**Section02 Email Verification**

**Notes: -**

**\*we will use Papercut as SMTP server locally on port 25 (download online)**

**Steps:-**

**1-we install the nugget package called NETCore.MailKit**

**2-on the startup.cs we add the following code**

**(to add the mailkit middleware and to apply the emai**

**using IdentityExample.Data;**

**using Microsoft.AspNetCore.Builder;**

**using Microsoft.AspNetCore.Hosting;**

**using Microsoft.AspNetCore.Identity;**

**using Microsoft.EntityFrameworkCore;**

**using Microsoft.Extensions.Configuration;**

**using Microsoft.Extensions.DependencyInjection;**

**using Microsoft.Extensions.Hosting;**

**using NETCore.MailKit.Extensions;**

**using NETCore.MailKit.Infrastructure.Internal;**

**namespace IdentityExample{**

**public class Startup{**

**public Startup(IConfiguration configuration){Configuration = configuration;}**

**public IConfiguration Configuration { get; }**

**// This method gets called by the runtime. Use this method to add services to the container.**

**public void ConfigureServices(IServiceCollection services){**

**//Database Layer**

**//initialize part with the database on the memory (communication with the database)**

**services.AddDbContext<AppDbContext>(config =>{config.UseInMemoryDatabase("Memory");});**

**//Identity Layer**

**//we inject identity repostiories that is interface of method collection to interact with the Identity**

**//AddDefaultTokenProviders: is the default Token providers that used to generate Tokens when**

**//we set the configuration of the cookie**

**//reset password , change emial , etc..**

**services.AddIdentity<IdentityUser, IdentityRole>(config =>{**

**config.Password.RequiredLength = 4;**

**config.Password.RequireDigit = false;**

**config.Password.RequireNonAlphanumeric = false;**

**config.Password.RequireUppercase = false;**

**//to indicate that the email is required to login / register**

**//so the previous login / register on the IdentityExample not working on our sample**

**config.SignIn.RequireConfirmedEmail = true;})**

**.AddEntityFrameworkStores<AppDbContext>()**

**.AddDefaultTokenProviders();**

**//Identity with Cookie Configuration**

**//we also configure the cookie as the previous section of basic authentication**

**//the cookie name , the login path when there is no cookie found**

**services.ConfigureApplicationCookie(config =>{**

**config.Cookie.Name = "Identity.Cookie";**

**config.LoginPath = "/Home/Login";});**

**//we get the configuration of the Email section and use it inside the MailKit middleware as below**

**services.AddMailKit(config => {config.UseMailKit(Configuration.GetSection("Email").Get<MailKitOptions>());});**

**services.AddControllersWithViews();}**

**// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.**

**public void Configure(IApplicationBuilder app, IWebHostEnvironment env){**

**if (env.IsDevelopment()){app.UseDeveloperExceptionPage();}**

**//apply the routing middleware**

**app.UseRouting();**

**//must set the authentication before authorization**

**//means : how are you?**

**app.UseAuthentication();**

**//means : are you allowed?**

**app.UseAuthorization();**

**//we use map default controller route**

**app.UseEndpoints(endpoints =>{endpoints.MapDefaultControllerRoute();});}}}**