**Sprawozdanie Kostyra Mateusz**

Utworzenie projektu  
Obraz zawierający tekst, zrzut ekranu, monitor, czarny

Opis wygenerowany automatycznie

Dodanie klasy VideoCassete oraz jej uzupełnienie  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Dodanie klasy VideoCasseteApi oraz jej uzupełnienie  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Dodanie konstruktora parametrycznego do klasy VideoCassete  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Dodaj kilka filmów  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Utwórz metody, która będzie pobierała wszystkie elementy z listy  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Nad klasą dodaj adnotację requestMaping z parametrem, którym będzie adres za pomocą którego będzie można odpytać tworzone Api  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Do adnotacji metody getAll - GetMapping dopisz ścieżkę /all  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Zmodyfikuj wyrażenie aby użyć typu Optional, który zabezpiecza przed wyjątkiem związanym z wystąpieniem wartości typu null.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Uruchom aplikację Postmana i przetestuj jej działanie. Do pobierania elementów służy metoda GET  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Uruchom jeszcze raz

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przejdź do implementacji metody addVideo realizującej dodawanie elementów, przy użyciu metody http POST - użyj adnotacji PostMapping. Elementem dodawanym będzie obiekt klasy VideoCassettes. Będzie to metoda webowa więc do parametru przekazanego metodzie dodaj adnotację RequestBody, co oznacza,  
że ten obiekt zostanie przesłany do aplikacji w postaci serializowanej.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Następnie zaimplementuj metodę do modyfikacji obiektu - użyj adnotacji PutMapping. Metodzie podajemy id i na podstawie podanych wartości będzie ona nadpisywać dany element.  
Obraz zawierający tekst

Opis wygenerowany automatycznie  
  
  
Przy implementacji ostatniej metody, która będzie służyła do usuwania obiektu z listy – użyj adnotacji DeleteMapping.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Użyj POST do dodania elementu  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Pobierz wszystkie elementy z listy  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Przetestuj metodę, która służy do usuwania elementów  
Obraz zawierający tekst, zrzut ekranu, wewnątrz, monitor

Opis wygenerowany automatycznie  
  
  
Pobierz wszystkie elementy z listy aby sprawdzić czy faktycznie element o indeksie 1 został usunięty  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Po przetestowaniu API aplikacji przystąpimy do realizacji aplikacji zapisującej dane w bazie danych. W pliku application.properties ustaw właściwości dotyczące nawiązywanego połączenia z bazą danych.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

W momencie, gdy chcemy obiekty klasy przechowywać w bazie danych należy klasę zamienić na encję.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Utwórz repozytorium - interfejs VideoCassetteRepo  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Uporządkuj kod tworzonego projektu, aby miał strukturę warstwową, co ułatwi zadanie utrzymania aplikacji, jej rozwoju i poprawy błędów.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Tworzenie warstwy pośredniej miedzy warstwą dostępu do danych a api stanowi dobra praktykę - będzie nią klasa VideoCassetteManager.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Utwórz metody, które będą odpowiadały metodom znajdującym się w repozytorium i będą wywoływane w ramach implementacji metod klasy api aplikacji.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Druga metoda o nazwie findAll ma pobierać wszystkie elementy z bazy danych. Typ Iterable oznacza elementy, które da się iterować czyli jakiegoś typu kolekcje.  
Obraz zawierający tekst

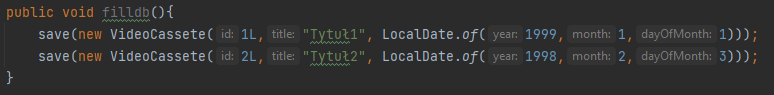
Opis wygenerowany automatycznie

Zaimplementuj metodę realizującą dodawanie elementu do bazy – repozytorium udostępnia metodę save. Metoda zwraca element, który został zapisany w bazie.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Zaimplementuj w serwisie metodę do usuwania elementów, która na podstawie dostarczonego id usunie dany element z bazy danych.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Skopiuj z konstruktora klasy api tworzenie dwóch obiektów i wklej w klasie menagera w ramach metody fillDB. Użyj metody save z klasy serwisu, aby dodać te dwa obiekty - filmy do bazy danych.  


Do metody dodaj specjalną adnotację eventListener z atrybutem AppliactionReadyEvent, która oznacza nasłuchiwanie zdarzenia uruchomienia aplikacji. Ciało metody fillDB oznaczone tą adnotacją, zostanie wykonane po uruchomieniu aplikacji, co spowoduje, że na jej starcie w bazie danych zostaną zapisane dwa filmy.  
Obraz zawierający tekst

Opis wygenerowany automatycznie  
  
  
W klasie api „zastąp” listę - bazą danych. Nie będziemy już potrzebowali listy, bo informacje o kasetach będą przechowywane w bazie danych - usuń listę i dotychczasowy konstruktor w którym dodajemy kasety do listy. W to miejsce utwórz deklarację obiektu klasy menagera, dodaj import, i w konstruktorze wstrzyknij instancję klasy menagera.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Metoda getAll powinna teraz korzystać z metody serwisu findAll i zwracać typ Iterable a nie listę.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Kolejna metoda getById nie będzie już potrzebowała korzystać   
z wyrażenia lambda – zostanie zastąpiona prostszym zapisem,   
w ramach którego wywołujemy metodę serwisu findAllById.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Aby zmodyfikować metodę dodającą obiekt/kasetę użyjemy metody   
z serwisu save, która zwraca dodany obiekt kasety wideo, wiec typ zwracany przez metodę add zamieniamy z boolean na VideoCassette.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Analogicznie modyfikujemy metodę update  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Przy modyfikacji ostatniej metody delete, korzystamy z metody z serwisu delete, która zwraca typ void.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Przejdź do bazy danych jest ona dostępna przez: /console.Obraz zawierający tekst

Opis wygenerowany automatycznie

Kliknij na nazwie bazy danych i naciśnij Run  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Po upewnieniu się, że w bazie danych są dwa elementy, wykonaj test metod w kliencie http – Postmanie. Najpierw pobierz wszystkie elementy.  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Następnie dodaj element  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Sprawdź czy element został dodany  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Sprawdź również z poziomu konsoli bazy danych czy element został dodany  
Obraz zawierający tekst, stół

Opis wygenerowany automatycznie

Przetestuj działanie metody PUT, która służy do modyfikacji pól danego elementu np. zmień tytuł którejś pozycji i datę  
Obraz zawierający tekst

Opis wygenerowany automatycznie  
  
Sprawdź wynik w Postmanie  
Obraz zawierający tekst

Opis wygenerowany automatycznie

W metodzie POST nie powinno się dodawać id bo on jest automatycznie generowany. Wykonaj zapytania  
Obraz zawierający tekst

Opis wygenerowany automatycznie  
  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Przetestuj działanie ostatniej metody delete – usuń np. 2 element z bazy  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający stół

Opis wygenerowany automatycznie