Giełda wymian studenckich

Dokumentacja techniczna

Spis treści

1. Dokumentacja serwera

1.1. Wprowadzenie

Na solucję „gielda-studentow”, składają się 4 projekty:

1. „gielda-studentow” - będący zasadniczą częścią aplikacji, zawierającą implementację API oraz warstwę logiki aplikacji
2. „StudentExchangeDataAccess” - zawierający klasy reprezentujące encje w bazie danych, wykorzystywane do mapowania obiektowo – relacyjnego, a także odpowiedzialny za połączenie z bazą danych
3. „UnitTests” - zawierający testy jednostkowe
4. „SeleniumTests” - zawieracjący testy end to end

1.2. Warstwa połączenia z bazą danych

Do mapowania obiektowo – relacyjnego wykorzystany został Entity Framework, którego główną klasą jest StudentExchangeDataContext, w którym zadeklarowane są klasy przeznaczone do mapowania, a także elementy, które nie mogły zostać zdefiniowane poprzez wykorzystanie adnotacji (np. nazwy tabel w przypadku relacji wiele do wielu). W celu połączenia z bazą danych, w konstruktorze klasy StudentExchangeDataContext podaje się tzw. „connection string”, który zależny jest bazy danych oraz typu połączenia z nią. W folderze Initializer znajduje się klasa MyInitializer, której zadaniem jest wprowadzenie do bazy danych testowych oraz resetowanie bazy danych przed każdorazowym uruchomieniem aplikacji. Ostatnim elementem projektu jest folder ValidationAttributes, w którym znajdują się utworzone atrybuty służące do walidacji pól encji.

1.3. Warstwa logiki aplikacji

Za przetwarzanie danych i odpowiednie nimi manipulowanie odpowiadają klasy znajdujące się w folderze Service głównego projektu. Wykorzystują do tego klasę StudentExchangeDataContext. Dla każdej hierarchii klas modelu utworzona jest jedna klasa Service o odpowiedniej nazwie.

1.4. Komunikacja serwera z aplikacjami klienckimi

Komunikacja serwera z innymi aplikacjami (np. stworzonym przez nas klientem webowym z wykorzystaniem frameworka AngularJS) odbywa się poprzez protokół HTTP. Obsługa żądań tego protokołu zaimplementowana jest w poszczególnych kontrolerach, znajdujących się w folderze Controllers. Odpowiedzi serwera mają odpowiedni kod. Jeśli odpowiedź powinna zawierać dane, to znajdują się one w ciele odpowiedzi w formacie JSON.

1.5. Komunikacja między warstwami

Do komunikacji między poszczególnymi warstwami aplikacji wykorzystany został pakiet Unity, realizujący wzorzec projektowy wstrzykiwanie zależności. Kontrolery posiadają wstrzyknięte zależności od serwisów, natomiast serwisy zależne są od kontekstu. Konfiguracja kontenera i cykli życia poszczególnych jego komponentów znajduje się w klasie WebApiConfig (folder App_Start).

2. Dokumentacja interfejsu użytkownika

2.1. Wprowadzenie

Część front-endowa została wykonana jako Single Page Application w technologii AngularJS. Zdecydowaliśmy się na ten rodzaj technologi z racji jej prostoty, a przy tym dużych możliwości.
Single Page Application (SPA) to technologia, która pozwala wyświetlać poszczególne elementy strony, bez potrzeby ponownego załadowania całej strony. Jest ona oparta na wzorcu MVC (Model-View-Controller).

2.2. Elementy wykorzystanej technologii

AngularJS posiada kilka elementów, które pomagają w łatwy sposób zarządzać całym projektem:

- Moduły - w łatwy sposób pozwalają zarządzać kodem, co jest ważną kwestią w kontekście przyszłego rozwoju aplikacji. Przy obecnej wielkości naszej aplikacji wystarczył jeden główny moduł (myApp).
- Widoki - odpowiadają głównie za wygląd stron oraz ich budowę. Zawierają zarówno kod w HTML, CSS jak i dyrektywy Angulara.
- Kontrolery - zawierają logikę zarządzającą danymi wyświetlanymi na stronie. Ważnym elementem kontrolerów są zapytania AJAXowe, poprzez które następuje komunikacja z serwerem.
- Routing – odpowiada za przypisanie odpowiedniego widoku do wyszukiwanego adresu URL w obrębie strony oraz łączy widoki z kontrolerami. Dzięki niemu wskazujemy w widoku aplikacji która część ma być dynamicznie zmieniania podczas przechodzenia po stronach aplikacji.

Do wykonania naszej aplikacji użyliśmy 18 widoków oraz odpowiadającej im ilości kontrolerów.

2.3. Struktura projektu

```
gielda-studentow/
    SPA/
        Controllers/        - katalog z kontrolerami
        Views/              - katalog z widokami
        myApp.js            - główny moduł
```