

Praca Dyplomowa Inżynierska

Bartosz Osiński
196097

Analiza działania oraz odtworzenie narzędzi front-endowych

Analysis and reconstruction of front-end tools

Praca dyplomowa na kierunku:
Informatyka

Praca wykonana pod kierunkiem
dr. inż. Waldemara Karwowskiego, prof. SGGW
Instytut Informatyki Technicznej
Katedra Sztucznej Inteligencji

Warszawa, rok 2022



SZKOŁA GŁÓWNA
GOSPODARSTWA
WIEJSKIEGO

Wydział Zastosowań
Informatyki
i Matematyki

Oświadczenie Promotora pracy

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis promotora

Oświadczenie autora pracy

Świadom/a odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2019 poz. 1231 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiato-
wej.

Data

Podpis autora pracy

Streszczenie

Stworzenie klasy \LaTeX -owej do użytku przy pisaniu pracy dyplomowej w SGGW

Tematem niniejszej pracy było zaimplementowanie klasy \LaTeX -owej pozwalającej na formatowanie tekstu zgodnie z wytycznymi nałożonymi przez uczelnię. Praca zawiera dwie główne części. Pierwsza z nich zawiera opis najważniejszych aspektów implementacji klasy. Natomiast druga część skupia się na sposobie użycia klasy przez osoby piszące prace dyplomowe.

Słowa kluczowe – LaTeX, klasa, praca dyplomowa, implementacja, SGGW, Szkoła Główna Gospodarstwa Wiejskiego

Summary

Creation of the \LaTeX Class to be Used When Writing a Thesis at the Warsaw University of Life Sciences – SGGW

The subject of this study was to implement a \LaTeX class that allows for text formatting according to the guidelines imposed by the University. The work consists of two main parts. The first one describes the most important aspects of the implementation. The second part focuses on how to use the class by people writing the theses.

Keywords – LaTeX, class, thesis, implementation, SGGW, Warsaw University of Life Sciences

Spis treści

1	Wstęp	9
1.1	Charakterystyka narzędzi front-endowych	9
1.2	Wybór typów narzędzi	10
2	Module bundler	11
2.1	Sposoby bundlowania	11
2.2	Transformacja kodu javascript przy użyciu parsera	11
2.3	Zależności cykliczne	11
2.4	Testowanie module bundlera	11
3	Generator CSSowych klas atomicznych	12
3.1	Czym są klasy atomiczne?	12
3.2	Powody używania generatora klas atomicznych	12
3.3	Sposób działania	12
4	Przykładowa aplikacja stworzona przy użyciu zaprezentowanych narzędzi	13
5	Podsumowanie i wnioski	14
6	Bibliografia	15

1 Wstęp

Współczesne aplikacje internetowe najczęściej składają się z dwóch części: front-endowej oraz back-endowej. Rozróżnienie między nimi polega na tym, gdzie kod jest wykonywany – jeśli jest on wykonywany po stronie serwera, mówimy o back-endzie, natomiast jeśli obliczenia wykonywane są na komputerze użytkownika, jest to front-end.

Programowanie front-endowe jest jedną z najszybciej rozwijających się dziedzin programowania. Technologie, które pięć lat temu były zdecydowanymi liderami na rynku dziś są uważane za przestarzałe. Tak szybka ewolucja technik developerskich połączona z potrzebą kompatybilności wstecznej aplikacji internetowych (Właściciele aplikacji chcą, aby działały one dla jak największej grupy klientów, a nie tylko dla tych, którzy posiadają najnowsze wersje przeglądarek) generuje problemy, które deweloperzy rozwiązują używając między innymi szerokiego zakresu narzędzi front-endowych.

1.1 Charakterystyka narzędzi front-endowych

Podstawową cechą narzędzi front-endowych – być może jedyną cechą wspólną dla większości narzędzi – jest sposób ich konfiguracji. Większość narzędzi jest dostosowana do uruchomienia na dwa sposoby:

1. Poprzez *Command Line Interface* – CLI – wykonując krótki skrypt w wierszu poleceń takim jak np. Bash
2. Jako kod, najczęściej w języku Javascript – narzędzie dostarczane jest developerowi jako biblioteka programistyczna. Programista tworzy plik, który wywołuje funkcje z dostarczonej biblioteki z odpowiednimi argumentami. Następnie za każdym razem, kiedy chce uruchomić narzędzie musi uruchomić stworzony przez siebie plik, np. przy użyciu środowiska Node.js, jeśli kod jest napisany w języku javascript.

Wiele narzędzi pozwala na uruchomienie ich przy użyciu obu tych metod: poprzez CLI w przypadkach, które nie wymagają skomplikowanej konfiguracji ani niestandardowych zachowań, oraz poprzez kod w bardziej skomplikowanych sytuacjach.

Wiele narzędzi łączy obie te metody w taki sposób, że programista tworzy plik, np. javascript, który eksportuje obiekt definiujący konfigurację, a następnie w wierszu poleceń uruchamia narzędzie przekazując mu tylko jeden argument – ścieżkę do wspomnianego wcze-

śniej pliku.

1.2 Wybór typów narzędzi

Praca ta ma na celu przeanalizowanie działania dwóch typów narzędzi frontendowych:

- Module bundler – Narzędzie, które łączy kod aplikacji rozdzielony na wiele plików w jeden plik, w ten sposób zamieniając kod przyjazny deweloperowi na wydajny kod wspierający starsze przeglądarki, które nie pozwalały na nowoczesne metody importowania i eksportowania symboli w plikach javascript. Przykładami istniejących narzędzi tego typu są *Webpack* oraz *Parcel*
- Generator CSS-owych klas atomicznych – Narzędzie, które przeszukuje kod aplikacji, szukając w niej nazw klas, dla których mogą być zdefiniowane jakieś style CSS, a następnie generuje plik CSS z odpowiednimi stylami. Przykładem może być *Tailwind*[1] działający w trybie *Just in Time* oraz *UnoCSS*[2]

Oba te typy narzędzi posiadają swoich przedstawicieli, na których będzie można się skupić podczas analizowania sposobu ich działania.

2 Module bundler

Module budler to – jak wspomniano wyżej – narzędzie służące do "sklejenia" kodu znajdującego się w wielu plikach w jeden duży plik. Służy to kilku celom:

- Zmniejszenie ilości zapytań do serwera – Duże, komercyjne aplikacje nierzadko składają się z setek lub tysięcy plików. Wdrożenie aplikacji rozdzielonej w taki sposób wiązało by się z wysyłaniem setek lub tysięcy zapytań z przeglądarki na serwer podczas każdorazowego odwiedzenia strony. Byłoby to duże obciążenie dla przeglądarki oraz ogromne obciążenie dla serwera.
- Utrzymanie kompatybilności wstecznej –

2.1 Sposoby bundlowania

2.2 Transformacja kodu javascript przy użyciu parsera

2.3 Zależności cykliczne

2.4 Testowanie module bundlera

3 Generator CSSowych klas atomicznych

3.1 Czym są klasy atomiczne?

3.2 Powody używania generatora klas atomicznych

3.3 Sposób działania

4 Przykładowa aplikacja stworzona przy użyciu zaprezentowanych narzędzi

5 Podsumowanie i wnioski

6 Bibliografia

- [1] *Just-in-Time Mode – Tailwind CSS* <https://v2.tailwindcss.com/docs/just-in-time-mode> (dostęp 28.02.2022)
- [2] *Repozytorium projektu UnoCSS w serwisie Github* <https://github.com/unocss/unocss> (dostęp 28.02.2022)
- [3] Axel Rauschmayer, *Exploring JS*

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym
w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)

