

Comparison of web application performance on the example of Laravel and Vaadin frameworks

Porównanie wydajności aplikacji internetowych na przykładzie szkieletów programistycznych Laravel i Vaadin

Jakub Radomski*

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

Abstract

The article presents the result of comparing the performance of web applications written with the use of Laravel and Vaadin frameworks. The aim of the study was to identify which framework offers better time efficiency and to verify the thesis: "A web application created using the Vaadin framework is more efficient than an application created using the Laravel framework". A research method based on a comparative analysis was used to conduct the study and with the help of two applications was tested the speed of resource generation, or the times obtained in work under load. Tests were carried out according to 13 research scenarios and the obtained results allowed to clearly verify the thesis was proven.

Keywords: Laravel; Vaadin; Web application; Performance

Streszczenie

Artykuł przedstawia wyniki porównania wydajności aplikacji internetowych napisanych przy użyciu szkieletów aplikacji Laravel oraz Vaadin. Celem badania było wskazanie, który szkielet oferuje lepszą wydajność czasową oraz zweryfikowanie postawionej tezy brzmiącej "Aplikacja internetowa stworzona przy użyciu platformy programistycznej Vaadin jest bardziej wydajna niż aplikacja stworzona za pomocą platformy Laravel". Do przeprowadzenia badania wykorzystano metodę badawczą opartą na analizie porównawczej i przy pomocy dwóch aplikacji zbadano między innymi prędkość generowania zasobów, czy czasy uzyskane w pracy pod obciążeniem. Przeprowadzono próby według 13 scenariuszy badań a otrzymane wyniki pozwoliły jednoznacznie zweryfikować, że postawiona teza została udowodniona.

Słowa kluczowe: Laravel; Vaadin; Aplikacja internetowa; Wydajność

*Corresponding author

Email address: jakub.radomski@pollub.edu.pl (J. Radomski)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

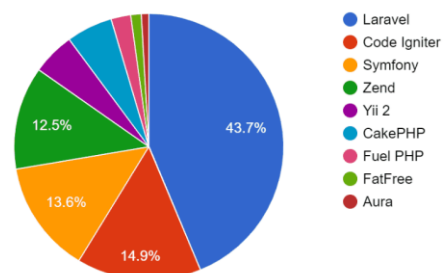
W dzisiejszych czasach programiści mają do wyboru mnogość narzędzi i języków a dodatkowo każdy język oferuje różne szkielety do budowy aplikacji mające na celu ułatwić ich tworzenie.

Aplikacja internetowa może zostać napisana przy użyciu różnych języków programowania. Ta sama aplikacja może powstać z użyciem języka PHP lub, na przykład wykorzystując język Java i będzie oferowała te same funkcje oraz ten sam interfejs użytkownika. Jednak nie każdy język, czy szkielet do budowy aplikacji cechuje się tą samą prostotą użycia, mnogością dostępnych gotowych rozwiązań, czy wydajnością działa aplikacji. Na tym ostatnim aspekcie skupia się ten artykuł, porównując wydajność popularnego wśród programistów PHP szkieletu Laravel (Rysunek 1) oraz zyskującego powoli na popularności szkieletu Vaadin (Rysunek 2).

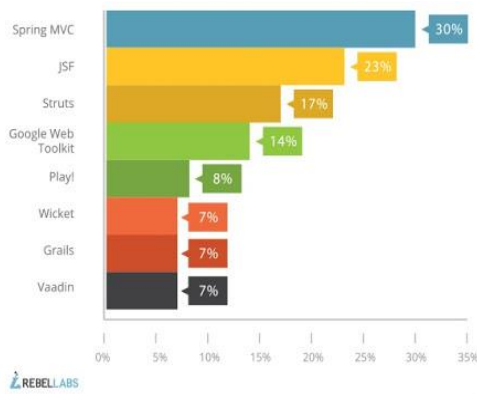
2. Przegląd literatury

Wybrane szkielety do tworzenia aplikacji internetowych nie miały bezpośredniego porównania wydajności i jest to jeden z powodów dla, którego zdecydowano się przeprowadzić to badanie. Laravel jako rozwiązanie, które zyskało sporą popularność na przestrzeni ostatnich

lat, doczekało się wielu artykułów naukowych. Poruszano w nich tematykę wydajności względem innych szkieletów do tworzenia aplikacji, subiektywną ocenę łatwości używania, czy liczby dostępnych gotowych rozwiązań jakie oferuje. Co jednak jest ciekawe, w kwestii wydajności można natrafić na prace, które są ze sobą sprzeczne.



Rysunek 1: Popularność szkieletów PHP w roku 2018 [1].



Rysunek 2: Popularność szkieletów Java w roku 2020 [2].

Artykuł napisany przez Przemysława Mincewicza [3] bada aplikacje napisane z użyciem szkieletów Laravel, Symfony oraz Zend pod kątem wydajności i możliwości. Porównana zostaje wydajność czasowa 3 aplikacji. Autor po wykonaniu 23 zaplanowanych scenariuszy badań dochodzi do wniosku, że Laravel “wypadł, zdecydowanie najslabiej ze wszystkich porównywanych frameworków”. Zostały zbadane takie aspekty jak czas dostępu do bazy danych, czy czas ładowania plików graficznych. W artykule zostało przedstawionych 9 wykresów dotyczących badań wydajności, gdzie w 4 z nich szkielet Laravel wcale nie przegrał z Symfony i Zend, więc z powyższym stwierdzeniem autora można się nie zgodzić. Z kolei artykuł zatytułowany “A Comparative study of PHP frameworks performance” [4] pokazuje szkielet Laravel z innej strony. Autorzy również porównywali szkielet Laravel, tym razem jednak z Symfony i Codeigniter. Doszli do wniosku, że jest szkielet Laravel jest najwydajniejszy względem dwóch pozostałych. Wniosek ten opracowali na podstawie zrealizowanych 4 scenariuszy badań. Zbadali aplikacje nie tylko pod względem wydajności czasowej ale również wykorzystanie pamięci wszystkich 3 aplikacji. Powstał również artykuł porównujący wydajność Laravel z platformą programistyczną Spring [5]. Jest to więc porównanie już dwóch różnych języków. Autorzy porównywali takie aspekty jak bezpieczeństwo czy wydajność czasową. Analizę wydajności przeprowadzili przy pomocy narzędzia JMeter oraz Chrome DevTools, dzięki któremu zbadali czasy odpowiedzi dla obydwu platform. Zbadali między innymi czas odpowiedzi strony głównej aplikacji, czy czas odpowiedzi dla wybranej podstrony. Prawie pięciokrotnie lepsze rezultaty uzyskał Spring. Ciekawe porównanie zrealizowano w pracy pod tytułem “Performance Comparison of QEC Network based JAVA Application and Web based PHP Application”, ponieważ badanie zrealizowano bez użycia szkieletów do budowy aplikacji, a więc tylko z wykorzystaniem języków Java i PHP [6]. Jest to kolejna praca, w której zbadano wydajność czasową przy użyciu narzędzia JMeter. Również w tym badaniu Java okazała się szybsza względem PHP. Platformę Spring porównał również w swoim artykule Łukasz Tomczyk [7]. Jednak w jego pracy platforma ta została zestawiona z platformą Vaadin. Do testów autor przygotował 9 scenariuszy. Autor skupił się na wydajności czasowej i podczas

realizacji zaplanowanych scenariuszy zbadał efektywność pracy z bazą danych oraz efektywność wczytywania zasobów w przeglądarce internetowej. W jego badaniu zarówno Vaadin, jak i Spring wykazywały miejsca w, których okazywały się lepsze niż druga badana platforma. Należy odnotować, że wszystkie badania przeprowadzone w wyżej wymienionych artykułach, każde zestawienie platform, czy języków zostało zrealizowane na aplikacjach, które posiadała możliwie identyczny wygląd, mechanikę działania oraz identyczną bazę danych w celu zapewnienia możliwie jak najbardziej miarodajnych wyników. Platformy programistyczne, jak również i języki z wykorzystaniem których zostały zbudowane, ciągle się zmieniają, otrzymują następną wersję. Należy mieć więc na uwadze, że ich wydajność będzie się stale zmieniać i rezultaty niektórych badań mogłyby być teraz inne.

3. Cel, teza i metoda badawcza

Celem badania było porównanie wydajności czasowej obydwu platform programistycznych poprzez sprawdzenie jakie wyniki uzyskują w testach sprawdzających, między innymi prędkość generowania zasobów, czy czasy uzyskane w pracy pod obciążeniem. Postawiono również tezę brzmiącą: “Aplikacja internetowa stworzona przy użyciu platformy programistycznej Vaadin jest bardziej wydajna niż aplikacja stworzona za pomocą platformy Laravel”. Dla potwierdzenia przedstawionej tezy wykorzystano metodę badań opartą na analizie porównawczej obu rozwiązań i w tym celu stworzono dwie aplikacje różniące się jedynie narzędziem użytym do ich stworzenia, natomiast baza danych, czy funkcjonalności zostały zachowane te same.

4. Aplikacje testowe

Aplikacje Laravel oraz Vaadin:

- stworzono w środowisku programistycznym IntelliJ z wykorzystaniem narzędzia Maven oraz przy użyciu środowiska PhpStorm z wykorzystaniem narzędzia Composer [8-11]. Zarówno Maven, jak i Composer pozwalają na automatyzację budowy projektów.
- współpracują z serwerem bazy danych MySQL [12].

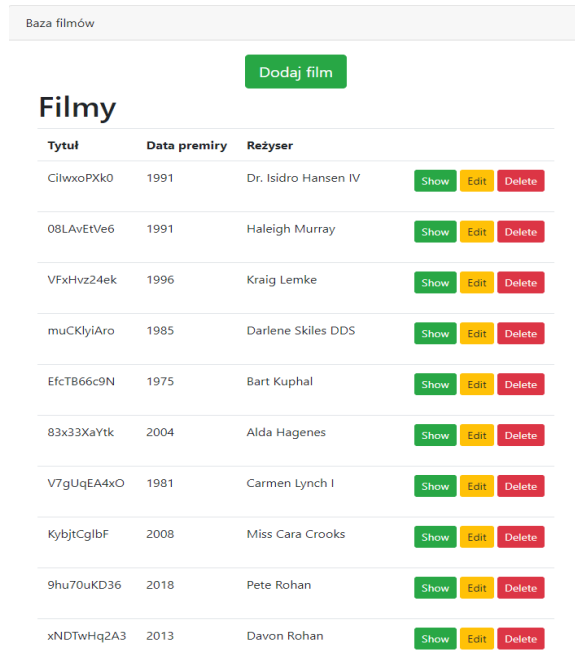
Obie aplikacje testowe posiadają typowe funkcjonalności CRUD (ang. Create, Read, Update, Delete), czyli tworzenie, edycję oraz usuwanie informacji o filmach bazy danych. Interfejs obu tych aplikacji jest bardzo podobny i przykładowa strona aplikacji została przedstawiona na Rysunku 3.

5. Analiza wydajności

W celu zbadania wydajności stworzonych aplikacji skorzystano z Chrome DevTools, narzędzie to jest wbudowane w przeglądarkę Google Chrome oraz programu ApacheBench, który pozwala symulować obciążenie aplikacji [13-14]. Analizę wydajnościową przedstawiono w postaci wykresów opisujących badane aspekty. Każdy z nich zawiera zestawienie wszystkich badanych narzędzi na tle dwóch zestawów danych. W badaniach

zwrócono uwagę na czas, opisany na wykresach w milisekundach (ms).

Założono 13 scenariuszy badań, których opisy przedstawiono w Tabeli 1. Każdy badany scenariusz składa się z 10 powtórzeń oraz wyliczonych średnich czasów dla każdego z badanych szkieletów.



Rysunek 3: Przykładowa strona aplikacji testowej.

Tabela 1: Scenariusze badania wydajności

Scenariusz	Opis
1	Generowanie strony z 10 rekordami
2	Generowanie strony ze 100 rekordami
3	Generowanie strony z 1000 rekordów
4	Generowanie strony z 10000 rekordów
5	Zapis 1 rekordu
6	Zapis 10 rekordów
7	Zapis 100 rekordów
8	Usunięcie rekordu z bazy
9	Generowanie strony startowej
10	10 pobrań strony startowej symulowanej przez 1 użytkownika w jednej chwili
11	1000 pobrań strony startowej symulowanej przez 50 użytkowników w jednej chwili
12	10 pobrań listy 10 filmów symulowanej przez 1 użytkownika w jednej chwili
13	1000 pobrań listy 10 filmów symulowanej przez 50 użytkowników w jednej chwili

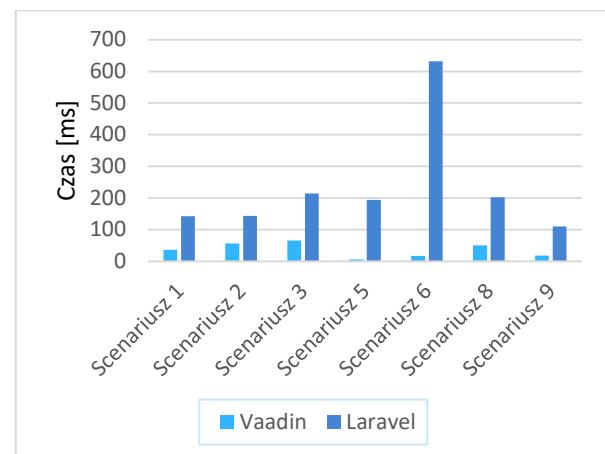
Badania zostały przeprowadzone na jednej jednostce pomiarowej, której parametry wraz z wersjami wykorzystanych technologii przedstawia Tabela 2.

Tabela 2: Parametry sprzętowe jednostki pomiarowej oraz wersje wykorzystanych technologii

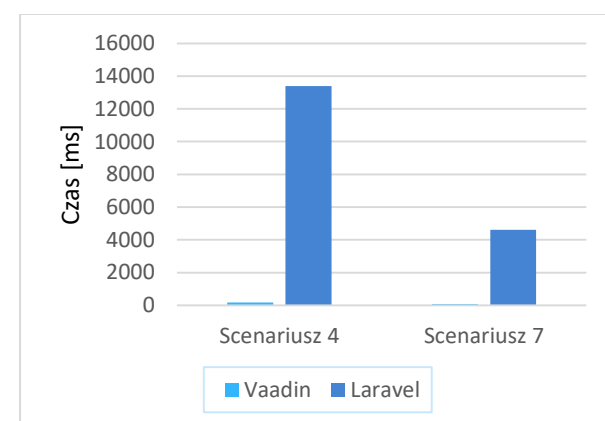
System operacyjny	Windows 10 Pro 64-bitowy
Procesor	Intel Core i5-9400F
Pamięć RAM	16 GB
PHP	Wersja 7.4
Java	Wersja 8
Laravel	Wersja 8.5
Vaadin	Wersja 14

5.1. Analiza z wykorzystaniem Chrome DevTools

Badanie z wykorzystaniem narzędzia Chrome DevTools dotyczy pierwszych 9 scenariuszy. Ich wyniki są przedstawione na Rysunkach 4 i 5. Zostały one podzielone na 2 oddzielne rysunki w celu zachowania lepszej czytelności wyników.



Rysunek 4: Średnie czasy operacji scenariuszy testowych.



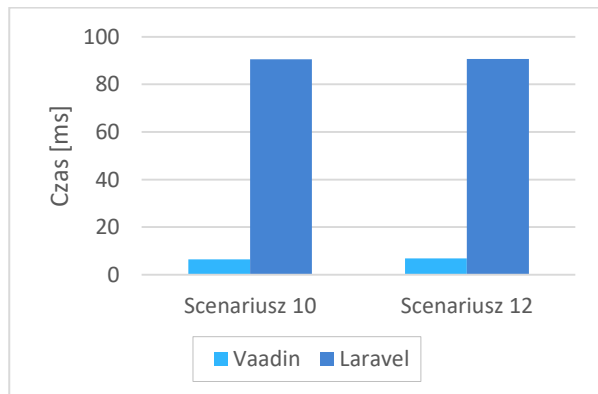
Rysunek 5: Średnie czasy operacji dla scenariusz 4 i 7.

Szkielet Vaadin osiągnął lepsze wyniki od szkieletu Laravel w każdym z 9 scenariuszy. Stopniowe zwiększanie liczby rekordów w 2 i 3 scenariuszu ma minimalny wpływ na otrzymywane czasy. Przewagę szkieletu Vaadin szczególnie widać w przypadkach zakładają-

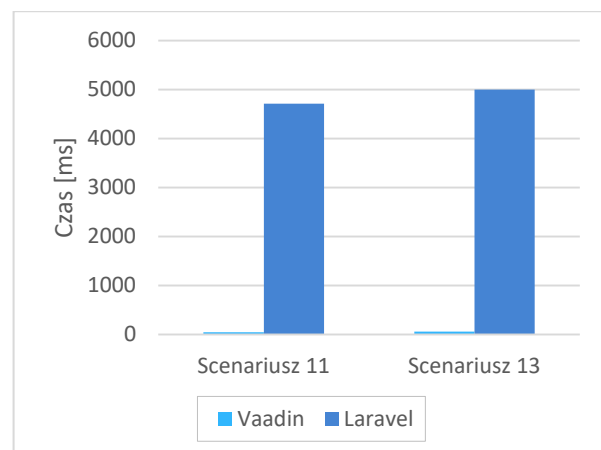
cych operacje na co raz większej liczbie rekordów tak jak podczas realizacji scenariusza 4 i 7.

5.2. Analiza z wykorzystaniem ApacheBench

Badanie z wykorzystaniem narzędzia ApacheBench dotyczy ostatnich 4 scenariuszy. Ich wyniki są przedstawione na Rysunkach 6 oraz 7. Zostały one podzielone na 2 oddzielne rysunki w celu zachowania lepszej czytelności wyników.



Rysunek 6: Średnie czasy operacji dla scenariusz 10 i 12.



Rysunek 7: Średnie czasy operacji dla scenariusz 11 i 13.

Również dla scenariuszy z wykorzystaniem narzędzia ApacheBench szkielet Vaadin osiągnął lepsze wyniki od szkieletu Laravel, wygrywając w każdym z pozostałych 4 scenariuszy.

6. Wnioski

Zaprezentowane wyniki badań dla podanych 13 scenariuszy jasno pokazują, że Laravel jest szkieletem mniej wydajnym. Vaadin uzyskał lepsze wyniki czasowe dla każdego prezentowanego scenariusza co jednoznacznie potwierdza postawioną na początku tezę mówiącą, że szkielet ten będzie szkieletem wydajniejszym.

Vaadin największą przewagę pokazuje w przypadku, gdy zaczęto zwiększać liczbę rekordów do 1000, 10000 w kwestii odczytu oraz do 100 podczas zapisu. Takie sytuacje, gdy trzeba było wygenerować stronę z co raz większą liczbą danych z bazy, nie stanowiły dla tego szkieletu problemu. Jest to spowodowane tym, że szkielet ten powstał głównie do zastosowań frontendowych,

więc operacje związane ze stroną klienta są mocną stroną tego szablonu.

Laravel pomimo tego, że oferuje obsługę strony klienta, między innymi z użyciem szablonów Blade, to jest to szkielet głównie do obsługi części serwerowej, a więc generowanie wizualne co raz większej liczby rekordów, jak to miało miejsce podczas scenariusza 3 oraz 4, może już sprawiać temu szkieletowi problemy. Również zapis nawet 100 rekordów do bazy nie był problemem dla szkieletu Vaadin. Wpływ na to może mieć fakt, że Laravel nie jest szkieletem, który stawia przede wszystkim na wydajność, ale skupia się w dużej mierze na tym by programiście udostępnić możliwie jak najwięcej gotowych rozwiązań, czy modułów, które ułatwią mu pracę nad aplikacją ale jednocześnie mogą mieć mniej pozytywny wpływ na prędkość jej działania.

W przypadku testów z wykorzystaniem narzędzia ApacheBench Vaadin również okazał się nieporównywalnie wydajniejszym szkieletem. Nawet testy symulujące pobieranie zasobów przez 50 użytkowników w tym samym momencie nie okazały się wyzwaniem dla tego szkieletu. Warty odnotowania jest również fakt, że im większa liczba użytkowników jest symulowana w danej chwili tym wyniki bardziej się różnią między pomiarami.

Literatura

- [1] Popularność szkieletów PHP z roku 2018, <https://www.excellentwebworld.com/>, [20.10.2021].
- [2] Popularność szkieletów Java z 2017, <https://www.jrebel.com/blog/java-web-framework-usage-stats>, [20.10.2021].
- [3] P. Mincewicz, M. Plechawska-Wójcik, Performance and possibility analysis of Laravel tool dedicated to create modern web applications, Journal of Computer Sciences Institute 7 (2018) 108-115, <https://doi.org/10.35784/jcsi.655>.
- [4] M. Laaziri, K. Benmoussa, S. Khouli, M. L. Kerkeb, A Comparative study of PHP frameworks performance, Procedia Manufacturing 32 (2019) 864-871, <https://doi.org/10.1016/j.promfg.2019.02.295>.
- [5] S. Jędrych, B. Jędruszek, B. Pańczyk, Comparative analysis of web applications development using JEE and PHP, Journal of Computer Sciences Institute 11 (2019) 86-90, <https://doi.org/10.35784/jcsi.145>.
- [6] S. Memon, R. B. Palh, M. Memon, H. Memon, Performance Comparison of QEC Network based JAVA Application and Web based PHP Application, International Journal of Advanced Computer Science and Applications 9 (8) (2018), <http://dx.doi.org/10.14569/IJACSA.2018.09.0870>.
- [7] Ł. Tomczyk, B. Pańczyk, Comparison of web applications development using Spring MVC and Vaadin, Journal of Computer Sciences Institute 6 (2018) 1-5, <https://doi.org/10.35784/jcsi.631>.
- [8] Oficjalna dokumentacja narzędzie IntelliJ, <https://www.jetbrains.com/idea/>, [23.10.2021].

- [9] Oficjalna dokumentacja narzędzia Maven, <https://maven.apache.org/guides/>, [23.10.2021].
- [10] Oficjalna dokumentacja narzędzia PhpStorm, <https://www.jetbrains.com/phpstorm/>, [23.10.2021].
- [11] Oficjalna dokumentacja narzędzia Composer, <https://getcomposer.org/doc/>, [23.10.2021].
- [12] Oficjalna dokumentacja narzędzia MySQL, <https://dev.mysql.com/doc/>, [23.10.2021].
- [13] Oficjalna dokumentacja narzędzie ApacheBench, <https://httpd.apache.org/docs/2.4/programs/ab.html>, [23.10.2021].
- [14] Oficjalna dokumentacja narzędzia Chrome DevTools, <https://developer.chrome.com/docs/devtools/>, [23.10.2021].