# Комп'ютерний практикум 3

# Виконання CRUD операцій над моделлю у MVC ASP.NET з використанням Entity Framework.

## Мета

Практичне застосування можливостей Entity Framework в MVC ASP.NET.

## Зміст

1. Створення нової сутності за допомогою Entity Framework.

2. Оновлення вже і існуючої у БД сутності.

3. Валідація даних.

4. Видалення сутності з БД.

## Література

* ASP.NET MVC <http://habrahabr.ru/post/175999/>
* <http://www.asp.net/mvc>
* Відео семпли у простій, зрозумілій формі <https://www.youtube.com/watch?v=KEibrCrpt6I&index=1&list=PLFE6E1A5CD2570B4C>

## Довідкова інформація:

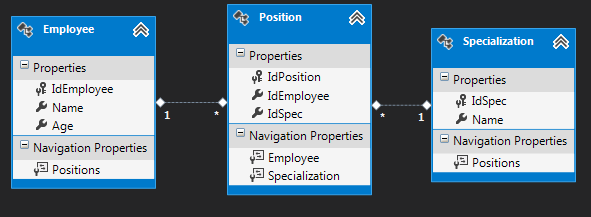
Семпли необхідні для виконання комп‘ютерного практикуму 3

# Приклад виконання CRUD операцій над моделью у MVC ASP.NET:

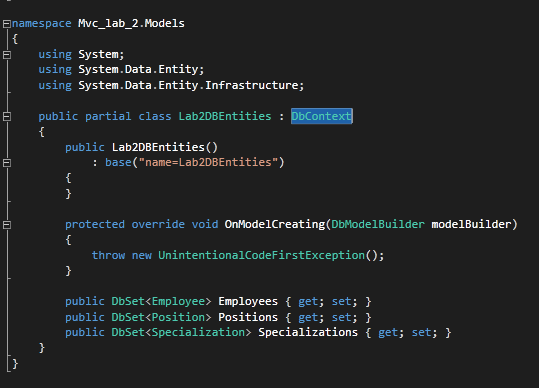
На основі таблиці, яка містить данні про Співробітників, створити MVC додаток, який буде виводити весь перелік співробітників, дозволить створювати нових співробітників, редагувати існуючих, видаляти співробітників та здійснювати валідацію даних при виконанні операцій створення та редагування сутностей.

Виконання:

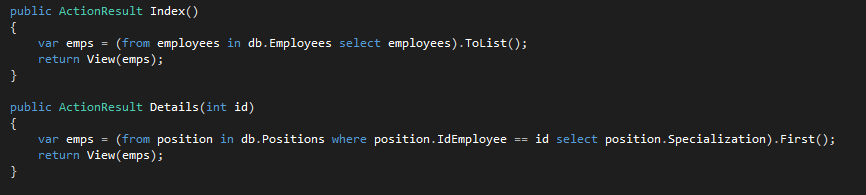
1. Відкриємо проект-приклад, що був створений у комп‘ютерному практикумі № 2. В цьому проекті було згенеровано модель, по заздалегідь створеній БД, та реалізовано необхідні екшини та представлення для відображення даних, що зберігаються у таблицях БД.

****

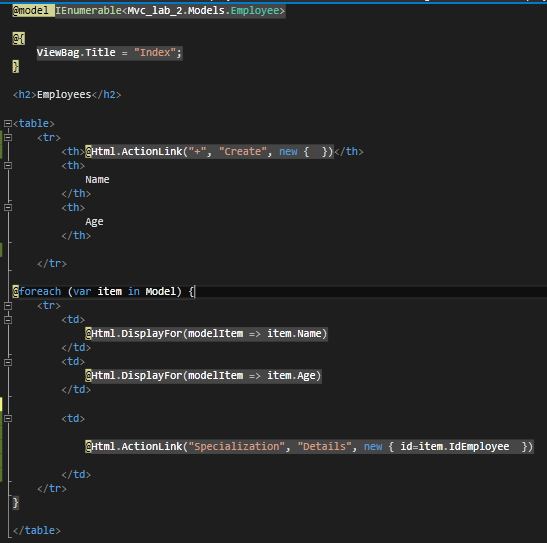
Структура БД.

****

Контекст БД.



Екшини HomeController для відображення даних БД.

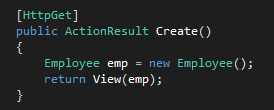


Представлення для відображення даних таблиці «Співробітники»

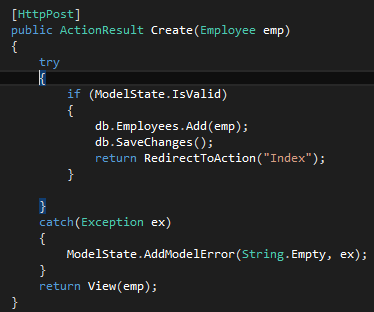
Далі саме над даними таблиці «Співробітники» будуть проводитися операції створення, редагування, видалення (CRUD) та валідації вводу даних.

2. Додамо у наш раніше створений HomeController екшини для створення нової сутності «Співробітник». Цих екшинів повинно бути два, перший для відображення форми вводу даних, який буде викликатися при запиті GET, другий для відправки даних форми на сервер, який буде викликатися при запиті POST.

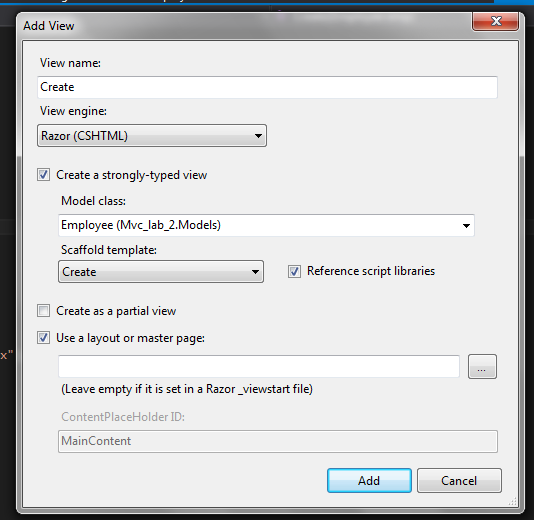
Код екшину Create для GET:



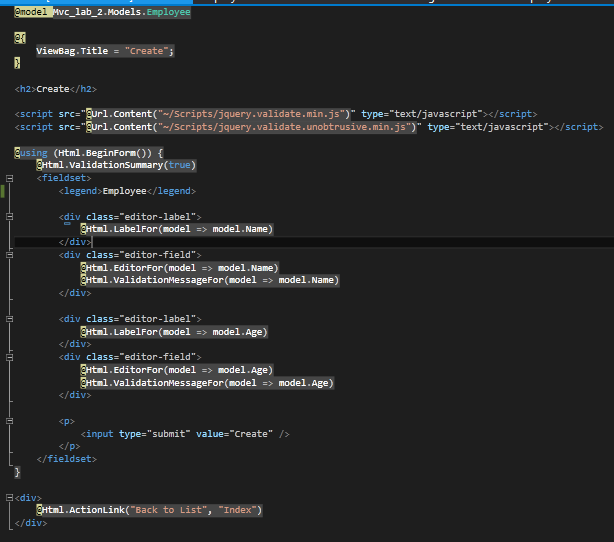
Код екшину Create для POST:



Створимо строго типізоване представлення типу ‘Create’ для відображення форми створення нової сутності.

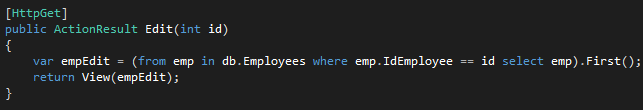


Код представлення Create:

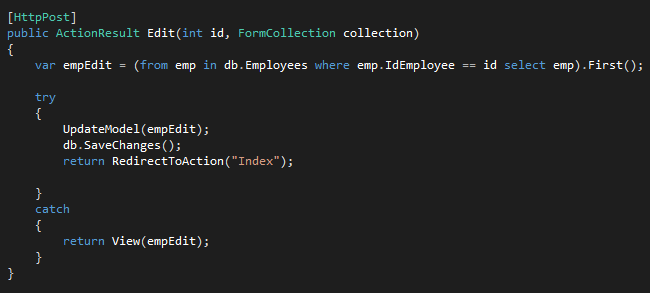


3. Додамо до контроллеру екшини та представлення для редагування вже існуючих сутностей. Для цього буде створено також два екшини для запитів GET і POST відповідно.

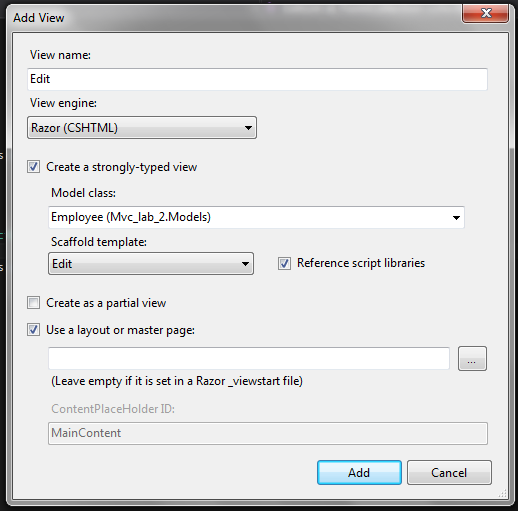
Код екшину Edit для GET:



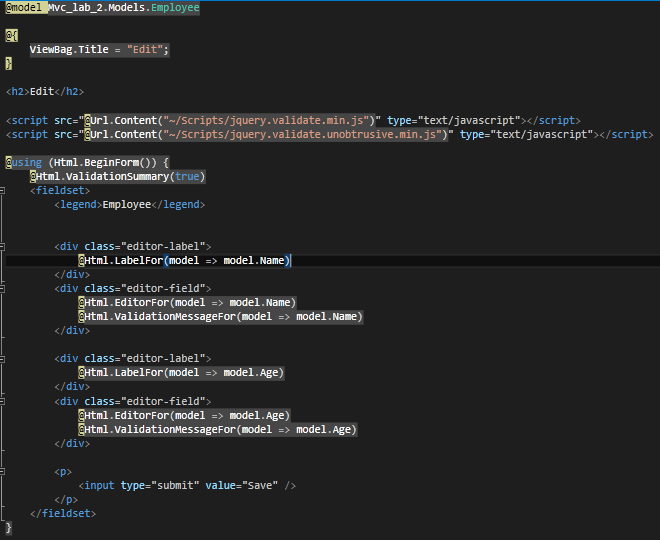
Код екшину Edit для POST:



Створимо строго типізоване представлення типу ‘Edite’ для відображення форми редагування сутності.



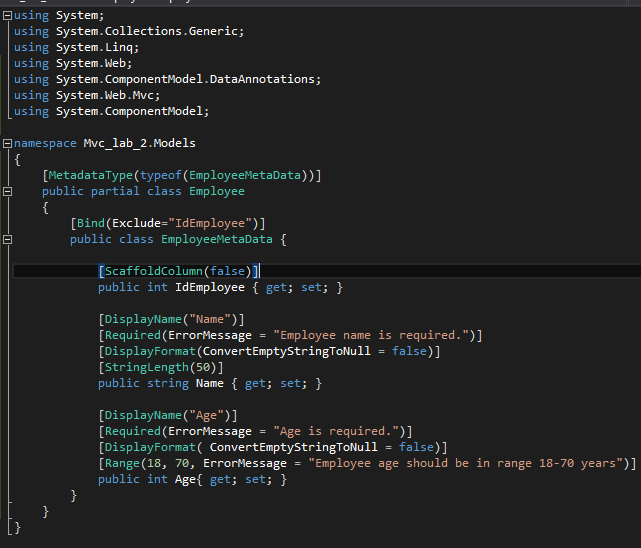
Код представлення Edit:



4. При створенні нових та редагуванні вже існуючих сутностей користувач може ввести в форму некоректні данні або не зрозуміти, які данні потрібно вводити, такі дії можливо призведуть до помилки та виникненню виключної ситуації, для попередження таких дій користувача необхідно накласти певні обмеження на данні, що вводяться користувачем, тобто перевіряти валідність введених даних.

Для реалізації валідації необхідно до папки Models додати новий класс з назвою «EmployeeValidation». В цьому файлі буде реалізований частковий класс Employee, на поля якого будуть накладені обмеження за допомогою спеціальних атрибутів.

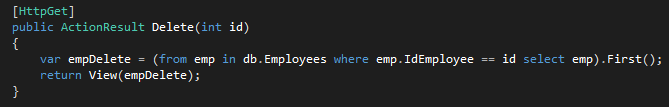
Код EmployeeValidation.cs:



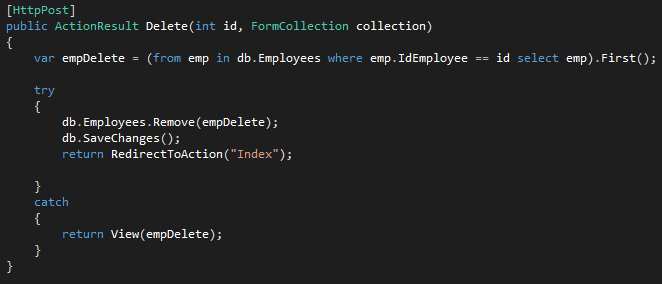
За допомогою цих атрибутів можливо задавати певний формат введення, обробляти порожні рядки, контролювати довжину певного рядка та накладати інші необхідні обмеження.

5. Додамо можливість видалення вже існуючих сутностей. Для цього у HomeController буде створено два екзини ‘Delete’ для запитів GET і POST відповідно.

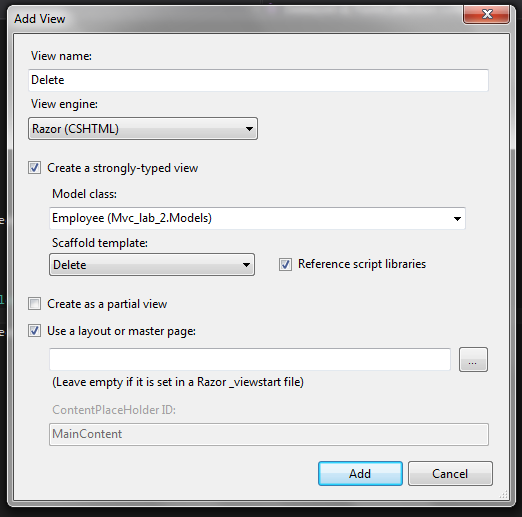
Код екшину Delete для GET:



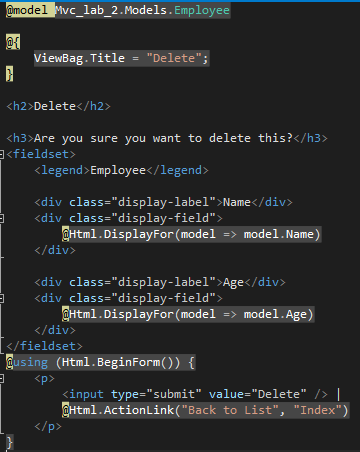
Код екшину Delete для POST:



Створимо строго типізоване представлення типу ‘Delete’ для відображення форми видалення сутності.

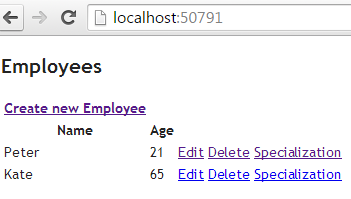


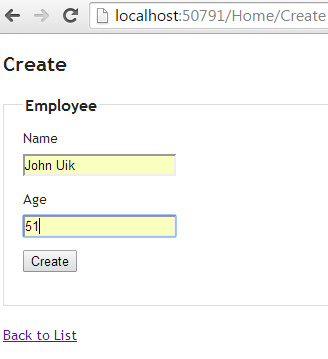
Код представлення Delete:

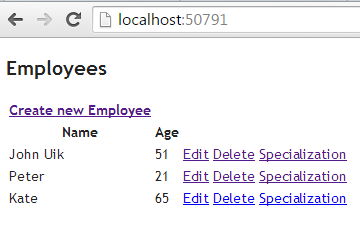


6. Зробимо білд проекту та запустимо наш додаток.

Створимо нового співробітника:

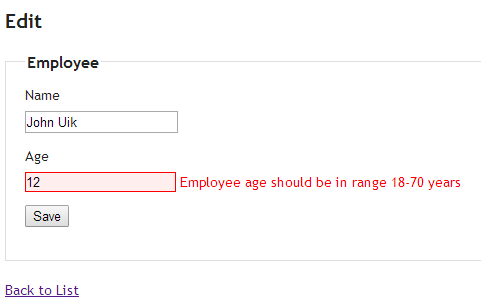




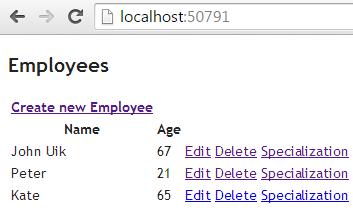


Нова сутність створена успішно.

Виконаємо редагування існуючої сутності:

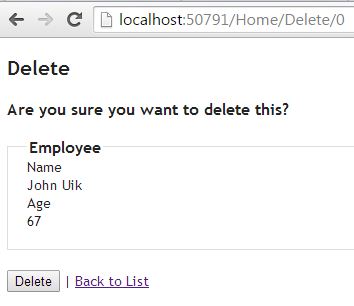


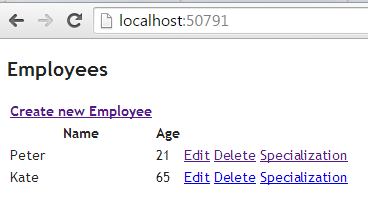
Валідація спрацювала на некоректні данні.



Редагування виконано успішно.

Виконаємо видалення існуючої сутності:





Видалення виконано успішно.

Всі CRUD операції виконуються успішно. 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:**

Створити MVC додаток, на основі створеної БД у практикумі № 2, який окрім виводу, буде давати можливість створювати нові записи до БД, редагувати існуючі та видаляти їх. Також реалізувати валідацію при створенні нових та редагуванні існуючих сутностей.