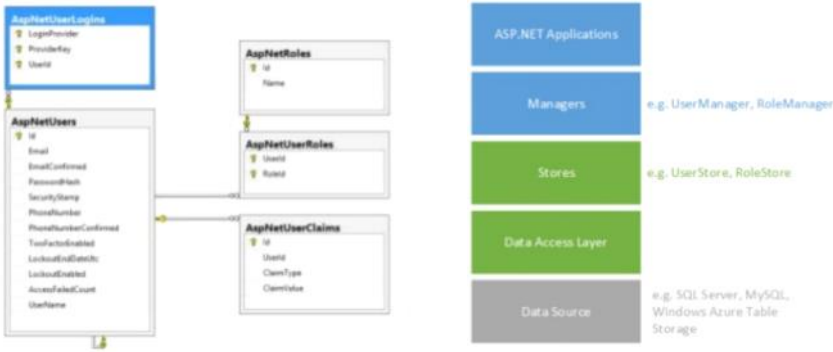


12 - ASP.Net Identity (Farklı Context)

13 Mayıs 2019 Pazartesi 08:22



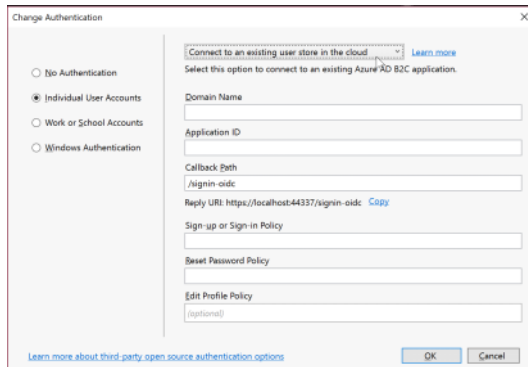
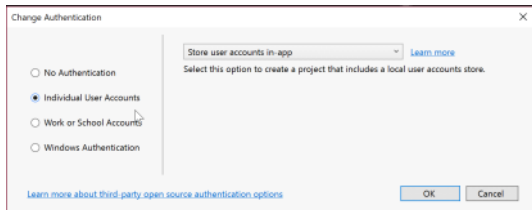
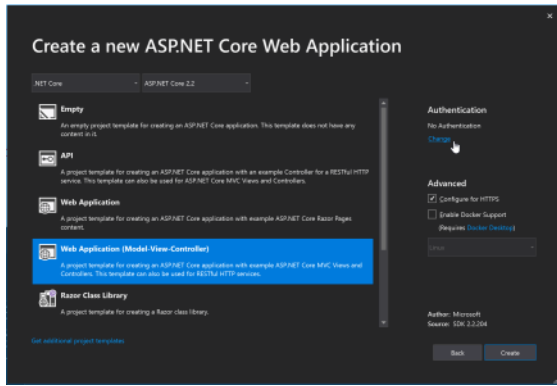
9 - Eticaret...

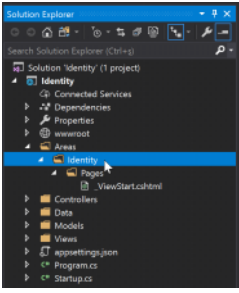


User Manager

SignInManager

Role Manager





```

19 public class Startup
20 {
21     public Startup(IConfiguration configuration)
22     {
23         Configuration = configuration;
24     }
25
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     public void ConfigureServices(IServiceCollection services)
30     {
31         services.Configure(CookiePolicyOptions)(options =>
32         {
33             // This lambda determines whether user consent for non-essential cookies is needed for a given request.
34             options.CheckConsentNeeded = context => true;
35             options.MinimumSameSitePolicy = SameSiteMode.None;
36         });
37
38         services.AddDbContext(options =>
39             options.UseSqlServer(
40                 Configuration.GetConnectionString("DefaultConnection")));
41
42         services.AddDefaultIdentity<IdentityUser>()
43             .AddDefaultUI(UIFramework.Bootstrap)
44             .AddEntityFrameworkStores<ApplicationDbContext>();
45
46         services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);
47     }
48
49     // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
50     public void Configure(IApplicationBuilder app, IHostingEnvironment env)
51     {
52         if (env.IsDevelopment())
53         {
54             app.UseDeveloperExceptionPage();
55             app.UseDatabaseErrorPage();
56         }
57         else
58         {
59             app.UseExceptionHandler("/Home/Error");
60             // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
61             app.UseHsts();
62         }
63
64         app.UseHttpsRedirection();
65         app.UseStaticFiles();
66         app.UseCookiePolicy();
67
68         app.UseAuthentication();
69
70         app.UseMvc(routes =>

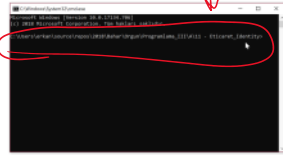
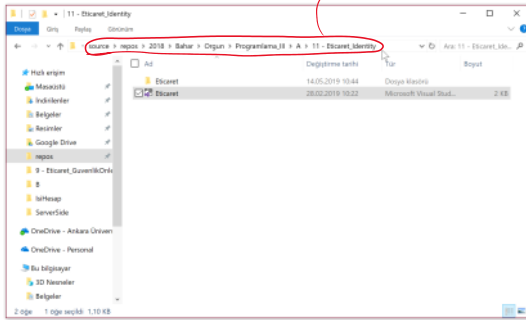
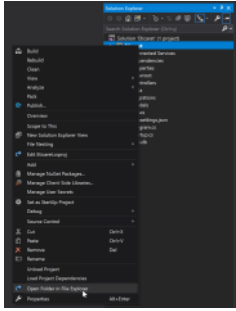
```

```

1
2
3 "ConnectionStrings": {
4     "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=aspnet-Identity-0a178710-
5     006d-4938-9f8a-19e7209220;Trusted_Connection=true;MultipleActiveResultSets=true"
6 }
7
8 "Logging": {
9     "LogLevel": {
10        "Default": "Warning"
11    }
12 }
13 "AllowedHosts": "*"

```

Komut ekranının (cmd) programın dışındaki başlaması için



dotnet tool install -g dotnet-aspnet-codegenerator
dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design
dotnet restore

codegenerator jektisi atipite

dotnet aspnet-codegenerator identity --useDefaultUI

```

Building project ...
Finding the generator 'identity'...
Running the generator 'identity'...
Runtime: 00:00:16.14

```

```

1 Support for ASP.NET Core Identity was added to your project
2 The code for adding Identity to your project was generated under Areas/Identity.
3 Configuration of the Identity related services can be found in the Areas/Identity/IdentityHostingStartup.cs file.
4 If your app was previously configured to use Identity, then you should remove the call to the AddIdentity method from your
  ConfigureServices method.
5
6 The generated UI requires support for static files. To add static files to your app:
7 1. Call app.UseStaticFiles() from your Configure method.
8
9 To use ASP.NET Core Identity you also need to enable authentication. To authentication to your app:
10 1. Call app.UseAuthentication() from your Configure method (after static files)
11 2. Call app.UseAuthorization() from your Configure method.
12
13 The generated UI requires MVC. To add MVC to your app:
14 1. Call services.AddMvc() from your ConfigureServices method
15 2. Call app.UseMvc() from your Configure method (after authentication)
16
17 The generated database code requires Entity Framework Core Migrations. Run the following commands:
18 1. dotnet ef migrations add CreateIdentitySchema
19 2. dotnet ef database update
20 Or from the Visual Studio Package Manager Console:
21 1. Add-Migration CreateIdentitySchema
22 2. Update-Database
23
24 Apps that use ASP.NET Core Identity should also use HTTPS. To enable HTTPS see https://go.microsoft.com/fwlink/?linkid=848054.

```

Projenize ASP.NET Çekirdek Kimliği desteği eklendi
 - Projenize Kimlik ekleme kodu Alanlar / Kimlik altında oluşturuldu.

Kimlik ile ilgili servislerin yapılandırılması, Areas / Identity / IdentityHostingStartup.cs dosyasında bulunabilir.

Uygulamanız daha önce Kimlik kullanacak şekilde yapılandırılmışsa, AddIdentity yöntemine yapılan çağrıyı ConfigureServices yönteminizden kaldırmalısınız.

Oluşturulan UI, statik dosyalar için destek gerektirir. Uygulamanıza statik dosyalar eklemek için:
 1. Yapılandır yönteminden app.StaticFiles () ögesini çağırın.

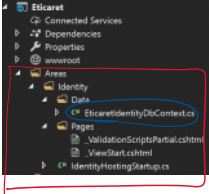
ASP.NET Çekirdek Kimliğini kullanmak için ayrıca kimlik doğrulamasını etkinleştirmeniz gerekir. Uygulamanızda kimlik doğrulaması yapmak için:
 1. Configure yönteminden services.AddMvc () Authentication () ögesini çağırın.

Oluşturulan UI, MVC gerektirir. MVC'yi uygulamanıza eklemek için:
 1. ConfigureServices yönteminden services.AddMvc () ögesini çağırın.
 2. Configure yönteminden applicationMvc () yöntemini çağırın (kimlik doğrulamasından sonra)

Üretilen veritabanı kodu Entity Framework Çekirdek Göçlerini gerektirir. Aşağıdaki komutları çalıştırın:
 1. dotnet ef migrations add CreateIdentitySchema
 2. dotnet ef veritabanı güncelleme

veya Visual Studio Paket Yöneticisi Konsolundan:
 1. Geçiş Ekleme CreateIdentitySchema
 2. Güncelleme Veritabanı

ASP.NET Çekirdek Kimliği kullanan uygulamalar da HTTPS kullanmalıdır. HTTPS'yi etkinleştirmek için <https://go.microsoft.com/fwlink/?linkid=848054> adresine bakın.



```

1 using System;
2 using Eticaret.Areas.Identity.Data;
3 using Microsoft.AspNetCore.Hosting;
4 using Microsoft.AspNetCore.Identity;
5 using Microsoft.AspNetCore.Identity.UI;
6 using Microsoft.EntityFrameworkCore;
7 using Microsoft.Extensions.Configuration;
8 using Microsoft.Extensions.DependencyInjection;
9
10 [assembly: HostingStartup(typeof(Eticaret.Areas.Identity.IdentityHostingStartup))]
11 namespace Eticaret.Areas.Identity
12 {
13     public class IdentityHostingStartup : IHostingStartup
14     {
15         public void Configure(IWebHostBuilder builder)
16         {
17             builder.ConfigureServices((context, services) => {
18                 services.AddDbContext<EticaretIdentityDbContext>(options => {
19                     options.UseSqlServer("Data Source=.;Initial Catalog=ETicaret;Integrated Security=True;User=root");
20                 });
21                 context.Configuration.GetConnectionString("EticaretIdentityDbContextConnection");
22                 services.AddDefaultIdentity<IdentityUser>()
23                     .AddEntityFrameworkStores<EticaretIdentityDbContext>();
24             });
25         }
26     }
27 }

```

```

1 {
2     "Logging": {
3         "LogLevel": {
4             "Default": "Warning"
5         }
6     },
7     "AllowedHosts": "*",
8     "ConnectionStrings": {
9         "EticaretIdentityDbContextConnection": "Server=(localdb);
10         Instance=(localdb);Database=Eticaret;Trusted_Connection=True;MultipleActiveResultSets=true"
11     }
12 }

```

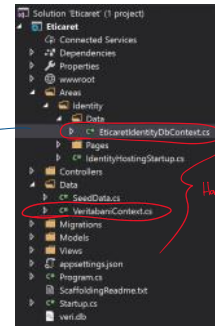
Handwritten notes:
 - Bu şekilde daha güvenli.
 - appsettings.json dosyasında kimlik bilgileri için bağlantı stringi kullanılır.
 - Bu şekilde daha güvenli.

```

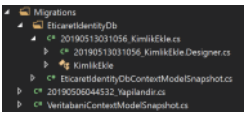
PM> add-migration KimlikEkle
More than one DbContext was found. Specify which one to use. Use the '-context' parameter for PowerShell commands and the '-context:' parameter for dotnet commands.
PM> add-migration KimlikEkle -context Eticaret.Areas.Identity.Data.EticaretIdentityDbContext
Microsoft.EntityFrameworkCore.Infrastructure[10493]
  Entity Framework Core 2.2.3-servicing-38854 initialized 'EticaretIdentityDbContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer' with options: None
To undo this action, use Remove-Migration.
PM> update-database
More than one DbContext was found. Specify which one to use. Use the '-context' parameter for PowerShell commands and the '-context:' parameter for dotnet commands.
PM> update-database -context Eticaret.Areas.Identity.Data.EticaretIdentityDbContext

```

Handwritten note: Birde farklı DbContext kullanılıyor aynı hanjisiyle çalıştırıyoruz özel olarak belirtmeliyiz. -Context ile



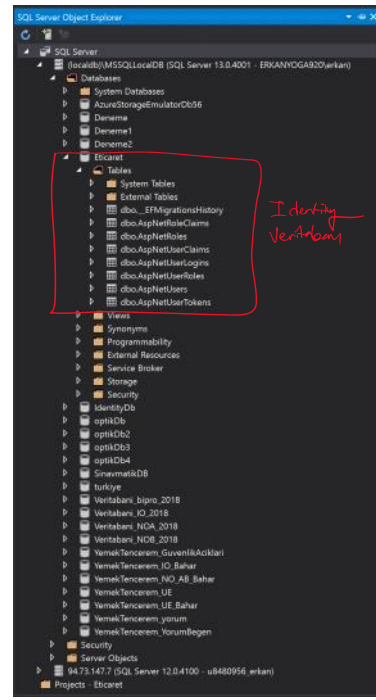
Handwritten note: Hangisi çalışıyor?



```

1 using System;
2 using Microsoft.EntityFrameworkCore.Metadata;
3 using Microsoft.EntityFrameworkCore.Migrations;
4
5 namespace Elicaret.Migrations.ElicaretIdentityDb
6 {
7     public partial class KimlikEkle : Migration
8     {
9         protected override void Up(MigrationBuilder migrationBuilder)
10        {
11            migrationBuilder.CreateTable(
12                name: "AspNetRoles",
13                columns: table => new
14                {
15                    Id = table.Column<string>(nullable: false),
16                    Name = table.Column<string>(maxLength: 256, nullable: true),
17                    NormalizedName = table.Column<string>(maxLength: 256, nullable: true),
18                    ConcurrencyStamp = table.Column<string>(nullable: true),
19                },
20                constraints: table =>
21                {
22                    table.PrimaryKey("PK_AspNetRoles", x => x.Id);
23                });
24            migrationBuilder.CreateTable(
25                name: "AspNetUsers",
26                columns: table => new
27                {
28                    Id = table.Column<string>(nullable: false),
29                    Username = table.Column<string>(maxLength: 256, nullable: true),
30                    NormalizedName = table.Column<string>(maxLength: 256, nullable: true),
31                    Email = table.Column<string>(maxLength: 256, nullable: true),
32                    NormalizedEmail = table.Column<string>(maxLength: 256, nullable: true),
33                    EmailConfirmed = table.Column<bool>(nullable: false),
34                    PasswordHash = table.Column<string>(nullable: true),
35                    SecurityStamp = table.Column<string>(nullable: true),
36                    ConcurrencyStamp = table.Column<string>(nullable: true),
37                    PhoneNumber = table.Column<string>(nullable: true),
38                    PhoneNumberConfirmed = table.Column<bool>(nullable: false),
39                    TwoFactorEnabled = table.Column<bool>(nullable: false),
40                    LockoutEnabled = table.Column<bool>(nullable: true),
41                    LockoutTime = table.Column<DateTimeOffset>(nullable: true),
42                    AccessFailedCount = table.Column<int>(nullable: false),
43                },
44                constraints: table =>
45                {
46                    table.PrimaryKey("PK_AspNetUsers", x => x.Id);
47                });
48            migrationBuilder.CreateTable(
49                name: "AspNetRoleClaims",
50                columns: table => new
51                {
52                    Id = table.Column<int>(nullable: false),
53                    RoleId = table.Column<string>(nullable: false),
54                    ClaimType = table.Column<string>(nullable: true),
55                    ClaimValue = table.Column<string>(nullable: true),
56                },
57                constraints: table =>
58                {
59                    table.PrimaryKey("PK_AspNetRoleClaims", x => x.Id),
60                    table.ForeignKey(
61                        name: "FK_AspNetRoleClaims_AspNetRoles_RoleId",
62                        column: x => x.RoleId,
63                        principalTable: "AspNetRoles",
64                        principalColumn: "Id",
65                        onDelete: ReferentialAction.Cascade);
66                });
67            migrationBuilder.CreateTable(
68                name: "AspNetUserClaims",
69                columns: table => new
70                {
71                    Id = table.Column<int>(nullable: false),
72                    UserId = table.Column<string>(nullable: false),
73                    ClaimType = table.Column<string>(nullable: true),
74                    ClaimValue = table.Column<string>(nullable: true),
75                },
76                constraints: table =>
77                {
78                    table.PrimaryKey("PK_AspNetUserClaims", x => x.Id),
79                    table.ForeignKey(
80                        name: "FK_AspNetUserClaims_AspNetUsers_UserId",
81                        column: x => x.UserId,
82                        principalTable: "AspNetUsers",
83                        principalColumn: "Id",
84                        onDelete: ReferentialAction.Cascade);
85                });
86            migrationBuilder.CreateTable(
87                name: "AspNetUserLogins",
88                columns: table => new
89                {
90                    LoginProvider = table.Column<string>(nullable: false),
91                    ProviderKey = table.Column<string>(nullable: false),
92                    ProviderDisplayName = table.Column<string>(nullable: true),
93                    UserId = table.Column<string>(nullable: false),
94                },
95                constraints: table =>
96                {
97                    table.PrimaryKey("PK_AspNetUserLogins", x => x.Id),
98                    table.ForeignKey(
99                        name: "FK_AspNetUserLogins_AspNetUsers_UserId",
100                       column: x => x.UserId,
101                       principalTable: "AspNetUsers",
102                       principalColumn: "Id",
103                       onDelete: ReferentialAction.Cascade);
104                });
105            migrationBuilder.CreateTable(
106                name: "AspNetUserRoles",
107                columns: table => new
108                {
109                    UserId = table.Column<string>(nullable: false),
110                    RoleId = table.Column<string>(nullable: false),
111                },
112                constraints: table =>
113                {
114                    table.PrimaryKey("PK_AspNetUserRoles", x => x.Id),
115                    table.ForeignKey(
116                        name: "FK_AspNetUserRoles_AspNetUsers_UserId",
117                        column: x => x.UserId,
118                        principalTable: "AspNetUsers",
119                        principalColumn: "Id",
120                        onDelete: ReferentialAction.Cascade);
121                    table.ForeignKey(
122                        name: "FK_AspNetUserRoles_AspNetRoles_RoleId",
123                        column: x => x.RoleId,
124                        principalTable: "AspNetRoles",
125                        principalColumn: "Id",
126                        onDelete: ReferentialAction.Cascade);
127                });
128            migrationBuilder.CreateTable(
129                name: "AspNetUserTokens",
130                columns: table => new
131                {
132                    UserId = table.Column<string>(nullable: false),
133                    LoginProvider = table.Column<string>(nullable: false),
134                    Name = table.Column<string>(nullable: false),
135                    Value = table.Column<string>(nullable: true),
136                },
137                constraints: table =>
138                {
139                    table.PrimaryKey("PK_AspNetUserTokens", x => x.Id),
140                    table.ForeignKey(
141                        name: "FK_AspNetUserTokens_AspNetUsers_UserId",
142                        column: x => x.UserId,
143                        principalTable: "AspNetUsers",
144                        principalColumn: "Id",
145                        onDelete: ReferentialAction.Cascade);
146                });
147            migrationBuilder.CreateTable(
148                name: "EFMigrationsHistory",
149                columns: table => new
150                {
151                    MigrationId = table.Column<string>(nullable: false),
152                    ProductVersion = table.Column<string>(nullable: true),
153                },
154                constraints: table =>
155                {
156                    table.PrimaryKey("PK_EFMigrationsHistory", x => x.MigrationId);
157                });
158            migrationBuilder.CreateIndex(
159                name: "ix_AspNetRoles",
160                table: "AspNetRoles",
161                column: "Id");
162            migrationBuilder.CreateIndex(
163                name: "ix_AspNetUsers",
164                table: "AspNetUsers",
165                column: "Id");
166            migrationBuilder.CreateIndex(
167                name: "ix_AspNetRoleClaims",
168                table: "AspNetRoleClaims",
169                column: "Id");
170            migrationBuilder.CreateIndex(
171                name: "ix_AspNetUserClaims",
172                table: "AspNetUserClaims",
173                column: "Id");
174            migrationBuilder.CreateIndex(
175                name: "ix_AspNetUserLogins",
176                table: "AspNetUserLogins",
177                column: "Id");
178            migrationBuilder.CreateIndex(
179                name: "ix_AspNetUserRoles",
180                table: "AspNetUserRoles",
181                column: "Id");
182            migrationBuilder.CreateIndex(
183                name: "ix_AspNetUserTokens",
184                table: "AspNetUserTokens",
185                column: "Id");
186            migrationBuilder.CreateIndex(
187                name: "ix_EFMigrationsHistory",
188                table: "EFMigrationsHistory",
189                column: "MigrationId");
190        }
191
192        protected override void Down(MigrationBuilder migrationBuilder)
193        {
194            migrationBuilder.DropTable(
195                name: "AspNetRoleClaims");
196            migrationBuilder.DropTable(
197                name: "AspNetUserClaims");
198            migrationBuilder.DropTable(
199                name: "AspNetUserLogins");
200            migrationBuilder.DropTable(
201                name: "AspNetUserRoles");
202            migrationBuilder.DropTable(
203                name: "AspNetUserTokens");
204            migrationBuilder.DropTable(
205                name: "AspNetRoles");
206            migrationBuilder.DropTable(
207                name: "AspNetUsers");
208            migrationBuilder.DropTable(
209                name: "EFMigrationsHistory");
210        }
211    }
212 }

```



```

1 using System;
2 using Elicaret.Areas.Identity.Data;
3 using Microsoft.AspNetCore.Hosting;
4 using Microsoft.AspNetCore.Identity;
5 using Microsoft.AspNetCore.Identity.UI;
6 using Microsoft.EntityFrameworkCore;
7 using Microsoft.Extensions.Configuration;
8 using Microsoft.Extensions.DependencyInjection;
9
10 [assembly: HostingStartup(typeof(Elicaret.Areas.Identity.IdentityHostingStartup))]
11 namespace Elicaret.Areas.Identity
12 {
13     public class IdentityHostingStartup : IHostingStartup
14     {
15         public void Configure(IWebHostBuilder builder)
16         {
17             builder.ConfigureServices((context, services) => {
18                 services.AddDbContext<ElicaretIdentityDbContext>(options =>
19                     options.UseSqlite(context.Configuration.GetConnectionString("Bizim_ElicaretIdentityDbContextConnection")));
20                 services.AddDefaultIdentity<IdentityUser>()
21                     .AddEntityFrameworkStores<ElicaretIdentityDbContext>();
22             });
23         }
24     }
25 }

```

```

1 {
2     "Logging": {
3         "LogLevel": {
4             "Default": "Warning"
5         }
6     },
7     "AllowedHosts": "*",
8     "ConnectionStrings": {
9         "ElicaretIdentityDbContextConnection": "Server=(localdb)\\
10         mssqllocaldb;Database=Elicaret;Trusted_Connection=True;MultipleActiveResultSets=true",
11         "Bizim_ElicaretIdentityDbContextConnection": "Data Source=ElicaretIdentity.db"
12     }
13 }

```

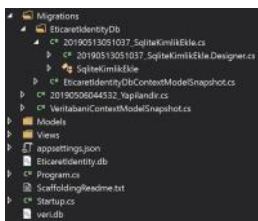
Service veritabanını postmekle yeteri

add-migration SqliteKimlikEkle -Context Elicaret.Areas.Identity.Data.ElicaretIdentityDbContext

```

PM> add-migration SqliteKimlikEkle
More than one DbContext was found. Specify which one to use. Use the "-Context:" parameter for PowerShell commands and the "--context:" parameter for dotnet commands.
PM> add-migration SqliteKimlikEkle -Context Elicaret.Areas.Identity.Data.ElicaretIdentityDbContext
Microsoft.EntityFrameworkCore.Infrastructure[10403]
Entity Framework Core 2.2.3-servicing-35854 initialized 'ElicaretIdentityDbContext' using provider 'Microsoft.EntityFrameworkCore.Sqlite' with options: None
To undo this action, use Remove-Migration.
PM> update-database
More than one DbContext was found. Specify which one to use. Use the "-Context:" parameter for PowerShell commands and the "--context:" parameter for dotnet commands.
PM> update-database -Context Elicaret.Areas.Identity.Data.ElicaretIdentityDbContext
Microsoft.EntityFrameworkCore.Infrastructure[10403]
Entity Framework Core 2.2.3-servicing-35854 initialized 'ElicaretIdentityDbContext' using provider 'Microsoft.EntityFrameworkCore.Sqlite' with options: None
Info: Executed DbCommand (18ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
- Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (18ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
PRAGMA foreign_keys=ON;

```



Item	Tip	Sema
Tablelar (9)		
AspNetRoleClaims		CREATE TABLE 'AspNetRoleClaims' ('Id' INTEGER NOT NULL CONSTRAINT 'PK_AspNetRoleClaims' PRIMARY KEY AUTOINCREMENT, 'RoleId' TEXT NOT NULL, 'ClaimType' TEXT NOT NULL, 'ClaimValue' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL)
AspNetRoles		CREATE TABLE 'AspNetRoles' ('Id' TEXT NOT NULL CONSTRAINT 'PK_AspNetRoles' PRIMARY KEY, 'Name' TEXT NOT NULL, 'NormalizedName' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL)
AspNetUserClaims		CREATE TABLE 'AspNetUserClaims' ('Id' INTEGER NOT NULL CONSTRAINT 'PK_AspNetUserClaims' PRIMARY KEY AUTOINCREMENT, 'UserId' TEXT NOT NULL, 'ClaimType' TEXT NOT NULL, 'ClaimValue' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL)
AspNetUserLogins		CREATE TABLE 'AspNetUserLogins' ('LoginProvider' TEXT NOT NULL, 'ProviderKey' TEXT NOT NULL, 'ProviderDisplayName' TEXT NOT NULL, 'UserId' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL)
AspNetUserRoles		CREATE TABLE 'AspNetUserRoles' ('UserId' TEXT NOT NULL, 'RoleId' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL, CONSTRAINT 'PK_AspNetUserRoles' PRIMARY KEY ('UserId', 'RoleId'))
AspNetUserTokens		CREATE TABLE 'AspNetUserTokens' ('UserId' TEXT NOT NULL, 'LoginProvider' TEXT NOT NULL, 'Name' TEXT NOT NULL, 'Value' TEXT NOT NULL, CONSTRAINT 'PK_AspNetUserTokens' PRIMARY KEY ('UserId', 'LoginProvider', 'Name'))
AspNetUsers		CREATE TABLE 'AspNetUsers' ('Id' TEXT NOT NULL CONSTRAINT 'PK_AspNetUsers' PRIMARY KEY, 'Username' TEXT NOT NULL, 'NormalizedUserName' TEXT NOT NULL, 'Email' TEXT NOT NULL, 'NormalizedEmail' TEXT NOT NULL, 'EmailConfirmed' INTEGER NOT NULL, 'PasswordHash' TEXT NOT NULL, 'SecurityStamp' TEXT NOT NULL, 'ConcurrencyStamp' TEXT NOT NULL, 'PhoneNumber' TEXT NOT NULL, 'PhoneNumberConfirmed' INTEGER NOT NULL, 'TwoFactorEnabled' INTEGER NOT NULL, 'LockoutEnabled' TEXT NOT NULL, 'LockoutTime' TEXT NOT NULL, 'AccessFailedCount' INTEGER NOT NULL)
EFMigrationsHistory		CREATE TABLE 'EFMigrationsHistory' ('MigrationId' TEXT NOT NULL CONSTRAINT 'PK_EFMigrationsHistory' PRIMARY KEY, 'ProductVersion' TEXT NOT NULL)
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
İndeksler (7)		
EmailIndex		CREATE INDEX 'EmailIndex' ON 'AspNetUsers' ('NormalizedEmail')
ix_AspNetRoleClaims_RoleId		CREATE INDEX 'ix_AspNetRoleClaims_RoleId' ON 'AspNetRoleClaims' ('RoleId')
ix_AspNetUserClaims_UserId		CREATE INDEX 'ix_AspNetUserClaims_UserId' ON 'AspNetUserClaims' ('UserId')
ix_AspNetUserLogins_UserId		CREATE INDEX 'ix_AspNetUserLogins_UserId' ON 'AspNetUserLogins' ('UserId')
ix_AspNetUserRoles_RoleId		CREATE INDEX 'ix_AspNetUserRoles_RoleId' ON 'AspNetUserRoles' ('RoleId')
RoleNameIndex		CREATE UNIQUE INDEX 'RoleNameIndex' ON 'AspNetRoles' ('NormalizedName')
UserNameIndex		CREATE UNIQUE INDEX 'UserNameIndex' ON 'AspNetUsers' ('NormalizedUserName')
ix_AspNetUsers		CREATE INDEX 'ix_AspNetUsers' ON 'AspNetUsers' ('Id')
ix_EFMigrationsHistory		CREATE INDEX 'ix_EFMigrationsHistory' ON 'EFMigrationsHistory' ('MigrationId')
ix_AspNetUsers_Username		CREATE INDEX 'ix_AspNetUsers_Username' ON 'AspNetUsers' ('Username')
ix_AspNetUsers_Email		CREATE INDEX 'ix_AspNetUsers_Email' ON 'AspNetUsers' ('Email')
ix_AspNetUsers_NormalizedEmail		CREATE INDEX 'ix_AspNetUsers_NormalizedEmail' ON 'AspNetUsers' ('NormalizedEmail')
ix_AspNetUsers_EmailConfirmed		CREATE INDEX 'ix_AspNetUsers_EmailConfirmed' ON 'AspNetUsers' ('EmailConfirmed')
ix_AspNetUsers_PasswordHash		CREATE INDEX 'ix_AspNetUsers_PasswordHash' ON 'AspNetUsers' ('PasswordHash')
ix_AspNetUsers_SecurityStamp		CREATE INDEX 'ix_AspNetUsers_SecurityStamp' ON 'AspNetUsers' ('SecurityStamp')
ix_AspNetUsers_ConcurrencyStamp		CREATE INDEX 'ix_AspNetUsers_ConcurrencyStamp' ON 'AspNetUsers' ('ConcurrencyStamp')
ix_AspNetUsers_PhoneNumber		CREATE INDEX 'ix_AspNetUsers_PhoneNumber' ON 'AspNetUsers' ('PhoneNumber')
ix_AspNetUsers_PhoneNumberConfirmed		CREATE INDEX 'ix_AspNetUsers_PhoneNumberConfirmed' ON 'AspNetUsers' ('PhoneNumberConfirmed')
ix_AspNetUsers_TwoFactorEnabled		CREATE INDEX 'ix_AspNetUsers_TwoFactorEnabled' ON 'AspNetUsers' ('TwoFactorEnabled')
ix_AspNetUsers_LockoutEnabled		CREATE INDEX 'ix_AspNetUsers_LockoutEnabled' ON 'AspNetUsers' ('LockoutEnabled')
ix_AspNetUsers_LockoutTime		CREATE INDEX 'ix_AspNetUsers_LockoutTime' ON 'AspNetUsers' ('LockoutTime')
ix_AspNetUsers_AccessFailedCount		CREATE INDEX 'ix_AspNetUsers_AccessFailedCount' ON 'AspNetUsers' ('AccessFailedCount')

2-TNR

User Manager

SignIn Manager

Role Manager

UserManager<IdentityUser> userManager = app.ApplicationServices.GetRequiredService<userManager<IdentityUser>>;

```

using System.Threading.Tasks;
namespace Eticaret.Data
{
    public static class SeedData
    {
        public async static void Seed(IApplicationBuilder app)
        {
            VeritabaniContext context = app.ApplicationServices.GetRequiredService<VeritabaniContext>();
            userManager<IdentityUser> userManager = app.ApplicationServices.GetRequiredService<userManager<IdentityUser>>();

            context.Database.Migrate(); //Bekleyen migrationları yürürlüğe koyar,ışletir

            if (!context.Urunler.Any())//ürünler tablosu boşsa
            {
                var kategoriler = new Kategori[] { };
                context.Kategoriler.AddRange(kategoriler);

                var urunler = new Urun[] { };
                context.AddRange(urunler);

                //var kullanicilar = new Kullanicil[]
                //{
                //    new Kullanicil{KAdi:"Ahmet",Sifre:"123",Turu:"Admin"},
                //    new Kullanicil{KAdi:"Mehmet",Sifre:"321",Turu:"User"}
                //};
                //context.AddRange(kullanicilar);

                await userManager.CreateAsync(new IdentityUser { UserName = "Ahmet" }, "Ahmet.123");
                await userManager.CreateAsync(new IdentityUser { UserName = "Mehmet" }, "Mehmet.123");

                context.SaveChanges();
            }
        }
    }
}
    
```

Id	KAdi	Sifre	Turu
1	Ahmet	123	Admin
2	Mehmet	321	User

Id	UserName	NormalizedUserName	Email	NormalizedEmail	IsConfirmed	PasswordHash	SecurityStamp	ConcurrencyStamp	PhoneNumber	PhoneNumberConfirmed	TwoFactorEnabled	Lockout
1	5b72d0-9ca...	AHMET	AHMET	AHMET	0	AQAAMEAACQAAAAENX9SBT7GKAfB8vz...	EKIHKHDXIAIS4AHp55SAZHCTHQ...	cd691740 ca...		0	0	
2	2b2ff115-fb5f...	MEHMET	MEHMET	MEHMET	0	AQAAMEAACQAAAAEDY74s0GZ3sRmQK9L...	LUNW6ESLWBAEMSA4HXZVWCRG2...	088d336f-045...		0	0	

Veritabanı ek pasword buik kullanicilaring sifreleri okunmuydu.

9 #milgen

ABC225 → Ahmet.123
 MZ3h x 5 → Mehmet.321

Hash tablosu getirme sifreleri için veri sifrelerin korunmak için ayrıca (Salt) kullanılır.

```

Startup.cs
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production scenarios.
        app.UseHsts();
    }

    app.UseHttpsRedirection();
    app.UseStaticFiles();
    app.UseAuthentication(); // Eklendi unutmayi!
    app.UseCookiePolicy();
    app.UseSession();
    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "default",
            template: "{controller=Home}/{action=Index}/{id?}");
    });

    SeedData.Seed(app);
}
    
```

```

13 public class KullaniciIslemleriController : Controller
14 {
15     private readonly VeritabaniContext _context;
16     private readonly SignInManager<IdentityUser> signInManager;
17
18     public KullaniciIslemleriController(VeritabaniContext context, SignInManager<IdentityUser> signInManager) //ctor
19     {
20         this.context = context;
21         this.signInManager = signInManager;
22     }
23     [HttpGet]
24     public IActionResult Giriş()
25     {
26         return View();
27     }
28
29     [HttpPost]
30     [ValidateAntiForgeryToken]
31     public async Task<ActionResult> Giriş(Kullanici kullanıcı)
32     {
33         var giriş = await signInManager.PasswordSignInAsync(kullanici.KAdi, kullanıcı.Sifre, false, true);
34         if (giris.Succeeded)
35         {
36             TempData["Mesaj"] = "Giriş başarılı";
37             return RedirectToAction("Index", "Home");
38         }
39         else
40         {
41             TempData["Mesaj"] = "Geçersiz kullanıcı adı veya şifre!";
42             return View();
43         }
44     }
45 }

```

Bağımlılık enjektörümü

3. TWR

```

1 using Microsoft.AspNetCore.Identity
2 {
3     Inject SignInManager<IdentityUser> signInManager
4 }
5 <DOCTYPE html>
6 <html>
7     <head>.../head>
24 <body>
25     <header>
26         <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
27             <div class="container">
28                 <a class="navbar-brand asp-area="" asp-controller="Home" asp-action="Index">Eticaret/asp>
29                 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"
30                     aria-expanded="false" aria-label="Toggle navigation">
31                     <span class="navbar-toggler-icon"></span>
32                 </button>
33                 <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
34                     <ul class="navbar-nav">
35                         <li class="nav-item">
36                             <a class="nav-link text-dark" asp-action="Cikis" asp-controller="KullaniciIslemleri"><i>çıkış</i></a>
37                         </li>
38                         <li class="nav-item">
39                             <a class="nav-link text-dark" asp-action="Giris" asp-controller="KullaniciIslemleri"><i>Giriş</i></a>
40                         </li>
41                         <li class="nav-item">
42                             <a class="nav-link text-dark" asp-action="Kayit" asp-controller="KullaniciIslemleri"><i>Kayıt</i></a>
43                         </li>
44                     </ul>
45                     <div class="text-right">
46                         <span class="text-dark"><i>Giriş yap</i></span>
47                     </div>
48                 </div>
49             </div>
50         </header>
51     </body>
52 </html>

```

Bağımlılık

çıkış: kullanıcının giriş işlemi başarılı ise

```

KullaniciIslemleriController.cs
public async Task<ActionResult> Cikis()
{
    HttpContext.Session.Clear(); // Sessiyon'u da temizlese iyi olur.
    await signInManager.SignOutAsync(); // Aşıl çıkışı sağlıyor kod
    TempData["Mesaj"] = "Geçerli oturum başarıyla sonlandırıldı.";
    return RedirectToAction("Index", "Home");
}

```

```
KategoriYonetimController.cs
Eticaret
Eticaret.Controllers.KategoriYonetimController.cs
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Mvc;
6 using Microsoft.AspNetCore.Mvc.Rendering;
7 using Microsoft.EntityFrameworkCore;
8 using Eticaret.Data;
9 using Eticaret.Models;
10 using Microsoft.AspNetCore.Authorization;
11
12 namespace Eticaret.Controllers
13 {
14     [Authorize] Sadece yetkili kullanıcılar
15     public class KategoriYonetimController : Controller
16     {
17         private readonly VeritabaniContext _context;
18
19         public KategoriYonetimController(VeritabaniContext context)
20         {
21             _context = context;
22         }
23
24         // GET: KategoriYonetimi
25         public async Task<IActionResult> Index()
26         {
27             return View(await _context.Kategoriler.ToListAsync());
28         }
29     }
30 }
```