Zadanie

Dostał\_ś zlecenie na stworzenie prostej aplikacji wspomagającej szkolną bibliotekę. Pracownik biblioteki stworzył już szablon aplikacji, tak aby nakreślić wszystkie potrzeby biblioteki. Biblioteka korzysta również z zewnętrznego API, z którym należy się zintegrować, aby uzyskać więcej informacji o książkach. Zmodyfikuj istniejący szablon, tak aby aplikacja była w pełni funkcjonalna.

W załączonych plikach znajduje się szablon aplikacji (folder LibraryAPI) oraz skrypt generujący bazę danych biblioteki. Szablon aplikacji składa się z:

* Folderu LibraryController wraz z plikiem LibraryController.cs w którym znajdują się wszystkie endpointy, które aplikacja powinna obsługiwać,
* Folderu Enums wraz z dwoma plikami BookGenres.cs oraz Statuses.cs informującymi o gatunkach książek dostępnych w bibliotece oraz statusach książek,
* Folderu Models wraz z plikami definiującymi klasy niezbędne do działania wszystkich kontrolerów,
* LibraryAPI.sln, Startup.cs, Program.cs oraz innych plików niezbędnych do uruchomienia aplikacji.

Skrypt generujący bazę danych biblioteki uruchomiony w aplikacji Microsoft SQL Server Management Studio po uruchomieniu stworzy bazę danych z tabelami:

* Books – znajdują się tutaj wszystkie książki wraz z informacjami o nich oraz ID ich obecnego statusu (z tabeli StatusHistory),
* Authors – znajdują się tutaj wszyscy autorzy książek dostępni w bibliotece,
* BookAuthor – jest to tabela łącząca rekordy z tabeli Books wraz z rekordami z tabeli Authors, jeśli dana książka ma nieznanego autora, rekordu z jej ID nie będzie w tej tabeli (klucze obce do tabel Books oraz Authors),
* StatusHistory – zawiera wszystkie statusy wszystkich książek, wraz z datą utworzenia takiego statusu i Id książki (klucz obcy do tabeli Books).

Wszystkie metody kontrolera powinny zostać zaimplementowane:

* GetBooks – Zwraca listę podstawowych informacji o książce (Id, tytuł oraz autor jeśli jest znany). Wynik tej metody powinien być możliwy do stronicowania (domyślna wielkość strony to 10 rekordów – np. dla wartości page = 2, limit = 20 metoda powinna zwrócić od 41 do 60 książki), a wyniki posortowane alfabetycznie po tytule.
* GetBookDetails – Zwraca szczegółowe informacje o wybranej książce (Id, tytuł, język, datę publikacji, gatunek, obecny status, autor jeśli jest znany, informację czy książka jest napisana w języku polskim oraz obecną cenę książki. Obecna cena książki jest możliwa do pobrania z zewnętrznego API za pomocą zapytania GET. Endpoint udostępniony dla Ciebie to: [http://60c35511917002001739e94a.mockapi.io/api/v1/Prices/{id}](http://60c35511917002001739e94a.mockapi.io/api/v1/Prices/%7bid%7d), gdzie za {id} powinieneś podstawić Id książki (np. [http://60c35511917002001739e94a.mockapi.io/api/v1/Prices/c1743884-4076-4a08-a4b1-00e5c30fd191](http://60c35511917002001739e94a.mockapi.io/api/v1/Prices/A65FAC80-7E3F-4ED7-BFC0-D30AA00A5521) - litery w guidzie powinny muszą być małe np. a zamiast A). Endpoint ten zwraca obiekt z Id oraz ceną książki. Nie każda książka posiada swoją cenę, a zewnętrzne API nie zawsze jest dostępne. Powinieneś obsłużyć takie sytuacje.
* GetBookStatuses – Zwraca listę wszystkich statusów wybranej książki posortowaną po dacie od najstarszego do najnowszego.
* InsertBook – Dodaje nową książkę do biblioteki oraz zwraca jej Id. Nowo dodana książka powinna posiadać status „InStock”, a data jego modyfikacji to data utworzenia książki. Jeśli przy dodawaniu książki zostanie podane również Id Autora, powiązanie takie powinno być zapamiętane w systemie, jednak Id autora nie jest wymagane przy dodawaniu nowej książki (brak tego autora oznacza, że jest on nieznany).
* ChangeBookStatus – zmienia status dla wybranej książki (wskazówka: w tabeli StatusHistory przy każdej zmianie statusu powinien pojawiać się nowy rekord, a CurrentStatusId w tabeli Documents powinien zostać zmodyfikowany).

Uwagi:

* Wykorzystaj Entity Framework oraz .NET Core 3.1,
* Nie modyfikuj definicji istniejących metod kontrolera,
* Nie modyfikuj istniejących klas w folderach Models oraz Enums (jednak w razie potrzeby możesz stworzyć nowe),
* Postaraj się aby Twoje rozwiązanie było otwarte na przyszłe modyfikacje, łatwo rozwijalne, jasne oraz czytelne,
* Nie bój się tworzyć nowych plików, folderów czy nawet projektów jeśli widzisz taką potrzebę 😊,
* Jeśli masz pomysł na usprawnienie lub chcesz się czymś pochwalić, wykorzystaj to,
* W razie jakichkolwiek problemów/niejasności nie bój się improwizować – w czasie rozmowy będziesz mógł uzasadnić swoje decyzje.