

# LABORATORIUM NIERELACYJNE BAZY DANYCH

**Data wykonania  
ćwiczenia:**

04.06.2023

**Rok studiów:**

3

**Semestr:**

6

**Grupa studencka:**

2

**Grupa laboratoryjna:**

2B

**Ćwiczenie nr.**

9

**Temat:** Tworzenie projektu MongoDB na platformie ASP.NET MVC 5

**Osoby wykonujące ćwiczenia:**

1. Igor Gawłowicz

Katedra Informatyki i Automatyki

# Tworzenie projektu MongoDB na platformie ASP.NET MVC 5

## 1. Wprowadzenie

Celem naszego sprawozdania jest opisanie szczegółowych kroków niezbędnych do stworzenia projektu ASP.NET MVC 5 wykorzystującego bazę danych MongoDB. Naszym celem jest umożliwienie operacji CRUD (Create, Read, Update, Delete) na danych przechowywanych w bazie MongoDB poprzez interfejs użytkownika aplikacji ASP.NET MVC 5.

## 2. Zadanie do wykonania

### 1. Tworzenie nowego projektu ASP.NET MVC 5

1. Otwieramy Visual Studio lub Visual Studio Code.
2. Wybieramy opcję utworzenia nowego projektu i wybieramy szablon ASP.NET MVC 5.
3. Nazywamy projekt i wybieramy lokalizację zapisu.

### 2. Instalacja pakietu MongoDB Driver dla C#

1. Otwieramy menedżer pakietów NuGet w naszym projekcie ASP.NET MVC 5.
2. Instalujemy pakiet MongoDB.Driver za pomocą poniższej komendy `Install-Package MongoDB.Driver`.

### 3. Konfiguracja połączenia z bazą danych MongoDB

1. Otwieramy plik `Web.config` w naszym projekcie ASP.NET MVC 5.
2. Dodajemy sekcję konfiguracji połączenia z bazą danych MongoDB:

```
<appSettings>
  <add key="MongoDBConnectionString" value="mongodb://localhost:27017" />
</appSettings>
```

### 4. Implementacja kontrolerów i widoków

1. Tworzymy kontrolery i widoki, aby umożliwić dodawanie, edycję i usuwanie danych z bazy MongoDB.

```
// Kontroler do obsługi operacji CRUD na danych z bazy MongoDB
public class ItemController : Controller
{
    private IMongoCollection<Item> _itemCollection;

    public ItemController()
    {
        var client = new
MongoClient(ConfigurationManager.AppSettings["MongoDBConnectionString"]);
        var database = client.GetDatabase("sklep");
        _itemCollection = database.GetCollection<Item>("Items");
    }
}
```

```

    }

    public ActionResult Index()
    {
        var items = _itemCollection.Find(_ => true).ToList();
        return View(items);
    }

    public ActionResult Create()
    {
        return View();
    }

    [HttpPost]
    public ActionResult Create(Item item)
    {
        _itemCollection.InsertOne(item);
        return RedirectToAction("Index");
    }

    public ActionResult Edit(string id)
    {
        var item = _itemCollection.Find(i => i.Id == id).FirstOrDefault();
        return View(item);
    }

    [HttpPost]
    public ActionResult Edit(Item item)
    {
        _itemCollection.ReplaceOne(i => i.Id == item.Id, item);
        return RedirectToAction("Index");
    }

    public ActionResult Delete(string id)
    {
        _itemCollection.DeleteOne(i => i.Id == id);
        return RedirectToAction("Index");
    }
}

```

2. Tworzymy widoki dla akcji kontrolera `ItemController` w folderze `Views/Item`.

```

@model IEnumerable<Item>

<!DOCTYPE html>
<html>
<head>
    <title>Lista przedmiotów</title>
</head>
<body>

    <h2>Lista przedmiotów</h2>

```

```

<p>
    @Html.ActionLink("Dodaj nowy przedmiot", "Create")
</p>

<table class="table">
    <tr>
        <th>Nazwa</th>
        <th>Opis</th>
        <th>Akcje</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>@item.Name</td>
            <td>@item.Description</td>
            <td>
                @Html.ActionLink("Edytuj", "Edit", new { id = item.Id }) |
                @Html.ActionLink("Usuń", "Delete", new { id = item.Id })
            </td>
        </tr>
    }
</table>

</body>
</html>

```

## Pytania kontrolne

1. Jakie pakiety NuGet są potrzebne do integracji z bazą danych MongoDB w projekcie ASP.NET MVC 5?
  - Do integracji z bazą danych MongoDB w projekcie ASP.NET MVC 5 potrzebny jest pakiet **MongoDB.Driver**.
2. Jak skonfigurować połączenie z bazą danych MongoDB w projekcie ASP.NET MVC 5?
  - Połączenie z bazą danych MongoDB w projekcie ASP.NET MVC 5 można skonfigurować poprzez dodanie sekcji konfiguracji w pliku **Web.config** i ustawienie właściwego połączenia w postaci connection stringa.
3. Jak można użyć klasy MongoClient w kontrolerach ASP.NET MVC 5 do nawiązania połączenia z bazą danych MongoDB?
  - W kontrolerach ASP.NET MVC 5 można użyć klasy **MongoClient**, aby nawiązać połączenie z bazą danych MongoDB, poprzez utworzenie obiektu **MongoClient** i przekazanie mu connection stringa z konfiguracji.
4. Jakie są podstawowe operacje CRUD w bazie danych MongoDB, i jak można je wykonać w projekcie ASP.NET MVC 5?

- Podstawowe operacje CRUD w bazie danych MongoDB to: Create (dodawanie), Read (odczytywanie), Update (aktualizacja) i Delete (usuwanie) danych. W projekcie ASP.NET MVC 5 operacje te można wykonać poprzez odpowiednie akcje w kontrolerach, które będą korzystały z metod dostępnych w obiekcie `IMongoCollection`.
5. Jakie dodatkowe funkcje mogą zostać dodane do aplikacji ASP.NET MVC 5 w oparciu o integrację z bazą danych MongoDB?
- Dodatkowe funkcje mogą obejmować filtrowanie danych, paginację, walidację danych, uwierzytelnianie i autoryzację oraz inne zaawansowane operacje dostępne w MongoDB, takie jak agregacje.
6. Jak można przetestować działanie aplikacji ASP.NET MVC 5 zintegrowanej z bazą danych MongoDB?
- Działanie aplikacji ASP.NET MVC 5 zintegrowanej z bazą danych MongoDB można przetestować poprzez uruchomienie aplikacji i przeprowadzenie testów funkcjonalnych, które sprawdzą poprawność działania operacji CRUD oraz innych funkcji dostępnych w aplikacji.
7. Jaką dokumentację należy przygotować po zakończeniu laboratorium, aby ułatwić zrozumienie procesu tworzenia aplikacji ASP.NET MVC 5 z wykorzystaniem MongoDB?
- Po zakończeniu laboratorium warto przygotować dokumentację zawierającą opis procesu tworzenia aplikacji ASP.NET MVC 5, instrukcje instalacji, opis konfiguracji, opis wykorzystanych funkcji MongoDB oraz przykłady kodu.
8. Jakie są korzyści integracji bazy danych MongoDB z aplikacją ASP.NET MVC 5 w porównaniu do tradycyjnych relacyjnych baz danych?
- Korzyści integracji bazy danych MongoDB z aplikacją ASP.NET MVC 5 mogą obejmować skalowalność, elastyczność schematu, wydajność operacji zapisu i odczytu oraz łatwość pracy z danymi w formacie dokumentów.
9. Jakie są potencjalne wyzwania związane z integracją bazy danych NoSQL, takiej jak MongoDB, z aplikacją opartą na platformie ASP.NET MVC 5, i jak można je rozwiązać?
- Potencjalne wyzwania mogą obejmować konieczność nauki nowych technologii i narzędzi, zarządzanie elastycznym schematem danych oraz optymalizację zapytań do bazy danych. Wyzwania te można rozwiązać poprzez odpowiednie szkolenie, stosowanie najlepszych praktyk w projektowaniu aplikacji oraz optymalizację zapytań i indeksów w bazie danych.