

Porównanie narzędzi do tworzenia aplikacji typu SPA na przykładzie Angular2 i React

Jadwiga Kalinowska*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Celem artykułu jest porównanie dwóch, najczęściej stosowanych narzędzi do wytwarzania aplikacji webowych typu SPA (Single Page Application). Analiza została przeprowadzona dla biblioteki ReactJS i dla frameworka Angular. Do badań wykorzystano aplikacje testowe o takiej samej funkcjonalności, zaimplementowane w obu technologiach. W porównaniu uwzględniono strukturę i wydajność aplikacji, wybrane metryki kodu, jakość dokumentacji oraz wsparcie społecznościowe.

Słowa kluczowe: Angular; React; SPA

* Autor do korespondencji.

Adres e-mail: Jadwiga.kalinowska@pollub.edu.pl

Comparison of tools for creating SPA applications using the examples of Angular2 and React

Jadwiga Kalinowska*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The aim of the article is to compare two, most commonly used tools for the Single Page Application development. The analysis was carried out for the ReactJS library and for the Angular framework. Two applications were implemented for research purposes. The React and Angular test application has the same functionality. The comparison included the structure and performance of applications, selected code metrics, quality of documentation and social support.

Keywords: Angular; React; SPA

*Corresponding author.

E-mail address: Jadwiga.kalinowska@pollub.edu.pl

1. Wstęp

Wzrost zapotrzebowania na portale internetowe wymusił stworzenie nowego podejścia do wytwarzania aplikacji jednostronicowych typu SPA (ang. Single Page Application). Architektura ta pozwala na integrację z użytkownikiem z wykorzystaniem dynamicznego nadpisywania treści, co wpływa znacząco na jej szybkość działania. Cała aplikacja ładowana jest jedynie przy pierwszym uruchomieniu w przeglądarce [4]. Ważnym aspektem a jednocześnie wadą tego rozwiązania jest ingerencja Java Script w strukturę DOM (ang. Document Object Model) [1].

Single Page Application powstała jako alternatywa dla wielostronicowych aplikacji typu MPA (Multi Page Application). MPA cechuje się tym, iż każda strona wysyła żądanie do serwera po czym aktualizowane są wszystkie dane. SPA jest idealnym rozwiązaniem dla platform typu Software as a Service (SaaS), czyli oprogramowanie jako usługa oraz dla platform społecznościowych. MPA zaleca się używać dla sklepów internetowych, stron biznesowych itp. [2].

Architektura SPA posiada gotowe biblioteki i frameworki, pozwala na separację warstwy serwerowej od klienckiej. Istotną cechą, która wpływa na popularność tego rozwiązania

jest możliwość ponownego wykorzystania kodu z aplikacji internetowej do tworzenia aplikacji mobilnych [2].

Narzędzia do wytwarzania aplikacji typu SPA dzielą się na dwa rodzaje: frameworki oraz biblioteki. Do najpopularniejszych można zaliczyć : AngularJS, Angular, Ember.js, Knockout.js, Meteor.js, ExtJS, Vue.js i React [3]. W niniejszym artykule porównano bibliotekę React i framework Angular.

Każde rozwiązanie SPA modyfikuje DOM. W tym celu, podczas aktualizacji aplikacji, tworzony jest własny, wirtualny DOM:

- React - kiedy do komponentu dostarczone są nowe właściwości (ang. props) lub zmienił się stan (ang. State) komponentu to React zostaje o tym poinformowany. React monitoruje różnice pomiędzy wstępnym, a obecnym wirtualnym DOM. Następnym krokiem jest wprowadzenie zmian w prawdziwym DOM, ale tylko w zmodyfikowanych węzłach [4].
- Angular 2 – proces ten odbywa się za pomocą biblioteki Zones.js. Biblioteka ta jest w stanie wykrywać zmiany zdarzenia przeglądarki (np. i - kliknięcie, najechanie kursorem na element, wprowadzenie tekstu z klawiatury) oraz żądania Ajax. Przy tworzeniu aplikacji do wszystkich

komponentów przywiązany zostaje detektor zmian. Po wykryciu zmian w szablonie komponentu, dokonane zostaje porównanie obecnej i poprzedniej wartości. W przypadku, gdy wartości nie są takie same - dokonywana jest aktualizacja DOM [4].

Oba rozwiązania mają szerokie zastosowania:

- React wykorzystywany jest w takich serwisach jak Facebook, Instagram, Netflix, Alibaba, Yahoo, E-Bay, Khan-Academy, AirBanB, Sony oraz Atlassian [5]; Angular wykorzystywany jest do tworzenia narzędzi dostarczanych przez Google.

2. Aplikacja testowa

Jednym z głównych zalet aplikacji typu SPA jest wysoka wydajność, która pozwala płynnie korzystać z aplikacji, bez opóźnień. Do wykonania analizy zostały stworzone bliźniacze aplikacje, posiadające identyczne funkcjonalności. Aplikacje wykorzystują nierelacyjną bazę danych czasu rzeczywistego Firebase oraz magazyn danych Firebase przechowujący pliki graficzne.

Główną funkcjonalnością aplikacji jest galeria zdjęć, wyświetlana w postaci wierszy, po 4 zdjęcia w każdym. Dodawanie zdjęć możliwe jest za pomocą odpowiedniego formularza, dostępnego dla zalogowanego użytkownika.

3. Analiza porównawcza

Do przeprowadzenia porównania obu narzędzi, wybrano 5 kryteriów. Są to:

- struktura aplikacji - obejmuje komponenty interfejsu użytkownika oraz komponenty organizacyjne umożliwiające dostęp do nich;
- metryki kodu - porównano liczbę linii kodu oraz rozmiary plików;
- dokumentacja - ważnym aspektem w analizie narzędzi typu jest dostępność i jakość dokumentacji;
- wsparcie społecznościowe - wzrost popularność programowania sprawił, iż powstały grupy społeczności, które dzielą się swoją wiedzą i pomagają innym rozwiązywać pojawiające się problemy. Największą społeczność można znaleźć na platformie StackOverflow [6];
- testy wydajnościowe – aplikację zostały przetestowane pod względem prędkości pobierania danych potrzebnych do działania aplikacji.

Do testów wykorzystano następujące przeglądarki:

- Google Chrome, w wersji 68.0.3440.106,
- Mozilla Firefox, w wersji 61.02 (64 bity),
- Microsoft Edge, w wersji 42.17134.1.0.

W obydwu aplikacjach stosowana jest wersja ECMAScript 6 i narzędzia deweloperskie udostępniane przez każdą z przeglądarek. W testach wykorzystano m.in. 570 KB obraz o rozdzielczości 1366x768 px. Testy zostały przeprowadzone dla 1, 5, 10, 25, 50 i 75 plików w komponencie galerii.

3.1. Struktura aplikacji

W obydwu przypadkach ciężko jest określić typ modelu architektonicznego. Angular2 oraz ReactJS nie wykorzystuje typowej architektury MVC lub MVVM. W przypadku Angular2 wynika to z tego iż jest on „platformą oraz frameworkiem do budowania aplikacji klienckich w HTML i TypeScript” [14]. Ogólnie architektura Angular2 składa się z modułów, komponentów (widoki) oraz serwisów.

Architektura React oparta jest jedynie na komponentach, ponieważ React jest jedynie biblioteką do tworzenia widoków. Zazwyczaj aplikacja musi zostać wzbogacona o router nawigujący po komponentach oraz Redux do kontrolowania stanów aplikacji JavaScript. Nie jest ważne umiejscowienie plików. Jednak ze względu na lepszą czytelność struktury, zachęca się programistów do przechowywania komponentów w folderze komponent.

W obydwu przypadkach pliki z bibliotekami znajdują się w folderach node_modules. Rozmiar bibliotek dla ReactJS to 240 MB, a dla Angular2 - 339 MB. W obydwu przypadkach biblioteki muszą znajdować się w folderze projektu.

3.2. Metryki kodu

W celu porównania liczby linii kodu wymaganych do poprawnego działania konkretnego komponentu, wykorzystano funkcję dodania zdjęć.

Podczas implementacji aplikacji React dla widoku dodawania zdjęć został zdefiniowany plik Upload.js. Plik jest klasą, która zawiera funkcję render(), renderującą widok komponentu, funkcję dodawania zdjęć i funkcję wyświetlającą pasek postępu. Plik zawiera 103 linie kodu.

W przypadku Angular należało stworzyć dwa pliki komponentu (ts, html), usługę (serwis) i model danych. Pierwszy plik upload.component.html, zawiera definicję wyświetlania komponentu (16 linii kodu). Drugi plik upload.component.ts (31 linii kodu), definiuje funkcję komponentu oraz wywołuje funkcję z serwisu upload.service.ts (48 linii kodu). Funkcja dodaje plik graficzny do bazy danych i magazynu danych Firebase. Ostatnim plikiem jest plik modelu danych upload.model.ts będący klasą Upload. Plik użytkowany jest w serwisie, ma 12 linii. Suma linii kodu plików w aplikacji Angular wynosi 107 linii (Tabela 1). Istotny jest także rozmiar bibliotek wykorzystywanych w obu projektach (Tabela 1).

Tabela 1. Porównanie metryk kodu dla pojedynczego komponentu

		React	Angular
Komponent dodawania plików graficznych	Rozmiar plików	3,43 KB	3,16 KB
	Liczba plików	1	4
	Liczba linii kodu	103	107
Rozmiar biblioteki		240MB	339MB
Rozmiar całego projektu		241MB	340MB

3.3. Dostępność dokumentacji

Zarówno Angular [7] jak i React [8] posiadają obszerne dokumentacje dostępne w języku angielskim. Twórcy zarówno Angular, jak i React zadbali o to, aby w zrozumiały sposób przedstawić podstawy korzystania z nich (np. klarownie opisano wymagane narzędzia, poszczególne kroki tworzenia oraz uruchomienia aplikacji). Obydwa narzędzia posiadają proste, lecz dokładne tutoriale, dzięki którym można zaznajomić się z podstawowymi zasadami oraz możliwościami narzędzi.

Angular2 i React są popularnym tematem artykułów, publikacji, tutoriali w formie wideo i wpisów na stronach związanych z programowaniem. Należy zwrócić uwagę, że w obydwu przypadkach większość materiałów dostępna jest jedynie w języku angielskim. Z powodu trudności w ustaleniu liczby pozycji w języku angielskim, a także polskim dla narzędzi typu SPA, w pracy do porównania dostępności książek skorzystano z internetowej księgarni informatycznej Helion.pl [9] oraz księgarni Google Books [10]. Google Books posiada około 20 milionów wyników dla React, natomiast dla Angular około 15 milionów. Dokładniejsze liczby dostarcza Helion.pl - posiada 21 pozycji dotyczących Angular oraz 19 w przypadku React w wersji Epub/Mobi. Helion w swojej ofercie dysponuje kursami video w liczbie – 5 pozycji dla Angular i 1 związaną z React. Kolejnym aspektem oceny jest liczba udostępnionych tutoriali w internecie – wykorzystano tu wyszukiwarkę Google. Dla React istnieje około 5 500 000 wyników, natomiast dla Angulara zaledwie niecałe 700 tysięcy. Zauważając wyszukiwanie do samego YouTube otrzymano około 4 miliony wyników dla React i blisko 300 tysięcy dla Angular (Tabela 2).

Tabela 2. Porównanie liczby dostępnych materiałów dla obu narzędzi

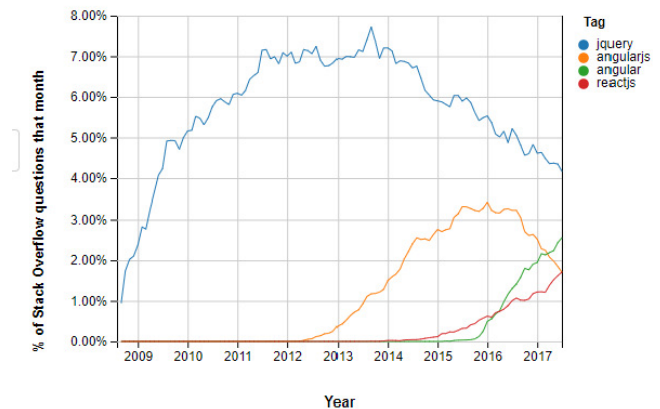
	Angular2	React
Google Books	15 000 000	20 000 000
Helion – Epub/Mobi	21	19
Helion – kurs video	5	1
Google – tutorial	700 000	5 500 000

3.4. Wsparcie społecznościowe

Największą społeczność można znaleźć na platformie StackOverflow [6]. StackOverflow udostępnia statystykę z wykorzystaniem tagów – słów kluczowych. Tagi zazwyczaj związane są z technologią, której dotyczy pytanie. Serwis posiada narzędzie, które sprawdza trendy zapytań zadawanych na StackOverflow [11]. Tag angular posiada 128099 zapytań [12] z czego 140 z dnia 25.08.2018, 1338 z okresu czasu 7 dni (od 19.08.2018 do 25.08.2018). Tag angular-cli posiada 4843 zapytań. Inne popularne tagi dla Angular2 to : angular5, angular-material, angular2-routing, angular6, angular-ui-bootstrap, angular2-template, angular-ui. Tag reactjs posiada 99538 zapytań [13]. Na dzień 25.08.2018 liczba zapytań wyniosła 159. W przeciągu 7 dni (19.08.2018 – 25.08.2018) liczba zapytań 1219. Popularne słowa kluczowe to: react-native, react-outer, react-redux. Tag create-react-app posiada 1338 zapytań.

Rys. 1 przedstawia zapytania dla najbardziej popularnych narzędzi JavaScript wg procentowych zapytań na

StackOverflow w latach 2009-2018. Wykres pokazuje, iż Angular2 i Reactjs są topowymi rozwiązaniami do tworzenia aplikacji.



Rys. 1. Procent pytań StackOverflow w danym miesiącu [14]

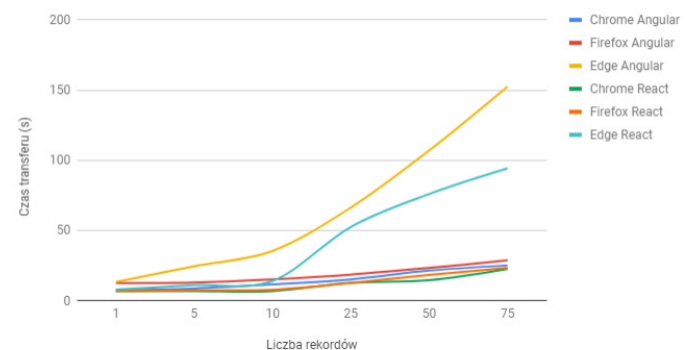
3.5. Wydajność aplikacji

W tabeli 3 przedstawiono czasy transferu danych dla obu aplikacji mierzone, narzędziami dostępnymi w analizowanych przeglądarkach.

Tabela 3. Porównanie czasów pobrania danych z bazy

	Angular			React		
	Czas transferu [s]			Czas transferu [s]		
Liczba rekordów	Google Chrome	Mozilla Firefox	Microsoft Edge	Google Chrome	Mozilla Firefox	Microsoft Edge
1	7,74	12,79	13,48	6,96	7	8,17
5	8,96	13,22	24,73	7,03	7,53	11,35
10	11,78	15,44	35,66	7,08	7,99	14,26
25	15,4	18,75	66,47	12,94	12,88	52,71
50	21,63	23,49	107,15	14,87	18,59	76,1
75	25,34	28,97	152,6	22,88	23,49	94,36

Najlepsze czasy zostały uzyskane przez Google Chrome, najgorsze - odnotowano w przypadku przeglądarki Microsoft Edge. Przeglądarka Mozilla Firefox uzyskała wyniki zbliżone do Google Chrome. W każdym z przypadków aplikacja Angular była nieznacznie wolniejsza w stosunku do React. Biorąc pod uwagę jedynie przeglądarki Google Chrome i Mozilla Firefox różnice w czasie transferu nie przekraczają 10s. W przypadku testów w przeglądarce Microsoft Edge dla 75 rekordów, aplikacja Angular potrzebowała aż 152,6 s, podczas gdy w Chrome rekordy załadowały się w 25 sekund. Na rysunku 2 przedstawiono czasy pobierania różnych liczb rekordów przez aplikacje w poszczególnych przeglądarkach.



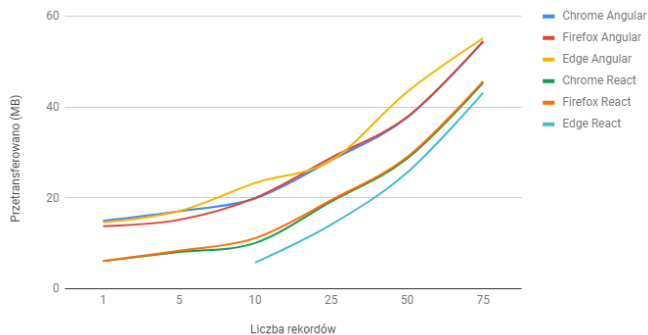
Rys. 2. Czas pobierania różnej liczby rekordów z bazy danych w przeglądarkach

Tabela 4. Analiza zużycia pamięci przy pobieraniu rekordów z bazy w przeglądarkach

Liczba rekordów	Angular2			ReactJS		
	Google Chrome	Mozilla Firefox	Microsoft Edge	Google Chrome	Mozilla Firefox	Microsoft Edge
1	15	13,79	14,64	6,1	6,15	0,751
5	17,1	15,22	17,12	8,1	8,38	3,01
10	19,9	20,04	23,31	10,09	11,17	5,79
25	28,3	28,9	28,27	19,2	19,53	14,14
50	37,7	37,86	43,39	28,7	29	25,61
75	54,4	54,57	55,2	45,4	45,71	43,17

W przypadku rozmiaru pamięci - aplikacja React osiąga lepsze wyniki niż aplikacja Angular. Potrzebuje też mniej pamięci do poprawnego działania. Zaskakujące jest jednak to, iż aplikacja React najlepsze wyniki osiągnęła w przeglądarce Microsoft Edge. Natomiast aplikacja Angular potrzebuje podobnego rozmiaru pamięci do transferu rekordów we wszystkich przeglądarkach.

Rysunek 3 porównuje rozmiar zużytej pamięci potrzebnej do pobrania rekordów z bazy danych.



Rys. 3. Graficzna analiza zużycia pamięci danych przy pobraniu rekordów z bazy w przeglądarkach

Tabela 5 przedstawia czas jakiego aplikacja potrzebuje do dodania pliku graficznego do bazy danych i magazynu danych Firebase, a także pobrania go oraz wyświetlenia w aplikacji. React w większości przypadków ma lepsze wyniki. Jedynie w przeglądarce Google Chrome aplikacja Angular okazała się szybsza. Różnica między tymi czasami jest niewielka. Obie aplikacje najwięcej czasu na dodanie zdjęcia potrzebowały w przeglądarce Mozilla Firefox.

Tabela 5. Czas dodawania rekordu z plikiem graficznym do bazy danych

Przeglądarka	Angular	React
	Czas transferu [s]	Czas transferu [s]
Google Chrome	4,16	4,26
Mozilla Firefox	5,74	5,11
Microsoft Edge	4,88	3,76

4. Wnioski

Angular2 jest lepszym narzędziem dla bardziej doświadczonego programisty. W tym przypadku nauka od podstaw jest dosyć trudna i wymagająca głębszej znajomości tematu. Ponieważ Angular2 jest frameworkiem - ciężko jest wdrożyć go do istniejącego projektu. Projekt Angular2 trzeba tworzyć od początku.

React jest łatwiejszym do nauki narzędziem niż Angular2. Ma on przewagę nad Angular ponieważ jest to tylko biblioteka. Pozwala to na wdrożenie komponentów React do istniejących już projektów.

Struktura aplikacji Angular2 jest idealna do dużych serwisów, natomiast struktura React dla mniejszych aplikacji. Komponent React opiera się tylko na jednym pliku. Natomiast komponenty Angular posiada 3 pliki, co pozwala na lepszy podział kodu.

Tabela 6 przedstawia ocenę punktową porównywanych narzędzi ze względu na wskazane kryteria. Skala ocen należy do przedziału 1-10, gdzie 1 jest najniższą oceną, a 10 oceną najwyższą.

Tabela 6. Ocena punktowa porównywanych narzędzi

	Angular2	React
Struktura aplikacji	8	7
Metryki kodu	7	9
Dokumentacja	9	9
Spółeczność	9	7
Testy wydajnościowe	7	10
Sumaryczna ocena	40	42

Narzędzia zdobyły zbliżony wynik w ostatecznej ocenie punktowej. Na podstawie przedstawionych tabel i wykresów można wywnioskować, że React lepiej wypada w testach wydajnościowych oraz ma mniejsze rozmiary plików aplikacji. Oba narzędzia posiadają dokumentację na podobnym poziomie. Angular2 jest z kolei narzędziem z większą społecznością.

Literatura

- [1] R. Nowacki, M. Plechawska-Wójcik Analiza porównawcza narzędzi do budowania aplikacji Single, 2016
- [2] Single Page Application (SPA) vs Multi Page Application (MPA): Pros and Cons, <http://merehead.com/blog/single-page-application-vs-multi-page-application/>, [03.08.2018]
- [3] "Single-page_application", https://en.wikipedia.org/wiki/Single-page_application [04.08.2018]
- [4] Angular vs. React - A comparison, <http://work.haufegroup.io/angular-VS-React/> [11.08.2018]
- [5] Vipul A M, Prathamesh Sonpatki, ReactJS by Example - Building Modern Web Applications with React, 2016
- [6] StackOverflow Overview 2018, <https://insights.stackoverflow.com/survey/2018/#overview> [25.08.2018]
- [7] Angular Docs Architecture, <https://angular.io/guide/architecture> [10.08.2018]
- [8] React. <https://reactjs.org/> [18.08.2018]
- [9] <https://helion.pl> [29.08.2018]
- [10] <https://books.google.pl/> [29.08.2018]
- [11] Stack Overflow Trends, <https://insights.stackoverflow.com/trends> [25.08.2018]
- [12] Stack Overflow angular, <https://stackoverflow.com/questions/tagged/angular> [25.08.2018]
- [13] Stack Overflow reactjs, <https://stackoverflow.com/questions/tagged/reactjs> [25.08.2018]
- [14] Introducing Stack Overflow Trends, StackOverflow, <https://stackoverflow.blog/2017/05/09/introducing-stack-overflow-trends/> [25.08.2018]