



Przemysław Kasprzyk

1 EF-DU-A

132758

Email: p.kasprzyk2008@gmail.com

Przedmiot: Aplikacje internetowe

Wersja: 3

Data: 10.01.2017

Rodzaj zajęć: projekt

Prowadzący: dr inż. Tomasz Rak

**System Zarządzania Dokumentami**

1. Cel projektu

Celem projektu jest utworzenie aplikacji internetowej służącej do zarządzania dokumentami. Aplikacja przeznaczona jest do użytku z poziomu przeglądarki internetowej.

2. Wymagania Funkcjonalne

System obsługiwany jest przez 3 typy użytkowników - administratora, moderatora, zalogowanego i niezalogowanego.

* System powinien obsługiwać następujące funkcjonalności
* Logowanie/rejestrację
* Dodawanie i wersjonowanie dokumentów
* Pobieranie dokumentów
* Proste szukanie dokumentów oraz przeszukiwanie sprecyzowane
* Wprowadzenie daty od kiedy dokument ba być opublikowany
* System ocen
* System komentarzy
* Możliwość dodawania słów kluczowych i opisów do dokumentów przez ich autora
* Możliwość ustawienia moderatorów sprawdzających treści dokumentu przed publikacją
* Możliwość ustawienia użytkownika zaufanego, nie potrzebującego moderacji
* Dodawanie zdjęć okładek jako plik lub za pomocą hiperłącza do pliku w sieci
* Możliwość otrzymywania powiadomień mailowych.
* Możliwość udostępniania dokumentu przez email z hiperłączem, email z załącznikiem, bądź wygenerowanie hiperłącza do pliku.
* Wstrzymanie dostępu użytkownikom naruszającym regulamin serwisu.
* Otrzymywanie powiadomień o wstrzymaniu dostępu, publikacji dokumentu, lub akcjach

Podział funkcjonalności użytkowników

* Niezalogowany użytkownik - ma możliwość przeglądania i udostępniania hiperłączy jedynie do dokumentów widocznych publicznie, może się zarejestrować, lub zalogować.
* Zalogowany użytkownik - rozszerza możliwości niezalogowanego, ma dostęp do pełnej bazy dokumentów. Nie ma ograniczeń w funkcjach udostępniania. O ile jego dostęp nie jest wstrzymany, może dodawać, publikować, komentować i oceniać dokumenty. Może edytować swój profil.
* Moderator - rozszerza możliwości użytkownika niezalogowanego ma możliwość sprawdzenia treści dokumentów i zatwierdzania ich do publikacji, a także zawieszenia uprawnień użytkownika
* Administrator - ma uprawnienia moderatora, a ponad to ma możliwość zmiany uprawnień użytkowników.

1. Wymagania niefunkcjonalne

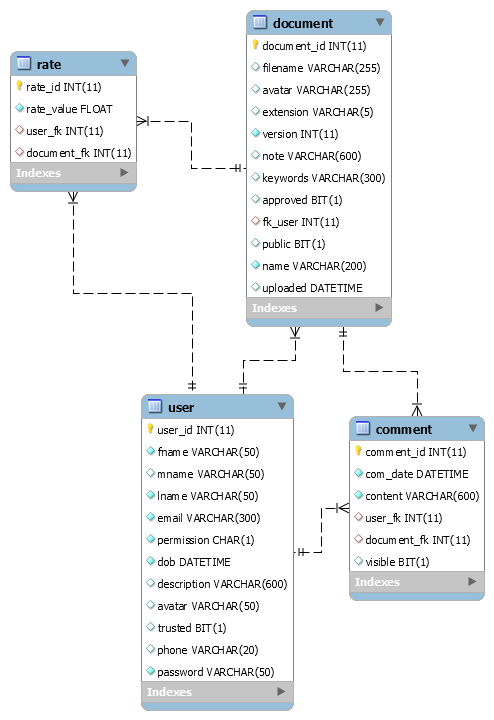
Warstwa serwerowa:

* Napisana w języku Java.
* Wykonana z użyciem frameworków Hibernate (wersja 4), Spring (wersja 4), Spring Data.
* Zastosowany system zarządzania zależnościami Apache Maven.
* Zastosowana implementacja serwera - Apache Tomcat/TomEE
* Relacyjna baza danych MySQL
* Pliki przechowywane bezpośrednio na serwerze (nie używana jest do przechowywania baza danych)
* System maszyny developerskiej - Windows 10

Warstwa prezentacji

* Napisana w języku TypeScript
* Wykonana z użyciem frameworka jQuery
* Bootstrap
* Zakładana kompatybilność z przeglądarkami Internet Explorer 11, Edge, Chrome 50, Firefox 49.

1. Diagram encji bazy danych



1. Struktura projektu

Projekt składa się z komponentu serwerowego i webowego. Komponent serwerowy składa się z 3 paczek - integration, domain i persistence. Dla paczek domain i persistence zastosowano domyślny modyfikator dostępu w miejscu gdzie nie ma zachodzić bezpośrednie połączenie pomiędzy warstwami aplikacji.

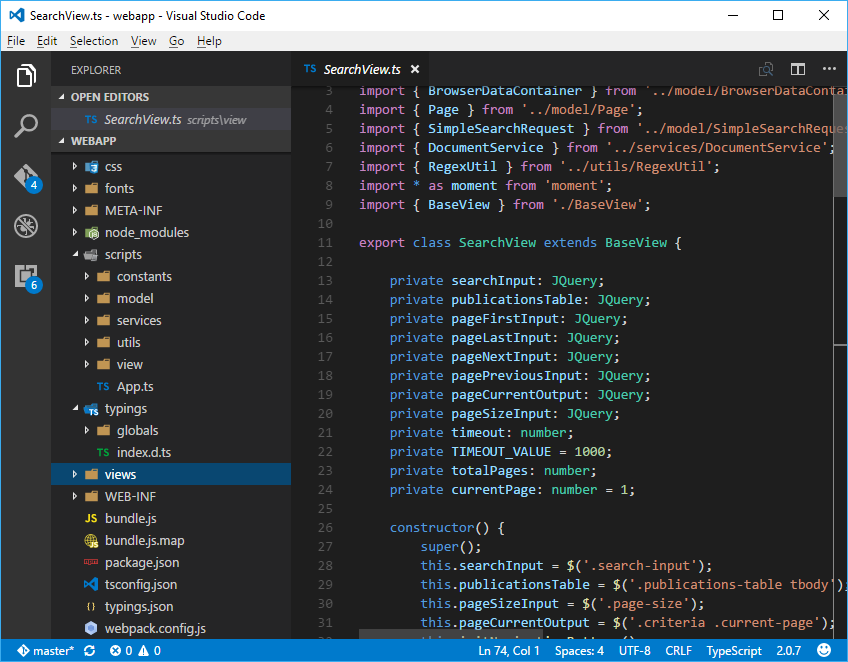
Integration - przechowuje elementy wspólne dla części serwerowej. Znajdują się tutaj elementy systemu zabezpieczeń aplikacji, konfiguracja projektu, a także interfejsy i klasy abstrakcyjne do wykorzystania w innych warstwach.

Persistence - zawiera modele JPA, repozytoria Spring Data JPA, a także serwisy których jedynym zadaniem jest zaciągnięcie danych.

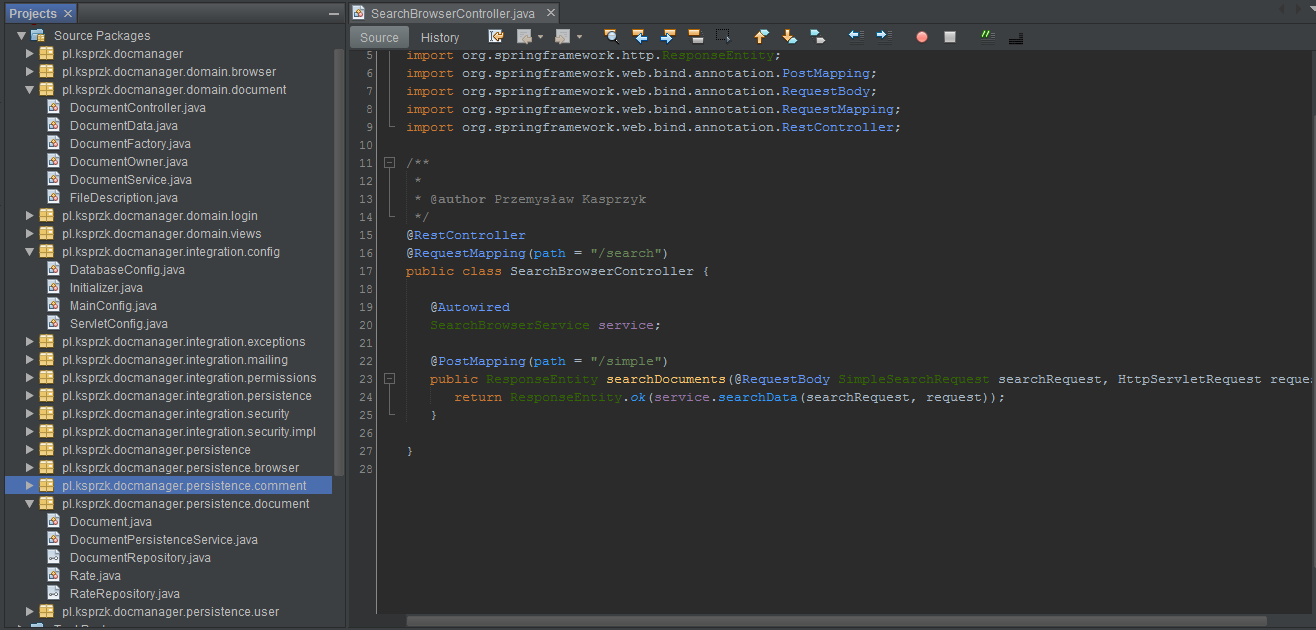
Domain - zawiera DTO, Kontrolery i mapping dla serwisów RESTowych, serwisy agregujące dane z serwisów paczki persistence, oraz fabryki przetwarzające JPA w DTO.

Komponent webowy został rozbity na klasy i interfejsy rozlokowane w katalogach model, service i view. Model zawiera interfejsy ułatwiające mapowanie DTO z serwera. Klasy z folderu service odpowiadają za wysyłanie i odbiór danych udostępnianych przez punkt dostępu serwera. Klasy z folderu view odpowiadają za zachowanie strony, oraz bindowanie danych pochodzących od użytkownika z serwisami.

Całość komponentu webowego uzupełniają widoki napisane w języku HTML, biblioteki JavaScript, zbiory styli CSS, i czcionek. Wszystkie utworzone w języku TypeScript skrypty są kompilowane a następnie składane do jednego dużego pliku bundle.js, celem ograniczenia ilości zapytań przeglądarki do serwera.



Rysunek 1 Widok struktury projekty od strony frontendu



Rysunek 2 Widok struktury projektu od strony backendu