

**Лабораторная работа 6**  
**по конструированию Интернет-приложений**  
**Разграничение прав**

30-04-2019

Цель лабораторной работы: научиться добавлять в веб-API возможности разграничения прав пользователей.

**Задания:**

- |   |   |
|---|---|
| 1. Добавить создание ролей и пользователей          | 1 |
| 2. Внести изменения для миграции данных             | 2 |
| 3. Выполнить миграцию данных                        | 3 |
| 4. Добавить роли для новых пользователей            | 4 |
| 5. Добавить разграничение прав по ролям             | 5 |
| 6. Добавить возврат 401 кода в авторизации          | 5 |
| 7. Добавить клиентскую обработку разграничения прав | 6 |

Общие сведения об Identity в ASP.NET Core

<https://docs.microsoft.com/ru-ru/aspnet/core/security/authentication/identity>

<https://metanit.com/sharp/aspnet5/16.1.php>

## 1. Добавить создание ролей и пользователей

Добавить создание пользователя и администратора при старте приложения. Добавить в Startup метод создающий две роли: пользователя и администратора, и пользователей для этих ролей.

```
using System.Threading.Tasks;
using System;
using Microsoft.Extensions.Logging;
...
private async Task CreateUserRoles(IServiceProvider serviceProvider)
{
    var roleManager =
serviceProvider.GetRequiredService<RoleManager<IdentityRole>>();
    var userManager =
serviceProvider.GetRequiredService<UserManager<User>>();

    // Создание ролей администратора и пользователя
    if (await roleManager.FindByNameAsync("admin") == null)
    {
        await roleManager.CreateAsync(new
IdentityRole("admin"));
    }
    if (await roleManager.FindByNameAsync("user") == null)
    {
        await roleManager.CreateAsync(new IdentityRole("user"));
    }

    // Создание Администратора
    string adminEmail = "admin@mail.com";
    string adminPassword = "Aa123456!";
    if (await userManager.FindByNameAsync(adminEmail) == null)
    {
        User admin = new User { Email = adminEmail, UserName =
adminEmail };
        IdentityResult result = await
userManager.CreateAsync(admin, adminPassword);
        if (result.Succeeded)
        {
            await userManager.AddToRoleAsync(admin, "admin");
        }
    }

    // Создание Пользователя
    string userEmail = "user@mail.com";
    string userPassword = "Aa123456!";
    if (await userManager.FindByNameAsync(userEmail) == null)
    {
        User user = new User { Email = userEmail, UserName =
userEmail };
        IdentityResult result = await
userManager.CreateAsync(user, userPassword);
        if (result.Succeeded)
        {

```

```

        await userManager.AddToRoleAsync(user, "user");
    }
}

public void Configure(IApplicationBuilder app,
    IHostingEnvironment env, ILoggerFactory loggerFactory, IServiceProvider
    services)
{
    CreateUserRoles(services).Wait();
}
}

```

## 2. Внести изменения для миграции данных

Добавить класс DesignTimeDbContextFactory как реализацию  
IDesignTimeDbContextFactory

```

using System.IO;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Design;
using Microsoft.Extensions.Configuration;

namespace ASPNetCoreApp.Models
{
    public class DesignTimeDbContextFactory :
        IDesignTimeDbContextFactory<BloggngContext>
    {
        public BloggngContext CreateDbContext(string[] args)
        {
            IConfigurationRoot configuration = new
            ConfigurationBuilder()
                .SetBasePath(Directory.GetCurrentDirectory())
                .AddJsonFile("appsettings.json")
                .Build();

            var builder = new
            DbContextOptionsBuilder<BloggngContext>();

            var connectionString =
            configuration.GetConnectionString("DefaultConnection");

            builder.UseSqlServer(connectionString);

            return new BloggngContext(builder.Options);
        }
    }
}

```

### 3. Выполнить миграцию данных

В консоли диспетчера пакетов

Выполнить создание миграции, если ее еще нет

**Add-Migration Initial**

Выполнить обновление/создание БД

**Update-Database**

Важно, чтобы сохранились данные в БД необходимо внести изменения в миграцию. Другой путь, создать БД заново, удалив текущую или переименовав строку подключения.

Результаты:

dbo.AspNetUsers [Данные]					
Максимальное количество строк: 1000					
	Id	UserName	Normalize...	Email	Norma
▶	72d326e45823	admin@mai...	ADMIN@M...	admin@mai...	ADMIN
	e40db966-6...	user@mail.c...	USER@MAI...	user@mail.c...	USER@
*	NULL	NULL	NULL	NULL	NULL

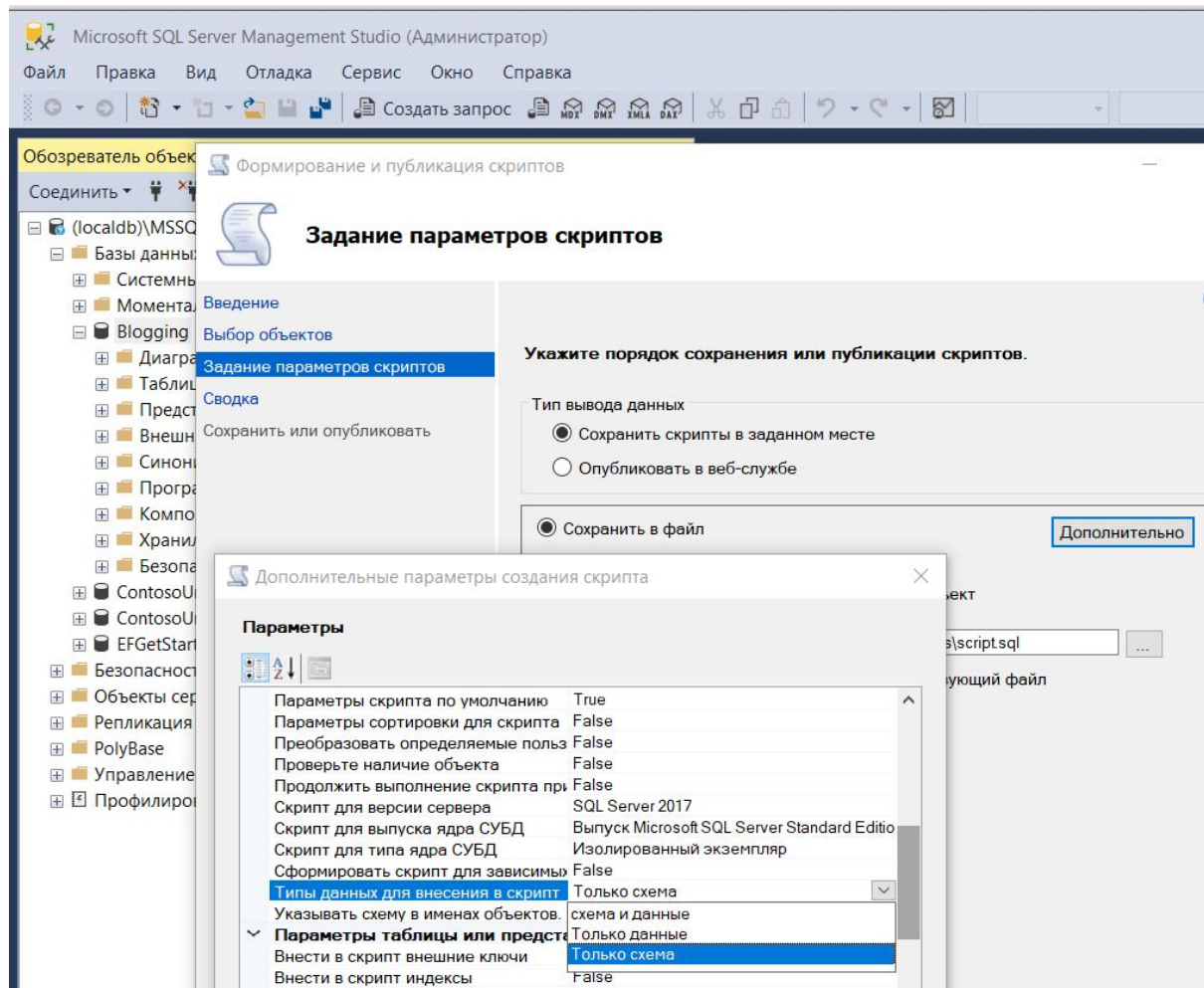
dbo.AspNetRoles [Данные]				
Максимальное количество строк: 1000				
	Id	Name	Normalize...	Concurrenc...
▶	425f3b031c08	admin	ADMIN	404c2168-f...
	5e4088ee-1...	user	USER	2e7192e6-a...
*	NULL	NULL	NULL	NULL

dbo.AspNetUserRoles [Данные]		
Максимальное количество строк: 1000		
	UserId	RoleId
	63d19886-dee1-4e6d-bbe5-72d326e45823	1343a14a-31db-4f3c-89fc-425f3b031c08
	e40db966-6abd-45b7-9b4b-2261b426775a	5e4088ee-1a54-47e3-baff-2c07a0b463da
▶*	NULL	NULL

Существующие данные БД можно перенести путем формирования скриптов.

Учебник. Создание скриптов для объектов в среде SQL Server Management StudioTutorial: Script objects in SQL Server Management Studio - SQL Server | Microsoft Docs

<https://docs.microsoft.com/ru-ru/sql/ssms/tutorials/scripting-ssms?view=sql-server-2017#script-a-database-by-using-the-generate-scripts-option>



#### 4. Добавить роли для новых пользователей

При регистрации нового пользователя устанавливать ему роль user по умолчанию  
Добавить в AccountController:

```
[HttpPost]
[Route("api/Account/Register")]
public async Task<IActionResult> Register([FromBody]
RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        User user = new User { Email = model.Email, UserName =
model.Email };
        // Добавление нового пользователя
        var result = await _userManager.CreateAsync(user,
model.Password);
        if (result.Succeeded)
        {
            await _userManager.AddToRoleAsync(user, "user");
        }
    }
}
```

#### 5. Добавить разграничение прав по ролям

Для метода удаления блога контроллера добавить атрибут [Authorize(Roles = "admin")]

```
using Microsoft.AspNetCore.Authorization;

...
[Authorize(Roles = "admin")]
[HttpDelete("{id}")]
public async Task<IActionResult> Delete([FromRoute] int id)
{
    ...
}
```

#### 6. Добавить возврат 401 кода в авторизации

В startup.cs добавить следующий код

```
public void ConfigureServices(IServiceCollection services)
{
    ...
    services.ConfigureApplicationCookie(options =>
    {
        options.Cookie.Name = "SimpleWebApp";
        options.LoginPath = "/";
        options.AccessDeniedPath = "/";
        options.LogoutPath = "/";
        options.Events.OnRedirectToLogin = context =>
        {
            context.Response.StatusCode = 401;
            return Task.CompletedTask;
        };
        options.Events.OnRedirectToAccessDenied = context =>
        {
            context.Response.StatusCode = 401;
        };
    });
}
```

```

        return Task.CompletedTask;
    };
});

```



Название	SimpleWebApp
Контент	ЭrgBX1aCqKPXxrtxt2sdpquDrJLjF2kEV_CimCsWE2Xk1Z045N74Nf3h03BMKt
Домен	localhost
Путь	/

## 7. Добавить клиентскую обработку разграничения прав

добавить стиль

```

.msgClass {
    color: red;
}

```

добавить блок вывода сообщения в index.html

```

<h3>Список блогов</h3>
<div id="actionMsg" class="msgClass"></div>

```

добавить обработку ответов в script.js

```

function createBlog() {
    let urlText = "";
    urlText = document.querySelector("#createDiv").value;
    var request = new XMLHttpRequest();
    request.open("POST", uri);
    request.onload = function () {
        // Обработка кода ответа
        var msg = "";
        if (request.status === 401) {
            msg = "У вас не хватает прав для создания";
        } else if (request.status === 201) {
            msg = "Запись добавлена";
            getBlogs();
        } else {
            msg = "Неизвестная ошибка";
        }
        document.querySelector("#actionMsg").innerHTML = msg;
    };
};

```

## Вход для зарегистрированных пользователей

Вы вошли как: admin@mail.com

Email:

Пароль:

Войти

Выйти

[Регистрация нового пользователя](#)

## Блоги

Количество блогов: 2

### Создание нового блога

URL:

Добавить

### Список блогов

Запись удалена

1 : <http://blogs.msdn.com/dotnet>

Изменить

Удалить

Today, we are releasing the .NET Core May 2018 Update. This update includes the .NET Core and .NET native version 2.0.