



Создание REST API



Создание приложения

api

×

▼

Очистить все

Все языки

▼

Все платформы


▼

Библиотека

▼

Точных соответствий не найдено.

Другие результаты на основе вашего поиска

 Веб-API ASP.NET Core (Майкрософт)
Шаблон проекта для создания приложения ASP.NET Core с контроллером для службы HTTP RESTful. Этот шаблон можно использовать для представлений MVC и контроллеров ASP.NET Core.

C#

Linux

macOS

Windows

Облако

Службы

WebAPI

Настроить новый проект

Веб-API ASP.NET Core (Майкрософт) C# Linux macOS Windows Облако

Имя проекта

SaloonApi

Расположение

C:\Users\311-8(student)\source\repos

...

Имя решения ⓘ

SaloonApi

☒ Поместить решение и проект в одном каталоге

Настройки приложения

Дополнительные сведения

Веб-API ASP.NET Core (Майкрософт)

C#

Linux

macOS

Windows

Облако

Служба

Платформа ⓘ

.NET 6.0 (долгосрочная поддержка)

Тип проверки подлинности ⓘ

Нет

☐ Настроить для HTTPS ⓘ

☐ Включить Docker ⓘ

Операционная система Docker ⓘ

Linux

☒ Использовать контроллеры (снимите флажок, чтобы использовать минимальные API) ⓘ

☒ Включить поддержку OpenAPI ⓘ

☐ Не использовать операторы верхнего уровня ⓘ

NuGet: SaloonApi

Обзор

Установлено

Обновления 1

Поиск (Ctrl+L)

☐ Включить предварительные



Swashbuckle.AspNetCore автор: Swashbuckle.AspNetCore
Swagger tools for documenting APIs built on ASP.NET Core

6.2.3

6.5.0

Вывод документации при старте приложения

The image shows a Visual Studio interface with three main panels:

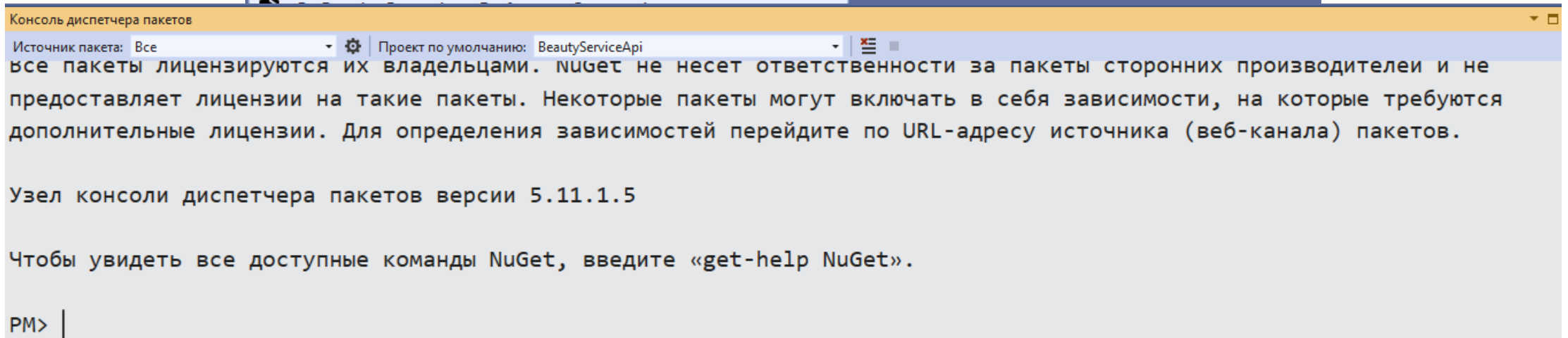
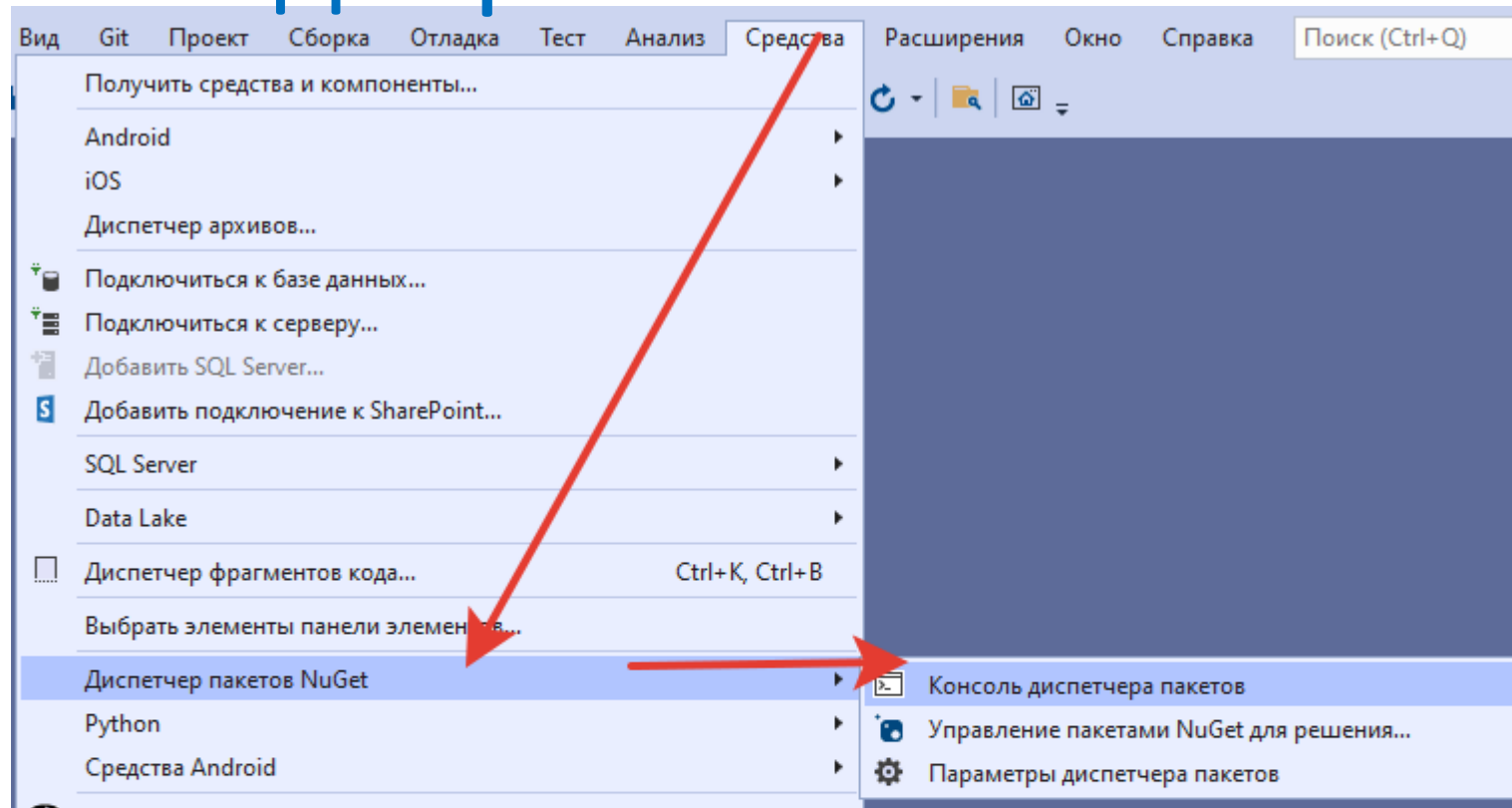
- launchSettings.json** (Left Panel):

```
1 {
2   "$schema": "https://json.schemastore.org/launchsettings.json",
3   "iisSettings": {
4     "windowsAuthentication": false,
5     "anonymousAuthentication": true,
6     "iisExpress": {
7       "applicationUrl": "http://localhost:30710",
8       "sslPort": 0
9     }
10  },
11  "profiles": {
12    "SaloonApi": {
13      "commandName": "Project",
14      "dotnetRunMessages": true,
15      "launchBrowser": true,
16      "launchUrl": "swagger",
17      "applicationUrl": "http://127.0.0.1:5098",
18      "environmentVariables": {
19        "ASPNETCORE_ENVIRONMENT": "Development"
20      }
21    },
22    "IIS Express": {
23      "commandName": "IISExpress",
24      "launchBrowser": true,
25      "launchUrl": "swagger",
26      "environmentVariables": {
27        "ASPNETCORE_ENVIRONMENT": "Development"
28      }
29    }
30  }
31 }
32 }
```
- Обозреватель решений** (Center Panel):
 - Решение "SaloonApi" (1 проекта 1)
 - Внешние источники
 - SaloonApi
 - Connected Services
 - Properties
 - launchSettings.json
 - Зависимости
 - Controllers
 - appsettings.json
 - C# Program.cs
 - C# WeatherForecast.cs
- Program.cs** (Right Panel):

```
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4
5 builder.Services.AddControllers();
6 // Learn more about configuring Swagger/OpenAPI at https://
7   aka.ms/aspnetcore/swashbuckle
8 builder.Services.AddEndpointsApiExplorer();
9 builder.Services.AddSwaggerGen();
10
11 var app = builder.Build();
12
13 // Configure the HTTP request pipeline.
14 if (app.Environment.IsDevelopment())
15 {
16     app.UseSwagger();
17     app.UseSwaggerUI();
18 }
19
20 app.UseAuthorization();
21
22 app.MapControllers();
23
24 app.Run();
```

Red arrows indicate the flow of configuration: one arrow points from the `launchSettings.json` file in the Solution Explorer to the `Program.cs` file, and another arrow points from the `launchSettings.json` file to the `builder.Services.AddSwaggerGen();` line in `Program.cs`.

Запуск команд через Power Shell



Пакет для создания аннотаций Swagger'а

Install-Package Swashbuckle.AspNetCore.Annotations -Version 6.2.3

```
Program.cs*  SaloonApi
1  var builder = WebApplication.CreateBuilder(args);
2
3  // Add services to the container.
4
5  builder.Services.AddControllers();
6  // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
7  builder.Services.AddEndpointsApiExplorer();
8  builder.Services.AddSwaggerGen(c => c.EnableAnnotations());
9
10 var app = builder.Build();
11
12 // Configure the HTTP request pipeline.
13 if (app.Environment.IsDevelopment())
14 {
15     app.UseSwagger();
16     app.UseSwaggerUI();
17 }
18
19 app.UseAuthorization();
20
21 app.MapControllers();
22
23 app.Run();
24
```

Registration

POST	/api/account/register	Регистрирует пользователя
POST	/api/account/confirm	Подтверждает Email

Authorization

POST	/api/account/login	Авторизовывает пользователя
------	--------------------	-----------------------------

Пакет для создания аннотаций Swagger'a

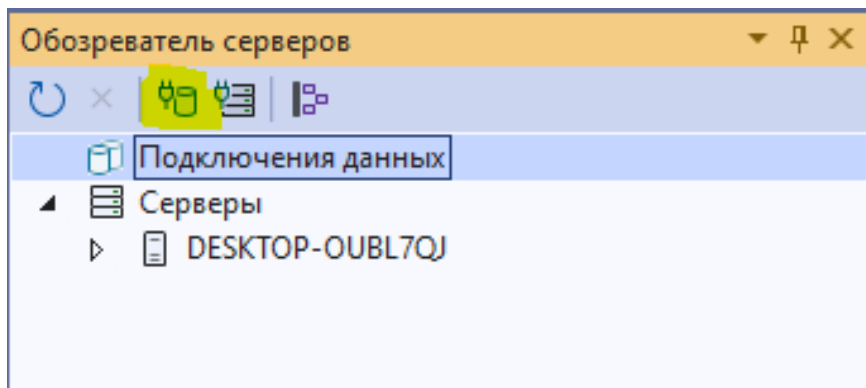
```
[SwaggerOperation(  
    Summary = "Регистрирует пользователя",  
    Description = "Полное описание",  
    Tags = new[] { "Registration" }  
)]
```

```
[SwaggerResponse(StatusCode.Status200  
    OK, "Пользователь зарегистрирован",  
    Type = typeof(RegisterModel))]  
[SwaggerResponse(StatusCode.Status400B  
    adRequest, "Необходимо указать логин")]
```

Registration		
POST	/api/account/register	Регистрирует пользователя
POST	/api/account/confirm	Подтверждает Email
Authorization		
POST	/api/account/login	Авторизовывает пользователя

- Summary - кратко о том, что делает метод
- Description - полное описание что делает метод.
- Tags - указываются теги для логической группировки операций

Подключение БД с помощью обозревателя серверов (Ctrl+Alt+S)



Добавить подключение

Введите данные для подключения к выбранному источнику данных или нажмите кнопку "Изменить", чтобы выбрать другой источник данных и (или) поставщик.

Источник данных:
Microsoft SQL Server (SqlClient) Изменить...

Имя сервера:
(localdb)\MSSQLLocalDB Обновить

Вход на сервер

Проверка подлинности: Проверка подлинности Windows

Имя пользователя:

Пароль:

☐ Сохранить пароль

Подключение к базе данных

☒ Выберите или введите имя базы данных:
BeautySaloon

☐ Прикрепить файл базы данных:
 Обзор...

Логическое имя:

Дополнительно...

Проверить подключение OK Отмена

Установка пакетов для реализации миграции БД

```
Install-Package Microsoft.EntityFrameworkCore.SqlServer -Version 7.0.0
```

```
Install-Package Microsoft.EntityFrameworkCore.Tools -Version 7.0.0
```

<https://learn.microsoft.com/en-us/ef/core/cli/powershell>

Миграция

Scaffold-DbContext "Server=ПараметрыПодключения"

Microsoft.EntityFrameworkCore.SqlServer -OutputDir ИмяКаталога

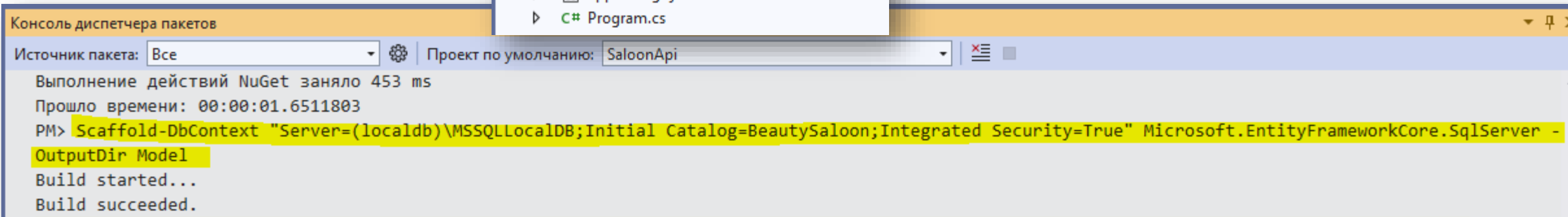
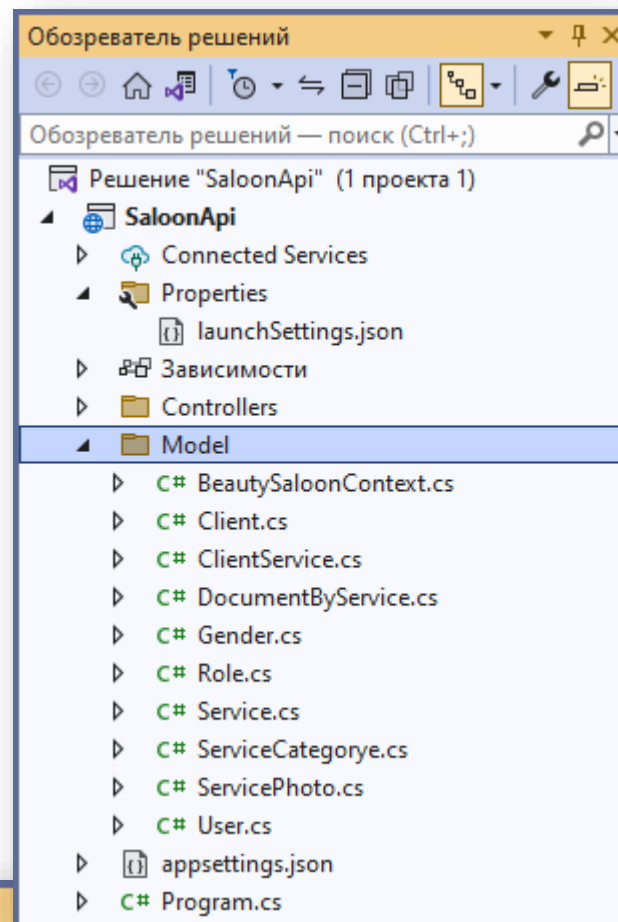
Пример команды:

Scaffold-DbContext "Server=(localdb)\MSSQLLocalDB;Initial
Catalog=BeautySaloon;Integrated Security=True"

Microsoft.EntityFrameworkCore.SqlServer -OutputDir Model

<https://learn.microsoft.com/en-us/ef/core/cli/powershell>

Миграция



BeautySaloonContext.cs

SaloonApi

SaloonApi.Model.BeautySaloonContext

BeautySaloonContext()

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

Ссылка: 0

public virtual DbSet<ServiceCategory> ServiceCategories { get; set; }

Ссылка: 0

public virtual DbSet<ServicePhoto> ServicePhotos { get; set; }

Ссылка: 0

public virtual DbSet<User> Users { get; set; }

Ссылка: 0

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see <https://go.microsoft.com/fwlink/?linkid=2131148>. For more guidance on storing connection strings, see <http://go.microsoft.com/fwlink/?LinkId=723263>.

=> optionsBuilder.UseSqlServer("Server=(localdb)\\MSSQLLocalDB;Initial Catalog=BeautySaloon;Integrated Security=True");

Ссылка: 0

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Client>(entity =>

{

entity.ToTable("Client");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.Birthday).HasColumnType("date");

entity.Property(e => e.Email).HasMaxLength(255);

entity.Property(e => e.FirstName).HasMaxLength(50);

Файл appsetting.json -изменение параметров приложения

```
"ConnectionStrings": {  
    "ИмяБД": "Server=ПараметрыПодключения"  
}
```



```
appsettings.json* X  
Схема: https://json.schemastore.org/appsettings.json  
1 {  
2   "Logging": {  
3     "LogLevel": {  
4       "Default": "Information",  
5       "Microsoft.AspNetCore": "Warning"  
6     }  
7   },  
8   "AllowedHosts": "*",  
9   "ConnectionStrings": {  
10    "BeautySaloon": "Server=(localdb)\\MSSQLLocalDB;Initial Catalog=BeautySaloon;Integrated  
11    Security=True"  
12  }  
13 }
```

Файл Startup.cs –добавление контекста БД

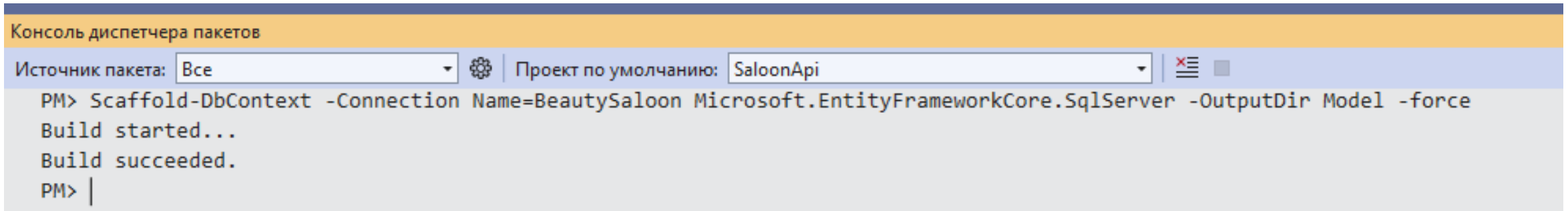
```
builder.Services.AddDbContext<ИмяКонтекстаБД>(options =>  
options.UseSqlServer(builder.Configuration.GetConnectionString(  
"ИмяБД")));
```



```
Program.cs*  X  
SaloonApi  
1  using Microsoft.EntityFrameworkCore;  
2  using Microsoft.Extensions.Configuration;  
3  using SaloonApi.Model;  
4  
5  var builder = WebApplication.CreateBuilder(args);  
6  
7  // Add services to the container.  
8  
9  builder.Services.AddControllers();  
10 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle  
11 builder.Services.AddEndpointsApiExplorer();  
12 builder.Services.AddSwaggerGen(c=>c.EnableAnnotations());  
13 builder.Services.AddDbContext<BeautySaloonContext>(options => options.UseSqlServer  
    (builder.Configuration.GetConnectionString("BeautySaloon")));  
14  
15 var app = builder.Build();  
16
```

Фиксация изменений

Scaffold-DbContext -Connection **Name=ИмяБД**
Microsoft.EntityFrameworkCore.SqlServer -OutputDir
Model -force



```
Консоль диспетчера пакетов
Источник пакета: Все | Проект по умолчанию: SaloonApi
PM> Scaffold-DbContext -Connection Name=BeautySaloon Microsoft.EntityFrameworkCore.SqlServer -OutputDir Model -force
Build started...
Build succeeded.
PM> |
```

BeautySaloonContext.cs

SaloonApiSaloonApi.Model.BeautySaloonContextBeautySaloonContext()

23

Ссылка: 0

24

public virtual DbSet<Gender> Genders { get; set; }

25

Ссылка: 0

26

public virtual DbSet<Role> Roles { get; set; }

27

Ссылка: 0

28

public virtual DbSet<Service> Services { get; set; }

29

Ссылка: 0

30

public virtual DbSet<ServiceCategory> ServiceCategories { get; set; }

31

Ссылка: 0

32

public virtual DbSet<ServicePhoto> ServicePhotos { get; set; }

33

Ссылка: 0

34

public virtual DbSet<User> Users { get; set; }

35

Ссылка: 0

36

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

37

=> optionsBuilder.UseSqlServer("Name=BeautySaloon");

38

Ссылка: 0

39

protected override void OnModelCreating(ModelBuilder modelBuilder)

40

{

41

modelBuilder.Entity<Client>(entity =>

42

{

43

entity.ToTable("Client");

44

Создание контроллера

Обозреватель решений

Обозреватель решений — поиск (Ctrl+;)

Решение "SaloonApi" (1 проекта 1)

SaloonApi

Connected Services

Properties

launchSettings.json

Зависимости

Controllers

Model

C# BeautySaloonContext.cs

C# Client.cs

C# ClientService.cs

C# DocumentByS

C# Gender.cs

C# Role.cs

C# Service.cs

C# ServiceCatego

C# ServicePhoto.

C# User.cs

Models

appsettings.json

C# Program.cs

C# WeatherForecast.

Контроллер...

Компонент Razor...

Создать элемент...

Ctrl+Shift+A

Существующий элемент...

Shift+Alt+A

Создать шаблонный элемент...

Создать папку

Из Cookiecutter...

Поддержка оркестратора контейнеров...

Поддержка Docker...

Телеметрия Application Insights...

Клиентская библиотека...

Модель машинного обучения...

Новый проект веб-задания Azure

Существующий проект как веб-задание Azure

Класс...

New EditorConfig

Новая конфигурация редактора (IntelliCode)

Просмотреть в браузере (Opera GX Internet Browser)

Ctrl+Shift+W

Просмотреть с помощью...

Добавить

Открыть элемент как корень обозревателя

Новое представление: Обозреватель решений

Исключить из проекта

Вырезать

Ctrl+X

Копировать

Ctrl+C

Удалить

Del

Переименовать

F2

Копировать полный путь

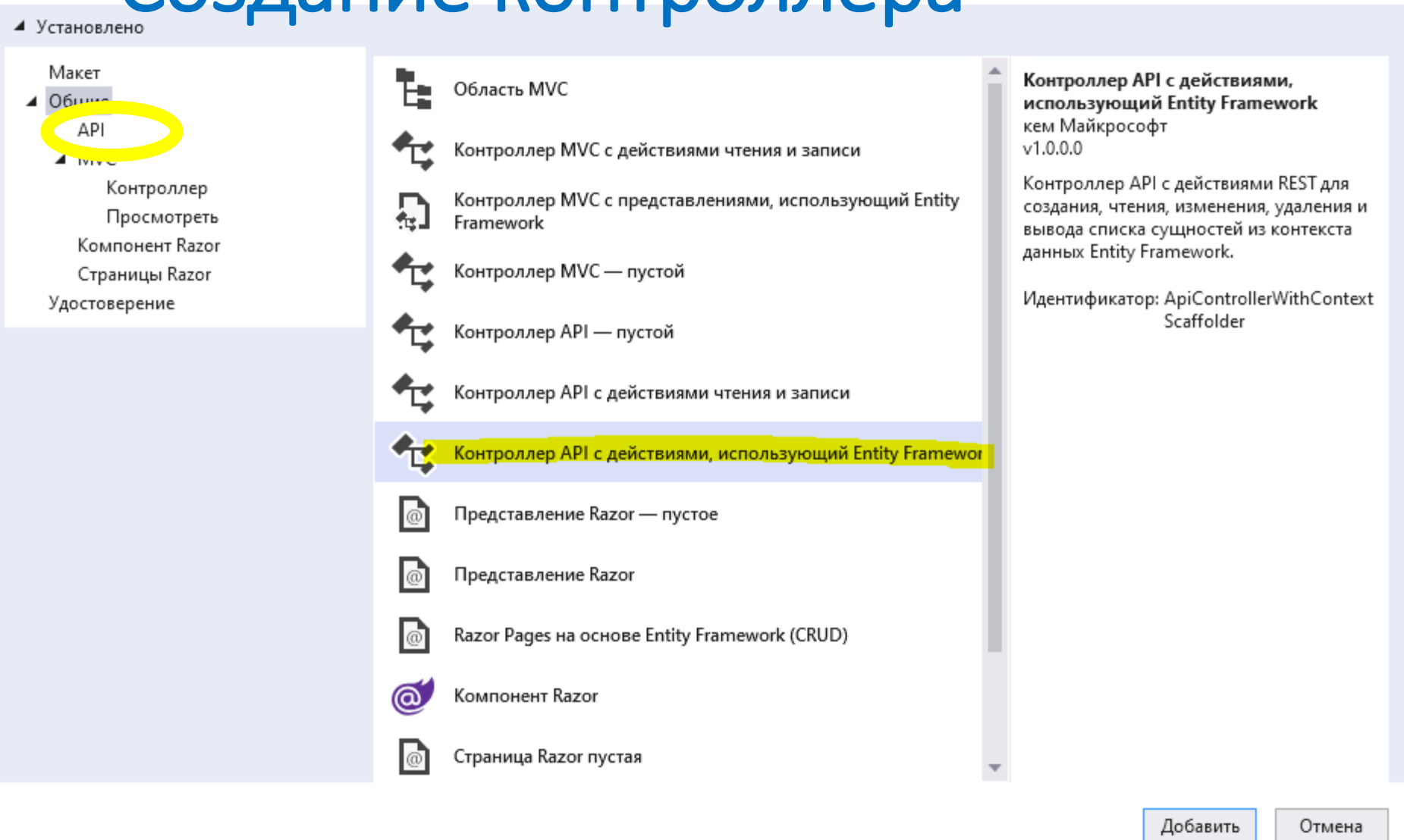
Открыть папку в проводнике

Открыть в терминале

Свойства

Alt+ВВОД

Создание контроллера



Добавление возможности просмотра подчиненных таблиц

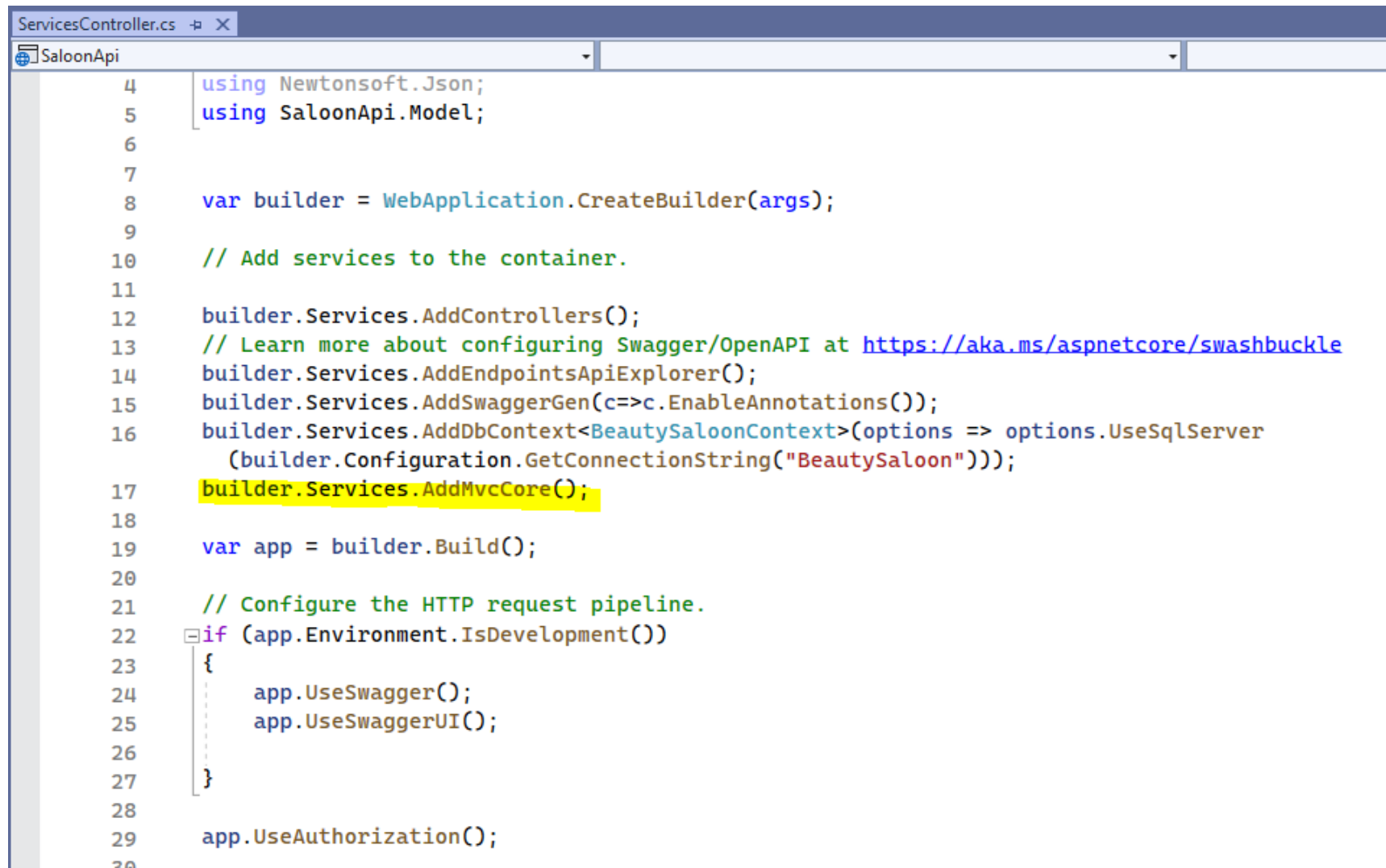
Предварительно необходимо установить библиотеку !!!!

Microsoft.AspNetCore.Mvc.NewtonsoftJson

```
Install-Package Microsoft.AspNetCore.Mvc.NewtonsoftJson -Version 6.0.14
```

Добавление возможности просмотра подчиненных таблиц

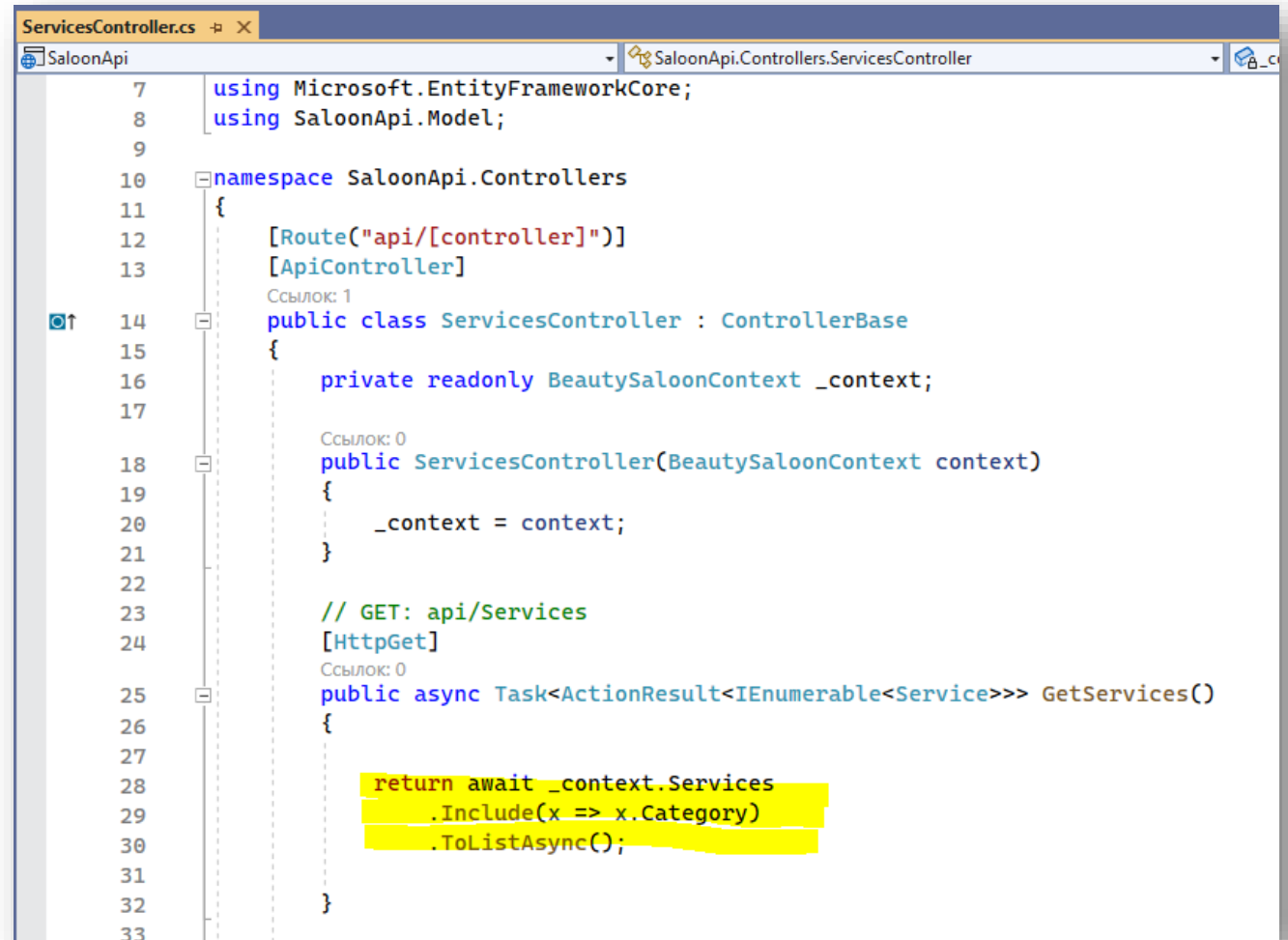
`builder.Services.AddMvcCore();`



```
ServicesController.cs  X
SaloonApi
4  using Newtonsoft.Json;
5  using SaloonApi.Model;
6
7
8  var builder = WebApplication.CreateBuilder(args);
9
10 // Add services to the container.
11
12 builder.Services.AddControllers();
13 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
14 builder.Services.AddEndpointsApiExplorer();
15 builder.Services.AddSwaggerGen(c => c.EnableAnnotations());
16 builder.Services.AddDbContext<BeautySaloonContext>(options => options.UseSqlServer
    (builder.Configuration.GetConnectionString("BeautySaloon")));
17 builder.Services.AddMvcCore();
18
19 var app = builder.Build();
20
21 // Configure the HTTP request pipeline.
22 if (app.Environment.IsDevelopment())
23 {
24     app.UseSwagger();
25     app.UseSwaggerUI();
26 }
27
28
29 app.UseAuthorization();
30
```

Вывод подчиненных таблиц

```
return await _context.ИмяТаблицы
    .Include(x=>x.ИмяПодчиненнойТаблицы)
    .ToListAsync();
```



```
ServicesController.cs
SaloonApi
SaloonApi.Controllers.ServicesController

7  using Microsoft.EntityFrameworkCore;
8  using SaloonApi.Model;
9
10 namespace SaloonApi.Controllers
11 {
12     [Route("api/[controller]")]
13     [ApiController]
14     public class ServicesController : ControllerBase
15     {
16         private readonly BeautySaloonContext _context;
17
18         public ServicesController(BeautySaloonContext context)
19         {
20             _context = context;
21         }
22
23         // GET: api/Services
24         [HttpGet]
25         public async Task<ActionResult<IEnumerable<Service>>> GetServices()
26         {
27
28             return await _context.Services
29                 .Include(x => x.Category)
30                 .ToListAsync();
31         }
32     }
33 }
```

[ApiController]

ссылка: 1

```
public class ServiceCategoriesController : ControllerBase
```

```
{  
    private readonly BeautySaloonContext _context;
```

Ссылка: 0

```
public ServiceCategoriesController(BeautySaloonContext context)
```

```
{  
    _context = context;  
}
```

```
// GET: api/ServiceCategories  
[HttpGet]
```

Ссылка: 0

```
public async Task<ActionResult<IEnumerable<ServiceCategories>>> GetServiceCategories()
```

```
{  
    return await _context.ServiceCategories.ToListAsync();  
}
```

```
// GET: api/ServiceCategories/5
```

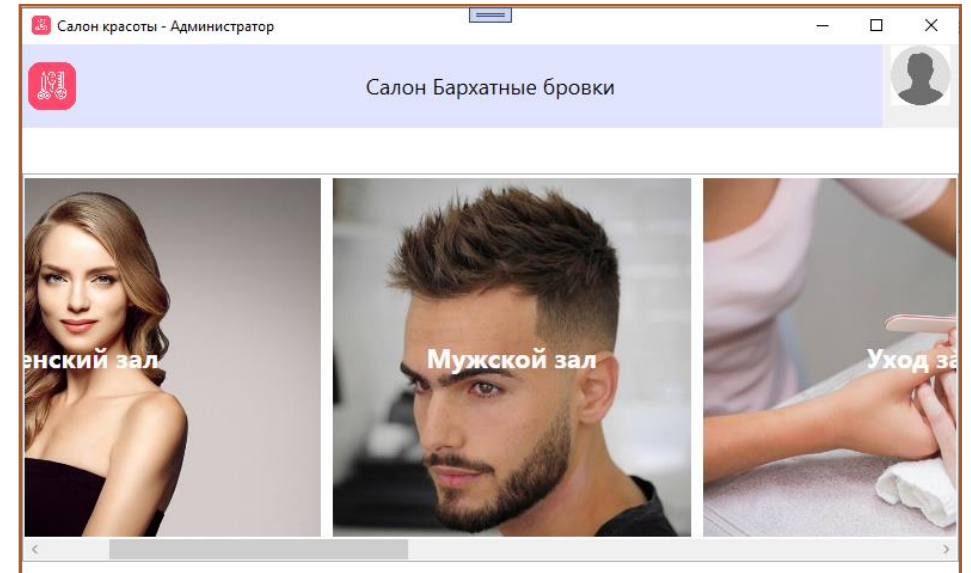
```
[HttpGet("{CategoryId}")]
```

Ссылка: 0

```
public async Task<ActionResult<ServiceCategories>> GetServiceCategories(int id)
```

```
{  
    var serviceCategories = await _context.ServiceCategories.FindAsync(id);  
  
    if (serviceCategories == null)  
    {  
        return NotFound();  
    }  
}
```

Вывод категорий



[ApiController]

ссылка: 1

```
public class ServicesController : ControllerBase
```

```
{
```

```
    private readonly BeautySaloonContext _context;
```

Ссылок: 0

```
    public ServicesController(BeautySaloonContext context)
```

```
    {
```

```
        _context = context;
```

```
    }
```

```
    // GET: api/Services
```

```
    [HttpGet]
```

Ссылок: 0

```
    public async Task<ActionResult<IEnumerable<Services>>> GetServices()
```

```
    {
```

```
        return await _context.Services.ToListAsync();
```

```
    }
```

```
    // GET: api/Services/5
```

```
    [HttpGet("{categoryId}")]
```

Ссылок: 0

```
    public async Task<ActionResult<IEnumerable<Services>>> GetServices(int categoryId)
```

```
    {
```

```
        var services = await _context.Services.Where(x=>x.CategoryId==categoryId).ToListAsync();
```

```
        if (services == null)
```

```
        {
```

```
            return NotFound();
```

```
        }
```

```
        return services;
```

```
    }
```

Вывод сервисов по заданной категории

Салон красоты - Администратор



Прайс услуг

Наращивание ресниц	1358.5000
Моделирование ресниц	1251.0000
Ламинирование ресниц	1305.0000
Русский спа массаж	2032.0000
Химические завивки	872.0000
Альгинатные маски	2370.0000
Криолифтинг	1090.0000

[ApiController]

ссылка: 1

```
public class UsersController : ControllerBase
```

```
{
```

```
    private readonly BeautySaloonContext _context;
```

Ссылка: 0

```
    public UsersController(BeautySaloonContext context)
```

```
    {
```

```
        _context = context;
```

```
    }
```

```
    // GET: api/Users
```

```
    [HttpGet]
```

Ссылка: 0

```
    public async Task<ActionResult<IEnumerable<Users>>> GetUsers()
```

```
    {
```

```
        return await _context.Users.ToListAsync();
```

```
    }
```

```
    // GET: api/Users/dima18/pass|
```

```
    [HttpGet("{login}/{password}")]
```

Ссылка: 0

```
    public async Task<ActionResult<Users>> GetUsers(string login, string password)
```

```
    {
```

```
        var users = await _context.Users.FirstOrDefaultAsync(x => x.UserLogin ==login &&  
            x.UserPassword==password);
```

```
        if (users == null)
```

```
        {
```

```
            return NotFound();
```

```
        }
```

```
        return users;
```

```
    }
```

Авторизация


```
// POST: api/Users
```

```
[HttpPost]
```

Ссылка: 0

```
public async Task<ActionResult<Users>> PostUsers(Users users)
```

```
{
    _context.Users.Add(users);
    try
    {
        await _context.SaveChangesAsync();
        return users;
    }
    catch (DbUpdateException)
    {
        if (UsersExists(users.UserLogin))
        {
            return BadRequest();
        }
    }
}
```

Регистрация

Салон красоты

RegistrationPage

Регистрация

Имя:

Фамилия:

Отчество:

Логин:

Пароль:

Повторите пароль:

Введите символы с картинки 1q2i1h
q2i1h

Статус: Не авторизован

POST

http://127.0.0.1:55409/api/users

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   ... "idrole": 2,
3   ... "userName": null,
4   ... "userLastName": null,
5   ... "userOtherName": null,
6   ... "userLogin": "demihprinx",
7   ... "userPassword": "cfhnjdfryu",
8   ... "idroleNavigation": null
9 }
```