# Комп'ютерний практикум 4

# Cтворення додатку MVC ASP.NET з використанням WebApi.

## Мета

Ознайомлення з можливостями WebApi практичне застосування їх в MVC ASP.NET.

## Зміст

1. Підключення вже існуючої бази даних за допомогою Entity Framework.

2. Створення прошарку Api.

3. Налаштування взаємодії між Api прошарком та Entity Framework.

## Література

* <http://www.asp.net/aspnet/samples/aspnet-web-api>
* <http://www.asp.net/web-api>

## Довідкова інформація:

Для виконання данного лабораторного практикуму необхідно встановити ASP.NET MVC4 або вище.

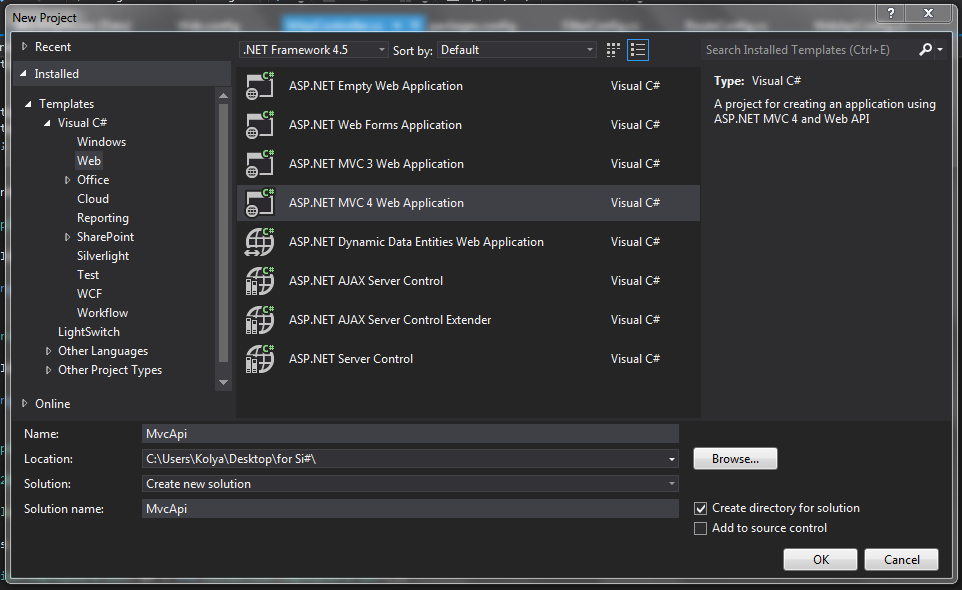
Приклад до виконання лабораторного практикуму 4

# Приклад виконання лабораторної роботи 4:

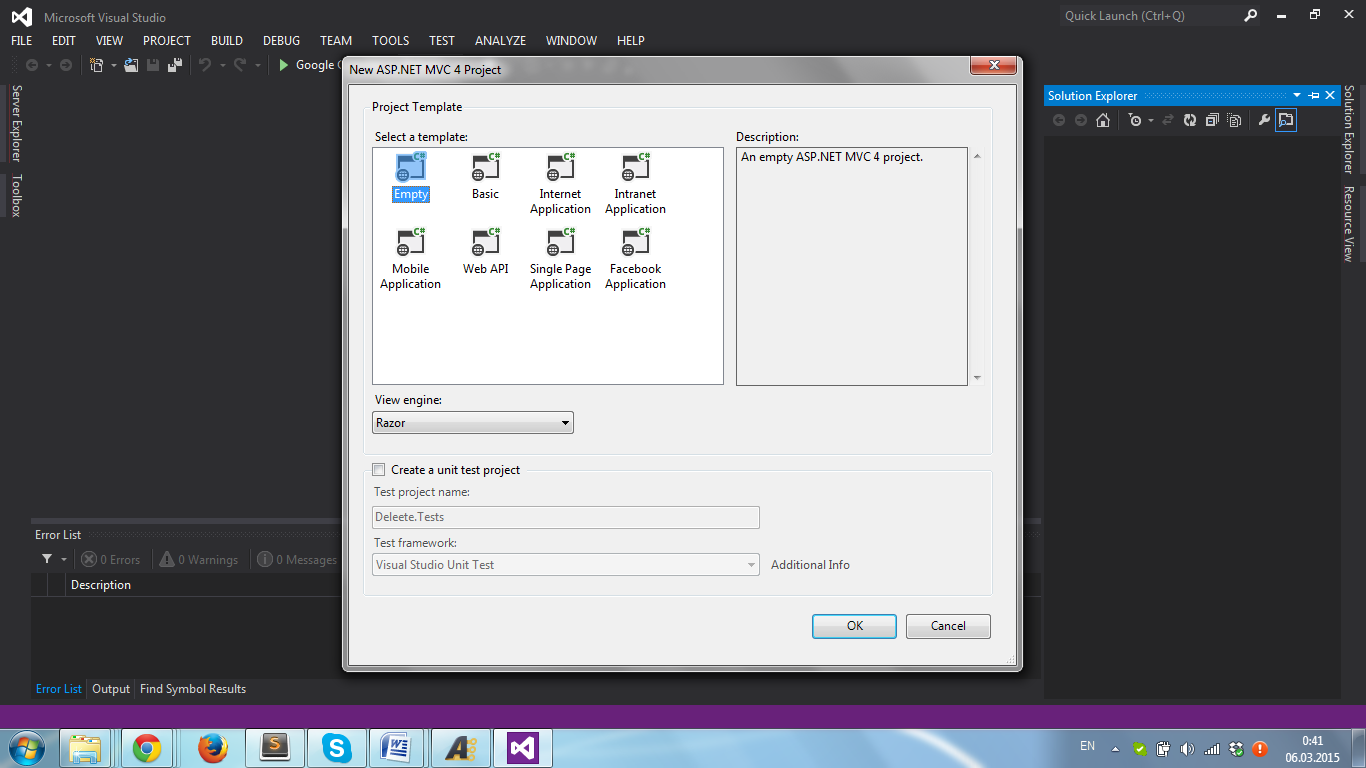
На основі створеної бази даних у лабораторних практикумах 2 і 3 реалізувати прошарок Api для отримання даних з БД (Перелік співробітників, спеціалізація співробітника) і повернення їх у форматі JSON.

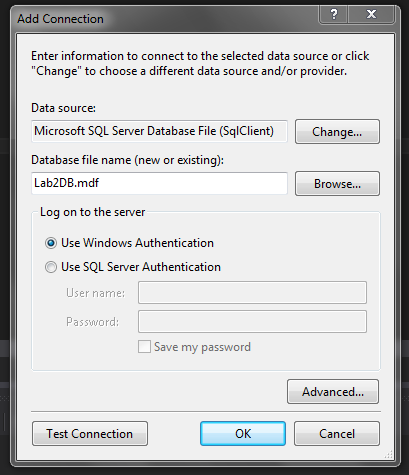
Виконання:

1.Сворити порожній проект ASP.NET MVC4

****

2. Обрати порожній проект. Для опису представлення буде використано синтаксис Razor.

****

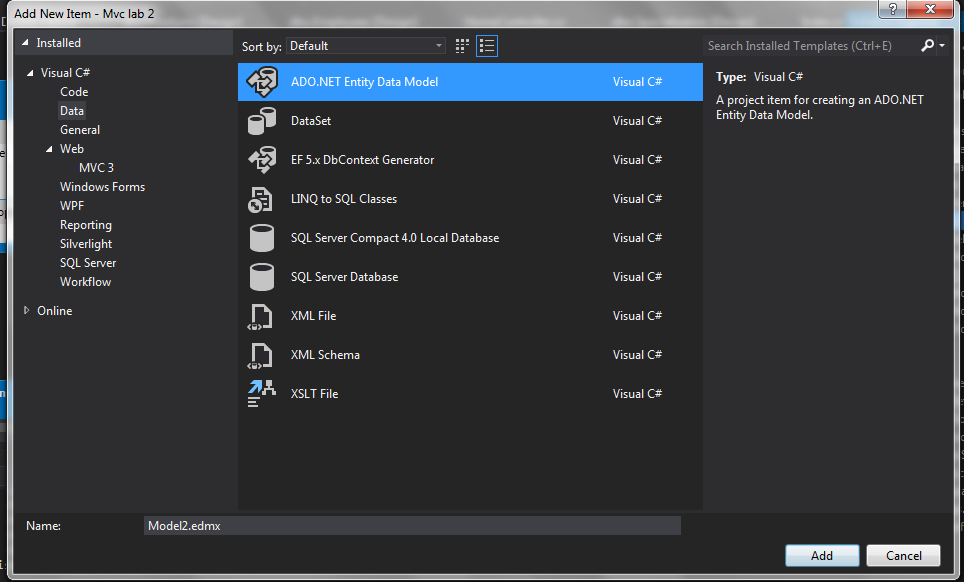
3. У проект підключаємо вже існуючу БД.

Обов‘язково натискаємо кнопку Test Connection, для перевірки з‘єднання.

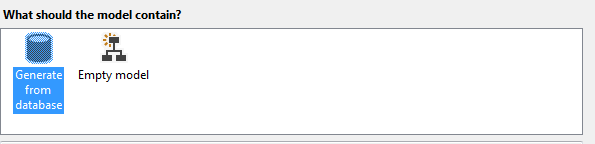
Після цього ваше з’єднання повинно з’явитися у вкладці Server Explorer.

Створена база даних не буде мати таблиць, тому їх треба створити, що і буде виконано на наступному кроці.

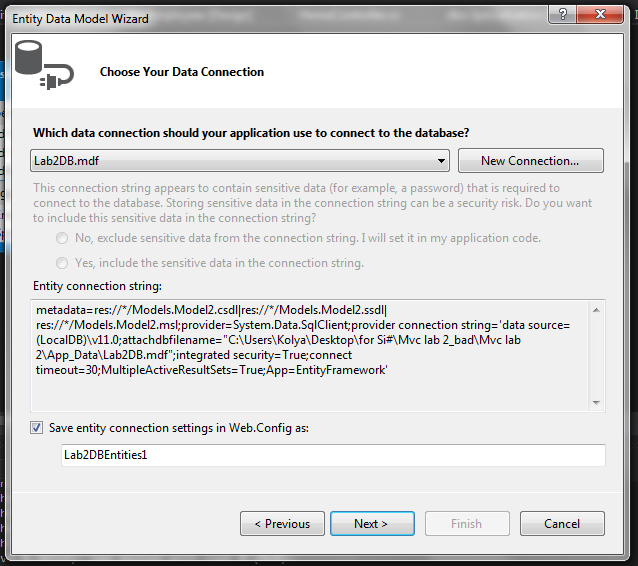
4. Для налагодження роботи між контролером та даними що зберігаються у БД необхідно створити модель, яка виступить цим зв‘язуючим прошарком. Для цього на теці ”Models” визиваємо контекстне меню і додаємо нову Entity Data Model.



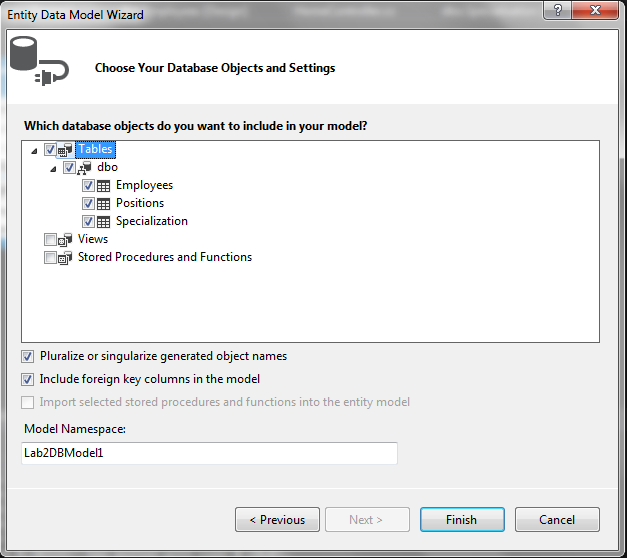
Далі виконуємо наступні дії:



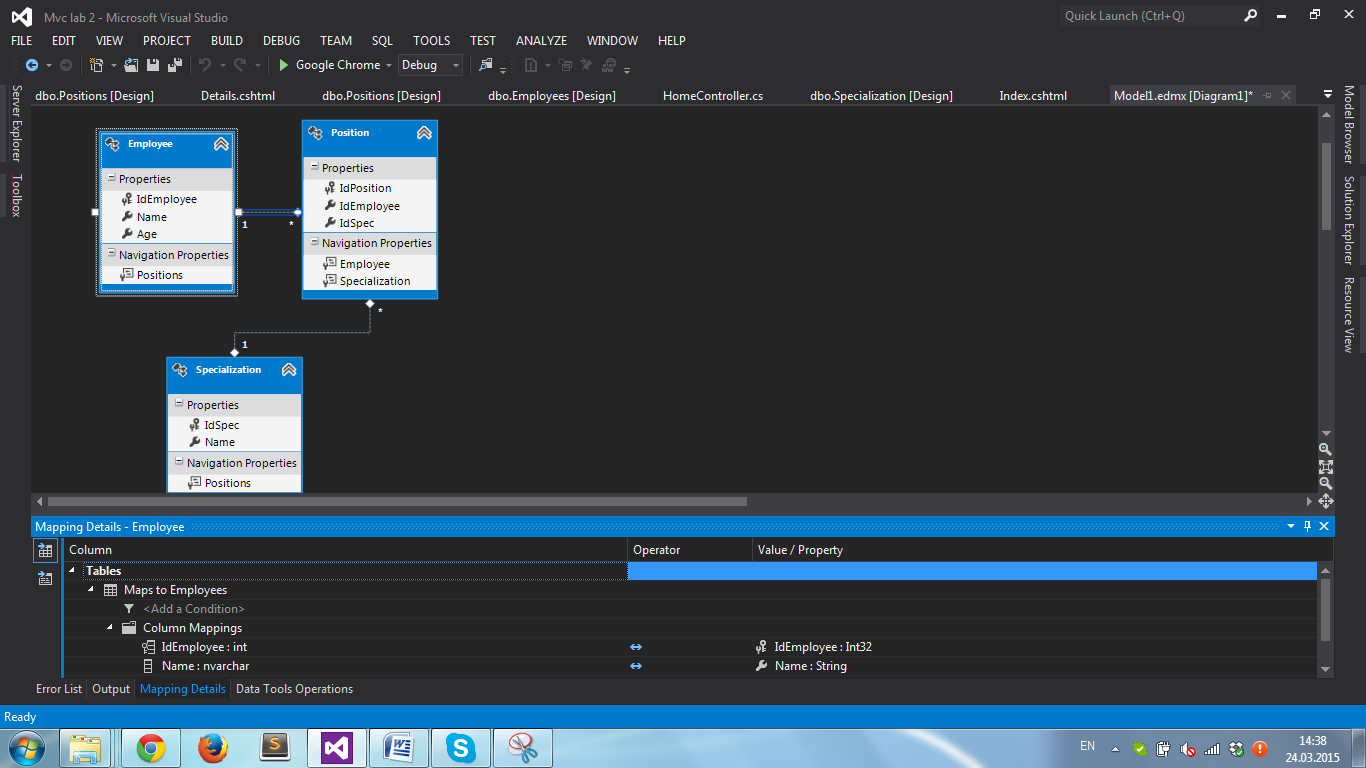
Обираємо наше підключення:



Обираємо всі таблиці, обираємо підключення зовнішніх ключів та генерацію одиничних відмінків у назвах класів.

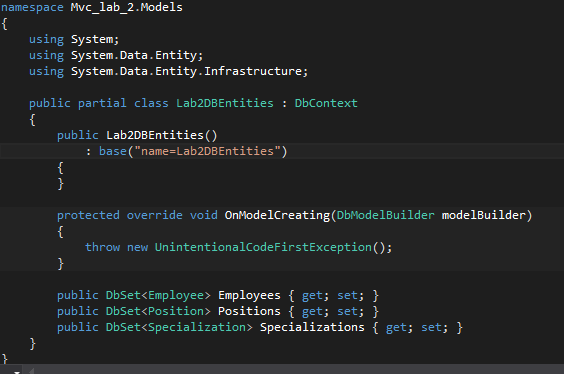


Після натискання кнопки завершити, отримаємо наступну модель:



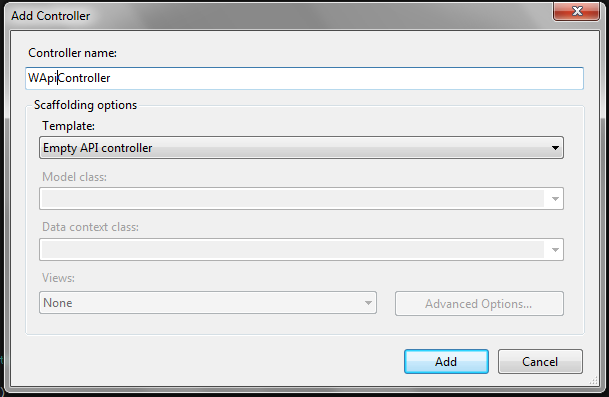
Обравши відповідну таблицю та натиснувши «Table maping» можна побачити у який тип згенерується кожне поле таблиці.

Також було згенеровано класи .Conext.cs, Employee.cs, Position.cs, Specialization.cs, які описують таблиці нашої БД і за допомогою яких ми будем взаємодіяти з нею через контролер.



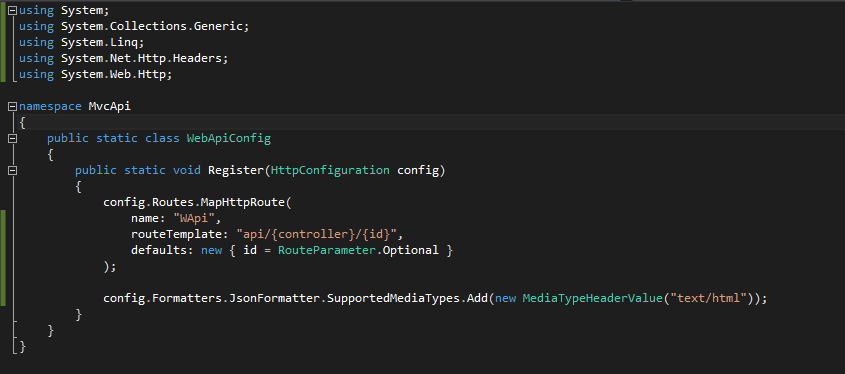
Обов‘язково треба перевірити у файлі Web.config наявність відповідної до поточного з‘єднання ConnectionString, якщо вона відсутня скопіювати її з минулих проектів.

5. Створимо новий контролер, назвемо його WApiController обравши тип контролера API:

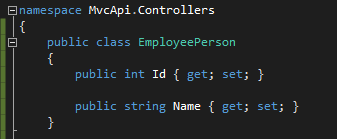


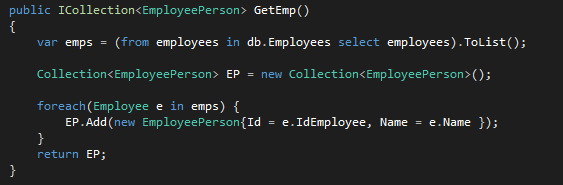
Методи WApi контролеру доступні за наступною адресою «localhost:port/api/WApi». За замовчуванням методи даного контроллеру будуть повертати данні у форматі XML, оскільки наше Api повинно повертати дані у вигляді JSON.

Для цього необхідно внести наступні зміни у файл App\_Start\WebApiConfig.cs:

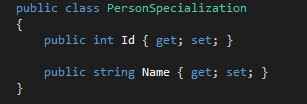


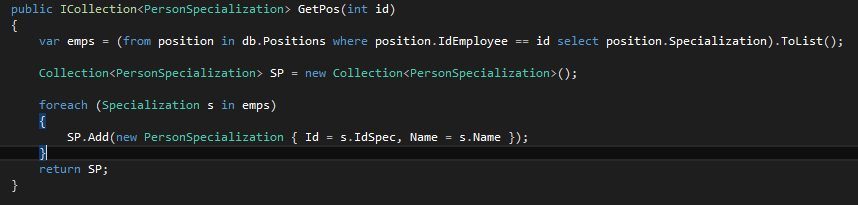
6. Додамо у наш Api контролер клас EmployeePerson який описує відображувану Api сутність співробітника та метод GetEmp який буде повертати перелік співробітників:



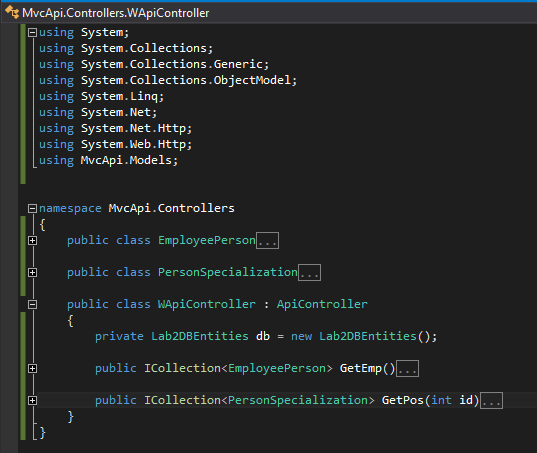


7. Додамо у наш Api контролер клас PersonSpecialization який описує відображувану Api сутність спеціалізації співробітника та метод GetPos який буде повертати перелік співробітників:



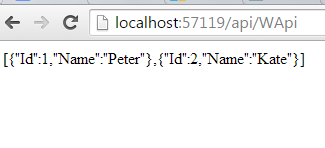


8. Весь код WApi контролера:



9. Виконаємо білд проекту, запустимо його та перевіримо роботу WApi контролеру, адреса контролера <http://localhost:57119/api/WApi>. За даною адресою виконається Api-функція GetEmp.

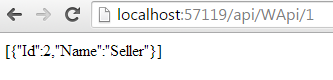
Отримаємо наступний результат:



Контролер повернув нам перелік співробітників у JSON форматі.

10. Передамо WApi контролеру параметр. Адресна строка буде мати вигляд <http://localhost:57119/api/WApi/1>. За даною адресою виконається Api-функція GetPos.

Отримаємо наступний результат:



Контролер повернув спеціалізацію співробітника у JSON форматі.

Додаток працює коректно.

## Вимоги до виконання роботи:

* Реалізувати додаток згідно варіанту
* Підготувати звіт до програми, який буде містити:
* зміст завдання
* текст додатку з коментарями
* результати перевірки коректності роботи додатку на різних значеннях параметрів (у тому числі і помилкових)
* формулювання призначення моделі, представлення та контролера.

**Завдання:**

На основі створеної бази даних у лабораторних практикумах 2 і 3 реалізувати прошарок Api для отримання даних з БД і повернення їх у форматі JSON.