

ZASTOSOWANIE FRAMEWORKA LARAVEL DO TWORZENIA PROGRAMÓW MODULARNYCH

Paweł Flis¹, Artur Hermanowicz¹

¹Uniwersytet Radomski im. K. Pułaskiego, Wydział Transportu, Elektrotechniki i Informatyki, Malczewskiego 29, 26-600 Radom, Polska

Streszczenie – W artykule przedstawiono zastosowanie frameworka Laravel do tworzenia programów modularnych. Laravel, będący frameworkiem opartym na języku PHP i architekturze Model-View-Controller (MVC), oferuje różnorodne narzędzia i mechanizmy wspierające modularność aplikacji. Omówiono kluczowe komponenty Laravel, takie jak Service Providers, Packages, Artisan Console, Routing & Middleware, Eloquent ORM & Migrations, Events & Listeners oraz IoC Container, które przyczyniają się do efektywnego zarządzania zależnościami oraz skalowalności i elastyczności aplikacji..

Słowa kluczowe – Laravel, modularność, framework, PHP, MVC, programowanie

WSTĘP

Tworzenie oprogramowania modularnego to jedno z największych wyzwań w dzisiejszej inżynierii oprogramowania. Modularność, definiowana jako podział aplikacji na mniejsze, niezależne moduły, jest kluczowa dla utrzymania wysokiej jakości kodu, jego skalowalności i łatwości w zarządzaniu. Niestety, osiągnięcie prawdziwej modularności często wiąże się z szeregiem problemów, takich jak złożoność integracji poszczególnych modułów, zapewnienie spójności danych czy zarządzanie zależnościami [1,2]. Na rynku istnieje kilka rozwiązań umożliwiających tworzenie programów modularnych, którymi to są frameworki takie jak Laravel, Zend czy też Symfony.

I. MODULARNOŚĆ

Modularność to podejście do projektowania aplikacji, w którym całość systemu składa się z niezależnych modułów, z których każdy odpowiada za określoną funkcjonalność. Moduły te mogą być rozwijane, testowane i wdrażane niezależnie od siebie. Kluczowe cechy programów modularnych obejmują:

- Niezależność komponentów

Każdy moduł w systemie działa jako oddzielna jednostka, która może być rozwijana i testowana samodzielnie. Dzięki temu możliwe jest aktualizowanie i modyfikowanie jednego modułu bez wpływu na resztę systemu.

- Wyraźnie zdefiniowane interfejsy

Interfejsy między modułami są jasno określone, co pozwala na łatwe wymienianie jednego modułu na inny, o ile spełnia on ten sam interfejs. To ułatwia integrację i zmniejsza ryzyko błędów.

- Enkapsulacja

Każdy moduł ukrywa swoje wewnętrzne szczegóły implementacyjne i udostępnia tylko niezbędne interfejsy

do komunikacji z innymi modułami. To podejście zapobiega przypadkowemu naruszeniu danych wewnętrznych modułu przez inne części systemu.

- Możliwość ponownego użycia komponentów

Moduły mogą być ponownie używane w różnych częściach systemu lub nawet w innych projektach. To zwiększa efektywność programowania, ponieważ raz napisany i przetestowany moduł może być używany wielokrotnie bez konieczności ponownego tworzenia kodu.

II. DOSTĘPNE FRAMEWORKI WPIERAJĄCE MODULARNOŚĆ

- Laravel

Laravel to framework, który zdobył ogromną popularność dzięki swojej prostocie i elegancji. Oferuje intuicyjne API (ang. application programming interface), jest to metoda na przesłanie danych pomiędzy aplikacjami[4], i wiele wbudowanych narzędzi, które przyspieszają rozwój aplikacji. Laravel wspiera modularność poprzez wykorzystanie paczek (packages) i systemu usług (service container). Jego dokumentacja jest bogata i łatwo dostępna, co czyni go przyjaznym dla programistów na każdym poziomie zaawansowania. Laravel jest również znany z doskonałej społeczności, która aktywnie rozwija nowe pakiety i rozszerzenia [5,6,7].

- Symfony

Symfony jest potężnym frameworkiem PHP, znanym ze swojej elastyczności i szerokiego zestawu komponentów. Jego modularna architektura pozwala na łatwe tworzenie aplikacji o dowolnej skali. Symfony wspiera najlepsze praktyki programistyczne, takie jak Dependency Injection, co ułatwia zarządzanie zależnościami między modułami. Jednakże, jego skomplikowana konfiguracja i stroma krzywa uczenia mogą stanowić barierę dla początkujących programistów [8,9].

- Zend Framework

Zend Framework, obecnie znany jako Laminas, jest wszechstronnym i bardzo elastycznym frameworkiem PHP. Jego architektura pozwala na używanie go jako kompletny framework lub jako zestaw niezależnych komponentów. Zend jest często wybierany do dużych, korporacyjnych projektów ze względu na swoją stabilność i wsparcie długoterminowe. Niemniej jednak, jego skomplikowana konfiguracja i mniejsza społeczność w porównaniu do Laravel mogą stanowić wyzwanie [10,11].

III. PORÓWNIANIE NARZĘDZI DO PROGRAMOWANIA MODULARNEGO

Laravel wyróżnia się na tle konkurencji dzięki swojej prostocie, elegancji i wsparciu dla nowoczesnych praktyk programistycznych. Jego bogata dokumentacja i duża społeczność zapewniają wsparcie na każdym etapie rozwoju aplikacji. Laravel oferuje szeroką gamę wbudowanych funkcji, takich jak Eloquent ORM, Blade templating engine czy system kolejek, które przyspieszają tworzenie aplikacji. Ponadto, Laravel Forge i Laravel Vapor ułatwiają wdrażanie i skalowanie aplikacji w chmurze, co jest kluczowe w dzisiejszym środowisku IT.

Symfony natomiast oferuje wysoką wydajność i obszerną dokumentację, ale jego złożoność może stanowić wyzwanie dla początkujących programistów. Jego elastyczność i wsparcie dla zaawansowanych wzorców projektowych sprawiają, że jest idealnym wyborem dla dużych i złożonych projektów.

Zend Framework jest elastyczny i dobrze zaprojektowany obiektowo, co czyni go odpowiednim dla dużych, korporacyjnych projektów. Jednakże, jego skomplikowana konfiguracja i mniejsza społeczność w porównaniu do Laravel mogą stanowić wyzwanie dla programistów szukających szybkiego wdrożenia.

IV. WNIOSKI

Laravel jest idealny dla deweloperów szukających równowagi między prostotą, a potężnymi funkcjami. Jego intuicyjność i wsparcie społeczności czynią go preferowanym wyborem. Symfony oferuje wysoką wydajność i obszerną dokumentację, ale jego złożoność może stanowić wyzwanie. Zend Framework (Laminas) jest elastyczny i dobrze zaprojektowany obiektowo, ale mniej odpowiedni do szybkiego rozwoju ze względu na skomplikowaną konfigurację i znacznie większe zapotrzebowanie na zasoby systemu. Laravel, dzięki swojej prostocie i wydajności, stanowi najlepszy wybór dla programistów dążących do tworzenia skalowalnych i elastycznych systemów.

APPLICATION OF THE LARAVEL FRAMEWORK FOR BUILDING MODULAR PROGRAMS

Creating modular software is one of the biggest challenges in today's software engineering. Modularity, defined as the division of an application into smaller, independent modules, is crucial for maintaining high code quality, scalability, and ease of management. Unfortunately, achieving true modularity often involves several issues, such as the complexity of integrating individual modules, ensuring data consistency, or managing dependencies. There are several solutions available in the market that help in creating modular software, such as frameworks like Laravel, Zend, and Symfony.

Key words: Laravel, modularity, framework, PHP, MVC, programming

BIBLIOGRAFIA

- [1] https://www.cri.agh.edu.pl/uczelnia/tad/PSI_STRONA/6_1.html (data dostępu 20.05.2024)
- [2] <https://techblog.ing.pl/blog/modularyzacja-kolejny-modny-trend-it-czy-jedyna-sluszna-droga> (data dostępu 20.05.2024)
- [3] Flis P., *Projekt i implementacja systemu wspomagającego tworzenie programów studiów*, praca inż., 2024 Uniwersytet Radomski
- [4] Gough J., Bryant D., Auburn M., *Architektura API. Projektowanie, używanie i rozwijanie systemów opartych na API* 2024 Helion
- [5] <https://laravel.com/docs/11.x> (data dostępu 20.05.2024)
- [6] <https://laravel.com/> (data dostępu 20.05.2024)
- [7] Stauffer M., *Laravel w działaniu. Budowa nowoczesnych aplikacji w PHP*, 2021 Helion
- [8] Gajda W., *Symfony 2 od podstaw*, 2012 Helion
- [9] <https://symfony.com/doc/current/index.html> (data dostępu 20.05.2024)
- [10] Gajda W., *Zend Framework od podstaw. Wykorzystaj gotowe rozwiązania PHP do tworzenia zaawansowanych aplikacji internetowych*, 2011 Helion
- [11] <https://framework.zend.com/> (data dostępu 20.05.2024)