# Комп'ютерний практикум 5

# Налаштування роутынгу у WebApi MVC ASP.NET та виконання використанням WebApi.

## Мета

Ознайомлення з роумінгом у WebApi та виконання CRUD операцій за допомогою WebAPI в MVC ASP.NET.

## Зміст

1. Налаштування роутінгу API контроллеру .

2. Реалізація CRUD операцій за допомогою WebAPI.

## 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. Плагін для Chrome Postman, або його аналоги у інших браузерах.

Приклад до виконання лабораторного практикуму 5

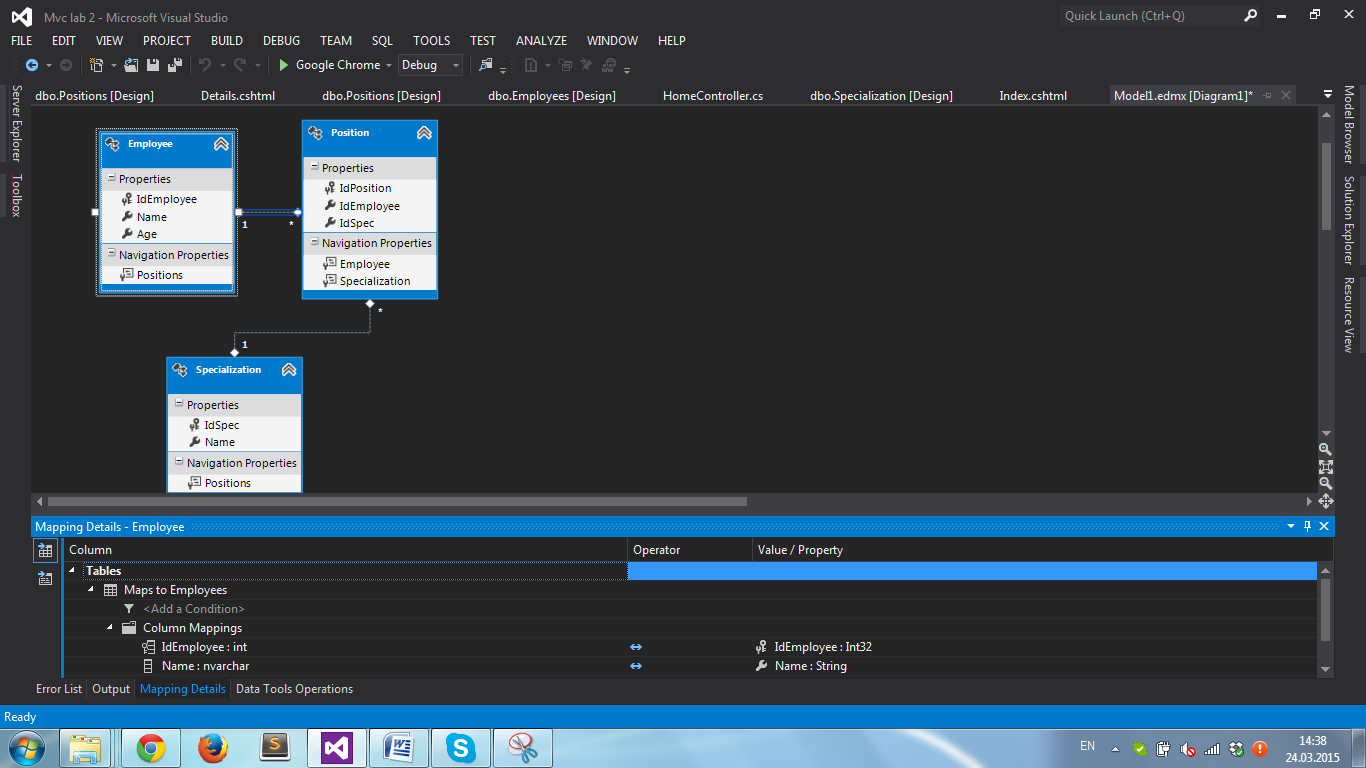
# Приклад виконання лабораторної роботи 5:

На основі лабораторного практикуму 4 реалізувати прошарок Api для додавання, оновлення, видалення та отримання даних з БД (Перелік спіробітників, спеціалізація співробітника) і повернення статусу виконаної операції у форматі JSON.

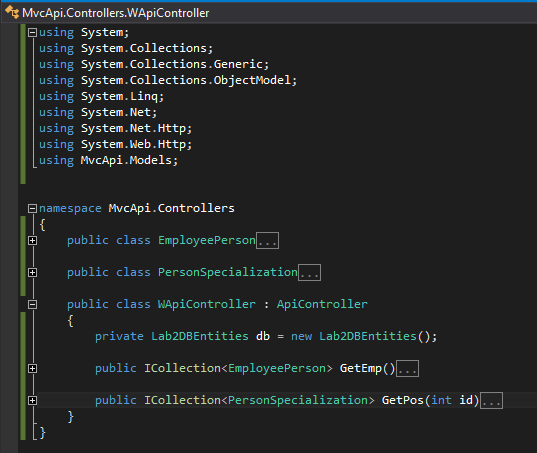
Виконання:

1. Відкриємо проект лабораторного практикуму 4. В цьому проекті було створено API контролер, під’єднано Entity модель раніше створеної БД та реалізовано операції отримання даних з БД:

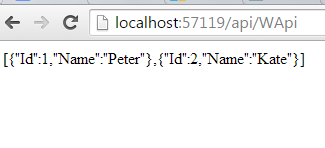
*Модель БД*

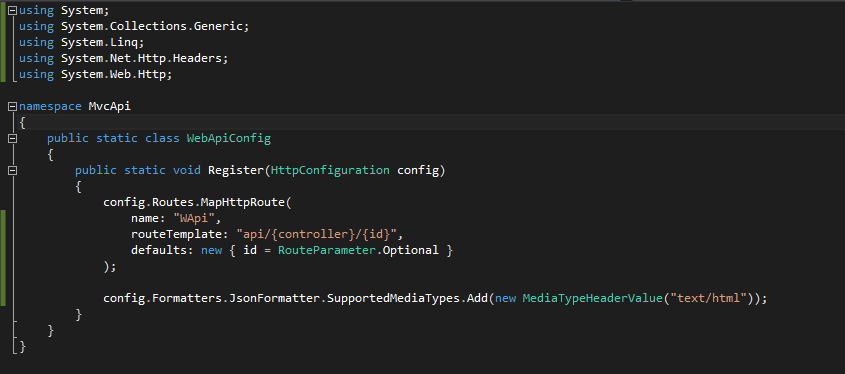


*API контроллер*

**

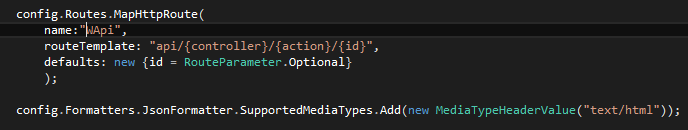
*Результат у форматі JSON*



2. В данному проекті API контролер сам обирає який Action викликати в залежності від отриманого шаблону параметрів. Конфігураційний файл при таких налаштуваннях має наступний вигляд:

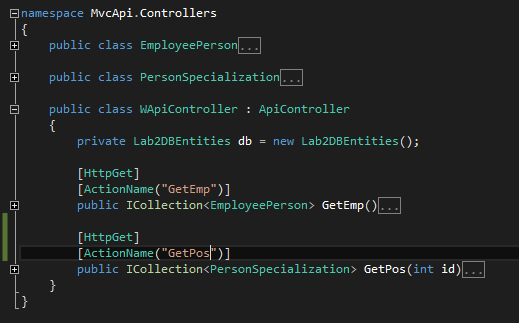
URL: 

Такий шаблон роутінгу не є досить гнучким та зручним. Зручніше було б власноруч обирати Action який треба викликати. В такому разі необхідно додати в існуючий шаблон додатковий прошарок «action», для цього треба внести певні зміни у конфігураційний файл роутінгу, після чого він матиме наступний вигляд:



Після внесення змін у конфігураційний файл роутінгу, треба додати відповідні назви екшинів у контролері. Для цього треба над кожним екшином додати атрибут ActionName з відповідною назвою екшину.

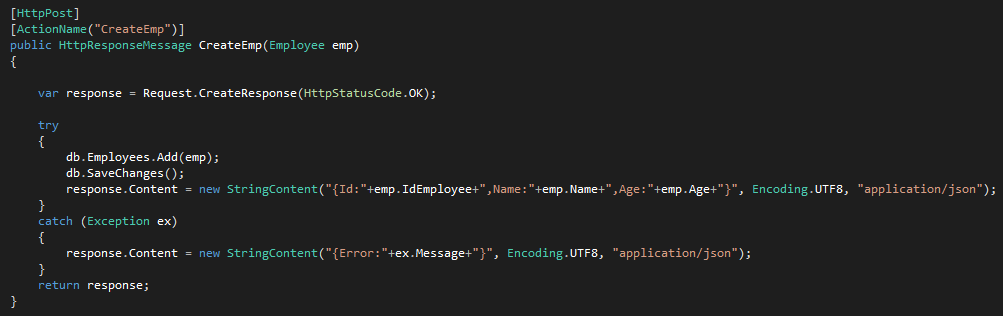
Після внесення змін контролер матиме наступний вигляд:

****

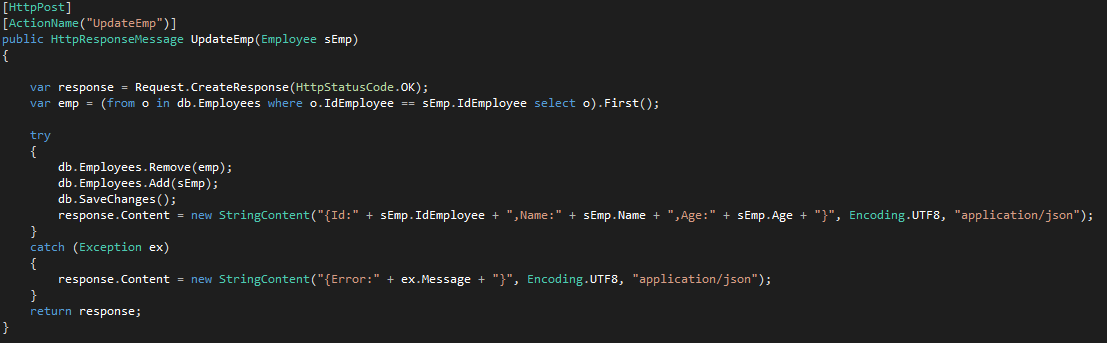
Новий шаблон роутінгу матиме наступний URL:

****

3. Реалізуємо можливість створення нового запису у таблиці БД. Для цього додамо у наш контролер екшн типу POST з назвою CreateEmp який на вхід буде приймати об’єкт Employee. Цей екшн матиме наступний вигляд:

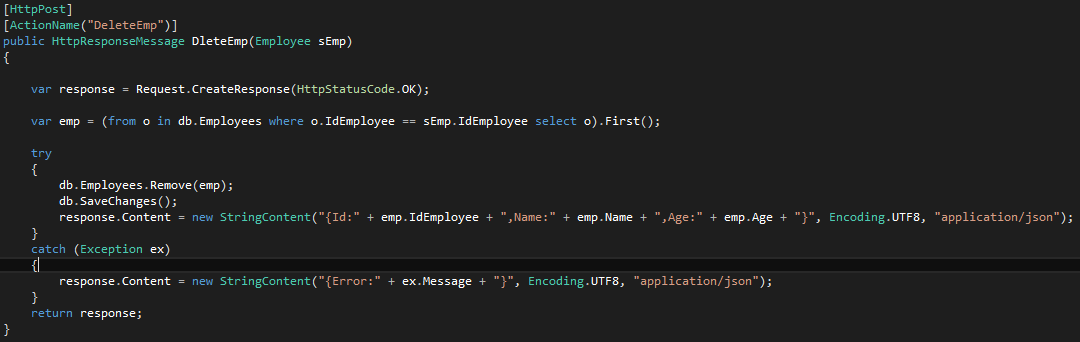


У разі вдалого створення нової сутності співробітника, екшн поверне нову сутність співробітника у форматі JSON, у разі помилки буде повернено текст помилки у форматі JSON.

4. Реалізуємо можливість оновлення вже існуючого запису у таблиці БД. Для цього додамо у наш контролер екшн типу POST з назвою UpdateEmp який на вхід буде приймати об’єкт Employee. Цей екшн матиме наступний вигляд:

У разі вдалого оновлення існуючої сутності співробітника, екшн поверне оновлену сутність співробітника у форматі JSON, у разі помилки буде повернено текст помилки у форматі JSON.

5. Реалізуємо можливість видалення існуючого запису з таблиці БД. Для цього додамо у наш контролер екшн типу POST з назвою DeleteEmp який на вхід буде приймати об’єкт Employee. Цей екшн матиме наступний вигляд:



У разі вдалого видалення сутності співробітника, екшн поверне видалену сутність співробітника у форматі JSON, у разі помилки буде повернено текст помилки у форматі JSON.

Після завершення внесення змін до нашого API контролеру необхідно виконати білд проекту та запустити його.

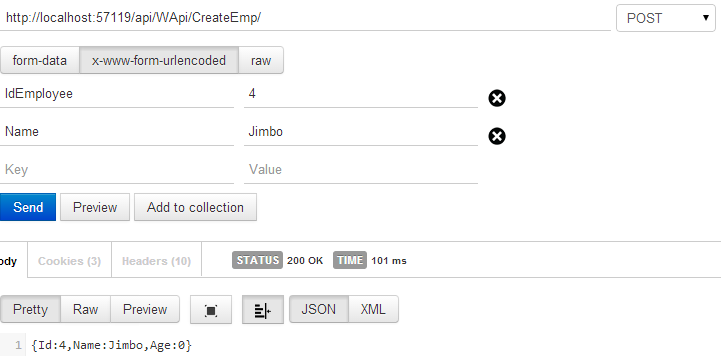
6. Для перевірки коректності роботи нашого Web-додатку варто скористатись плагіном Postman для зручної відправки POST запитів, та отримання результатів їх виконання.

Після запуску проекту, запустимо пагін Postman та зробимо GET запит на отримання переліку співробітників:



Отриманий результат є коректним, екшен працює правильно.

Зробимо POST запит на створення нової сутності співробітника:



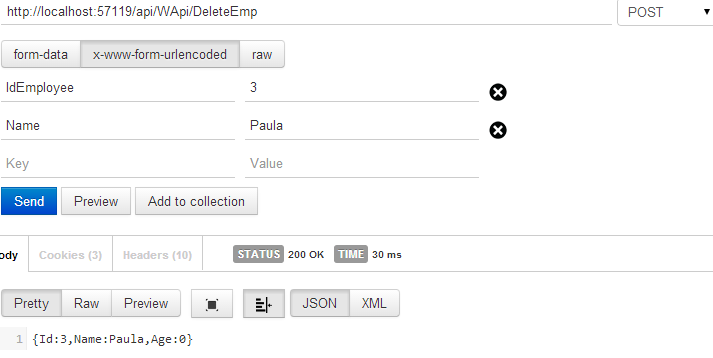
Отриманий результат є коректним, екшен працює правильно.

Зробимо POST запит на оновлення вже існуючої сутності співробітника:



Отриманий результат є коректним, екшен працює правильно.

Зробимо POST запит на видалення існуючої сутності співробітника:



Отриманий результат є коректним, екшен працює правильно.

Зробимо GET запит на отримання переліку співробітників, щоб переконатися у внесенні змін до нашої БД:



Всі екшени відпрацювали коректно, 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 лабораторного практикуму 4 реалізувати прошарок Api для додавання, оновлення, видалення та отримання даних з БД і повернення статусу виконанної операції у форматі JSON.