania w aplikacji Notes (212)
 Obsługa logowania do aplikacji Notes z użyciem Twittera (226)
Zestaw komponentów aplikacji Notes (232)
Podsumowanie (233)

Rozdział 9. Dynamiczna interakcja między klientem a serwerem z użyciem biblioteki Socket.IO (235)

Wprowadzenie do biblioteki Socket.IO (237)
Inicjalizowanie biblioteki Socket.IO za pomocą platformy Express (237)
Aktualizowanie strony głównej aplikacji Notes w czasie rzeczywistym (241)
 Użycie klasy EventEmitter w modelu z aplikacji Notes (241)
 Generowane w czasie rzeczywistym zmiany na stronie głównej aplikacji Notes (243)
 Operacje w czasie rzeczywistym w trakcie wyświetlania notatek (246)
Czat między użytkownikami i dodawanie komentarzy na temat notatek (249)
 Model danych do przechowywania komentarzy (250)
 Obsługa komentarzy w routerze z aplikacji Notes (253)

- Modyfikowanie szablonu do wyświetlania notatek pod kątem komentarzy (254)
- Podsumowanie (261)

Rozdział 10. Instalowanie aplikacji opartych na platformie Node.js (263)

- Architektura aplikacji Notes (264)
- Tradycyjne instalowanie usług na platformę Node.js w Linuxie (265)
 - Wymagania wstępne - przygotowanie baz danych (266)
 - Instalowanie platformy Node.js w systemie Ubuntu (267)
 - Konfigurowanie systemu PM2 na potrzeby zarządzania procesami platformy Node.js (271)
- Instalowanie mikrousług na platformę Node.js z użyciem systemu Docker (275)
 - Instalowanie Dockera na laptopie (277)
 - Tworzenie sieci AuthNet dla usługi uwierzytelniania użytkowników (281)
 - Tworzenie sieci FrontNet w aplikacji Notes (288)
 - Dostęp do dostosowanej do Dockera aplikacji Notes z poziomu innych urządzeń (293)
 - Instalacja w chmurze za pomocą narzędzia Docker Compose (299)
 - Instalowanie aplikacji w chmurze za pomocą narzędzia Docker Compose (303)
- Podsumowanie (308)

Rozdział 11. Testy jednostkowe (309)

- Testowanie kodu asynchronicznego (310)
- Asercje - najprostsza technika testowania (311)
- Testowanie modelu (312)
 - Wybrane narzędzia testowe: Mocha i Chai (312)
 - Zestaw testów modelu z aplikacji Notes (313)
- Używanie Dockera do zarządzania testowymi serwerami bazodanowymi (322)
 - Zarządzanie infrastrukturą testów za pomocą narzędzia Docker Compose (322)
 - Skrypty w pliku package.json dotyczące infrastruktury testów dostosowanej do Dockera (325)
 - Wykonywanie testów za pomocą narzędzia Docker Compose (327)
- Testowanie usług zaplecza zgodnych z architekturą REST (328)
- Bezobsługowe testowanie frontonu w przeglądarce za pomocą narzędzia CasperJS (331)
 - Konfigurowanie (332)
 - Ułatwianie testowania interfejsu użytkownika aplikacji Notes (333)
 - Skrypt testu aplikacji Notes za pomocą narzędzia CasperJS (333)
 - Uruchamianie testów interfejsu użytkownika za pomocą narzędzia CasperJS (336)
- Podsumowanie (337)

Skorowidz (339)