

## Ćwiczenia 4

W trakcie niniejszych ćwiczeń do wykonania jest prosta aplikacja REST API, która umożliwia wykonanie operacji pozwalających na modyfikowanie danych w bazie SQL Server. Razem z zadaniem załączony jest skrypt pozwalający na stworzenie tabelki Animals i wypełnienie jej danymi. Komunikacja z bazą danych powinna odbywać się poprzez klasy **SqlConnection/SqlCommand**.

Animal		
IdAnimal	int	PK
Name	nvarchar(200)	
Description	nvarchar(200)	N
Category	nvarchar(200)	
Area	nvarchar(200)	

Figure 1 - klucz główny jest automatycznie generowany (IDENTITY)

Dane serwera: **db-mssql16.pjwstk.edu.pl**

1. Dodaj **kontroler Animals**
2. Dodaj metodę/endpoint pozwalającą na uzyskanie listy zwierząt. Końcówka powinna reagować na żądanie typu **HTTP GET** wysłane na adres **/api/animals**
  - a. Końcówka powinna pozwolić na przyjęcie parametru w **query string**, który określa sortowanie. Parametr nazywa się **orderBy**. Przykład: **api/animals?orderBy=name**
  - b. Parametr jako dostępne wartości przyjmuje: name, description, category, area. Możemy sortować wyłącznie po jednej kolumnie. Sortowanie jest zawsze w kierunku „**ascending**”.
  - c. Domyślne sortowanie (kiedy w żądaniu nie zostanie przekazany parametr w query string) powinna odbywać się po kolumnie **name**.
3. Dodaj metodę/endpoint pozwalający na dodanie nowego zwierzęcia.
  - a. Metoda powinna odpowiadać na żądanie **HTTP POST** na adres **api/animals**
  - b. Metoda powinna przyjmować dane w postaci JSON

4. Dodaj metodę/endpoint pozwalający na **aktualizację danych** konkretnego zwierzęcia.
  - a. Metoda powinna odpowiadać na żądanie **HTTP PUT** wysłane na adres **/api/animals/{idAnimal}**
  - b. Metoda przyjmuje dane w postaci JSON'a
  - c. Zakładamy, że klucze główne nie ulegają modyfikacji (kolumna IdAnimal)
5. Dodaj metodę/endpoint do **usuwania danych** na temat konkretnego zwierzęcia.
  - a. Metoda powinna odpowiadać na żądanie **HTTP DELETE** wysłane na adres **/api/animals/{idAnimal}**
6. Pamiętaj o poprawnych kodach **HTTP**
7. Postaraj się skorzystać z wbudowanego mechanizmu do **DependencyInjection**.
8. Dbaj o walidację danych
9. Dbaj o nazewnictwo i styl