

SPIS TREŚCI

PRZEDMOWA	9
WPROWADZENIE	10
ROZDZIAŁ 1 JAK ZMIENIA SIĘ NOWOCZESNA ARCHITEKTURA SIECI?	12
Powstanie aplikacji statycznych	14
Poznaj grubego klienta	17
ROZDZIAŁ 2 DLACZEGO JAVASCRIPT JEST DOBRYM WYBÓREM DLA NOWOCZESNYCH APLIKACJI?	20
Co to jest AngularJS?	22
Co to jest Node.js?	25
Co to jest Express?	33
Co to jest MongoDB?	35
ROZDZIAŁ 3 ROZPOCZYNAAMY PRACĘ NAD PROJEKTEM APLIKACJI SPOŁECZNOŚCIOWEJ	40
Utworzenie statycznej imitacji strony wyświetlającej najnowsze posty	42
Użycie AngularJS w aplikacji	43
Dodawanie nowych postów	46
Co dalej?	49
ROZDZIAŁ 4 UTWORZENIE API NODE.JS	50
Punkt wyjścia	52
Utworzenie postów za pomocą API	54
Obsługa modeli MongoDB za pomocą Mongoose	55
Użycie modeli Mongoose wraz z API	56
Co dalej?	59
ROZDZIAŁ 5 INTEGRACJA NODE.JS I ANGULARJS	60
Komponent \$http	62
Użycie \$http do odczytu postów z API	63
Udostępnianie pliku posts.html za pomocą Node.js	64
Użycie \$http do zapisu postów w API	65
Zmiana kolejności wyświetlania postów	66
Uporządkowanie pliku server.js	67
Uporządkowanie kodu AngularJS	71
Co dalej?	75

ROZDZIAŁ 6	AUTOMATYZACJA ZA POMOCĄ GULP	76
	Wprowadzenie do Grunt i Gulp	78
	„Witaj, świecie” w Gulp	79
	Przygotowanie kodu JavaScript za pomocą Gulp	80
	Kompilacja CSS za pomocą Gulp	87
	Zadanie dev w Gulp	89
	Inne wtyczki Gulp	91
	Co dalej?	92
ROZDZIAŁ 7	UTWORZENIE UWIERZYTELNIANIA W NODE.JS	94
	Wprowadzenie uwierzytelniania na podstawie tokenu	96
	JSON Web Token (JWT)	97
	Użycie BCrypt	101
	Uwierzytelnianie z użyciem MongoDB	103
	Co dalej?	106
ROZDZIAŁ 8	DODANIE ROUTINGU I UWIERZYTELNIANIA KLIENTA	108
	Routing	110
	Utworzenie formularza logowania	113
	Uwierzytelnianie Express	116
	Zdarzenia AngularJS	119
	Uwierzytelnianie postów aplikacji społecznościowej	121
	HTML5 pushstate	123
	Rejestracja	124
	Wylogowanie	125
	Zapamiętaj mnie	126
	Klucz zewnętrzny użytkownika	127
	Co dalej?	128
ROZDZIAŁ 9	OBSŁUGA POWIADOMIEŃ ZA POMOCĄ WEBSOCKET	130
	Wprowadzenie do WebSocket	132
	Jak działa WebSocket?	133
	Do czego można wykorzystać WebSocket?	134
	WebSocket w budowanej aplikacji społecznościowej	135
	WebSocket w AngularJS	139
	Architektura WebSocket	141
	Dynamiczna nazwa hosta WebSocket	146
	Co dalej?	147

ROZDZIAŁ 10 WYKONYWANIE TESTÓW E2E	148
Konfiguracja narzędzia Protractor	150
Frameworki testowania w JavaScript	151
Utworzenie prostego testu Protractor	152
Przygotowanie w narzędziu Protractor definicji oczekiwanego zachowania	162
Wtyczka chai-as-promised	164
Kiedy należy wykonywać testy typu E2E?	165
Co dalej?	166
ROZDZIAŁ 11 TESTOWANIE SERWERA NODE.JS	168
To nie całkiem są testy jednostkowe	170
Framework Mocha dla Node.js	171
Kontroler Post	173
SuperTest	174
Router bazowy	175
Użycie routera bazowego wraz z SuperTest	176
Modele w testach kontrolerów	177
Testowanie kontrolera z uwierzytelnieniem	179
Pokrycie kodu	181
Polecenie npm test	183
JSHint	184
Co dalej?	185
ROZDZIAŁ 12 TESTOWANIE KODU ANGULARJS	186
Użycie narzędzia Karma	188
Bower	189
Konfiguracja narzędzia Karma	191
Podstawowy test usługi	193
Testowanie HTTP za pomocą narzędzia Karma	194
Użycie narzędzia Karma do przetestowania kontrolera	196
Testowanie za pomocą komponentów spy	200
Co dalej?	202
ROZDZIAŁ 13 WDROŻENIE W HEROKU	204
Platforma jako usługa	206
Jak działa Heroku?	207
12 czynników w aplikacji	208
Wdrożenie aplikacji w Heroku	209

MongoDB w Heroku	211
Redis w Heroku	212
Kompilacja zasobów	214
Klaster Node.js	216
Co dalej?	217

ROZDZIAŁ 14 WDRÓŻENIE W DIGITAL OCEAN	218
Co to jest Digital Ocean?	220
Architektura jeden kontra wiele serwerów	221
Fedora 22	222
Utworzenie serwera	223
Instalacja Node.js	225
Instalacja MongoDB	226
Instalacja Redis	227
Uruchomienie aplikacji społecznościowej	229
Uruchomienie aplikacji społecznościowej za pomocą systemd	230
Wdrożenie bez przestoju serwera	231
Migracja do architektury wieloserwerowej	235
Co dalej?	237
Podsumowanie	238
SKOROWIDZ	239