

Microsoft .NET: ASP.NET Web API, Entity Framework (Code First from Database), LINQ, REST

Do realizacji ćwiczeń potrzebne jest zintegrowane środowisko programistyczne Microsoft Visual Studio 2015.

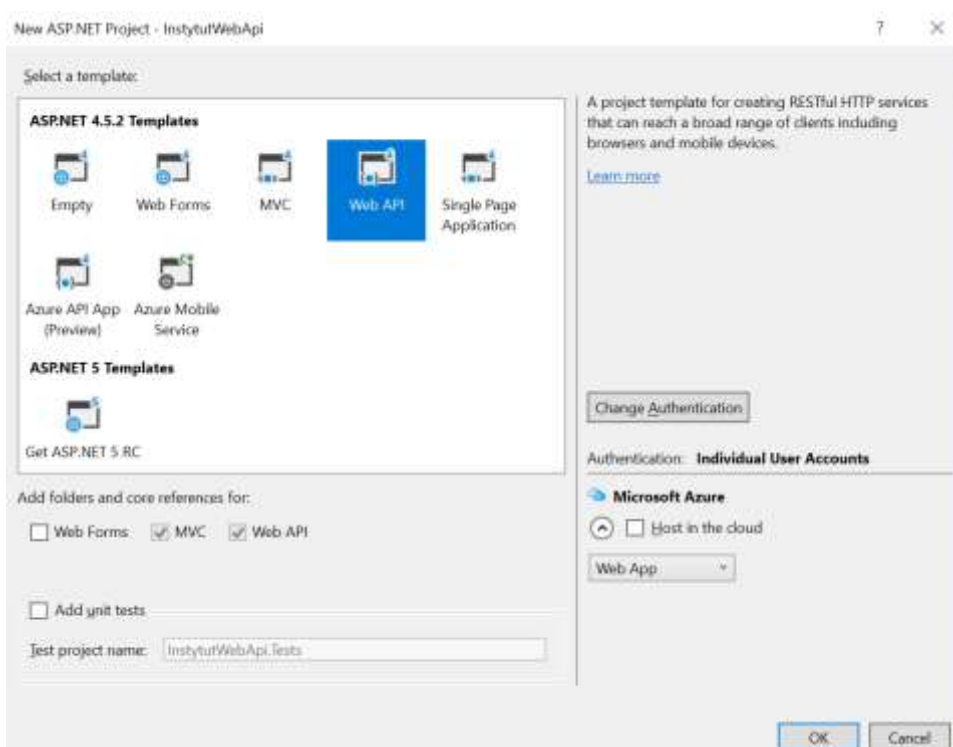
Celem ćwiczeń jest zapoznanie się z technologią ASP.NET Web API oraz składnią zapytań LINQ. Warstwa dostępu do danych będzie oparta na ADO.NET Entity Framework z wykorzystaniem strategii Code First from database.

1. Utworzenie nowego projektu (Project).

- a) Uruchom narzędzie Microsoft Visual Studio.
- b) Z menu głównego wybierz File→New→Project. Wybierz szablon ASP.NET Web Application z grupy szablonów dla języka Visual C#. Zaakceptuj zaproponowany katalog lub zmień go na inny gdy nie masz prawa zapisu w proponowanym katalogu. Kliknij przycisk **OK**.



- c) W kolejnym kroku kreatora projektu z listy szablonów ASP.NET wybierz Web API. Pozostałe ustawienia w tym kroku pozostaw domyślne. Kliknij przycisk **OK**.

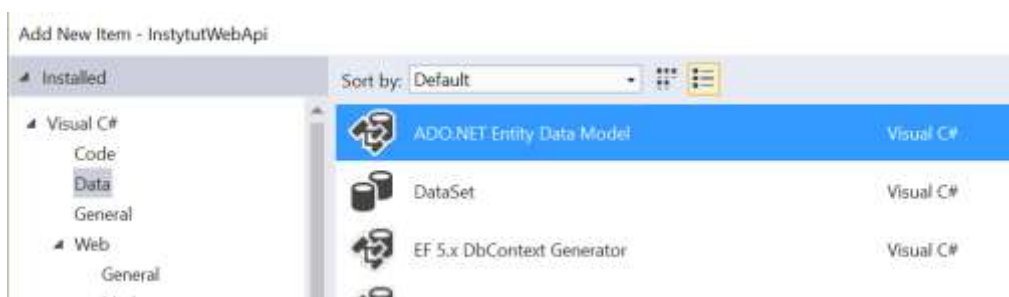


2. Dodanie pliku danych SQL Server Express do projektu

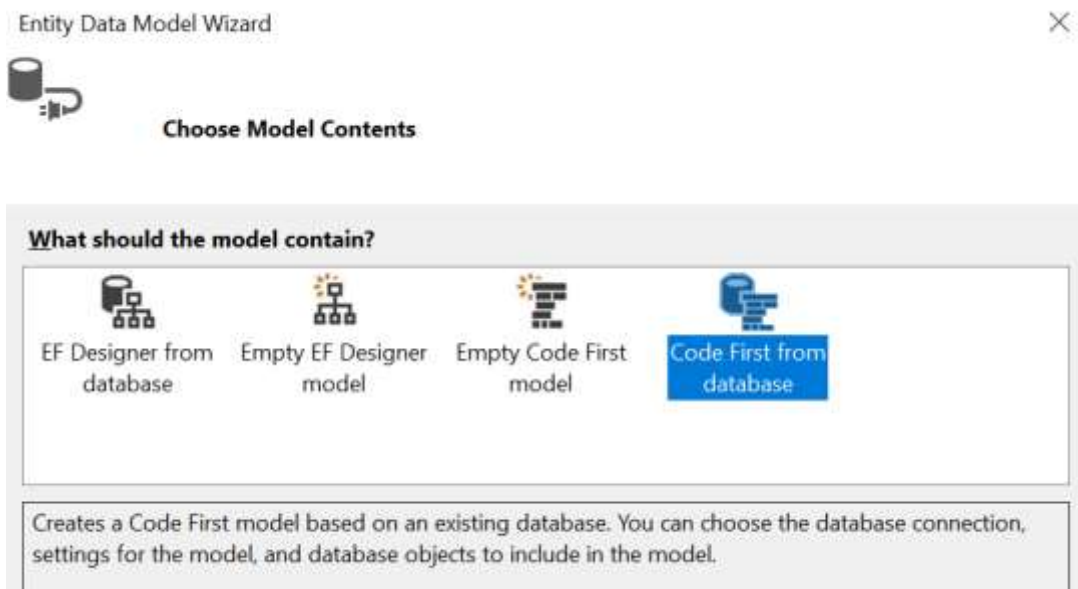
- Pobierz na dysk lokalny plik bazy danych `Institute.mdf`. W panelu **Solution Explorer** wywołaj prawym klawiszem myszy menu kontekstowe dla folderu `App_Data` i wybierz opcję **Add Existing Item**. Wskaż pobrany plik `Institute.mdf`.
- Po dodaniu pliku do projektu wywołaj dla niego w panelu **Solution Explorer** prawym klawiszem myszy menu kontekstowe i wybierz opcję **Open**. Rozwiń gałąź **Tables** i upewnij się, że plik bazy danych zawiera tabele `Employees`, `Departments` i `Jobs`.

3. Utworzenie modelu encji (Entity Data Model).

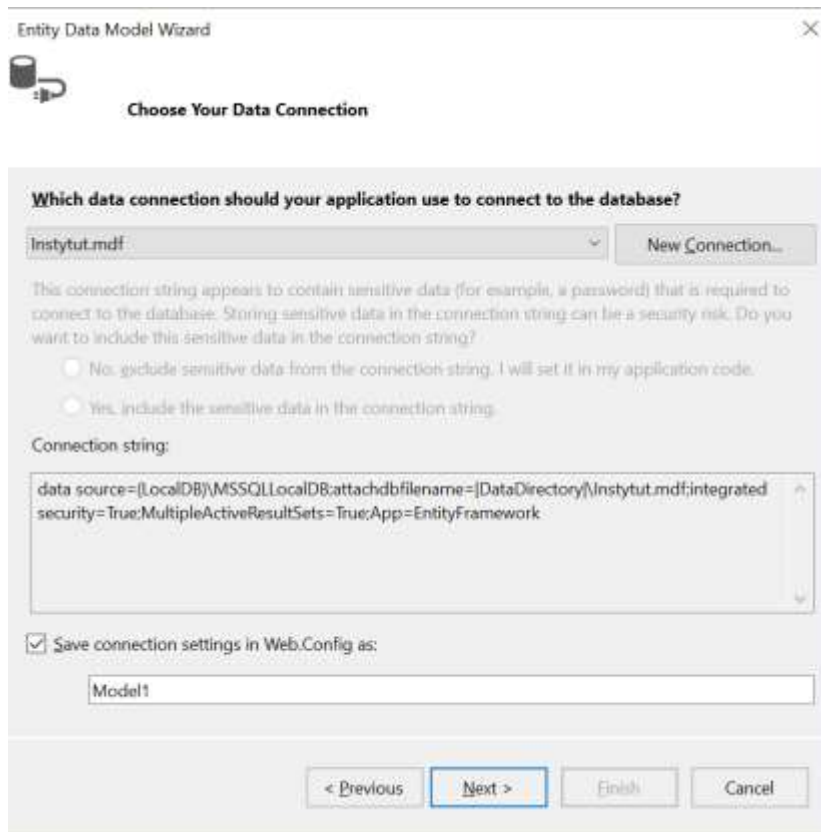
- Z menu kontekstowego dla węzła projektu w panelu **Solution Explorer** wybierz opcję **Add New Item**. Wybierz szablon **ADO.NET Entity Data Model** dla języka `Visual C#` i pozostaw zaproponowaną nazwę pliku. Kliknij przycisk **Add**.



- W oknie dialogowym kreatora modelu wybierz opcję generacji modelu **Code First** na podstawie bazy danych i przejdź do kolejnego kroku.

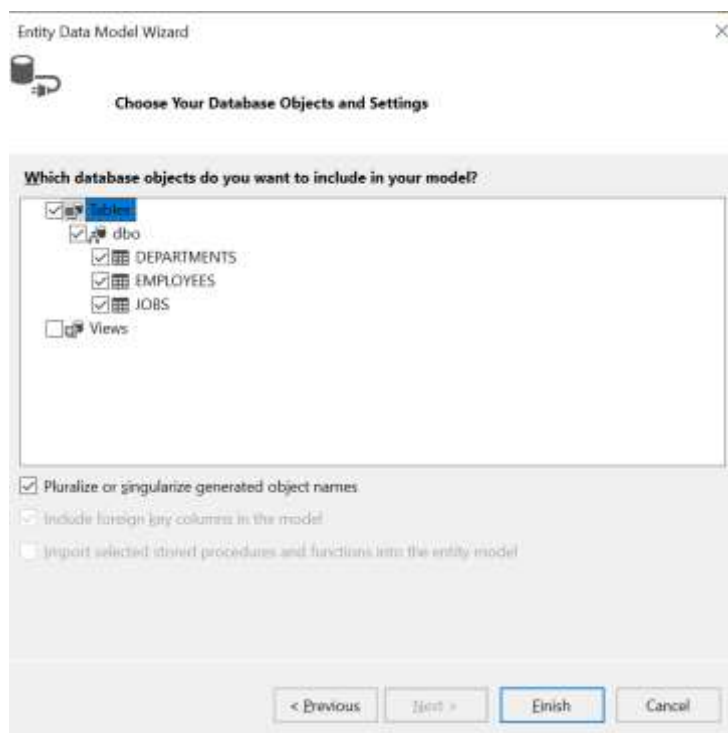


- c) W następnym kroku kreatora wybierz połączenie z dołączonym do projektu plikiem bazy danych. Pozostaw zaznaczone pole wyboru dodania parametrów połączenia w pliku konfiguracyjnym aplikacji. Przejdź do kolejnego kroku kreatora.



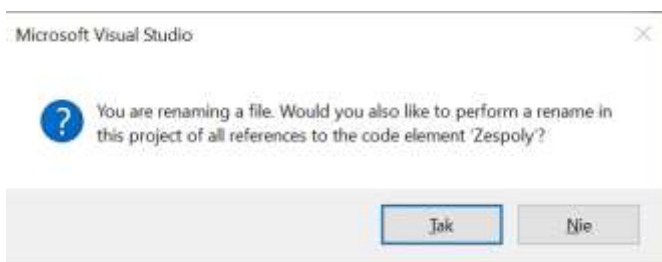
The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard'. Below the title bar is a sub-header 'Choose Your Data Connection'. The main question is 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'Instytut.mdf' and a 'New Connection...' button is to its right. Below this, a warning message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (selected) and 'Yes, include the sensitive data in the connection string.'. Below the radio buttons is a text box labeled 'Connection string:' containing the string: 'data source=(localdb)\MSSQLLocalDB;attachdbfilename=|DataDirectory|\Instytut.mdf;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework'. At the bottom, there is a checkbox 'Save connection settings in Web.Config as:' which is checked, followed by a text box containing 'Model1'. Navigation buttons at the bottom are '< Previous', 'Next >' (highlighted with a blue border), 'Finish', and 'Cancel'.

- d) W ostatnim kroku kreatora zaznacz wszystkie trzy tabele zawarte w bazie danych. Zaznacz pole wyboru Pluralize or singularize generated object names i kliknij Finish.



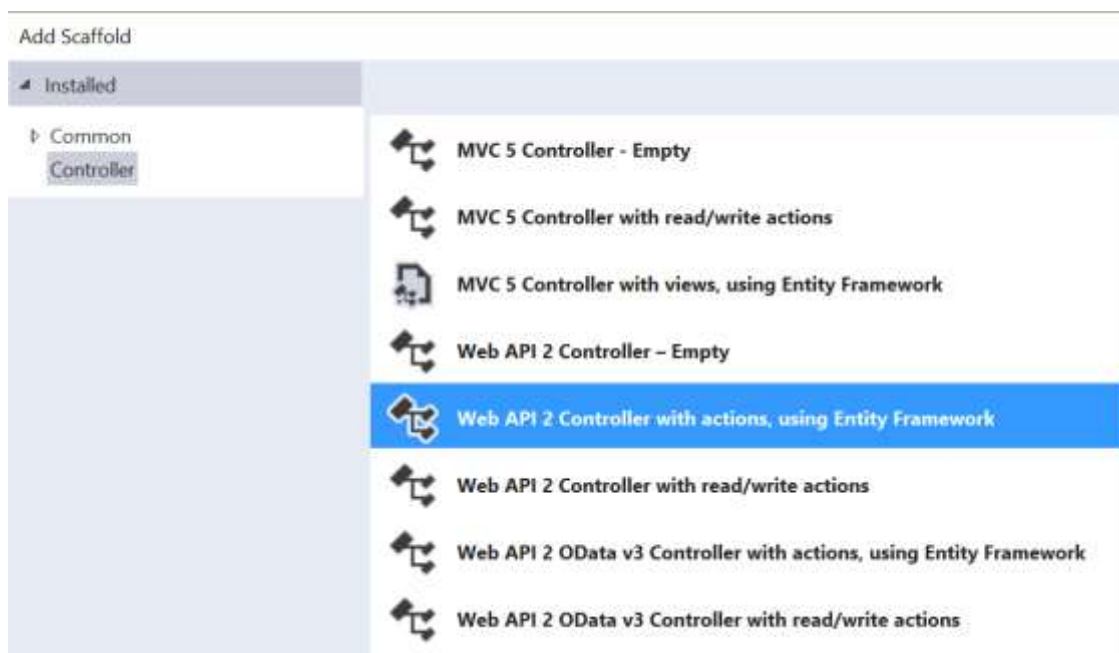
The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Database Objects and Settings' step. The title bar reads 'Entity Data Model Wizard'. Below the title bar is a sub-header 'Choose Your Database Objects and Settings'. The main question is 'Which database objects do you want to include in your model?'. A list of objects is shown with checkboxes: 'Tables' (checked), 'dbo' (checked), 'DEPARTMENTS' (checked), 'EMPLOYEES' (checked), 'JOBS' (checked), and 'Views' (unchecked). Below the list, there are three checkboxes: 'Pluralize or singularize generated object names' (checked), 'Include foreign key columns in the model' (unchecked), and 'Import selected stored procedures and functions into the entity model' (unchecked). Navigation buttons at the bottom are '< Previous', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

- e) Obejrzyj wygenerowaną klasę modelu i klasy encji. Ponieważ nazwy tabel w bazie danych były zapisane wielkimi literami, również nazwy encji są zapisane wielkimi literami. Zmień po kolei nazwy wszystkich trzech encji na “Employee”, “Department”, “Job” za pomocą opcji Rename dla pliku w panelu Solution Explorer. Gdy pojawi się okienko z pytaniem czy w związku ze zmianą nazwy pliku mają być dokonane również zmiany w kodzie, wyraż na to zgodę.



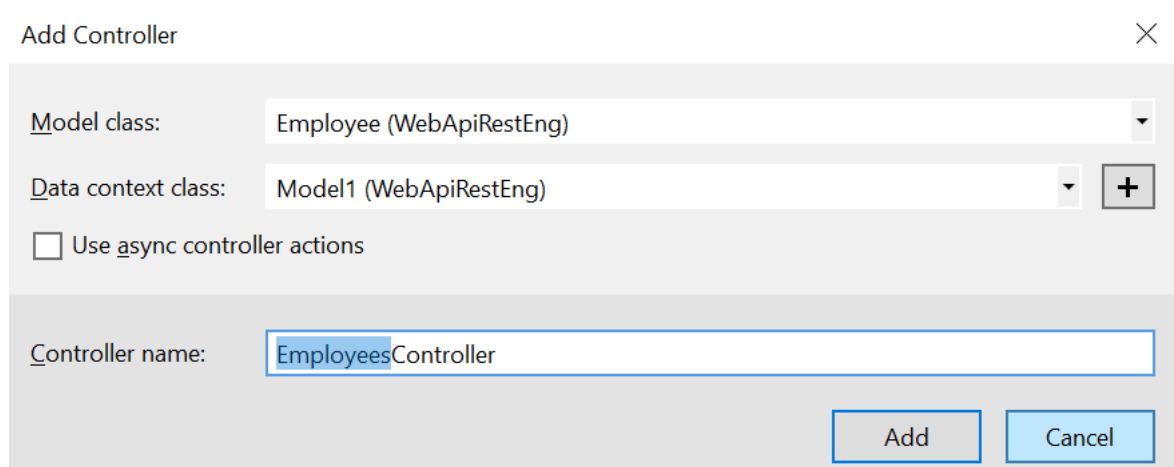
Upewnij się czy teraz wszystkie trzy klasy encji mają pożądane nazwy i czy odwołania do nich w klasie modelu są poprawne. Zapisz wszystkie pliki (Save All) i przebuduj aplikację (Build Solution).

4. Z poziomu węzła Controllers projektu wywołaj operację Add→Controller. Wybierz szablon Web API 2 Controller with actions, using Entity Framework i kliknij Add.



5. W kreatorze dodawania kontrolera:

- a) Jako klasę modelu wskaż klasę encji Employee
- b) Jako klasę kontekstu wybierz Model1
- c) Nie zaznaczaj pola wyboru zlecającego utworzenie asynchronicznych akcji kontrolera
- d) Zaakceptuj proponowaną nazwę kontrolera EmployeesController



Add Controller

Model class: Employee (WebApiRestEng)

Data context class: Model1 (WebApiRestEng) +

☐ Use async controller actions

Controller name: EmployeesController

Add Cancel

6. Analogicznie utwórz kontroler dla encji Department, ale tym razem zaznacz pole wyboru zlecające utworzenie asynchronicznych akcji kontrolera.

7. Przeanalizuj i porównaj kod poszczególnych metod obu kontrolerów, zwracając szczególną uwagę na konstrukcje związane z asynchronicznym przetwarzaniem akcji kontrolera dotyczącego zespołów.

8. Uruchom aplikację i z poziomu przeglądarki wykonaj poniższe zadania:

- a) Przejdź ze strony startowej do strony z dokumentacją udostępnionego API i ją przeanalizuj
- b) Pobierz informacje o wszystkich pracownikach wprowadzając w pasku adresu przeglądarki odpowiedni adres
- c) Zmodyfikuj adres tak aby pobrać tylko dane pracownika o identyfikatorze 140
- d) Pobierz informacje o wszystkich zespołach wprowadzając w pasku adresu przeglądarki odpowiedni adres
- e) Zmodyfikuj adres tak aby pobrać tylko dane zespołu o identyfikatorze 30
- f) Sprawdź co się stanie gdy podany będzie nieistniejący identyfikator zespołu lub pracownika.

9. Odszukaj w kodzie aplikacji miejsce, w którym zdefiniowane są adresy REST zasobów udostępnianych przez utworzone kontrolery Web API.

10. Odszukaj kod odpowiedzialny za zachowanie aplikacji przy odwołaniu do nieistniejącego zasobu.

11. Zmodyfikuj kod kontrolera obsługującego pracowników tak aby lista pracowników była zwracana w kolejności malejących płac podstawowych. (Dodaj odpowiednią konstrukcję LINQ).