

Model ASP.NET MVC Core 3.1 Web App - Deel 6

1 Authenticatie en Autorisatie

1.1 Aanmaken model voor ApplicationUser

Onder de folder models, voeg een nieuwe class toe voor ApplicationUser

```
public class ApplicationUser : IdentityUser
{
    public string Name { get; set; }
    public string StreetAddress { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public string PostalCode { get; set; }
}
```

1.2 Aanpassen ApplicationDbContext

Voeg aan de class ApplicationDbContext DbSet public properties toe voor SubCategory, MenuItem, Coupon en ApplicationUser

```
public DbSet<ApplicationUser> ApplicationUser { get; set; }
```

1.3 Aanpassen Startup.cs

Vervang in ConfigureServices IdentityUser door ApplicationIdentityUser. We specificëren eveneens IdentityRole bij AddIdentity service omdat we rollen zullen gebruiken in onze app

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));

    services.AddIdentity<IdentityUser, IdentityRole>()
        .AddDefaultTokenProviders()
        .AddEntityFrameworkStores<ApplicationDbContext>();

    services.AddControllersWithViews();
    services.AddRazorPages().AddRazorRuntimeCompilation();
}
```

1.4 Database migraties toevoegen voor de aangepaste AspNetUsers tabel en database updaten

Indien je applicatie goed werkt, kan je nu de tabellen in de database creëren.

We doen dit via de **NuGet Package Manager Console** door eerst data-migraties op te zetten en daarna update-database commando

```
PM> add-migration ChangedAspNetUsersToDb
```

```
PM> update-database
```

1.5 Aanpassen van de Register Razor Page en Code-behind

1. Open Areas/Identity/Pages/Account/Register.cshtml.cs en voeg de extra properties toe aan het Input model:

```
public class InputModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    public string Email { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "Het {0} moet minstens {2} en maximaal
{1} tekens bevatten.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Wachtwoord")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Bevestig Wachtwoord")]
    [Compare("Password", ErrorMessage = "Het wachtwoord en bevestiging van
wachtwoord komen niet overeen")]
    public string ConfirmPassword { get; set; }

    [Required]
    [Display(Name = "Naam")]
    public string Name { get; set; }
    [Display(Name = "Straat")]
    public string StreetAddress { get; set; }
    [Display(Name = "TelNr")]
    public string PhoneNumber { get; set; }
    [Display(Name = "Gemeente")]
    public string City { get; set; }
    [Display(Name = "Provincie")]
    public string State { get; set; }
    [Display(Name = "Postcode")]
    public string PostalCode { get; set; }
}
```

2. Wijzig in de Register.cshtml.cs eveneens de creatie van IdentityUser object naar een ApplicationUser (in methode OnPostAsync)

```
public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    if (ModelState.IsValid)
    {
        //var user = new IdentityUser { UserName = Input.Email, Email =
Input.Email };
        var user = new ApplicationUser
        {
            UserName = Input.Email,
            Email = Input.Email,
        }
    }
}
```

```

        Name = Input.Name,
        City = Input.City,
        StreetAddress = Input.StreetAddress,
        State = Input.State,
        PostalCode = Input.PostalCode,
        PhoneNumber = Input.PhoneNumber
    };
    var result = await _userManager.CreateAsync(user, Input.Password);
    if (result.Succeeded)
    {...
```

3. Open Areas/Identity/Pages/Account/Register.cshtml en voeg extra form fields toe voor de properties van de ApplicationUser (afgeleide class van IdentityUser):

```

@page
@model RegisterModel
```

```

@{
    ViewData["Title"] = "Registreer";
}
```

```

<br />
<h2 class="text-info">Een nieuwe account aanmaken</h2>
<br />

<form method="post" asp-route-returnUrl="@Model.ReturnUrl">
    <div class="border backgroundColorWhite">
        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.Name" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.Name" class="form-control" />
            </div>
            <span asp-validation-for="Input.Name" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.Email" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.Email" class="form-control" />
            </div>
            <span asp-validation-for="Input.Email" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.PhoneNumber" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.PhoneNumber" class="form-control" />
            </div>
        </div>
    </div>
</form>
```

```

        </div>
        <span asp-validation-for="Input.PhoneNumber" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.StreetAddress" class="col-form-
label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.StreetAddress" class="form-control" />
        </div>
        <span asp-validation-for="Input.StreetAddress" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.City" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.City" class="form-control" />
        </div>
        <span asp-validation-for="Input.City" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.State" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.State" class="form-control" />
        </div>
        <span asp-validation-for="Input.State" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.PostalCode" class="col-form-
label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.PostalCode" class="form-control" />
        </div>
        <span asp-validation-for="Input.PostalCode" class="text-
danger"></span>
    </div>

    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.Password" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.Password" class="form-control" />
        </div>
        <span asp-validation-for="Input.Password" class="text-
danger"></span>
    </div>

    <div class="form-group row">
        <div class="col-2">

```

```

        <label asp-for="Input.ConfirmPassword" class="col-form-label"></label>
    </div>
    <div class="col-5">
        <input asp-for="Input.ConfirmPassword" class="form-control" />
    </div>
    <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
</div>

    <div class="form-group row">
        <div class="col-5 offset-2">
            <button type="submit" class="btn btn-primary form-control">Registreer</button>
        </div>
    </div>
</div>
</form>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

2 Authorisatie

2.1 De namen van de verschillende rollen gaan we bijhouden in constanten in een Utility class.

Maak een nieuwe folder Utility aan en maak daarin een nieuwe static class SD:

```

namespace PittigRestoMVC.Utility
{
    namespace Pittig.Utility
    {
        public static class SD
        {
            public const string DefaultFoodImage = "default_food.png";

            public const string ManagerUser = "Manager";
            public const string KitchenUser = "Kitchen";
            public const string FrontDeskUser = "FrontDesk";
            public const string CustomerEndUser = "Customer";
        }
    }
}

```

2.2 Database initialiseren met rollen

We willen deze rollen in de database plaatsen (Table ASPNetRoles).

Er zijn verschillende manieren om dit te doen.

We kunnen dit doen via een Initializer te gebruiken, die de rollen zal creëren indien deze nog niet bestaan:

Maak onder de folder Data een Interface IDbInitializer en class DbInitializer:

```

namespace PittigRestMVC.Data

```

```

{
    public interface IDbInitializer
    {
        void Initialize();
    }
}

namespace PittigRestoMVC.Data
{
    public class DbInitializer : IDbInitializer
    {
        private readonly ApplicationDbContext _db;
        private readonly UserManager<IdentityUser> _userManager;
        private readonly RoleManager<IdentityRole> _roleManager;

        public DbInitializer(ApplicationDbContext db, UserManager<IdentityUser>
userManager, RoleManager<IdentityRole> roleManager)
        {
            _db = db;
            _roleManager = roleManager;
            _userManager = userManager;
        }

        public async void Initialize()
        {
            if (_db.Database.GetPendingMigrations().Count() > 0)
            {
                _db.Database.Migrate();
            }

            if (_db.Roles.Any(r => r.Name == SD.ManagerUser)) return;

            _roleManager.CreateAsync(new
IdentityRole(SD.ManagerUser)).GetAwaiter().GetResult();
            _roleManager.CreateAsync(new
IdentityRole(SD.FrontDeskUser)).GetAwaiter().GetResult();
            _roleManager.CreateAsync(new
IdentityRole(SD.KitchenUser)).GetAwaiter().GetResult();
            _roleManager.CreateAsync(new
IdentityRole(SD.CustomerEndUser)).GetAwaiter().GetResult();

            _userManager.CreateAsync(new ApplicationUser
            {
                UserName = "admin@gmail.com",
                Email = "admin@gmail.com",
                Name = "Admin GMAIL",
                EmailConfirmed = true,
                PhoneNumber = "1112223333"
            }, "Admin123*").GetAwaiter().GetResult();

            IdentityUser user = await _db.Users.FirstOrDefaultAsync(u => u.Email ==
"admin@gmail.com");

            await _userManager.AddToRoleAsync(user, SD.ManagerUser);
        }
    }
}

```

```

    }
}

```

2.3 Pas Startup.cs aan:

1. In ConfigureServices voegen we de DbInitializer class toe via AddScoped

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));

    services.AddIdentity<IdentityUser, IdentityRole>()
        .AddDefaultTokenProviders()
        .AddEntityFrameworkStores<ApplicationDbContext>();

    services.AddScoped<IDbInitializer, DbInitializer>();
    services.AddControllersWithViews();
    services.AddRazorPages().AddRazorRuntimeCompilation();
}

```

2. In Configure voegen roepen we deze aan:

```

public void Configure(IApplicationBuilder app, IWebHostEnvironment env,
    IDbInitializer dbInitializer)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseDatabaseErrorPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for
        production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();
    dbInitializer.Initialize();
    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern:
                "{area=Customer}/{controller=Home}/{action=Index}/{id?}");
        endpoints.MapRazorPages();
    });
}
}

```

2.4 Run de web app en controleer of dat een User 'Admin@gmail.com' met Rol 'MANAGER' aan de database is toegevoegd:

SQLQuery3.sql - (L...3CPIU4L\hcop (76)) * SQLQuery2.sql - (L...3CPIU4L\hcop (80)) SQLQuery1.sql - (L...3CPIU4L\hcop (66))

```

/***** Script for SelectTopNRows command from SSMS *****/

select * FROM [AspNetUsers]
select * FROM [AspNetRoles]
SELECT * FROM [AspNetUserRoles]
  
```

Results Messages

	Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	PasswordHash	SecurityStamp
1	45bf14d0-26e5-4824-8733-712f6aee4e95	admin@gmail.com	ADMIN@GMAIL.COM	admin@gmail.com	ADMIN@GMAIL.COM	1	AQAAAAEAAACcQAAAAEJlhFEAYQImQqTUJPrx+QWkk216JwyC...	13B7J5XE2LGYIYL

	Id	Name	NormalizedName	ConcurrencyStamp
1	16effe64-feb2-421b-9133-b9d066832ba9	Manager	MANAGER	96792067-b3da-45f2-8c34-fa073c23debb
2	28f3c2b6-3724-442e-8180-7b4f3992717f	Customer	CUSTOMER	6eb4b63f-401d-46a1-918f-2a03a8e17ff9
3	a9ff489-0a23-4766-bd03-5c9cdfc480a0	FrontDesk	FRONTDESK	239d697b-df9c-4bcd-80db-1248af4df59
4	b1740646-7928-428d-9e19-9269ec65a4c1	Kitchen	KITCHEN	d06402f0-6183-42ba-a3b8-0d291b787993

	Userid	Roleid
1	45bf14d0-26e5-4824-8733-712f6aee4e95	16effe64-feb2-421b-9133-b9d066832ba9

2.5 Toegang regelen tot controllers op basis van Rollen

Om enkel ingelogde users met de Manager rol toelating te geven tot de functionaliteiten van het content-management gedeelte:

Voeg bovenaan elke Controller class (Categorie, SubCategorie, MenuItem, Coupon) het volgende attribuut toe:

```
[Authorize(Roles = SD.ManagerUser)]
```

2.6 Verbergen/tonen van Razor Content op basis van Rollen

We gaan op basis van de rol bepaalde delen van de Razor Views al dan niet tonen (bv de dropdown list voor web content beheer):

Open Views/Shared/_Layout.cshtml en pas deze aan:

```
@using PittigRestoMVC.Utility
```

.....

```

<partial name="_LoginPartial" />
<ul class="navbar-nav flex-grow-1">
@if (User.IsInRole(SD.ManagerUser))
{
    <li class="nav-item dropdown text-white-50">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropDownMenuLink"
role="button"
                                data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
  
```

Inhoud beheeren

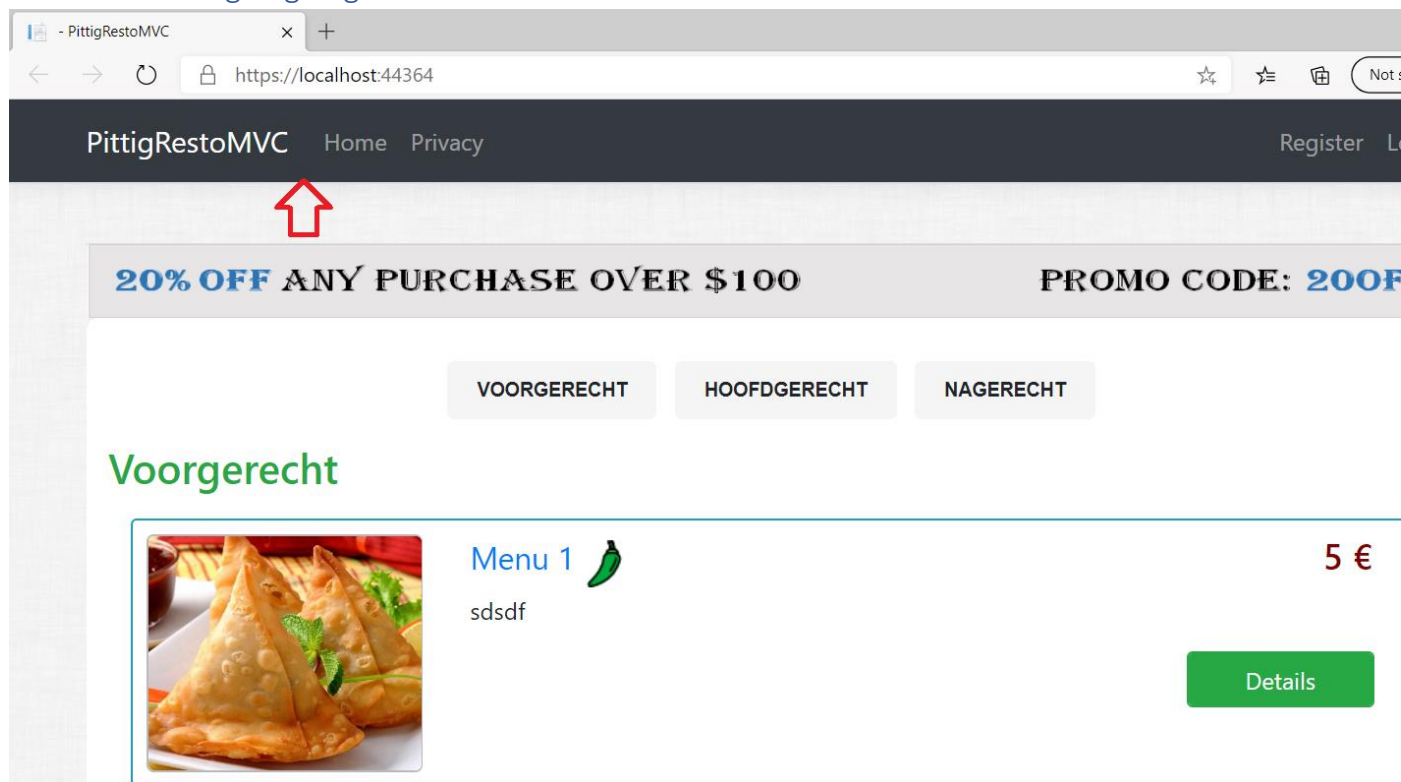

```

        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropDownMenuLink">
            <a class="dropdown-item" asp-action="Index" asp-controller="Category"
asp-area="Admin">categorie</a>
            <a class="dropdown-item" asp-action="Index" asp-controller="SubCategory"
asp-area="Admin">subcategorie</a>
            <a class="dropdown-item" asp-action="Index" asp-controller="MenuItem"
asp-area="Admin">Menu</a>
            <a class="dropdown-item" asp-action="Index" asp-controller="Coupon" asp-
area="Admin">Kortingen</a>
            <a class="dropdown-item" asp-action="Index" asp-controller="User" asp-
area="Admin">Personeel</a>
        </div>
    </li>
}
    <li class="nav-item">
        <a class="nav-link" asp-area="" asp-controller="Home" asp-
action="Index">Home</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy</a>
    </li>
</ul>

....

```

- 2.7 Start de app en controleer dat je de dropdown lijst 'Inhoud beheren' niet zichtbaar is voor niet ingelogde gebruikers:



2.8 Log in met een User met Manager rol en controleer dropdown lijst 'Inhoud beheren' zichtbaar wordt:

Log in - PittigRestoMVC

https://localhost:44364/Identity/Account/Login

PittigRestoMVC Home Privacy Register Login

Log in

Use a local account to log in.

Email

admin@gmail.com

Password

Admin123*

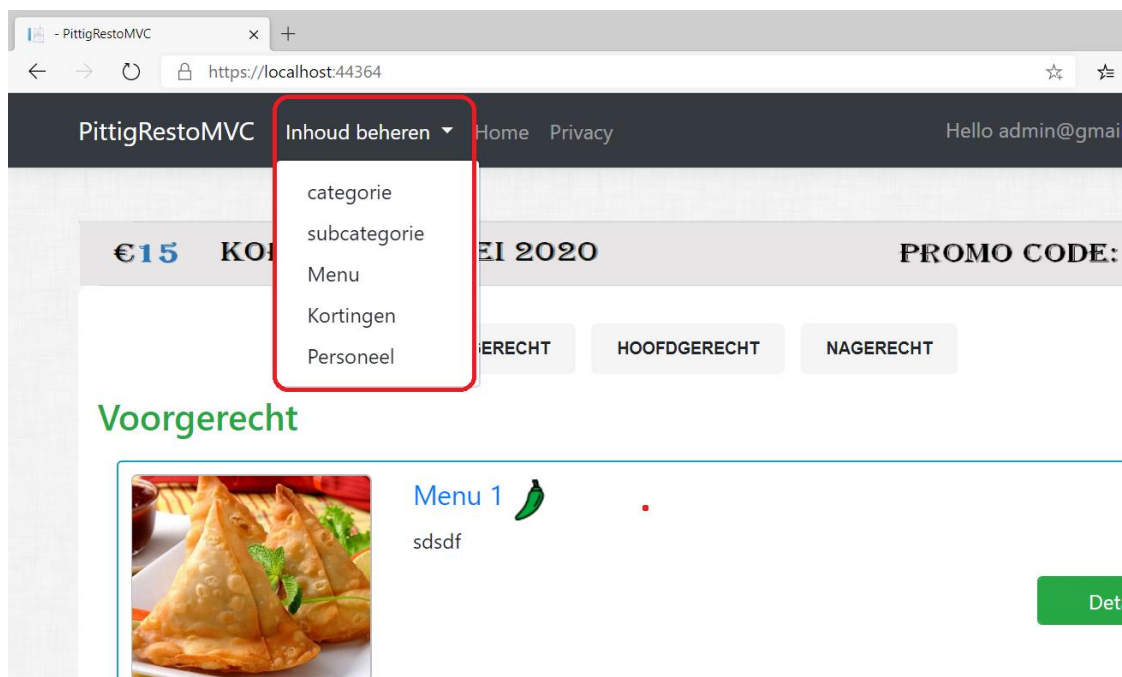
☐ Remember me?

Log in

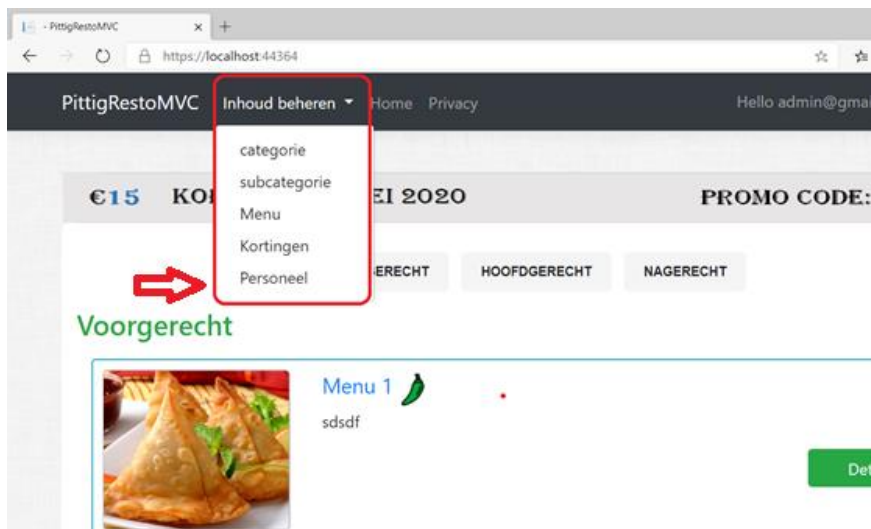
[Forgot your password?](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.



3 Personeel beheren



3.1 Aanmaken UserController

1. Maak onder de folder Areas/Admin/Controllers een nieuwe lege MVC Controller aan met naam 'UserController':

```
using System;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PittigRestoMVC.Data;
using PittigRestoMVC.Utility;

namespace PittigRestoMVC.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.ManagerUser)]
    public class UserController : Controller
    {
        private readonly ApplicationDbContext _db;

        public UserController(ApplicationDbContext db)
        {
            _db = db;
        }

        public async Task<IActionResult> Index()
        {
            var claimsIdentity = (ClaimsIdentity)this.User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

            return View(await _db.ApplicationUser.Where(u => u.Id != claim.Value).ToListAsync());
        }

        public async Task<IActionResult> Lock(string id)
        {
            if (id == null)
            {
                return NotFound();
            }

            var applicationUser = await _db.ApplicationUser.FirstOrDefaultAsync(m => m.Id == id);

            if (applicationUser == null)
            {
                return NotFound();
            }
        }
    }
}
```

```

    }

    applicationUser.LockoutEnd = DateTime.Now.AddYears(100);

    await _db.SaveChangesAsync();

    return RedirectToAction(nameof(Index));
}

public async Task<IActionResult> Unlock(string id)
{
    if (id == null)
    {
        return NotFound();
    }

    var applicationUser = await _db.ApplicationUser.FirstOrDefault(m => m.Id == id);

    if (applicationUser == null)
    {
        return NotFound();
    }

    applicationUser.LockoutEnd = DateTime.Now;

    await _db.SaveChangesAsync();

    return RedirectToAction(nameof(Index));
}
}
}

```

2. Maak onder de folder Areas/Admin/Views een nieuwe folder 'User' aan en daarin een Razor View 'Index.cshtml':

```

@model IEnumerable<ApplicationUser>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br /><br />
<div class="border backgroundWhite">
    <div class="row">
        <div class="col-6">
            <h2 class="text-info"> Gebruikerslijst</h2>
        </div>
        <div class="col-6 text-right">
            <a asp-area="Identity" asp-page="/Account/Register" class="btn btn-info"><i
class="fas fa-plus"></i> &nbsp; Nieuw Personeelslid</a>
        </div>
    </div>
<br />
<div>
    @if (Model.Count() > 0)
    {
        <table class="table table-striped border">
            <tr class="table-secondary">
                <th>
                    @Html.DisplayNameFor(m => m.Name)
                </th>
                <th>
                    @Html.DisplayNameFor(m => m.Email)
                </th>
            </tr>

```

```

        @Html.DisplayNameFor(m => m.PhoneNumber)
    </th>
</th></th>
</tr>
@foreach (var item in Model)
{
    <tr>
        <td>
            @Html.DisplayFor(m => item.Name)
        </td>
        <td>
            @Html.DisplayFor(m => item.Email)
        </td>
        <td>
            @Html.DisplayFor(m => item.PhoneNumber)
        </td>
        <td>
            @if (item.LockoutEnd == null || item.LockoutEnd <
DateTime.Now)
            {
                <a class="btn btn-success text-white" asp-action="Lock"
asp-route-id="@item.Id">
                    <i class="fas fa-lock-open"></i>
                </a>
            }
            else
            {
                <a class="btn btn-danger text-white" asp-action="UnLock"
asp-route-id="@item.Id">
                    <i class="fas fa-lock"></i>
                </a>
            }
        </td>
    </tr>
}
</table>
}
else
{
    <p>Er bestaan nog geen gebruikers...</p>
}
</div>
</div>

```

4 Instellen van User Rollen toevoegen aan Register Razor pagina

1. De Manager rol mag Users Registreren en een rol toekennen. Hiervoor wordt een radiobutton-list toegevoegd aan de Register.cshtml Razor page:

Voeg ook een using directive (bovenaan) toe naar PittigRestoMVC.Utility namespace(voor het gebruiken van de class SD):

```

@page
@using PittigRestoMVC.Utility
@model RegisterModel

```

```

@{
    ViewData["Title"] = "Registreen";
}

<br />
<h2 class="text-info">Een nieuwe account aanmaken</h2>
<br />

<form method="post" asp-route-returnUrl="@Model.ReturnUrl">
    <div class="border backgroundWhite">
        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.Name" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.Name" class="form-control" />
            </div>
            <span asp-validation-for="Input.Name" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.Email" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.Email" class="form-control" />
            </div>
            <span asp-validation-for="Input.Email" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.PhoneNumber" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.PhoneNumber" class="form-control" />
            </div>
            <span asp-validation-for="Input.PhoneNumber" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.StreetAddress" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.StreetAddress" class="form-control" />
            </div>
            <span asp-validation-for="Input.StreetAddress" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.City" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.City" class="form-control" />
            </div>
            <span asp-validation-for="Input.City" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.State" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.State" class="form-control" />
            </div>
            <span asp-validation-for="Input.State" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Input.PostalCode" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Input.PostalCode" class="form-control" />
            </div>
            <span asp-validation-for="Input.PostalCode" class="text-danger"></span>
        </div>
    </div>
</form>

```

```

        </div>
        <div class="col-5">
            <input asp-for="Input.PostalCode" class="form-control" />
        </div>
        <span asp-validation-for="Input.PostalCode" class="text-danger"></span>
    </div>

    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.Password" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.Password" class="form-control" />
        </div>
        <span asp-validation-for="Input.Password" class="text-danger"></span>
    </div>

    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.ConfirmPassword" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.ConfirmPassword" class="form-control" />
        </div>
        <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
    </div>
    @if (User.IsInRole(SD.ManagerUser))
    {
        <div class="form-group row">
            <div class="col-2">
            </div>
            <div class="col-5">
                <input type="radio" name="rdUserRole" value="@SD.KitchenUser" checked /> Keuken
                <input type="radio" name="rdUserRole" value="@SD.FrontDeskUser" checked /> Balie
                <input type="radio" name="rdUserRole" value="@SD.ManagerUser" checked /> Manager
            </div>
        </div>
    }
    <div class="form-group row">
        <div class="col-5 offset-2">
            <button type="submit" class="btn btn-primary form-control">Registreer</button>
        </div>
    </div>
</div>
</form>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

4.1 Toevoegen van Role aan Razor code behind:

1. Open Register.cshtml.cs, de code behind van de Register Razor page en pas de methode OnPostAsync aan:

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Text.Encodings.Web;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Identity.UI.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.Extensions.Logging;

```

```

using PittigRestoMVC.Models;
using PittigRestoMVC.Utility;

namespace PittigRestoMVC.Areas.Identity.Pages.Account
{
    [AllowAnonymous]
    public class RegisterModel : PageModel
    {
        private readonly SignInManager<IdentityUser> _signInManager;
        private readonly UserManager<IdentityUser> _userManager;
        private readonly ILogger<RegisterModel> _logger;
        private readonly IEmailSender _emailSender;
        private readonly RoleManager<IdentityRole> _roleManager;

        public RegisterModel(
            UserManager<IdentityUser> userManager,
            SignInManager<IdentityUser> signInManager,
            ILogger<RegisterModel> logger,
            IEmailSender emailSender,
            RoleManager<IdentityRole> roleManager)
        {
            _userManager = userManager;
            _signInManager = signInManager;
            _logger = logger;
            _emailSender = emailSender;
            _roleManager = roleManager;
        }

        [BindProperty]
        public InputModel Input { get; set; }

        public string returnUrl { get; set; }

        public class InputModel
        {
            [Required]
            [EmailAddress]
            [Display(Name = "Email")]
            public string Email { get; set; }

            [Required]
            [StringLength(100, ErrorMessage = "Het {0} moet minstens {2} en
maximimaal {1} tekens bevatten.", MinimumLength = 6)]
            [DataType(DataType.Password)]
            [Display(Name = "Wachtwoord")]
            public string Password { get; set; }

            [DataType(DataType.Password)]
            [Display(Name = "Bevestig Wachtwoord")]
            [Compare("Password", ErrorMessage = "Het wachtwoord en bevestiging van
wachtwoord komen niet overeen")]
            public string ConfirmPassword { get; set; }

            [Required]
            [Display(Name = "Naam")]
            public string Name { get; set; }
            [Display(Name = "Straat")]
            public string StreetAddress { get; set; }
        }
    }
}

```



```

        [Display(Name = "TelNr")]
        public string PhoneNumber { get; set; }
        [Display(Name = "Gemeente")]
        public string City { get; set; }
        [Display(Name = "Provincie")]
        public string State { get; set; }
        [Display(Name = "Postcode")]
        public string PostalCode { get; set; }
    }

    public void OnGet(string returnUrl = null)
    {
        ReturnUrl = returnUrl;
    }

    public async Task<IActionResult> OnPostAsync(string returnUrl = null)
    {
        string role = Request.Form["rdUserRole"].ToString();
        role = String.IsNullOrEmpty(role) ? SD.CustomerEndUser : role;
        if (ModelState.IsValid)
        {
            var user = new ApplicationUser
            {
                UserName = Input.Email,
                Email = Input.Email,
                Name = Input.Name,
                City = Input.City,
                StreetAddress = Input.StreetAddress,
                State = Input.State,
                PostalCode = Input.PostalCode,
                PhoneNumber = Input.PhoneNumber
            };
            var result = await _userManager.CreateAsync(user, Input.Password);
            _logger.LogInformation("User created a new account with password.");
            if (result.Succeeded)
            {
                await _userManager.AddToRoleAsync(user, role);
                if (role == SD.CustomerEndUser)
                {
                    await _signInManager.SignInAsync(user, isPersistent: false);
                    return RedirectToAction("Index", "Home", new { area = "Customer" });
                }

                return RedirectToAction("Index", "User", new { area = "Admin" });
            }

            //var code = await
            _userManager.GenerateEmailConfirmationTokenAsync(user);
            //var callbackUrl = Url.Page(
            //    "/Account/ConfirmEmail",
            //    pageHandler: null,
            //    values: new { userId = user.Id, code = code },
            //    protocol: Request.Scheme);

```

```

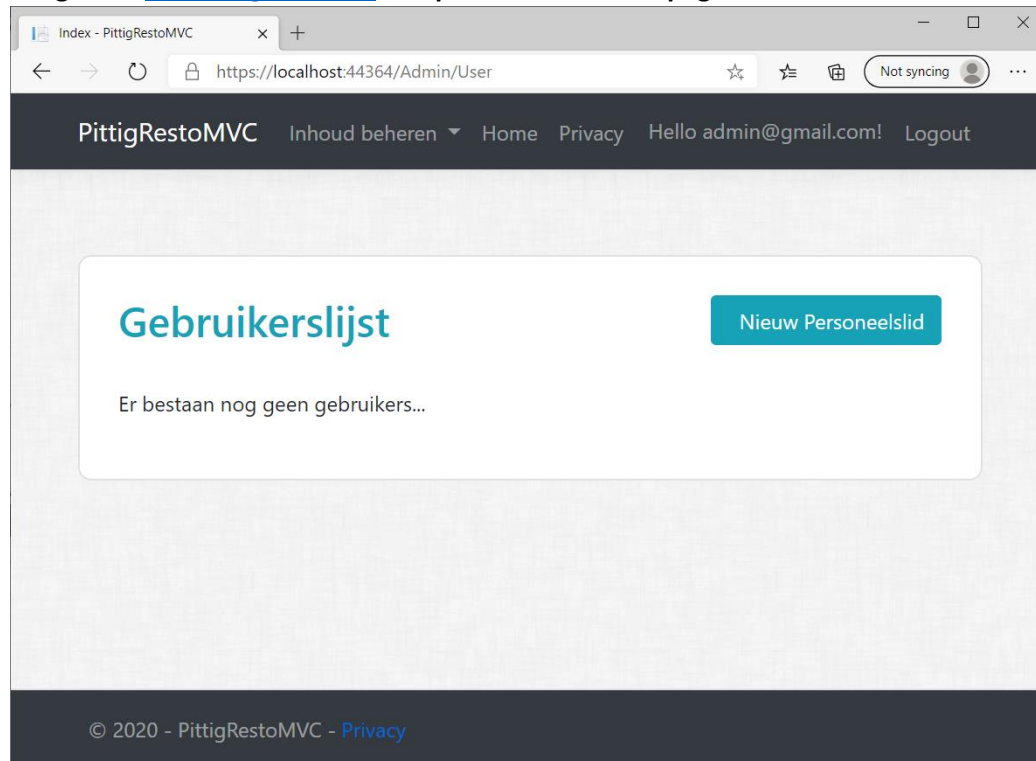
        //await _emailSender.SendEmailAsync(Input.Email, "Confirm your
email",
        //    $"Please confirm your account by <a
href='{HtmlEncoder.Default.Encode(callbackUrl)}'>clicking here</a>.");
    }
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError(string.Empty, error.Description);
    }
}

// If we got this far, something failed, redisplay form
return Page();
}
}
}

```

4.2 Test de app uit:

1.log in als admin@gmail.com en open de Personeels-pagina:



2. Klik op de button “Nieuw personeelslid” en voeg een paar gebruikers toe met verschillende rol bv een user met Manager-rol en user met Balie-rol:

Registreer - PittigRestoMVC

https://localhost:44364/Identity/Account/Register

PittigRestoMVC Inhoud beheren Home Privacy Hello admin@gmail.com! Logout

Een nieuwe account aanmaken

Naam: Helena Coppieters
 Email: hcoppieters@hotmail.com
 TelNr: 12345647
 Straat: Street
 Gemeente: Gent
 Provincie: O-VL
 Postcode: 8500
 Wachtwoord:
 Bevestig Wachtwoord:

☐ Keuken ☐ Balie ☒ Manager

Registreer

Index - PittigRestoMVC

https://localhost:44364/Admin/User

PittigRestoMVC Inhoud beheren Home Privacy Hello admin@gmail.com! Logout

Gebruikerslijst

Nieuw Personeelslid

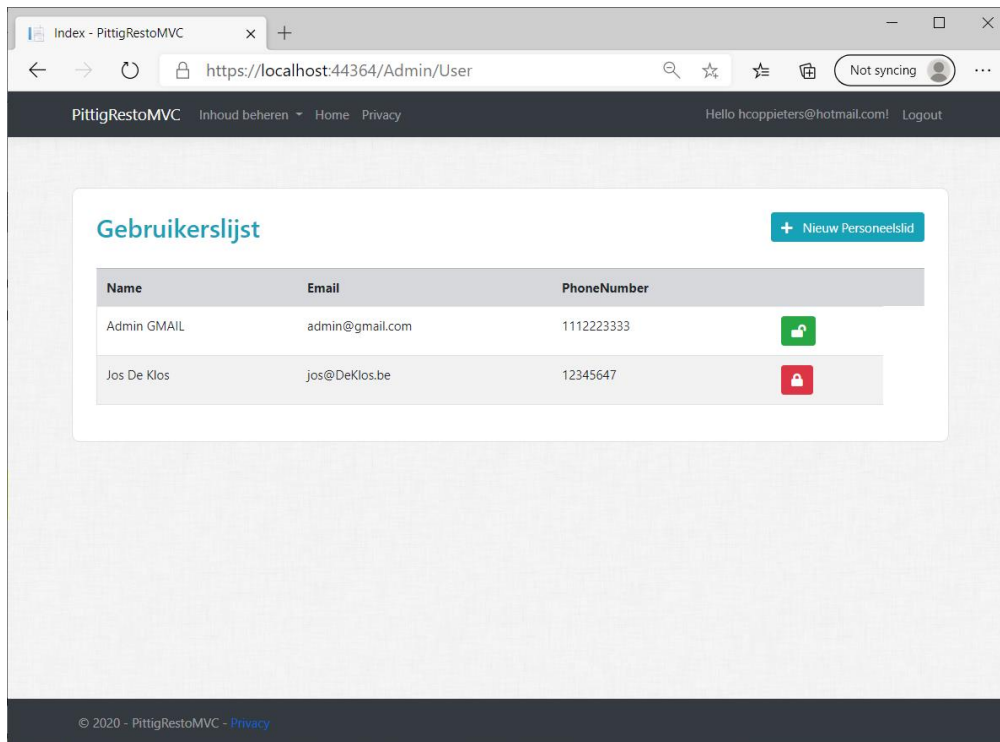
Name	Email	PhoneNumber	
Helena Coppieters	hcoppieters@hotmail.com	12345647	
Jos De Klos	jos@DeKlos.be	12345647	

© 2020 - PittigRestoMVC - Privacy

3. De lock-icoontjes, kunnen bv van de website font-awesome worden gehaald:

Open Views/Shared/_Layout.cshtml en voeg hiervoor de volgende lijn toe aan de head:

```
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - PittigRestoMVC</title>
  <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3B1XeVIU"
crossorigin="anonymous">
  <link rel="stylesheet" href="~/css/site.css" />
</head>
```



4. Controleer in de database of de users en hun rollen correct zijn aangemaakt:

SQLQuery3.sql - (L...3CPIU4L\hcoppp (76))* SQLQuery2.sql - (L...3CPIU4L\hcoppp (80)) SQLQuery1.sql - (L...3CPIU4L\hcoppp (66))

```

/***** Script for SelectTopNRows command from SSMS *****/

select * FROM [AspNetUsers]
select * FROM [AspNetRoles]
SELECT * FROM [AspNetUserRoles]

```

175 %

Results Messages

	Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	PasswordHash
1	45bf14d0-26e5-4824-8733-712f6aee4e95	admin@gmail.com	ADMIN@GMAIL.COM	admin@gmail.com	ADMIN@GMAIL.COM	1	AQAAAAEAAACcQ/
2	af0940fe-4500-4153-a85f-649d9dccc0301	hcoppieters@hotmail.com	HCOPIETERS@HOTMAIL.COM	hcoppieters@hotmail.com	HCOPIETERS@HOTMAIL.COM	0	AQAAAAEAAACcQ/
3	cdfa87a6-ad42-4d34-8393-7219e365fc8c	jos@DeKlos.be	JOS@DEKLOS.BE	jos@DeKlos.be	JOS@DEKLOS.BE	0	AQAAAAEAAACcQ/

	Id	Name	NormalizedName	ConcurrencyStamp
1	16effe64-feb2-421b-9133-b9d066832ba9	Manager	MANAGER	96792067-b3da-45f2-8c34-fa073c23debb
2	28f3c2b6-3724-442e-8180-7b4f3992717f	Customer	CUSTOMER	6eb4b63f-401d-46a1-918f-2a03a8e17ff9
3	a9fff489-0a23-4766-bd03-5c9cdfc480a0	FrontDesk	FRONTDESK	239d697b-df9c-4bcd-80db-1248af4cf59
4	b1740646-7928-428d-9e19-9269ec65a4c1	Kitchen	KITCHEN	d06402f0-6183-42ba-a3b8-0d291b787993

	UserId	RoleId
1	45bf14d0-26e5-4824-8733-712f6aee4e95	16effe64-feb2-421b-9133-b9d066832ba9
2	af0940fe-4500-4153-a85f-649d9dccc0301	16effe64-feb2-421b-9133-b9d066832ba9
3	cdfa87a6-ad42-4d34-8393-7219e365fc8c	a9fff489-0a23-4766-bd03-5c9cdfc480a0

5 Emails Sturen

5.1 Email Service toevoegen en registreren als Singleton

1. Maak een nieuwe folder 'Services' aan en maak hierin een class EmailOptions

```

namespace PittigRestoMVC.Service
{
    public class EmailOptions
    {
        public string SendGridKey { get; set; }
    }
}

```

2. Maak onder de folder 'Services' een class EmailSender (afgeleide class van IEmailSender)

(Installeer de NuGet Package SendGrid)

```

using Microsoft.AspNetCore.Identity.UI.Services;
using Microsoft.Extensions.Options;
using SendGrid;
using SendGrid.Helpers.Mail;
using System.Threading.Tasks;

```

```

namespace PittigRestoMVC.Service
{

```

```

public class EmailSender : IEmailSender
{
    public EmailOptions Options { get; set; }

    public EmailSender(IOptions<EmailOptions> emailOptions)
    {
        Options = emailOptions.Value;
    }

    public Task SendEmailAsync(string email, string subject, string message)
    {
        return Execute(Options.SendGridKey, subject, message, email);
    }

    private Task Execute(string sendGridKey, string subject, string message,
string email)
    {
        var client = new SendGridClient(sendGridKey);
        var msg = new SendGridMessage()
        {
            From = new EmailAddress("admin@Pittig.com", "Pittig Restaurant"),
            Subject = subject,
            PlainTextContent = message,
            HtmlContent = message
        };
        msg.AddTo(new EmailAddress(email));
        return client.SendEmailAsync(msg);
    }
}

```

3. Open Startup.cs en registreer in de methode ConfigureServices de EmailSender als Singleton

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));

    services.AddIdentity<IdentityUser, IdentityRole>()
        .AddDefaultTokenProviders()
        .AddEntityFrameworkStores<ApplicationDbContext>();

    services.AddScoped<IDbInitializer, DbInitializer>();
    services.AddSingleton<IEmailSender, EmailSender>();
    services.Configure<EmailOptions>(Configuration);
    services.AddControllersWithViews();
    services.AddRazorPages().AddRazorRuntimeCompilation();
}

```

4. Creatie van API Key voor SendGrid

Maak een Account aan op <https://sendgrid.com/>

Klik onder Settings op API Keys

Vul een Naam in (bv Syntra-West), selecteer de nodige permissies en Creëer de API Waarde

6 Referenties

<https://aspnetcore.readthedocs.io/en/latest/security/authentication/identity.html>

<https://www.stevejgordon.co.uk/asp-net-core-identity-token-providers>

<https://medium.com/@xsoheilalizadeh/asp-net-core-identity-deep-dive-stores-e0e54291b51d>

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/acconconfirm?view=aspnetcore-3.1&tabs=visual-studio>

<https://fontawesome.com/6?next=%2Fv5.5.0%2Ficons%2Flock-open>

<http://bekenty.com/send-e-mail-in-asp-net-core-with-sendgrid/>

<https://hostlaunch.io/docs/how-to-get-a-sendgrid-api-key/>

<https://github.com/dotnet/AspNetCore.Docs/issues/7954>

<https://www.codeproject.com/Articles/1276970/NET-Core-Razor-Page-Email-Form-using-SendGrid-and>

https://sendgrid.com/docs/API_Reference/index.html