

**WYŻSZA SZKOŁA INFORMATYKI I ZARZĄDZANIA
„COPERNICUS” WE WROCŁAWIU**

WYDZIAŁ INFORMATYKI, ADMINISTRACJI I FIZJOTERAPII

Kierunek studiów: **Informatyka**

Poziom studiów: **Studia drugiego stopnia-inżynierskie**

Specjalność: **Inżynieria systemów informatycznych**

PRACA DYPLOMOWA INŻYNIERSKA

Daniel Słaby

**Analiza porównawcza nowoczesnych rozwiązań
technologicznych aplikacji SPA za pomocą
jednakowej aplikacji zaimplementowanej w
każdym z wybranych rozwiązań**

**Comparative analysis of modern technological solutions of SPA
applications using the same application implemented in each of the
selected solutions**

Ocena pracy:
(ocena pracy dyplomowej, data, podpis promotora)

.....
(pieczęć uczelni)

Promotor:

Dr inż. Grzegorz Debita

WROCŁAW 2020

Spis treści

1	Wstęp	2
1.1	Wprowadzenie	2
1.2	Cel pracy	2
1.3	Motywacja	2
1.4	Zakres	3
	Część przeglądowa	4
2	Część przeglądowa	5
2.1	Przykładowa sekcja	5
2.1.1	Przykładowa subsekcja	5
2.2	Druga sekcja	6
	Część praktyczna	7
3	Część praktyczna	8
3.1	8
3.1.1	8
4	Podsumowanie	9
	Bibliografia	10
	Spis rysunków	14
	Oświadczenie o udostępnianiu pracy dyplomowej	15
	Oświadczenie autorskie	16

1. Wstęp

1.1 Wprowadzenie

Technologia front-end ewoluuje w zastraszającym tempie. 20 października 2010 rozpoczęła się nowa era technologii webowych wraz z wydaniem biblioteki AngularJS. Był to swoisty początek aplikacji SPA (Single Page Application) i całkowitej zmiany podejścia do tworzenia webowych aplikacji klienckich.

Dzisiaj większość narzędzi jakie używamy w życiu codziennym albo w pełni znajduje się w przeglądarce albo ma już na niej swoje odpowiedniki. Znaczna część aplikacji z których korzystamy na co dzień wykorzystuje te narzędzia jako rdzeń swojego działania. Ze znanych aplikacji z jakich korzysta na co dzień użytkownik możemy wymienić jako przykład odpowiednio: React: Facebook, Instagram, Netflix, WhatsApp Angular 2: GitHub Community Forum, Microsoft Office Home, Google Marketing Platform VueJS: FontAwesome, ChatWoot, Moonitor, Leave Dates

1.2 Cel pracy

W pracy tej, przeanalizowano trzy najpopularniejsze rozwiązania do tworzenia aplikacji SPA. Każde z nich pozwala na stworzenie takiej samej aplikacji z punktu widzenia końcowego użytkownika. Każda z nich jednak w inny sposób rozwiązuje problemy zarządzania zasobami oraz optymalizacji renderowania. W tym celu skonstruuje narzędzie pozwalające na badanie dynamicznych aplikacji webowych i postaram się odpowiedzieć na pytanie, czy różnica wydajności pomiędzy frameworkami jest tak duża, że powinna mieć wpływ na wybór konkretnego rozwiązania w ogólnym przypadku?

1.3 Motywacja

Za motywacją do stworzenia tejże pracy stały dwa główne powody. Pierwszym z nich jest chęć porównania trzech najczęściej wybieranych narzędzi w środowisku front-end. Każde z nich rozwiązuje ten sam problem, jednak nie jest do jasne, jak wygląda ich wydajność w przypadku użytkowania aplikacji. Jest to bowiem problem który trudno zbadać, co prowadzi nas do drugiego powodu. Badanie aplikacji typu SPA w momencie ich działania jest niezwykle trudnym zadaniem, dlatego też, celem praktycznym pracy jest stworzenie narzędzia do rozwiązywania tego problemu.

1.4 Zakres

Zadanie będzie polegać na zaprojektowaniu odpowiedniej aplikacji która zawiera kilka znanych problemów wydajnościowych występujących w świecie aplikacji SPA. Następnie implementacja takiej aplikacji w każdym z wybranych rozwiązań. Kolejno na specjalnie stworzonej maszynie wirtualnej z ograniczonymi zasobami, wykonane zostaną testy wydajnościowe za pomocą wbudowanego profilera.

Mając już konkretne dane, wyniki zostaną przeanalizowane porównując które rozwiązanie jest najwydajniejsze w konkretnym przypadku co pozwoli na odpowiedź na pytanie: czy różnica wydajności pomiędzy narzędziami jest tak duża, że powinna mieć wpływ na wybór konkretnego rozwiązania w ogólnym przypadku?

Część przeglądowa

2. Część przeglądowa

2.1 Przykładowa sekcja

2.1.1 Przykładowa subsekcja

Quia ipsam non animi placeat amet sed. Cumque aut ratione velit expedita occaecati rerum qui. Cumque aut et quo repellat. Dolorem minima molestiae accusamus laboriosam qui. Neque quam qui natus optio quae. Przykładowy Refferal do obrazka nr ??.

Criteria	Frequency
Cache performance	2
Compatibility	4
Developer guidelines	3
Documentation	6
Does “in-house” experience exists?	2
Simple to use	6
Maintainability	5
Maturity	4
Modularity	3
Performance	5
Popularity	8
Portability	4
Scalability	3
Security	2
Framework’s size	1
Testability	3
Who is the developer?	4

Rysunek 2.1. Grafika przedstawiająca ocenę wpływu wybranego elementu na atrakcyjność danego narzędzia

2.2 Druga sekcja

Quia ipsam non animi placeat amet sed. Cumque aut ratione velit expedita occaecati rerum qui. Cumque aut et quo repellat. Dolorem minima molestiae accusamus laboriosam qui. Neque quam qui natus optio quae. Przykładowy Refferal do obrazka nr 2.1.

Tabela 2.1. Przykładowa tabela

Nazwa komponentu	Licencja
Graylog 2.5	GNU General Public License v3
MongoDB 3.4	GNU Affero General Public License v3
Elasticsearch 6.5.1	Elastic License [?]
rsyslog 8.24.0	GNU General Public License v3
NXLog CE 2.10.2150	NXLog Public License
Docker 18.09.1	Apache License 2.0

Część praktyczna

3. Część praktyczna

3.1

3.1.1

Do listingów też można zrobić odniesienie np poniżej mamy listing nr 3.1

Listing 3.1 . Przykładowy listing pliku konfiguracyjnego

```
<Extension _syslog>
  Module xm_gelf
</Extension>

<Input in>
  Module im_msvistalog
</Input>

<Output out>
  Module      om_udp
  Host        192.168.21.211
  Port        12201
  OutputType  GELF
</Output>

<Route 1>
  Path        in => out
</Route>
```

4. Podsumowanie

Bibliografia

- [1] JSON.org, <https://www.json.org/>
- [2] Peleke Sengstacke *JavaScript Transpilers: What They Are and Why We Need Them*, Kwiecień 25, 2016
<https://scotch.io/tutorials/javascript-transpilers-what-they-are-why-we-need->
- [3] Sławomir Kołodziej *What Are Single Page Applications? What Is Their Impact on Users' Experience and Development Process?*, 3 Lipiec 2019
<https://www.netguru.com/blog/what-are-single-page-applications>
- [4] Eric Molin, *Comparison of Single-Page Application Frameworks*, Instytut KTH w Sztokholmie
<https://pdfs.semanticscholar.org/fa9f/f75f32de61cddafa8805ea433d4d8a0e20da.pdf>
- [5] <https://octoverse.github.com/>
- [6] Maja Nowak, *Reasons, Why Vue.js Is Getting More Traction Every Month*, 19 Grudzień 2018,
<https://www.monterail.com/blog/reasons-why-vuejs-is-popular>
- [7] <https://github.com/angular/angular/blob/master/CHANGELOG.md>
- [8] <https://github.com/facebook/react/blob/master/CHANGELOG.md>
- [9] *Probabilistyczny opis błędu jako podstawa definiowania niepewności pojedynczego wyniku pomiaru*
<http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-article-BSW4-0034-0011>
- [10] Google Developers, *Lighthouse*
<https://developers.google.com/web/tools/lighthouse>
- [11] Kyle Simpson *You Don't Know JS: Async & Performance*
http://cdn.lxqnsys.com/05_You_Don't%20Know_JS_Async_&_Performance.pdf

- [12] Jadwiga Kalinowska, Beata Pańczyk, *Porównanie narzędzi do tworzenia aplikacji typu SPA na przykładzie Angular2 i React*, Politechnika Lubelska, Instytut Informatyki
<http://www.diva-portal.org/smash/get/diva2:1037481/FULLTEXT01.pdf>
- [13] Paul Lewis, *Google Developers, Rendering Performance*,
<https://developers.google.com/web/fundamentals/performance/rendering>
- [14] *ReactJS, Lists and Keys*,
<https://reactjs.org/docs/lists-and-keys.html>
- [15] *ReactJS, Virtual DOM and Internals*,
<https://reactjs.org/docs/faq-internals.html>
- [16] *ReactJS, Reconciliation*,
<https://reactjs.org/docs/reconciliation.html>
- [17] GNU.org, *Makefile*,
https://www.gnu.org/software/make/manual/html_node/Introduction.html
- [18] NpmJs.com, *npm | build amazing things*,
<https://www.npmjs.com/>
- [19] *ReactJS, Optimizing Performance*,
<https://reactjs.org/docs/optimizing-performance.html>
- [20] magnifier.pl, *Konteneryzacja - czym jest i dlaczego staje się tak popularna?*, 24 Październik. 2019,
<https://magnifier.pl/konteneryzacja-docker-kubernetes/>
- [21] MDN web docs, *Promise - JavaScript*,
https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise
- [22] MDN web docs, *What is a web server?*, 18 Stycznia 2019,
https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
- [23] Docker Docs, *Docker*,
<https://docs.docker.com/get-started/#docker-concepts>

- [24] Nginx, *Nginx for Windows*,
<http://nginx.org/en/docs/windows.html>
- [25] Nginx, *Linux packages*,
http://nginx.org/en/linux_packages.html
- [26] Wiki Alpine Linux, *Docker*,
<https://wiki.alpinelinux.org/wiki/Docker>
- [27] Docker Hub, *Ubuntu*,
https://hub.docker.com/_/ubuntu
- [28] Meggin Kearney, Addy Osmani, Kayce Basques, Jason Miller, *Measure Performance with the RAIL Model | Web Fundamentals*, 12 Luty. 2019,
<https://developers.google.com/web/fundamentals/performance/rail>
- [29] Jesus Castello, *Ruby Templating Engines: ERB, HAML & Slim - RubyGuides*,
<https://www.rubyguides.com/2018/11/ruby-erb-haml-slim/>
- [30] Whatsabyte, *What Are Threads in a Processor?*, 24 Sierpień 2018,
<https://whatsabyte.com/blog/processor-threads/>
- [31] Dimiter Petrov, *A tale of flaky Cypress tests*, 24 Październik 2019,
<https://dimiterpetrov.com/blog/a-tale-of-flaky-cypress-tests/>
- [32] Selenium Dev, *The Selenium project and tools*,
https://www.selenium.dev/documentation/en/introduction/the_selenium_project_and_tools/
- [33] Docker Docs, *Overview of Docker Compose*,
<https://docs.docker.com/compose/>
- [34] Docker Documentation, *Networking in Compose*.
<https://docs.docker.com/compose/networking/>
- [35] MDN - Mozilla, *Memory Management*, 4 Marca 2020,
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Memory_Management
- [36] MDN - Mozilla, *Performance - Web APIs*, 30 Styczeń 2020,
<https://developer.mozilla.org/en-US/docs/Web/API/Performance>

- [37] MDN - Mozilla, *DOMHighResTimeStamp* - *Web APIs*, 19 Lusty 2020,
<https://developer.mozilla.org/en-US/docs/Web/API/DOMHighResTimeStamp>
- [38] MDN - Mozilla, *First paint* - *MDN Web Docs Glossary*, 10 Marca. 2020,
https://developer.mozilla.org/en-US/docs/Glossary/First_paint

Spis rysunków

2.1	Grafika przedstawiająca ocenę wpływu wybranego elementu na atrakcyjność danego narzędzia	5
-----	--	---

Wrocław, dnia 2020-05-08

Wydział Informatyki, Administracji i Fizjoterapii

Kierunek studiów: **informatyka (INF)**

Daniel Słaby

.....

(imię i nazwisko studenta)

6781

.....

(nr albumu)

**OŚWIADCZENIE O UDOSTĘPNIANIU PRACY
DYPLOMOWEJ**

Tytuł pracy dyplomowej: Analiza porównawcza nowoczesnych rozwiązań technologicznych aplikacji SPA za pomocą jednakowej aplikacji zaimplementowanej w każdym z wybranych rozwiązań

Wyrażam zgodę (nie wyrażam zgody)¹ na udostępnianie mojej pracy dyplomowej.

.....

(podpis studenta)

¹Niepotrzebne skreślić.

Wrocław, dnia 2020-05-08

Wydział Informatyki, Administracji i Fizjoterapii

Kierunek studiów: **informatyka (INF)**

Daniel Słaby

.....

(imię i nazwisko studenta)

6781

.....

(nr albumu)

OŚWIADCZENIE AUTORSKIE

Oświadczam, że niniejszą pracę dyplomową pod tytułem:

Analiza porównawcza nowoczesnych rozwiązań technologicznych aplikacji SPA za pomocą jednakowej aplikacji zaimplementowanej w każdym z wybranych rozwiązań

napisałem/am samodzielnie. Nie korzystałem/am z pomocy osób trzecich, jak również nie dokonałem/am zapożyczeń z innych prac.

Wszystkie fragmenty pracy takie jak cytaty, ryciny, tabele, programy itp., które nie są mojego autorstwa, zostały odpowiednio zaznaczone i zamieszczono w pracy źródła ich pochodzenia. Treść wydrukowanej pracy dyplomowej jest identyczna z wersją pracy zapisaną na przekazywanym nośniku elektronicznym.

Jednocześnie przyjmuję do wiadomości, że jeżeli w przypadku postępowania wyjaśniającego zebrany materiał potwierdzi popełnienie przeze mnie plagiatu, skutkować to będzie niedopuszczeniem do dalszych czynności w sprawie nadania mi tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz złożenie zawiadomienia o popełnieniu przestępstwa.

.....

(podpis studenta)