

Analiza wydajnościowa frameworka Symfony do tworzenia nowoczesnych aplikacji webowych na podstawie wybranych wersji

Wójcik Aleksander, Wolski Mateusz*, Jakub Bartłomiej Smółka

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

Streszczenie. Tematyką badaną w artykule jest sprawdzenie wydajności frameworka Symfony do tworzenia aplikacji webowych. Przedstawiono przegląd literatury mówiącej o Symfony oraz jego najpopularniejsze moduły. Na podstawie stworzonych trzech identycznych aplikacji testowych napisanych we frameworkach Symfony 2.8, 3.4 oraz 4.2 porównano je ze sobą pod względem wydajnościowym. Aplikacja testowa została napisana w formacie bloga. Został wykorzystany styl architektury oprogramowania, znany jako API. Dzięki temu możliwe jest przeprowadzenie zaplanowanych testów. Badanie wydajności narzędzia Symfony dla poszczególnych wersji sporządzono na podstawie między innymi czasu ładowania danych z bazy oraz ich wyszukiwania w kolekcji danych, dodatkowymi testami było pobieranie danych z pliku csv oraz zapisywanie ich do pliku csv.

Słowa kluczowe: framework Symfony; język PHP; MVC; tworzenie aplikacji w frameworku Symfony; testy wydajnościowe

*Autor do korespondencji.

Adresy e-mail: olekwójcik1@gmail.com, maaateusz94@gmail.com

Performance analysis of the Symfony framework for creating modern web application based on selected versions

Wójcik Aleksander, Wolski Mateusz*, Jakub Bartłomiej Smółka

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

Abstract. The subject reached in the article is to check the performance of the Symfony framework for creating web applications. An overview of the literature about Symfony and the most popular modules. Based on the created three identical test applications written in the Symfony 2.8, 3.4 and 4.2 frameworks, they were compared with each other in terms of performance. The test application was written in the blog format. The software architecture style known as the API has been used. Related to this, it is possible to conduct scheduled tests. Symfony's performance testing for individual versions was based on, the time of loading data from the database and their search in the data collection, additional tests were to download data from a csv file and save them to a csv file.

Keywords: framework Symfony; programming language PHP; MVC; creating of the web application in framework Symfony; performance tests

*Corresponding author.

E-mail addresses: olekwójcik1@gmail.com, maaateusz94@gmail.com

1. Wstęp

W dzisiejszych czasach przesyłanie informacji pomiędzy aplikacjami np. webowymi, a mobilnymi staje się coraz bardziej popularne. Większość systemów informatycznych korzysta z relacyjnych baz danych, dodatkowo znaczna ich część posiada bazy dane zaprojektowane w sposób obiektowy, a są to proste firmowe strony internetowe kończąc na zaawansowanych systemach obsługujących sklepy internetowe. Dzięki technologii API REST, dane oraz funkcjonalności można integrować między wieloma systemami informatycznymi.

2. Przegląd literatury

Wraz ze wzrostem zapotrzebowania na udoskonalanie aplikacji webowych coraz bardziej popularne zaczęły być frameworki [1]. Pozwalają one na zwiększenie wydajności programowania. Pierwszą główną funkcją frameworków jest wykorzystanie wbudowanych metod, które zajmują setki linii kodu oraz są bardzo czasochłonne w przypadku

samodzielnego przygotowania. Drugim aspektem, w którym można znaleźć bardzo dobre zastosowanie frameworków są kwestie bezpieczeństwa, do których wykorzystywani są użytkownicy/programiści stający się testerami własnych aplikacji. W przypadku odkrycia luki w bezpieczeństwie twórcy frameworka utworzyli żywą społeczność, dającą możliwość raportowania o nieścisłościach. Kolejnym zaletą korzystania z frameworków jest ich darmowy dostęp, ponieważ frameworki pozwalają pisać programistom kod szybciej, z mniejszą ilością błędów, to ostateczny koszt wytworzenia aplikacji będzie mniejszy. Ostatnią korzyścią korzystania z frameworków jest ich rozbudowany zespół wsparcia, dokumentacja czy duże fora zwolenników danych ram programowania, dzięki temu można często znaleźć rozwiązania do różnych problemów.

3. Przegląd wybranych technologii

REST to wzór projektowy zdefiniowany w celu ułatwienia tworzenia i organizowania systemów rozproszonych. Kluczowym słowem z tej definicji powinien być styl, ponieważ ważnym aspektem REST jest to, że to styl

architektoniczny - nie jest wytyczną, nie jest standardem, oznaczałoby, że istnieje zestaw twardych reguł, które należy zastosować, aby ostatecznie uzyskać architekturę REST [2]. Główną ideą REST jest to, że system ten należy do systemów rozproszonych, zorganizowanych RESTfully. Główne cechy takiego systemu to:

- wydajność: styl komunikacji zaproponowany przez REST ma być wydajny i prosty, umożliwiając zwiększenie wydajności systemów, które go implementują,
- skalowalność: każdy system rozproszony powinien być w stanie obsłużyć ten aspekt wystarczająco dobrze, a prosta interakcja zaproponowana przez REST pozwala na to w dużym stopniu,
- prostota interfejsu: prosty interfejs umożliwia prostsze interakcje między systemami, które z kolei mogą przynosić korzyści podobne do wcześniej wspomnianych,
- modyfikowalność komponentów: rozproszony charakter systemu i oddzielenie problemów zaproponowanych przez REST umożliwia modyfikowanie komponentów niezależnie od siebie przy minimalnym koszcie i ryzyku,
- przenośność: REST może być zaimplementowana i wykorzystana przez każdy rodzaj technologii.

Framework, czyli szkielet do budowy aplikacji. Definiuje on strukturę aplikacji oraz ogólny mechanizm jej działania, a także dostarcza zestaw komponentów i bibliotek ogólnego przeznaczenia do wykonywania określonych zadań.

Główne zalety wykorzystania frameworka [7]:

- zdefiniowana z góry struktura, która jest znana oraz przemyślana przez wiele osób,
- społeczność, która przyczynia się do ulepszenia ramowej bazy kodu
- opracowany zestaw funkcji, które są niezbędne do wykorzystania przy nowych projektach,
- moduły, biblioteki, wtyczki, które można dodać do aplikacji,
- integracja z ORM (Object-Relational Mapping),
- wstępnie ustalone wzorce projektowe w aplikacji,
- często framework narzuca dostosowanie się do pewnego wzorca projektowego, co prowadzi do lepiej zorganizowanego kodu, lepsza testowalność kodu.

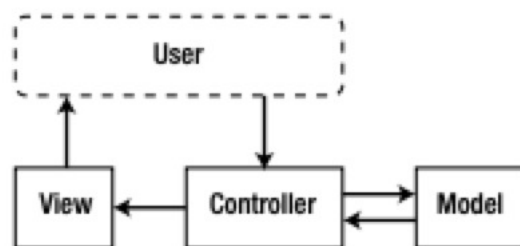
MVC (Model-View-Controller) to wzorzec projektowania oprogramowania zbudowany na bazie połączenia trzech głównych typów komponentów, w języku programowania takim jak PHP, często z silnym naciskiem na paradygmaty oprogramowania obiektowego (OOP). Te trzy typy komponentów są luźno określane jako modele, widoki i kontroler [3].

Model jest miejscem, gdzie zachowywana jest cała logika biznesowa aplikacji. Logika biznesowa może być czymś specyficznym dla aplikacji przechowującej dane lub korzystającej z usług stron trzecich w celu spełnienia wymagań biznesowych. Jeśli aplikacja powinna uzyskać dostęp do informacji w bazie danych, kod odpowiedzialny za to będzie przechowywany w modelu. Jeśli jest potrzeba na

przykład do pobrania danych giełdowych, kod ten będzie również przechowywany w modelu.

Widok to miejsce, gdzie przechowywane są wszystkie elementy interfejsu użytkownika naszej aplikacji. Może to obejmować kod HTML, arkusze stylów CSS lub pliki JavaScript. Wszystko, co widzi użytkownik lub elementy z którymi wchodzi w interakcję, może być przechowywane w widoku [4].

Architektura MVC składa się z następujących elementów model, widok, kontroler została przedstawiona na rys. 1. Kontroler jest komponentem łączącym modele i widoki. Izolują one logikę biznesową modelu od elementów interfejsu użytkownika w widoku oraz obsługują to w taki sposób, aby aplikacja mogła reagować na interakcję użytkownika w widoku. Kontrolery są pierwszym punktem wejścia w to trio komponentów, ponieważ żądanie jest najpierw przekazywane kontrolerowi, który następnie tworzy modele i widoki wymagane do spełnienia żądania w aplikacji [3].



Rys. 1. Architektura MVC

PHP to skryptowy język programowania - napisane w nim programy nie są kompilowane do postaci kodu maszynowego, lecz wykonywane przez interpreter PHP. PHP został stworzony do tworzenia dynamicznych stron WWW [5].

PHP jest projektem typu open-source. Każdy może pobrać za darmo jego kopię, zainstalować i używać bez żadnych ograniczeń. Do kodu źródłowego zapewniony jest pełny dostęp - jeżeli programista ma odpowiednie zdolności, może go modyfikować do woli oraz nadsyłać własne propozycje zmian do osób nadzorujących projekt. Dzięki takiej wolności PHP rozwija się bardzo dynamicznie, a w Internecie można znaleźć setki modyfikacji oraz dodatkowych modułów [6].

Symfony wspierane przez SensioLabs, zostało założone przez Fabien Potencier pod koniec 2004 roku w celu szybszego tworzenia serwisów internetowych. Wkrótce po jego utworzeniu zdecydował się na udostępnienie oprogramowania na licencji MIT. Obecnie jest jednym z najbardziej popularnych frameworków PHP [12]. Framework Symfony jest rozwiązaniem wieloplatformowym, oznacza to, że można z niego korzystać niezależnie od posiadanego systemu operacyjnego. Do głównych cech Symfony, na pewno można zaliczyć [7]:

- wzorzec projektowy MVC - opisany dokładniej w rozdziale 2.2,
- programowanie obiektowe,

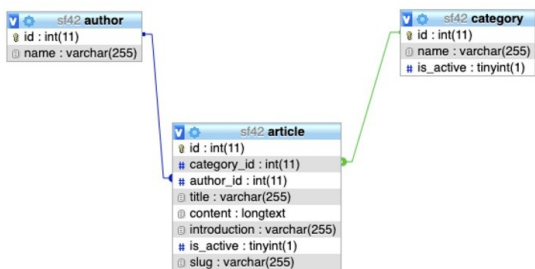
- łatwość instalacji i konfiguracji,
- zarządzanie sesjami,
- wbudowana internacjonalizacja,
- system formularzy,
- integracja z Doctrine.

4. Metoda badawcza

Praca składa się z trzech takich samych aplikacji testowych napisanych w trzech wybranych wersjach frameworka Symfony. Przygotowana aplikacja została stworzona w oparciu o popularny format bloga. Został wykorzystany styl architektury oprogramowania, znany jako API. Dzięki temu możliwe jest przeprowadzenie następujących testów:

- obciążenie wydajnościowe aplikacji wyznaczającej ciąg Fibonacciego, przy $n=1000$, gdzie n oznacza wygenerowane liczby ciągu Fibonacciego,
- pobranie 1000 pierwszych aktywnych artykułów z bazy danych MySQL,
- pobranie listy artykułów o ograniczeniu wyników do 50 dla każdego testu, z zastosowaniem losowych filtrów kategorii oraz autorów,
- generowanie pliku csv z danymi pobranymi z przygotowanej bazy danych,
- import danych z pliku csv do bazy danych,
- Testowa baza obejmuje następujące tabele:
- Article – informację na temat artykułów,
- Author – dane poszczególnych autorów,
- Category – informację na temat kategorii.

Na potrzeby testów została wygenerowana baza danych, zawierająca ok. 21 000 rekordów losowych danych. Schemat bazy danych na rys. 2.



Rys. 2. Schemat bazy danych

Opis środowiska testowego na którym zostały przeprowadzone testy, znajduje się w tabeli 1.

Tabela 1. Specyfikacja urządzenia fizycznego

Nazwa	Wartość
Procesor	Intel Core i5
Taktowanie	2,3 GHz
Liczba rdzeni	4
Pamięć RAM	8 GB
Pamięć flash	512 GB SDD
System operacyjny	macOS Mojave 10.14.4

Eksperyment polegał na przeprowadzeniu badań na trzech takich samych aplikacjach testowych napisanych w następujących wersjach frameworka Symfony 2.8, 3.4 i 4.2. Wersja 2.8 oraz 3.4 są wersjami LTS (Long-Term Support), czyli z najdłuższym wsparciem autora. Natomiast wersja 4.2 była w trakcie badań najnowszą stabilną wersją. Aplikacje testowe mają za zadanie zrealizować te same scenariusze. Każdy scenariusz, realizowany w powyższych frameworkach, pozwala na odmierzenie czasu w milisekundach jaki potrzebuje aplikacja na wykonanie przygotowanego badania. W zależności od testów zostały przygotowane dwie procedury testowe, pierwsza z nich dotyczyła testów eksportowania i importowania danych z pliku csv. Opis scenariusza został przedstawiony w tabeli 2.

Tabela 2. Opis pierwszej procedury testowej

Lp	Opis
1.	Uruchomienie środowiska testowego.
2.	Wczytanie niezbędnych danych do wykonania testu.
3.	Przygotowanie metody odpowiedzialnej za przeprowadzenie testu.
4.	Odczytanie aktualnego czasu systemowego i ustawienie go jako punkt start.
5.	Wywołanie przygotowanej metody.
6.	Odczytanie aktualnego czasu systemowego i ustawienie go jako punkt stop.
7.	Zapisanie wyniku testu.

Druga procedura testowa została przygotowana dla pozostałych badań. Opis tego scenariusza został przedstawiony w tabeli 3.

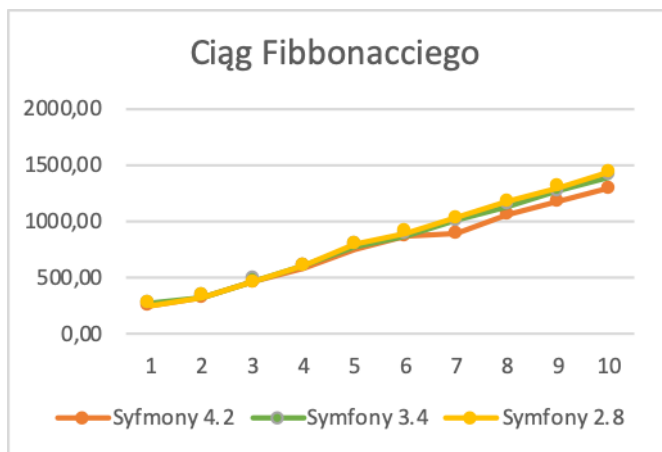
Tabela 3. Opis drugiej procedury testowej

Lp	Opis
1.	Uruchomienie środowiska testowego.
2.	Wczytanie niezbędnych danych do wykonania testu.
3.	Konfiguracja programu testowego JMeter – w zależności od przeprowadzonego testu.
4.	Wykonanie testu.
5.	Zapisanie danych.

5. Rezultaty badań

5.1. Scenariusz 1

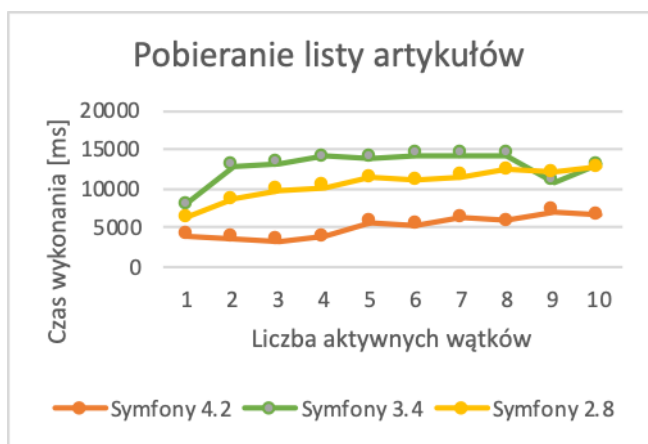
Pierwszy scenariusz polegał na obciążeniu aplikacji wyznaczającej ciąg Fibonacciego, przy $n=1000$, gdzie n oznacza wygenerowane liczby ciągu.



Rys. 3. Wykres czasu wykonania aplikacji obciążonej ciągiem Fibbonacciego

5.2. Scenariusz 2

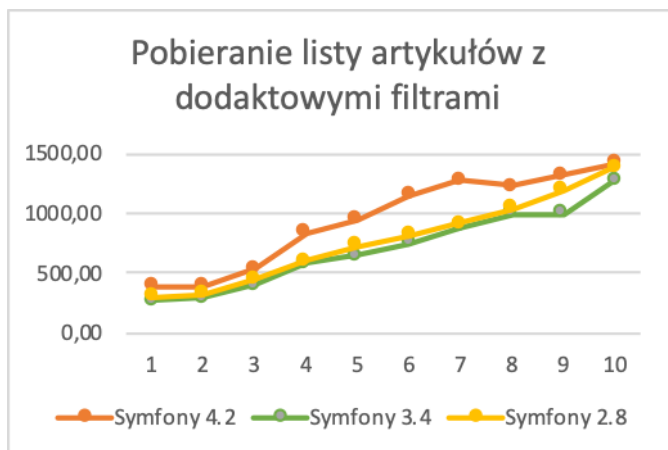
Drugi scenariusz polegał na pobraniu 1000 pierwszych aktywnych artykułów z bazy danych MySQL.



Rys. 4. Wykres wykonania zapytania dla poszczególnych wersji frameworka wyświetlania listy artykułów

5.3. Scenariusz 3

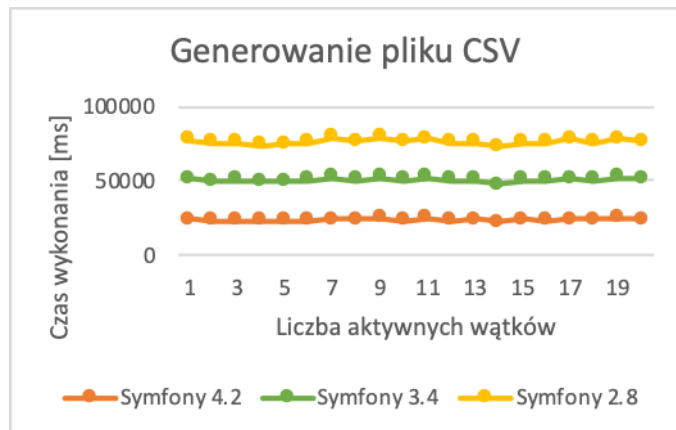
Trzeci scenariusz polegał na pobraniu listy artykułów o ograniczeniu wyników do 50 dla każdego testu, zastosowano dodatkowe filtry kategorii oraz autorów.



Rys. 5. Wykres przedstawiający pobieranie listy artykułów z dodatkowymi filtrami dla poszczególnych wersji frameworka

5.4. Scenariusz 4

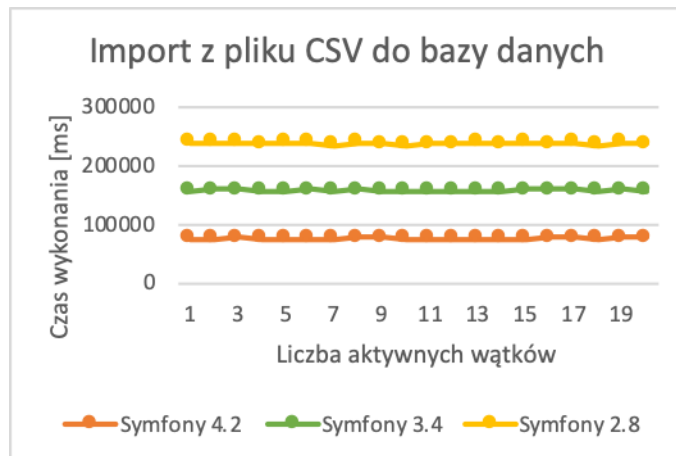
Czwarty scenariusz polegał na generowaniu pliku csv z danych pobranych z przygotowanej wcześniej bazy danych.



Rys. 6. Wykres czasów importu danych z bazy danych do pliku csv

5.5. Scenariusz 5

Piąty scenariusz polegał na odczytaniu danych z pliku csv i zapisaniu ich do bazy danych.



Rys. 7. Wykres przedstawiający czas importu danych z pliku csv do bazy danych

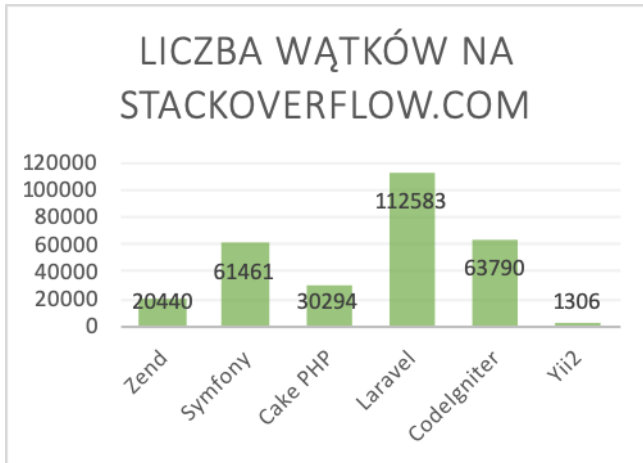
5.6. Dostępność dokumentacji

Każda wersja Symfony 2.8, 3.4 oraz 4.2 posiadają obszerne dokumentacje dostępne w języku angielskim [9]. Twórcy zadbałi o to, aby w zrozumiały sposób przedstawić podstawy korzystania z nich (np. klarownie opisano wymagane narzędzia, poszczególne kroki tworzenia, instalacje oraz uruchomienia aplikacji). Poszczególne narzędzia posiadają proste, lecz dokładne materiały szkoleniowe w formie wideo. Dzięki którym można zaznajomić się z podstawowymi zasadami oraz możliwościami narzędzi. Jednakże twórcy Symfony, dodatkowo nawiązali współpracę z firmą zewnętrzną do udostępniania materiałów [10].

5.7. Wsparcie społeczności

Największą społeczność można znaleźć na platformie StackOverflow [11]. 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]. Na Rys. 8 został przedstawiony liczba poszczególnych wątków dla wybranych technologii dotyczących najpopularniejszych frameworków PHP [11].



Rys. 8. Wykres liczby wątków na stackoverflow.com

6. Wnioski

Cel pracy, jakim było zaimplementowanie trzech jednakowych aplikacji w trzech różnych wersjach Symfony oraz przeprowadzenie wyżej opisanych testów został osiągnięty. Do badania zostały wyselekcjonowane 3 różne wersje tj. 2.8, 3.4 oraz 4.2. w każdej z tych wersji zostały napisane takie same aplikacje. Aby uwiarygodnić wyniki, wszystkie aplikacje były uruchamiane na wyizolowanym środowisku Docker.

Z otrzymanych wyników można jednoznacznie stwierdzić, że framework Symfony 4.2 wykonuje operację na dużej ilości danych znacznie szybciej od pozostałych. Jednakże w przypadku testów na próbie mniejszej ilości danych wyniki są do siebie zbliżone.

Testy zapisu i odczytu danych z i do plików csv dały jednoznaczne wyniki – Symfony 4.2 potrafi wykonać te operację trzy razy szybciej niż jego poprzednicy.

Z testów wynika, że wersją Symfony 4.2 ma znaczną przewagę przy bardziej wymagających działaniach, takich jak zwracanie dużej ilości danych z API. Natomiast, przy mniej wymagających operacjach, wszystkie badane wersje mają bardzo zbliżone wyniki.

Literatura

- [1] Natalya Prokofyeva, V. B. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Ryga.
- [2] Doglio, F. (2015). Rest 101. In: Pro REST API Development with Node.js. . Apress, Berkeley, CA .
- [3] Pitt, C. (2012). Introduction to MVC. In: Pro PHP MVC. Apress, Berkeley, CA.
- [4] Introduction to framework Symfony https://symfony.com/doc/current/create_framework/introduction.html [01.06.2019]
- [5] Introducing PHP. In: Beginning PHP and MySQL. (2010). Apress.
- [6] Beginning PHP an MySQL From Novice to Professional <https://www.apress.com/gp/book/9781430231141> [02.06.2019]
- [7] Russell, C. (2016). Frameworks. In: PHP Development Tool Essentials. Apress, Berkeley, CA.
- [8] Symfony Documentation Getting Started with frameowrk <https://symfony.com/doc/current/index.html#gsc.tab=0>
- [9] Symfony <https://symfony.com/doc/> [02.06.2019]
- [10] Symfony Cats Turtorial <https://symfonycasts.com> [31.05.2019]
- [11] Stack Overflow <https://stackoverflow.com> [01.06.2019]
- [12] Phpbenchmarks, Steevan Barboyon <https://www.phpbenchmarks.com/en> [06.06.2019]