

O'REILLY®

Wydanie II

React od podstaw

Nowoczesne wzorce
tworzenia aplikacji



Helion 

Eve Porcello, Alex Banks

Tytuł oryginału: Learning React: Modern Patterns for Developing React Apps, 2nd Edition

Tłumaczenie: Agnieszka Górczyńska

ISBN: 978-83-283-7289-4

© 2021 Helion SA

Authorized Polish translation of the English edition of Learning React, 2nd Edition ISBN 9781492051725 © 2020 Alex Banks and Eve Porcello

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/reaop2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wprowadzenie	9
1. Poznajemy Reacta	13
Solidne podstawy	14
Przeszłość i przyszłość biblioteki React	14
Zmiany w wydaniu drugiej książki	15
Praca z plikami	16
Repozytorium pliku	16
React Developer Tools	16
Instalacja Node.js	17
2. JavaScript dla Reacta	19
Deklarowanie zmiennych	20
Słowo kluczowe const	20
Słowo kluczowe let	21
Szablon ciągu tekstowego	22
Tworzenie funkcji	24
Deklaracje funkcji	24
Wyrażenie funkcji	24
Parametry domyślne	26
Funkcja strzałki	26
Kompilacja kodu napisanego w JavaScriptcie	29
Obiekty i tablice	30
Destrukturyzacja obiektu	30
Destrukturyzacja tablicy	32
Usprawnienie literału obiektu	32
Operator rozszczepienia	33
Asynchroniczny JavaScript	35
Proste obietnice za pomocą fetch	35
Słowa kluczowe async i await	37
Tworzenie obietnicy	37

Klasy	38
Moduły ES6	40
CommonJS	41
3. Programowanie funkcyjne w JavaScriptcie	43
Co to jest programowanie funkcyjne?	44
Styl imperatywny kontra styl deklaratywny	45
Koncepcje funkcyjne	48
Niemodyfikowalność	48
Czyste funkcje	50
Konwersja danych	52
Funkcje wyższego rzędu	58
Rekurencja	59
Kompozycja	61
Zebranie wszystkiego w całość	63
4. Jak działa React?	69
Konfiguracja strony	69
Elementy Reacta	70
ReactDOM	72
Elementy potomne	73
Komponenty Reacta	77
Komponenty Reacta — podróż po historii	80
5. React i JSX	83
Elementy Reacta jako JSX	83
Wskazówki pomocne w pracy z JSX	84
Mapowanie tablicy za pomocą JSX	85
Babel	85
Przepisy kulinarne w postaci kodu JSX	87
Fragmenty Reacta	93
Wprowadzenie do narzędzia webpack	94
Tworzenie projektu	96
Wczytywanie paczki	104
Mapowanie źródła	105
Narzędzie create-react-app	105
6. Zarządzanie informacjami o stanie Reacta	109
Utworzenie komponentu StarRating	109
Zaczep useState	111
Refaktoryzacja dla zapewnienia zaawansowanych możliwości wielokrotnego użycia	115

Informacje o stanie w drzewie komponentów	117
Przekazywanie informacji o stanie w dół drzewa komponentów	118
Przekazywanie interakcji z powrotem w drzewie komponentów	120
Tworzenie formularzy	124
Użycie ref	124
Komponent kontrolowany	126
Tworzenie zaczepu niestandardowego	127
Dodawanie danych kolorów do informacji o stanie	129
Kontekst Reacta	130
Umieszczenie danych kolorów w kontekście	132
Pobieranie kolorów za pomocą useContext	133
Dostawca kontekstu zawierającego informacje o stanie	134
Zaczepy niestandardowe w kontekście	136
7. Usprawnianie komponentów za pomocą zaczepów	139
Wprowadzenie zaczepu useEffect	139
Tablica zależności	142
Głębokie sprawdzanie zależności	146
Kiedy należy używać useLayoutEffect?	151
Reguły, które należy stosować podczas pracy z zaczepami	153
Usprawnienie kodu za pomocą useReducer	155
Zaczep useReducer stosowany do obsługi skomplikowanych informacji o stanie	157
Usprawnienie wydajności działania komponentu	160
shouldComponentUpdate i PureComponent	162
Kiedy przeprowadzać refaktoryzację?	163
8. Wykorzystywanie danych	165
Żądanie danych	165
Wysyłanie danych w żądaniu	167
Przekazywanie plików za pomocą metody fetch()	167
Autoryzacja żądania	168
Zapisywanie danych lokalnie	169
Obsługa informacji o stanie obietnic	172
Generowane właściwości	174
Listy wirtualizowane	177
Tworzenie zaczepu useFetch	181
Tworzenie komponentu Fetch	183
Obsługa wielu żądań	185
Zachowywanie wartości	186
Żądania typu kaskadowego	190
Ograniczanie szybkości działania sieci	193

Równocześnie wykonywane żądania	194
Oczekiwanie na wartości	196
Anulowanie żądania	197
Wprowadzenie do GraphQL	200
API GraphQL serwisu GitHub	200
Wykonanie żądania GraphQL	202
9. Technologia Suspense	209
Granice błędu	211
Podział kodu	214
Wprowadzenie — komponent Suspense	216
Użycie komponentu Suspense z danymi	217
Zgłaszanie obietnicy	220
Tworzenie źródła danych typu Suspenseful	224
Algorytm Fiber	228
10. Testowanie kodu tworzonoego z użyciem Reacta	231
ESLint	231
Wtyczki ESLinta	234
Prettier	236
Konfiguracja Prettiera w projekcie	237
Prettier w VSCode	238
Sprawdzanie typów w aplikacji pisanej z użyciem Reacta	239
PropTypes	239
Flow	242
TypeScript	245
Programowanie sterowane testami	248
TDD i nauka	249
Wykorzystanie frameworka Jest	249
Create React App i testy	249
Testowanie komponentów Reacta	253
Zapytania	256
Testowanie zdarzeń	257
Stosowanie pokrycia kodem	259
11. Router Reacta	261
Dołączanie routera	262
Właściwości routera	265
Trasy zagnieżdżone	267
Użycie przekierowania	269
Parametry routingu	271

12. React i serwer	275
Izomorficzny kontra uniwersalny	275
Domeny serwera i klienta	276
Serwer generujący kod Reacta	278
Generowanie danych za pomocą Next.js	284
Gatsby	288
Przyszłość Reacta	290

Poznajemy Reacta

Co powoduje, że bibliotekę JavaScriptu można uznać za dobrą? Czy to jest liczba gwiazdek otrzymanych w serwisie GitHub? Czy liczba pobrań biblioteki za pomocą menedżera pakietów npm? A może to będzie liczba komunikatów w serwisie Twitter dotyczących biblioteki? Jak wybrać najdoskonalsze narzędzie do opracowania najlepszego produktu? Skąd wiadomo, że nie marnujemy czasu? Na jakiej podstawie można określić, że coś jest dobre?

Gdy biblioteka React ujrzała światło dzienne, można było spotkać się z wieloma bardzo pochlebnymi opiniami na jej temat, choć pojawiała się również sporo sceptycznych. To była nowa biblioteka, a nowości bywają irytujące.

W odpowiedzi na głosy krytyczne Pete Hunt z zespołu odpowiedzialnego za tworzenia Reacta opublikował artykuł zatytułowany *Why React?*, w którym zaproponował, aby poświęcić tej bibliotece 5 minut. W ten sposób chciał zachęcić programistów do wypróbowania jej, zanim uznają, że tworzący ją zespół posunął się za daleko.

To prawda, że React to mała biblioteka i że nie jest dostarczana ze wszystkim, co może być potrzebne do zbudowania aplikacji. Dlatego warto poświęcić jej owe 5 minut.

To prawda, że podczas pracy z Reactem tworzony jest kod przypominający HTML umieszczony bezpośrednio w kodzie JavaScriptu. Prawdą jest również, że te znaczniki wymagają etapu preprocesora, zanim zostaną wykonane w przeglądarce WWW. Prawdopodobnie będziesz potrzebować narzędzia kompilacji, takiego jak webpack. Dlatego poświęć tej bibliotece 5 minut.

Skoro React ma już niemalże dekadę, wiele zespołów uznało, że ta technologia jest całkiem dobra, ponieważ programiści tych zespołów poświęcili jej 5 minut. Mamy tutaj na myśli Ubera, Twittera i Airbnb, czyli ogromne firmy, które wypróbowały bibliotekę React oraz uznały, że może ona pomóc w szybszym utworzeniu lepszych produktów. Czyż nie taki był cel opracowania tej biblioteki? Nie wysyłanie komunikatów w serwisie Twitter. Nie zdobywanie gwiazdek. Nie pobieranie. Celem jest tworzenie świetnych rozwiązań, które będą lubiane przez użytkowników. Priorytetem programistów jest dostarczanie oprogramowania i bycie z niego dumnym. Jeżeli zajmujesz się takimi rzeczami, prawdopodobnie polubisz pracę z Reactem.

Solidne podstawy

Niezależnie od tego, czy dopiero poznasz Reacta, czy też sięgnąłeś po tę książkę, by poznać najnowsze funkcje tej biblioteki, chcemy, aby zapewniła Ci solidne fundamenty dla wszystkich przyszłych projektów, które będziesz budować z jej użyciem. Celem, jaki nam przyświecał podczas pisania książki, było uniknięcie zamieszania w trakcie procesu nauki, czemu ma służyć uporządkowanie zagadnień w odpowiedniej kolejności i tym samym przygotowanie tzw. mapy drogowej nauki.

Zanim zagłębisz się w tajniki biblioteki React, najpierw musisz poznać JavaScript. Nie mamy tutaj na myśli całego JavaScriptu i każdego stosowanego w nim wzorca, ale powinieneś potrafić komfortowo pracować z tablicami, obiektami i funkcjami, aby materiał przedstawiony w tej książce był dla Ciebie użyteczny.

W następnym rozdziale poznasz nowszą składnię JavaScriptu, dzięki której będziesz mieć dostęp do najnowszych funkcji tego języka, zwłaszcza tych najczęściej wykorzystywanych podczas pracy z biblioteką React. Następnie przejdziemy do funkcyjnego JavaScriptu, aby w ten sposób umożliwić Ci poznanie paradygmatu, który doprowadził do powstania Reacta. Miłym skutkiem ubocznym pracy z Reactem jest to, że możesz stać się jeszcze lepszym programistą JavaScriptu poprzez promowanie wzorców, które są czytelne, nadają się do wielokrotnego użycia, a także umożliwiają przetestowanie kodu. To coś jakby lekkie i pożyteczne pranie mózgu.

Kolejnym krokiem będzie przedstawienie podstawowych koncepcji, jakie kryją się za biblioteką React, dzięki którym nauczysz się tworzyć interfejs użytkownika za pomocą komponentów. Następnie przejdziemy do tematu przygotowywania tych komponentów oraz dodawania logiki z właściwościami i informacjami o stanie. Omówimy zaczepy Reacta umożliwiające wielokrotne ponowne użycie logiki w różnych komponentach.

Po zaprezentowaniu podstaw przejdziemy do budowy aplikacji pozwalającej użytkownikom dodawać, edytować i usuwać kolory. Dowiesz się, jak zaczepy i komponent `Suspense` pomagają w pobieraniu danych. Podczas tworzenia tej aplikacji przedstawimy wiele różnych narzędzi stosowanych w ekosystemie biblioteki React do obsługi zadań takich jak routing, testowanie i generowanie po stronie serwera.

Mamy nadzieję, że dzięki temu będziesz mógł szybciej rozpocząć pracę w ekosystemie biblioteki React — nie tylko przejrzeć podstawy, ale faktycznie poznać narzędzia i zdobyć umiejętności niezbędne do tworzenia rzeczywistych aplikacji React.

Przeszłość i przyszłość biblioteki React

Bibliotekę React utworzył Jordan Walke, czyli inżynier oprogramowania z firmy Facebook. Po raz pierwszy biblioteka ta została wykorzystana w 2011 roku, w kanale wiadomości Facebooka, a następnie, w 2012 roku, w serwisie Instagram, po jego przejściu przez Facebooka. Na konferencji JSConf w 2013 roku ogłoszono, że React to biblioteka typu open source, i w ten sposób dołączyła ona do zatłoczonej kategorii bibliotek interfejsu użytkownika, takich jak m.in. jQuery, Angular, Dojo, Meteor. W tamtym czasie React była uznawana za element widoku (ang. *view*) w architekturze

MVC. Innymi słowy w aplikacjach JavaScriptu komponenty Reacta działały na warstwie widoku lub interfejsu użytkownika.

Od tamtego momentu React zaczął stopniowo się rozpowszechniać. W styczniu 2015 roku firma Netflix ogłosiła, że używa Reacta do tworzenia interfejsu użytkownika. W tym samym miesiącu została wydana biblioteka React Native, przeznaczona do tworzenia aplikacji mobilnych. Facebook wydał również ReactVR, czyli narzędzie umożliwiające wykorzystanie Reacta w znacznie większej liczbie narzędzi generujących treści. W latach 2015 i 2016 pojawiła się ogromna liczba popularnych narzędzi, np. React Router, Redux i Mobx, przeznaczonych do obsługi zadań takich jak routing i zarządzanie informacjami o stanie. W końcu React był przedstawiany jako biblioteka zawierająca implementację określonego zestawu funkcjonalności, a nie jako narzędzie do niemalże każdego zadania.

Kolejnym ogromnym wydarzeniem na osi czasu biblioteki React było wydanie React Fiber w 2017 roku. Fiber to utworzony zupełnie od początku algorytm generowania, którego działanie można uznać za niemalże magiczne. Zmodyfikowano praktycznie wszystkie wewnętrzne komponenty Reacta, a skutkiem była zmiana wszystkiego poza publicznie dostępnym API. Dzięki tej zmianie biblioteka stała się znacznie nowocześniejsza i bardziej wydajna w działaniu, same modyfikacje zaś nie miały niekorzystnych implikacji dla użytkowników.

Z kolei w 2019 roku byliśmy świadkami pojawienia się zaczepów Hooks, które zapewniają nowy sposób dodawania i współdzielenia między komponentami logiki odpowiedzialnej za obsługę informacji o stanie. Pojawił się również komponent Suspense, który zapewnił optymalizację asynchronicznego generowania treści przez Reacta.

W przyszłości na pewno zostaną wprowadzone kolejne zmiany. Jednym z powodów tak ogromnego sukcesu, jaki osiągnęła biblioteka React, jest jednak silny zespół od lat pracujący nad jej rozwojem. Ten zespół jest jednocześnie ambitny i ostrożny, nieustannie wprowadza optymalizacje, a przy tym analizuje, jaki wpływ w społeczności mają poszczególne zmiany w bibliotece.

Czasami zmiany wprowadzane w bibliotece React lub powiązanych z nią narzędziach okazują się niezgodne ze zmianami występującymi w poprzednich wersjach. Tak naprawdę niektóre zmiany wprowadzone w przyszłych wersjach biblioteki mogą spowodować, że przykładowe fragmenty kodu zamieszczone w tej książce nie będą działać. Dokładne informacje o wersjach zostały zamieszczone w pliku *package.json*, więc dzięki temu możesz zainstalować pakiety w odpowiednich wersjach.

Informacje dotyczące nowych zmian wprowadzonych w bibliotece znajdziesz również w oficjalnym blogu biblioteki React na stronie <https://reactjs.org/blog/>. Po wydaniu nowej wersji biblioteki członkowie tworzącego ją zespołu przygotowują nowy wpis, w którym są prezentowane wprowadzone zmiany. Ten blog jest tłumaczony na coraz więcej języków. Dlatego jeśli angielski nie jest Twoim językiem ojczystym, przetłumaczone wersje dokumentacji znajdziesz na stronie <https://reactjs.org/languages/>.

Zmiany w wydaniu drugim książki

Właśnie czytasz drugie wydanie tej książki. Odnieśliśmy wrażenie, że konieczna jest jej aktualizacja, ponieważ w ciągu ostatnich kilku lat byliśmy świadkami dość intensywnego rozwoju biblioteki React. Naszym celem jest skoncentrowanie się na wszystkich obecnie dostępnych najlepszych praktykach, których stosowanie zaleca zespół tworzący Reacta. Ponadto podzielimy się wiedzą

o funkcjonalnościach, które są już uznawane za przestarzałe. Istnieje ogromna ilość kodu utworzonego wiele lat temu w starym stylu Reacta. Ten kod nadal działa doskonale i musi być konserwowany. We wszystkich takich przypadkach będziemy wspominać na pasku bocznym o przestarzałych funkcjach na wypadek, gdybyś musiał pracować ze starymi aplikacjami napisanymi z użyciem Reacta.

Praca z plikami

Z tego podrozdziału dowiesz się, jak przebiega praca z plikami w książce, a także jak zainstalować użyteczne narzędzia Reacta.

Repozytorium pliku

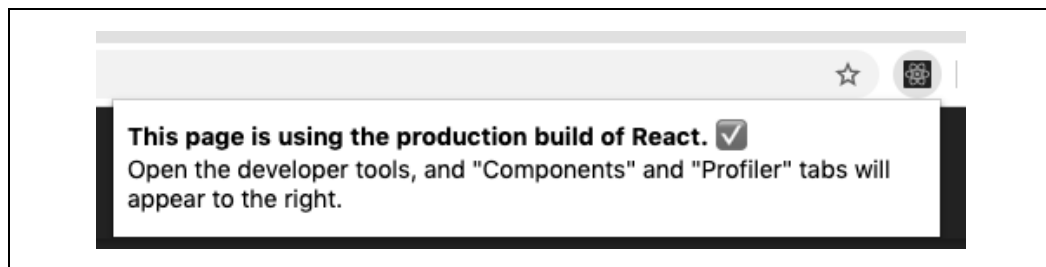
W repozytorium GitHub zawierającym materiały przygotowane dla książki (<ftp://ftp.helion.pl/przyklady/reaop2.zip>) wszystkie pliki kodu źródłowego zostały umieszczone w oddzielnych podkatalogach dla poszczególnych rozdziałów.

React Developer Tools

Gorąco zalecamy zainstalowanie wtyczki React Developer Tools, która ułatwia pracę z projektami Reacta. Jest to wtyczka dla przeglądarek WWW Chrome i Firefox, a także w postaci oddzielnej aplikacji przeznaczonej do użycia z Safari, IE i React Native. Po zainstalowaniu narzędzi programistycznych możesz analizować komponenty Reacta, właściwości widoku, szczegółowe informacje o stanie, a nawet widoki witryn internetowych wykorzystujących w produkcji bibliotekę React. Narzędzia te okazują się naprawdę użyteczne podczas debugowania i poznawania sposobów zastosowania Reacta w innych projektach.

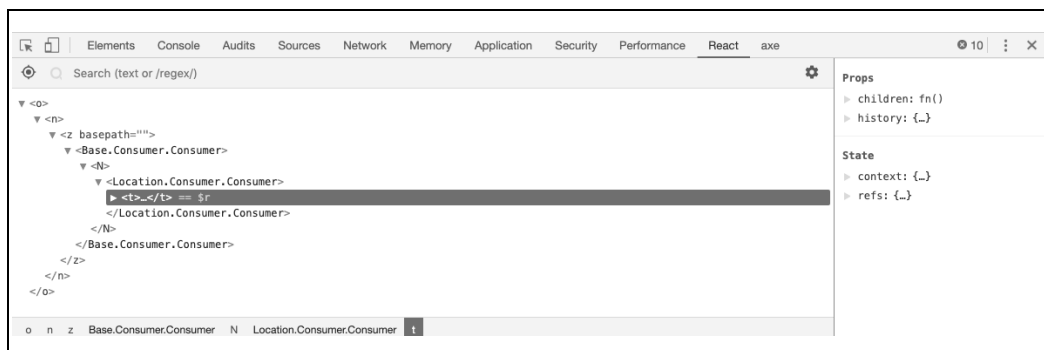
Aby zainstalować te narzędzia, przejdź do repozytorium w serwisie GitHub na stronie <https://github.com/facebook/react-devtools>. Znajdziesz tam łącza prowadzące do wtyczek dla przeglądarek WWW Chrome (<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>) i Firefox (<https://addons.mozilla.org/en-US/firefox/addon/react-devtools/>).

Po zainstalowaniu wtyczki będziesz w stanie zobaczyć, czy witryna internetowa została zbudowana z użyciem technologii React. Zawsze gdy ikona Reacta jest podświetlona na pasku narzędzi przeglądarki WWW (zobacz rysunek 1.1, dana witryna internetowa korzysta z biblioteki React).



Rysunek 1.1. Wyświetlenie w przeglądarce WWW Chrome okna dialogowego, które zostało wygenerowane przez rozszerzenie React Developer Tools

Następnie po przejściu do narzędzi dla programistów zobaczysz nową kartę, o nazwie *React* (zobacz rysunek 1.2). Jej kliknięcie spowoduje wyświetlenie wszystkich komponentów tworzących aktualną stronę w przeglądarce WWW.



Rysunek 1.2. Analiza modelu DOM za pomocą React Developer Tools

Instalacja Node.js

Node.js to środowisko uruchomieniowe JavaScriptu używane do tworzenia aplikacji pełnego stosu. Jest to oprogramowanie typu open source, które można zainstalować na wielu różnych platformach, m.in. Windows, macOS i Linux. Skorzystamy z niego w rozdziale 12., podczas tworzenia serwera Express.

Musisz mieć zainstalowane środowisko Node.js, przy czym nie musisz być ekspertem od Node.js, aby móc korzystać z biblioteki React. Jeżeli nie jesteś pewien, czy masz zainstalowane Node.js w systemie, przejdź do okna powłoki, a następnie wydaj polecenie:

```
$ node -v
```

Po wydaniu tego polecenia wygenerowane dane wyjściowe powinny zawierać numer wersji Node.js, najlepiej gdyby to była wersja 8.6.2 lub nowsza. Natomiast jeśli otrzymasz komunikat informujący o nieznanym poleceniu, oznacza to, że Node.js nie zostało zainstalowane w systemie. Ten problem można łatwo rozwiązać przez zainstalowanie Node.js z witryny internetowej projektu (<https://nodejs.org/en/>). Wystarczy wykonać kolejne kroki zautomatyzowanego programu instalacyjnego, a następnie po ponownym wydaniu polecenia `node -v` powinieneś zobaczyć numer wersji.

npm

Po zainstalowaniu Node.js otrzymasz również menedżer pakietów npm (ang. *node package manager*). W społeczności programistów JavaScriptu inżynierowie dzielą się kodem projektów typu open source, aby w ten sposób uniknąć konieczności ponownego tworzenia frameworków, bibliotek lub funkcji pomocniczych. React to również przykład użytecznej biblioteki dostępnej za pomocą menedżera pakietów npm, z którego będziemy korzystać do zainstalowania wielu pakietów podczas lektury książki.

Większość obecnie spotykanych projektów JavaScriptu zawiera kolekcje plików z plikiem o nazwie *package.json*. Zawiera on opis projektu i listę jego wszystkich zależności. Jeżeli w katalogu zawierającym plik *package.json* wydasz polecenie `npm install`, menedżer npm zainstaluje wszystkie pakiety wymienione w pliku *package.json*.

Jeśli zupełnie od początku rozpoczynasz pracę nad projektem i chcesz zawrzeć w nim niezbędne zależności, wydaj następujące polecenie:

```
$ npm init -y
```

To spowoduje inicjalizację projektu i utworzenie w nim pliku *package.json*. Następnie możesz przystąpić do samodzielnej instalacji niezbędnych zależności za pomocą menedżera pakietów npm. Aby zainstalować pakiet za pomocą npm, trzeba wydać następujące polecenie:

```
$ npm install nazwa-pakietu
```

Natomiast usunięcie pakietu nastąpi po wydaniu polecenia:

```
$ npm remove nazwa-pakietu
```

Yarn

Alternatywą dla npm jest Yarn. Narzędzie to wydał w 2016 roku Facebook we współpracy z firmami Exponent, Google i Tilde. Projekt pomaga Facebookowi, a także innym firmom w niezawodnym zarządzaniu zależnościami. Znając sposób działania menedżera pakietów npm, możesz bardzo łatwo rozpocząć pracę z Yarnem. Przede wszystkim musisz zainstalować to narzędzie globalnie, używając do tego npm:

```
$ npm install -g yarn
```

Dzięki temu możesz instalować pakiety. Podczas instalowania zależności wymienionych w pliku *package.json* zamiast `npm install` wydaj polecenie `yarn`.

Oto polecenie umożliwiające zainstalowanie konkretnego pakietu za pomocą Yarn:

```
$ yarn add nazwa-pakietu
```

Jeżeli chcesz usunąć zależność, wydaj następujące polecenie:

```
$ yarn remove nazwa-pakietu
```

Yarn jest używany w środowisku produkcyjnym przez firmę Facebook, a także znalazł zastosowanie w wielu projektach, takich jak np. React, React Native i Create React App. Jeżeli kiedykolwiek natrafisz na projekt zawierający plik o nazwie *yarn.lock*, teraz już wiesz, że użyto w nim narzędzia Yarn. Podobnie jak w przypadku polecenia `npm install`, zależności projektu można zainstalować przez wydanie polecenia `yarn`.

W ten sposób przygotowałeś środowisko do pracy z projektami tworzonymi z użyciem Reacta i możesz przystąpić do poznawania tej biblioteki. W rozdziale 2. zamieściliśmy krótkie wprowadzenie do najnowszej składni stosowanej w języku JavaScript, która jest najczęściej stosowana także w kodzie piśnianym z użyciem Reacta.

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

React: śmiało sięgaj po efektywność i nowoczesność!

JavaScript jest obecnie pełnowartościowym językiem programowania — wszechstronnym, elastycznym i pozwalającym na tworzenie wydajnego kodu. Społeczność skupiona wokół JavaScriptu udostępniła wiele przydatnych frameworków i bibliotek. Wśród nich wyjątkowo ciekawą propozycją jest React — popularna biblioteka przeznaczona do tworzenia nowoczesnych, dynamicznych aplikacji. To narzędzie, które w stosunkowo prosty sposób pozwala na pełne wykorzystanie możliwości nowoczesnych przeglądarek i urządzeń mobilnych — wystarczy wiedzieć, jak technologie JavaScript, CSS i HTML działają w przeglądarce WWW.

W tej przystępnej, zajmująco napisanej książce znajdziesz nie tylko dokładne omówienie biblioteki React, ale też podstawowe koncepcje JavaScriptu, testowania i debugowania oraz zarządzania informacjami o stanie. Przedstawiono tu aktualne i wnikliwe informacje o najlepszych praktykach i wzorcach stosowanych podczas tworzenia nowoczesnego kodu Reacta. Pokazano, jak budować interfejs użytkownika pozwalający na zmianę treści strony bez jej odświeżania. Zaprezentowano również podstawy programowania funkcyjnego z użyciem najnowszych funkcji oferowanych przez ECMAScript. Dzięki temu przewodnikowi dowiesz się, jak tworzyć komponenty Reacta, korzystać z routera Reacta, a także generować i testować serwery. Poszczególne informacje zostały uporządkowane i podane w sposób ułatwiający natychmiastowe rozpoczęcie pracy z tą świetną biblioteką.

W książce między innymi:

- kluczowe koncepcje programowania funkcyjnego w JavaScriptcie
- działanie Reacta w przeglądarce WWW
- tworzenie warstwy prezentacji aplikacji
- debugowanie aplikacji Reacta
- zaczepty Reacta i korzystanie z routingu
- aplikacja Reacta klient-serwer

Eve Porcello i Alex Banks są inżynierami oprogramowania, wykładowcami i współzałożycielami firmy programistycznej Moon Highway. Stworzyli wiele kursów dostępnych w serwisach LinkedIn Learning i egghead.io, często też występują jako prelegenci i instruktorzy podczas organizowanych na całym świecie warsztatów w zakresie stosowania technologii JavaScript, Node.js, React i GraphQL.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!
SZKOLENIA
AKADEMIA IT & BUSINESS
HELIONSZKOLENIA.PL

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-283-7289-4



INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 69,00 zł