

				
<p><b>POLITECHNIKA ŚLĄSKA WYDZIAŁ INŻYNIERII MATERIAŁOWEJ</b> <b>KATEDRA INFORMATYKI PRZEMYSŁOWEJ</b></p>				
<p><b>Programowanie Bazodanowe</b></p>				
Rok akademicki	Semestr	Kierunek	Studenci	Prowadzący
2023/2024	II	MIP	Łukasz Zamojski	Dr inż. Piotr Ciepliński

## Programowanie Bazodanowe – sprawozdanie z laboratoriów

## SQL:

### Opis:

W pliku Program.cs ustawiono wstrzyknięcie implementacji serwisów obsługujących bazę danych poprzez procedury SQL.

```
using BLL.ServiceInterfaces;
using BLL_DB;
using DAL;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
builder.Services.AddDbContext<WebshopContext>();
builder.Services.AddScoped<IProducts, Products>();
builder.Services.AddScoped<IBasket, Basket>();
// Learn more about configuring Swagger/OpenAPI at https://docs.microsoft.com/en-us/aspnet/core/fundamentals/swagger
builder.Services.AddEndpointsApiExplorer();
```

Dostęp serwisów do danych odbywa się za pośrednictwem klasy SqlConnection.

```
private readonly string _connectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=WebshopSel;Integrated Security=True;Connect Timeout=30;Encrypt=False;";
Odwolania: 2 | lukazam890, 27 dni temu | 1 autor, liczba zmian: 2
public void addToBasket(BasketPositionRequestDTO basket)
{
    using (SqlConnection connection = new SqlConnection(_connectionString))
    {
        string query = "exec AddToBasket " + basket.UserID + ", " + basket.ProductID + ", " + basket.Amount + ", " + basket.Price;
        SqlCommand sqlCommand = new SqlCommand(query, connection);
        sqlCommand.Connection.Open();
        sqlCommand.ExecuteNonQuery();
        sqlCommand.Connection.Close();
    }
}

Odwolania: 2 | lukazam890, 49 dni temu | 1 autor, liczba zmian: 2
public void changeNumberOfProducts(int id, int numberOfProducts)
{
    using (SqlConnection connection = new SqlConnection(_connectionString))
    {
        string query = "exec ChangeNumber " + id + ", " + numberOfProducts;
        SqlCommand sqlCommand = new SqlCommand(query, connection);
        sqlCommand.Connection.Open();
        sqlCommand.ExecuteNonQuery();
        sqlCommand.Connection.Close();
    }
}
}
```

Operacje na danych są przeprowadzane przez procedury SQL znajdujące się na serwerze bazy danych.

```
CREATE PROCEDURE [dbo].[AddToBasket]
    @userid int,
    @productid int,
    @amount int,
    @price int
AS
INSERT INTO BasketPositions (UserID, ProductID, Amount, Price)
VALUES (@userid, @productid, @amount, @price);
RETURN 0
```

## Products:

Odczyt listy produktów:

### Products

GET /api/Products

Parameters

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5107/api/Products' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5107/api/Products
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "id": 1,   "name": "Product1",   "price": 123,   "image": "10",   "isActive": true,   "groupId": 1,   "basketPositions": null }, {   "id": 2,   "name": "Product2",   "price": 134,   "image": "10",   "isActive": true,   "groupId": 3,   "basketPositions": null }, {   "id": 3,   "name": "Product3",   "price": 1256,   "image": "10",   "isActive": true,   "groupId": 3,   "basketPositions": null } }</pre></div> <div>Response headers</div> <div><pre>content-type: application/json; charset=utf-8 date: Wed, 15 May 2024 12:17:21 GMT server: Kestrel transfer-encoding: chunked</pre></div>

Responses

Code	Description	Links
200	Success	No links

## Dodawanie nowego produktu:

POST

/api/Products

Parameters

No parameters

Request body

application/json

```
{  "name": "Test10",  "price": 321,  "image": "10",  "isActive": true,  "groupID": 1}
```

Execute

Responses

Code	Description	Links
200	Success	No links

Code

Details

200

Response body

```
{  "id": 2,  "name": "Product2",  "price": 156,  "image": "10",  "isActive": true,  "groupID": 3,  "basketPositions": null},  {  "id": 3,  "name": "Product3",  "price": 1256,  "image": "10",  "isActive": true,  "groupID": 3,  "basketPositions": null},  {  "id": 4,  "name": "Test10",  "price": 321,  "image": "10",  "isActive": true,  "groupID": 1,  "basketPositions": null}
```

Response headers

content-type: application/json; charset=utf-8

Download

## Dezaktywacja produktu:

GET

/api/Products/disable

Parameters

Cancel

Name	Description
id	integer(\$int32) (query)

Execute

Responses

Code	Description	Links
200	Success	No links

Response body

```
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 3,
  "name": "Product3",
  "price": 1256,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": false,
  "groupID": 1,
  "basketPositions": null
}
}
```

Download

Response headers

## Aktywacja produktu:

GET /api/Products/enable

Parameters

Cancel

Name	Description
id	
integer(\$int32)	4
(query)	

Execute

Responses

Code	Description	Links
200	Success	No links

Code

Details

200

Response body

```
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 3,
  "name": "Product3",
  "price": 1256,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
}
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 15 May 2024 12:21:18 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Usunięcie produktu:

DELETE

/api/Products

Parameters

Cancel

Name	Description
id	
integer(\$int32)	
(query)	

Execute

Responses

Code	Description	Links
200	Success	No links

Response body

```
{
  {
    "id": 1,
    "name": "Product1",
    "price": 123,
    "image": "10",
    "isActive": true,
    "groupID": 1,
    "basketPositions": null
  },
  {
    "id": 2,
    "name": "Product2",
    "price": 134,
    "image": "10",
    "isActive": true,
    "groupID": 3,
    "basketPositions": null
  },
  {
    "id": 4,
    "name": "Test10",
    "price": 321,
    "image": "10",
    "isActive": true,
    "groupID": 1,
    "basketPositions": null
  }
}
```

Download

## Basket:

Utworzenie basket:

POST

/api/Basket

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{  "productId": 4,  "userId": 1,  "amount": 2,  "price": 321}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \  'http://localhost:5107/api/Basket' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '{  "productId": 4,  "userId": 1,  "amount": 2,  "price": 321  }'
```

Request URL

http://localhost:5107/api/Basket

Server response

Code	Details
200	<div><div>Response headers</div><div><pre>content-length: 0  date: Wed, 15 May 2024 12:23:11 GMT  server: Kestrel</pre></div></div>

Responses

Code	Description	Links
200	Success	No links

## Zmiana liczby produktów w Basket

GET

/api/Basket/number

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	
(query)	
numberOfProducts	
integer(\$int32)	
(query)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5107/api/Basket/number?userId=1&numberOfProducts=1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket/number?userId=1&numberOfProducts=1
```

Server response

Code	Details
200	<div>Response headers<div>content-length: 0 date: Wed, 15 May 2024 12:52:21 GMT server: Kestrel</div></div>

Responses

Code	Description
------	-------------

Links

## Utworzenie Order

GET

/api/Basket/Order

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	
(query)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5107/api/Basket/Order?userId=1' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5107/api/Basket/Order?userId=1
```

Server response

Code	Details
200	<div>Response body<div><pre>{   "id": 11,   "userID": 1,   "date": "2024-05-15T14:47:08.373333",   "positions": [     {       "id": 11,       "orderID": 11,       "amount": 1,       "price": 321,       "productID": 4     }   ],   "isPaid": false }</pre></div><div>Download</div></div>

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 15 May 2024 12:47:07 GMT
server: Kestrel
transfer-encoding: chunked
```



## Opłacenie zamówienia:

**PUT** **/api/Basket/pay**

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	
value	
number(\$double)	321
(query)	

ExecuteClear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:5107/api/Basket/pay?userId=1&value=321' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket/pay?userId=1&value=321
```

Server response

Code	Details
200	Response headers
	content-length: 0 date: Wed, 15 May 2024 12:50:13 GMT server: Kestrel

## Usunięcie basket:

**DELETE** **/api/Basket**

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5107/api/Basket?userId=1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket?userId=1
```

Server response

Code	Details
200	Response headers
	content-length: 0 date: Wed, 15 May 2024 12:54:06 GMT server: Kestrel

Responses

Code	Description	Links
200	Success	No links

## Potwierdzenie odnotowania zapłaty w bazie:

	ID	UserID	Date	isPaid
▶	11	1	15.05.2024 14:5...	True
⚙	NULL	NULL	NULL	NULL

Potwierdzenie usunięcia basket:

	ID	ProductID	UserID	Amount	Price	UserGroupID
⚙	NULL	NULL	NULL	NULL	NULL	NULL

## Entity Framework:

### Opis:

W pliku program.cs wstrzyknięto serwisy z obsługą Entity Framework.

```
using BLL.ServiceInterfaces;
using BLL_EF;
using DAL;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
builder.Services.AddDbContext<WebshopContext>();
builder.Services.AddScoped<IProducts, Products>();
builder.Services.AddScoped<IBasket, Basket>();
// Learn more about configuring Swagger/OpenAPI at https://docs.microsoft.com/aspnet/core/fundamentals/swagger
```

Dostęp do danych odbywa się poprzez DbContext.

```
Odsłonięto: 9 | lukasz@90: 56 dni temu | 1 osoba autorów: 2, liczba zmian: 4
public class WebshopContext : DbContext
{
    Odsłonięto: 5 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<BasketPosition> BasketPositions { get; set; }
    Odsłonięto: 3 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<Order> Orders { get; set; }
    Odsłonięto: 2 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<OrderPosition> OrderPositions { get; set; }
    Odsłonięto: 10 | pk, 63 dni temu | 1 autor, liczba zmian: 3
    public DbSet<Product> Products { get; set; }
    Odsłonięto: 0 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<ProductGroup> ProductGroups { get; set; }
    Odsłonięto: 0 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<User> Users { get; set; }
    Odsłonięto: 0 | pk, 70 dni temu | 1 autor, 1 zmiana
    public DbSet<UserGroup> Groups { get; set; }

    Odsłonięto: 0 | lukasz@90: 56 dni temu | 1 osoba autorów: 2, liczba zmian: 2
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer("Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=WebshopSel;Integrated Security=True;Connect Timeout=30;Encrypt=False;Trust Server Certificate=False;Application Int
    }

    Odsłonięto: 0 | pk, 70 dni temu | 1 autor, 1 zmiana
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<UserGroup>().HasMany(g => g.Users).WithOne(u => u.Group).OnDelete(DeleteBehavior.ClientSetNull);
        modelBuilder.Entity<ProductGroup>().HasMany(g => g.Products).WithOne(p => p.Group).OnDelete(DeleteBehavior.SetNull);
        modelBuilder.Entity<ProductGroup>().HasOne(p => p.Parent).WithMany(r => r.Childrens).OnDelete(DeleteBehavior.NoAction);
        modelBuilder.Entity<BasketPosition>().HasOne(b => b.Product).WithMany(p => p.BasketPositions).OnDelete(DeleteBehavior.Restrict);
        modelBuilder.Entity<BasketPosition>().HasOne(b => b.User).WithMany(u => u.BasketPositions).OnDelete(DeleteBehavior.Restrict);
        modelBuilder.Entity<User>().HasMany(u => u.Orders).WithOne(o => o.User).OnDelete(DeleteBehavior.Cascade);
        modelBuilder.Entity<Order>().HasMany(o => o.Positions).WithOne(p => p.Order).OnDelete(DeleteBehavior.Cascade);
    }
}
```

Operacje na danych odbywają się w implementacjach serwisów z wykorzystaniem LINQ.

Odwołania: 2 | lukazam890, 49 dni temu | Liczba autorów: 2, liczba zmian: 3

```
public void addToBasket(BasketPositionRequestDTO basket)
{
    Product? product = _context.Products.FirstOrDefault(p => p.ID == basket.ProductID);
    if (product == null || !product.IsActive)
        return;
    BasketPosition basketTemp = new BasketPosition
    {
        ProductID = basket.ProductID,
        UserID = basket.UserID,
        Amount = basket.Amount,
        Price = basket.Price,
    };
    _context.BasketPositions.Add(basketTemp);
    _context.SaveChanges();
}
```

Odwołania: 2 | lukazam890, 56 dni temu | Liczba autorów: 2, liczba zmian: 2

```
public void changeNumberOfProducts(int id, int numberOfProducts)
{
    if (numberOfProducts <= 0)
        return;
    BasketPosition? basket = _context.BasketPositions.FirstOrDefault(p => p.UserID == id);
    if (basket == null) return;
    basket.Amount = numberOfProducts;
    _context.SaveChanges();
}
```

## Products:

Odczyt listy produktów:

### Products

GET /api/Products

Parameters

No parameters

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5107/api/Products' \
-H 'accept: text/plain'
```

Request URL

```
http://localhost:5107/api/Products
```

Server response

CodeDetails

200

Response body

```
{
  "id": 1,
  "name": "Product1",
  "price": 123,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
},
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
}
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 15 May 2024 13:07:20 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

CodeDescriptionLinks

## Dodanie produktu:

POST

/api/Products

Cancel

Reset

Parameters

No parameters

Request body

application/json

```
{
  "name": "TestEE",
  "price": 4321,
  "image": "10",
  "isActive": true,
  "groupId": 1
}
```

Execute

Responses

Code	Description	Links
200	Success	No links

Response body

```
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupId": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupId": 1,
  "basketPositions": null
},
{
  "id": 5,
  "name": "TestEE",
  "price": 4321,
  "image": "10",
  "isActive": true,
  "groupId": 1,
  "basketPositions": null
}
}
```

Download

## Dezaktywacja produktu:

GET

/api/Products/disable

Cancel

Parameters

Name	Description
id	integer(\$int32) (query)

5

Execute

Clear

Responses

Curl

Response body

```
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
},
{
  "id": 5,
  "name": "TestEF",
  "price": 4321,
  "image": "10",
  "isActive": false,
  "groupID": 1,
  "basketPositions": null
}
}
```

Download

Aktywacja produktu:

GET /api/Products/enable

Parameters

Cancel

Name	Description
id	integer(\$int32) 5 (query)

Execute Clear

Responses

Response body

```
{
  "id": 2,
  "name": "Product2",
  "price": 134,
  "image": "10",
  "isActive": true,
  "groupID": 3,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
},
{
  "id": 5,
  "name": "TestEF",
  "price": 4321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
}
}
```

Download

Usunięcie produktu:

DELETE /api/Products

Parameters

Cancel

Name	Description
id	integer(\$int32) 2 (query)

Execute

Responses

Response body

```
{
  "id": 1,
  "name": "Product1",
  "price": 123,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
},
{
  "id": 4,
  "name": "Test10",
  "price": 321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
},
{
  "id": 5,
  "name": "TestEF",
  "price": 4321,
  "image": "10",
  "isActive": true,
  "groupID": 1,
  "basketPositions": null
}
}
```

Download

## Basket:

Dodanie basket:

POST /api/Basket

Parameters

No parameters

Request body

application/json

```
{
  "productID": 5,
  "userID": 1,
  "amount": 2,
  "price": 4321
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5107/api/Basket' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "productID": 5,
    "userID": 1,
    "amount": 2,
    "price": 4321
  }'
```

Request URL

http://localhost:5107/api/Basket

Server response

## Zmiana liczby produktów:

GET

/api/Basket/number

⌵

Parameters

Cancel

Name	Description
userId integer(\$int32) (query)	<input type="text" value="1"/>
numberOfProducts integer(\$int32) (query)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5107/api/Basket/number?userId=1&numberOfProducts=1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket/number?userId=1&numberOfProducts=1
```

Server response

Code	Details
------	---------

## Utworzenie order:

GET

/api/Basket/Order

⌵

Parameters

Cancel

Name	Description
userId integer(\$int32) (query)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5107/api/Basket/Order?userId=1' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5107/api/Basket/Order?userId=1
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "id": 13,   "userID": 1,   "date": "2024-05-15T15:17:39.4660251+02:00",   "positions": [     {       "id": 13,       "orderID": 13,       "amount": 1,       "price": 4321,       "productID": 5     }   ],   "isPaid": false }</pre> <div><div>Download</div></div>

Response headers

Responses



## Opłaceniu zamówienia:

**PUT** /api/Basket/pay

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	
value	
number(\$double)	4321
(query)	

ExecuteClear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:5107/api/Basket/pay?userId=1&value=4321' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket/pay?userId=1&value=4321
```

Server response

## Usunięcie basket:

**DELETE** /api/Basket

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5107/api/Basket?userId=1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5107/api/Basket?userId=1
```

Server response

Code	Details
200	Response headers <div>content-length: 0 date: Wed, 15 May 2024 13:14:09 GMT server: Kestrel</div>


Responses

Code	Description	Links
200	Success	No links

## Potwierdzenie w bazie opłacenia zamówienia:

	ID	UserID	Date	isPayed
▶	13	1	15.05.2024 15:1...	True
⚙	NULL	NULL	NULL	NULL

Potwierdzenie usunięcia basket:

	ID	ProductID	UserID	Amount	Price	UserGroupID
	NULL	NULL	NULL	NULL	NULL	NULL

## Wnioski:

Entity Framework jest bardzo użytecznym narzędziem do obsługi baz danych w .NET. Entity Framework to ORM, stanowi on pomost pomiędzy obiektową reprezentacją danych w programie a relacyjną w bazie danych. Korzystanie ze Entity Framework niesie za sobą wiele korzyści w porównaniu do używania standardowych poleceń SQL

Oprogramowanie aplikacji za pomocą Entity Framework jest dużo szybsze niż korzystanie z SQL. Poprzez korzystanie z obiektowej logiki, zaplanowanie organizacji kodu jest dużo prostsze. Korzystanie z metod wbudowanych w LINQ wydaje się być bardziej intuicyjne niż korzystanie ze zbioru funkcji SQL. Dużym ułatwieniem i skróceniem czasu pracy jest podejście Code First. Wystarczy zadeklarować klasy reprezentujące dane, a Entity Framework poprzez mechanizm migracji sam utworzy odpowiednie tabele i relacje pomiędzy nimi.

Entity Framework gwarantuje wyższy poziom bezpieczeństwa niż korzystanie z poleceń SQL. Procedury SQL są podatne na ataki SQL Injection, kiedy wraz z danymi użytkownik wkleja złośliwe polecenia. Entity Framework sam w sobie jest odporny na takie zagrożenia.

Kolejną zaletą jest uniwersalność, ponieważ Entity Framework może współpracować z wieloma różnymi technologiami baz danych. Są to między innymi MSSQL, MySQL, SQLite. Przejście pomiędzy różnymi technologiami nie wymaga wprowadzania w programie wielkich zmian.