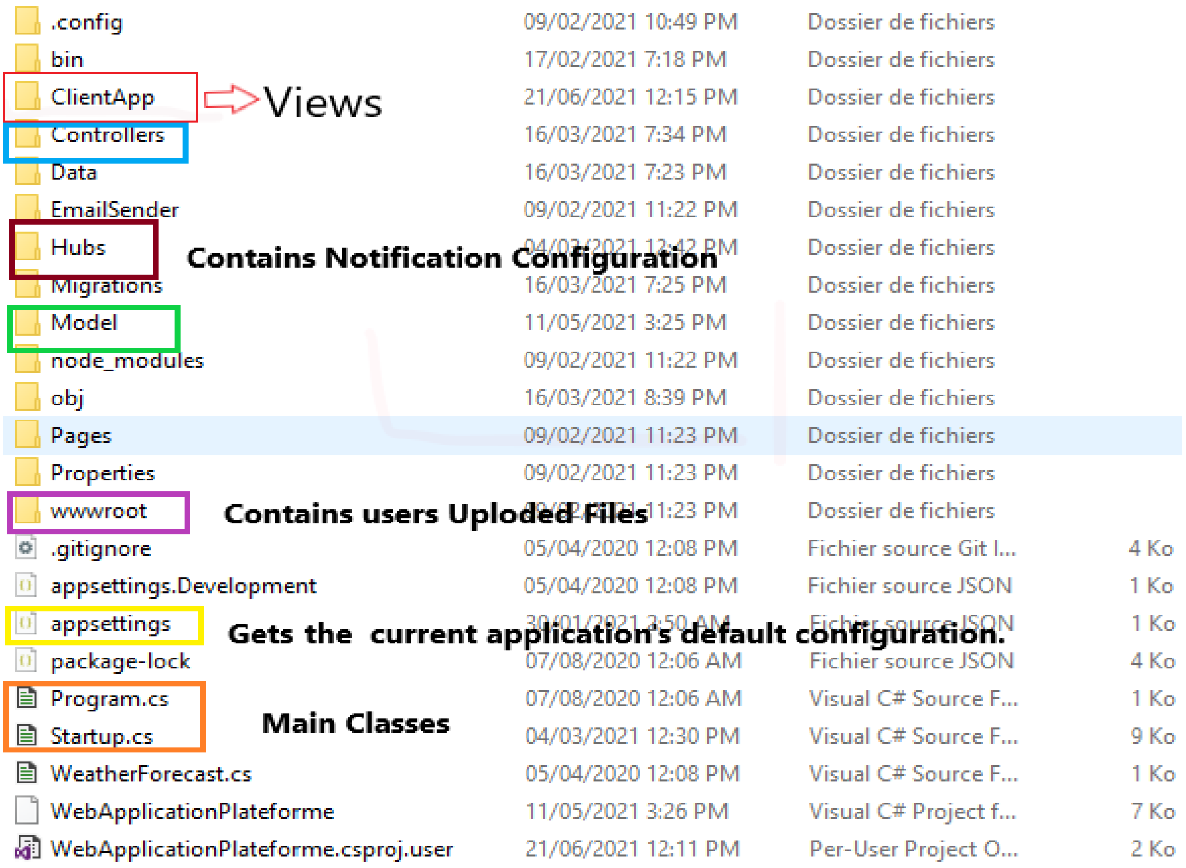
Application Architecture

Our application has an MVC architecture, stands for Model, View, and Controller, it separates our application into three components Model, View, and Controller.



.config	09/02/2021 10:49 PM	Dossier de fichiers	
bin	17/02/2021 7:18 PM	Dossier de fichiers	
ClientApp	21/06/2021 12:15 PM	Dossier de fichiers	
Controllers	16/03/2021 7:34 PM	Dossier de fichiers	
Data	16/03/2021 7:23 PM	Dossier de fichiers	
EmailSender	09/02/2021 11:22 PM	Dossier de fichiers	
Hubs	04/03/2021 12:42 PM	Dossier de fichiers	
Migrations	16/03/2021 7:25 PM	Dossier de fichiers	
Model	11/05/2021 3:25 PM	Dossier de fichiers	
node_modules	09/02/2021 11:22 PM	Dossier de fichiers	
obj	16/03/2021 8:39 PM	Dossier de fichiers	
Pages	09/02/2021 11:23 PM	Dossier de fichiers	
Properties	09/02/2021 11:23 PM	Dossier de fichiers	
wwwroot	09/02/2021 11:23 PM	Dossier de fichiers	
.gitignore	05/04/2020 12:08 PM	Fichier source Git I...	4 Ko
appsettings.Development	05/04/2020 12:08 PM	Fichier source JSON	1 Ko
appsettings	30/01/2021 3:50 AM	Fichier source JSON	1 Ko
package-lock	07/08/2020 12:06 AM	Fichier source JSON	4 Ko
Program.cs	07/08/2020 12:06 AM	Visual C# Source F...	1 Ko
Startup.cs	04/03/2021 12:30 PM	Visual C# Source F...	9 Ko
WeatherForecast.cs	05/04/2020 12:08 PM	Visual C# Source F...	1 Ko
WebApplicationPlateforme	11/05/2021 3:26 PM	Visual C# Project f...	7 Ko
WebApplicationPlateforme.csproj.user	21/06/2021 12:11 PM	Per-User Project O...	2 Ko

Views

Contains Notification Configuration

Contains users Uploded Files

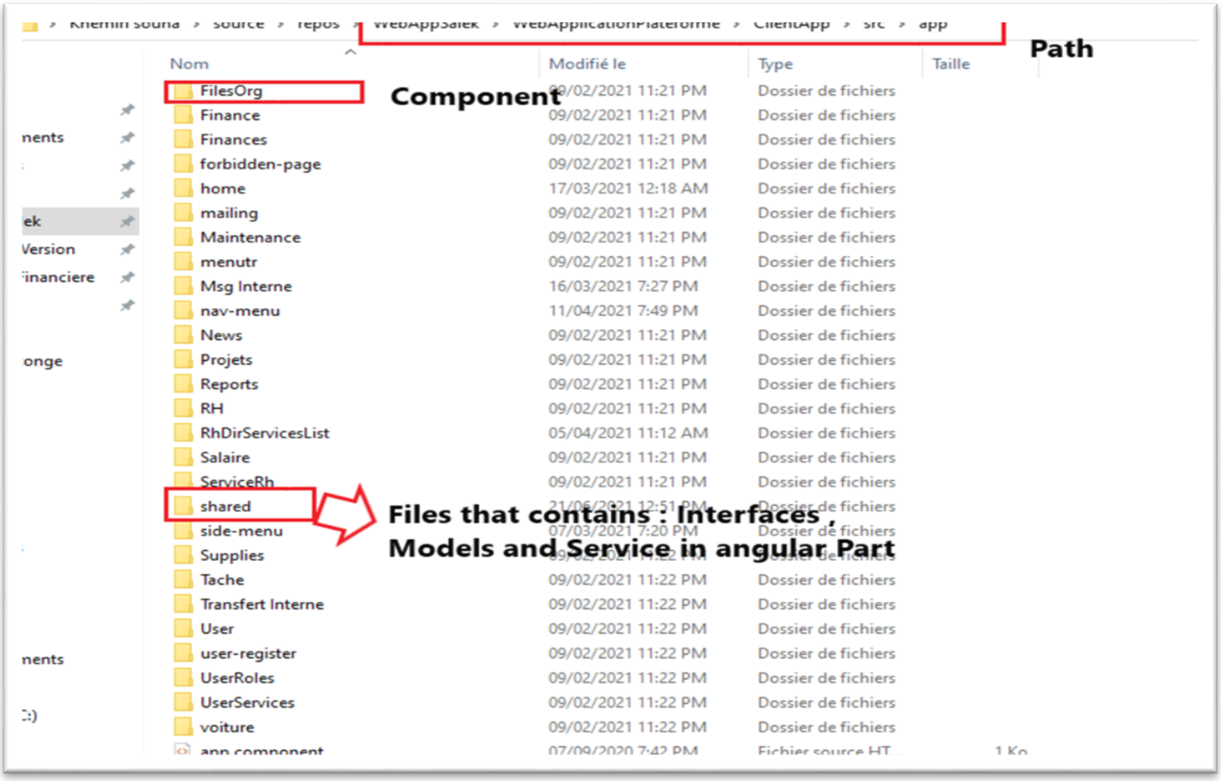
Gets the current application's default configuration.

Main Classes

Frontend Part   :

1.   Views :

It's a user interface, it displays model data to the user and also enables them to modify. It's the Frontend part with angular which contains components with HTML, CSS and TS files.


Khemin souha › source › repos › WebAppSalek › WebApplicationPlateforme › ClientApp › src › app
nents
ek
Version
inanciere
onge
nents
:)

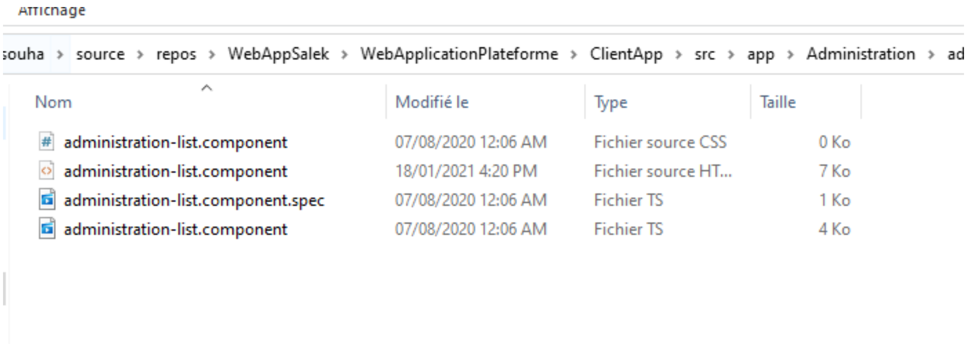
Path

Component

Files that contains : Interfaces , Models and Service in angular Part

Nom	Modifié le	Type	Taille
FilesOrg	09/02/2021 11:21 PM	Dossier de fichiers	
Finance	09/02/2021 11:21 PM	Dossier de fichiers	
Finances	09/02/2021 11:21 PM	Dossier de fichiers	
forbidden-page	09/02/2021 11:21 PM	Dossier de fichiers	
home	17/03/2021 12:18 AM	Dossier de fichiers	
mailing	09/02/2021 11:21 PM	Dossier de fichiers	
Maintenance	09/02/2021 11:21 PM	Dossier de fichiers	
menutr	09/02/2021 11:21 PM	Dossier de fichiers	
Msg Interne	16/03/2021 7:27 PM	Dossier de fichiers	
nav-menu	11/04/2021 7:49 PM	Dossier de fichiers	
News	09/02/2021 11:21 PM	Dossier de fichiers	
Projets	09/02/2021 11:21 PM	Dossier de fichiers	
Reports	09/02/2021 11:21 PM	Dossier de fichiers	
RH	09/02/2021 11:21 PM	Dossier de fichiers	
RhDirServicesList	05/04/2021 11:12 AM	Dossier de fichiers	
Salaire	09/02/2021 11:21 PM	Dossier de fichiers	
ServiceRh	09/02/2021 11:21 PM	Dossier de fichiers	
shared	21/06/2021 12:51 PM	Dossier de fichiers	
side-menu	07/03/2021 7:20 PM	Dossier de fichiers	
Supplies	09/02/2021 11:21 PM	Dossier de fichiers	
Tache	09/02/2021 11:22 PM	Dossier de fichiers	
Transfert Interne	09/02/2021 11:22 PM	Dossier de fichiers	
User	09/02/2021 11:22 PM	Dossier de fichiers	
user-register	09/02/2021 11:22 PM	Dossier de fichiers	
UserRoles	09/02/2021 11:22 PM	Dossier de fichiers	
UserServices	09/02/2021 11:22 PM	Dossier de fichiers	
voiture	09/02/2021 11:22 PM	Dossier de fichiers	
app.component	07/09/2020 7:42 PM	Fichier source HT...	1 Ko

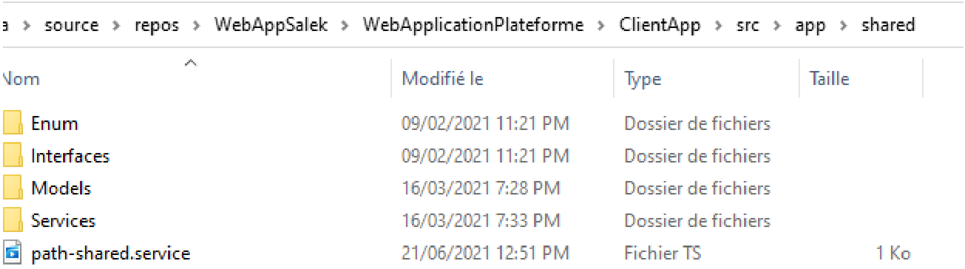
1.1 : Component :


Affichage
souha › source › repos › WebAppSalek › WebApplicationPlateforme › ClientApp › src › app › Administration › ad

Nom	Modifié le	Type	Taille
administration-list.component	07/08/2020 12:06 AM	Fichier source CSS	0 Ko
administration-list.component	18/01/2021 4:20 PM	Fichier source HT...	7 Ko
administration-list.component.spec	07/08/2020 12:06 AM	Fichier TS	1 Ko
administration-list.component	07/08/2020 12:06 AM	Fichier TS	4 Ko

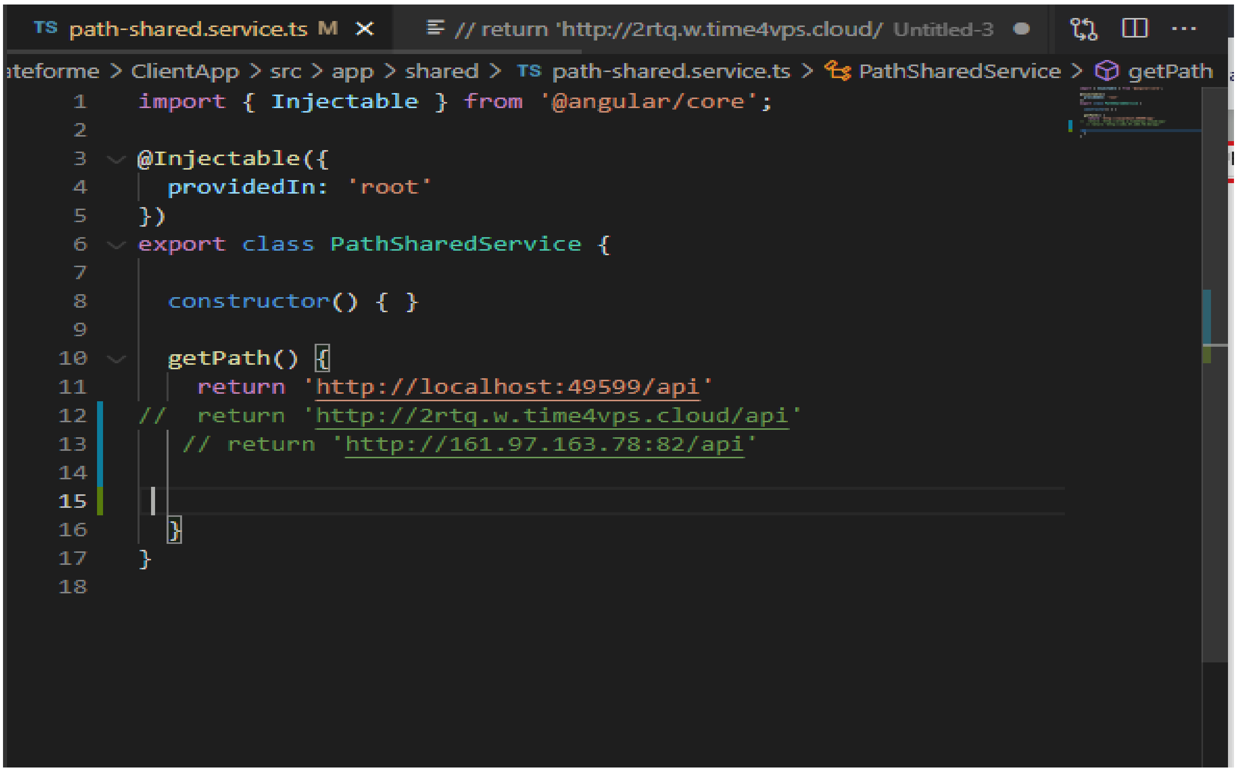
1.2 : Shared :

It Contains Interfaces, Models , and Services .


a › source › repos › WebAppSalek › WebApplicationPlateforme › ClientApp › src › app › shared

Nom	Modifié le	Type	Taille
Enum	09/02/2021 11:21 PM	Dossier de fichiers	
Interfaces	09/02/2021 11:21 PM	Dossier de fichiers	
Models	16/03/2021 7:28 PM	Dossier de fichiers	
Services	16/03/2021 7:33 PM	Dossier de fichiers	
path-shared.service	21/06/2021 12:51 PM	Fichier TS	1 Ko

Path file is the one who is responsible for defining the web api path.



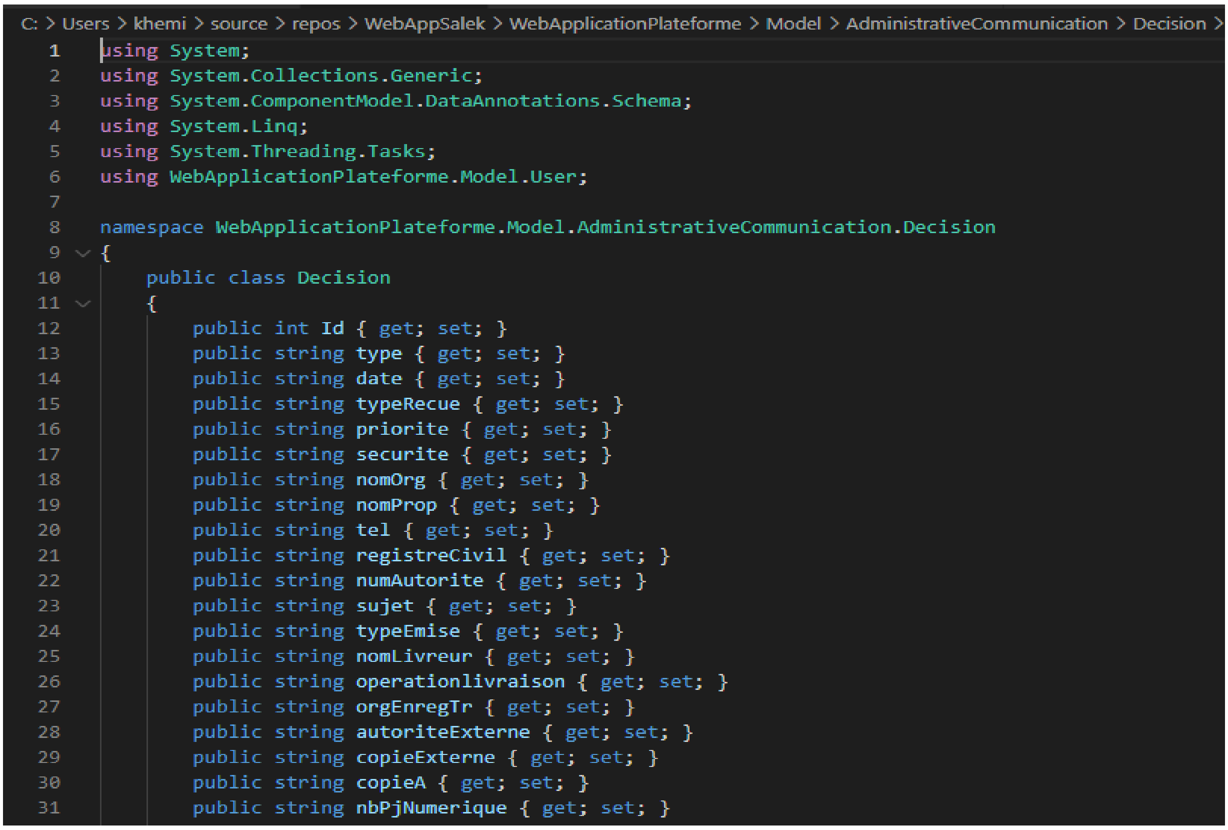
TS path-shared.service.ts M ×
// return 'http://2rtq.w.time4vps.cloud/ Untitled-3
ıteforme > ClientApp > src > app > shared > TS path-shared.service.ts > PathSharedService > getPath

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class PathSharedService {

  constructor() { }

  getPath() {
    return 'http://localhost:49599/api'
//  return 'http://2rtq.w.time4vps.cloud/api'
   // return 'http://161.97.163.78:82/api'


  }
}

```

Backend Part :

2. Models:

Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.



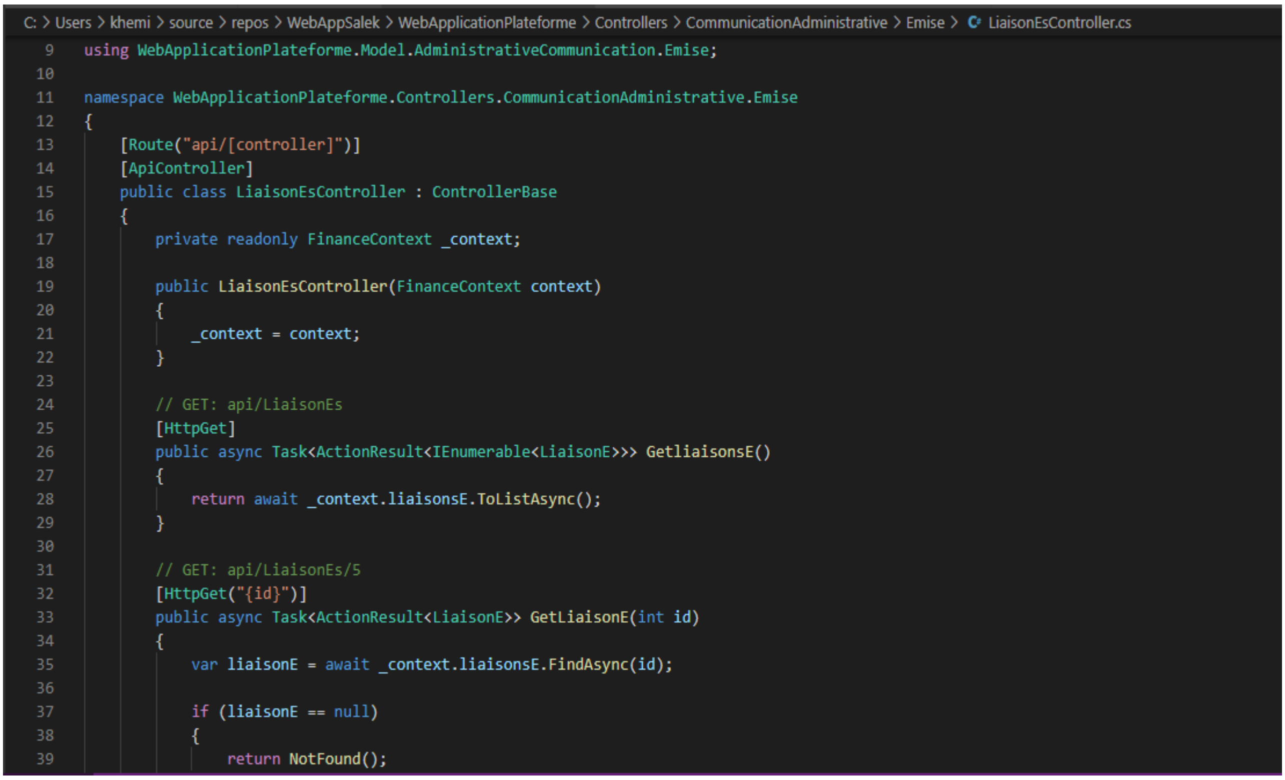
C: > Users > khemi > source > repos > WebAppSalek > WebApplicationPlateforme > Model > AdministrativeCommunication > Decision >

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;
using WebApplicationPlateforme.Model.User;

namespace WebApplicationPlateforme.Model.AdministrativeCommunication.Decision
{
    public class Decision
    {
        public int Id { get; set; }
        public string type { get; set; }
        public string date { get; set; }
        public string typeRecue { get; set; }
        public string priorite { get; set; }
        public string securite { get; set; }
        public string nomOrg { get; set; }
        public string nomProp { get; set; }
        public string tel { get; set; }
        public string registreCivil { get; set; }
        public string numAutorite { get; set; }
        public string sujet { get; set; }
        public string typeEmise { get; set; }
        public string nomLivreur { get; set; }
        public string operationlivraison { get; set; }
        public string orgEnregTr { get; set; }
        public string autoriteExterne { get; set; }
        public string copieExterne { get; set; }
        public string copieA { get; set; }
        public string nbPjNumerique { get; set; }
```

3. Controllers :

The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

It Contains the Get, Put , Post , Delete Methods and provides an web api path to connect users requests.



C: > Users > khemi > source > repos > WebAppSalek > WebApplicationPlateforme > Controllers > CommunicationAdministrative > Emise > LiaisonEsController.cs

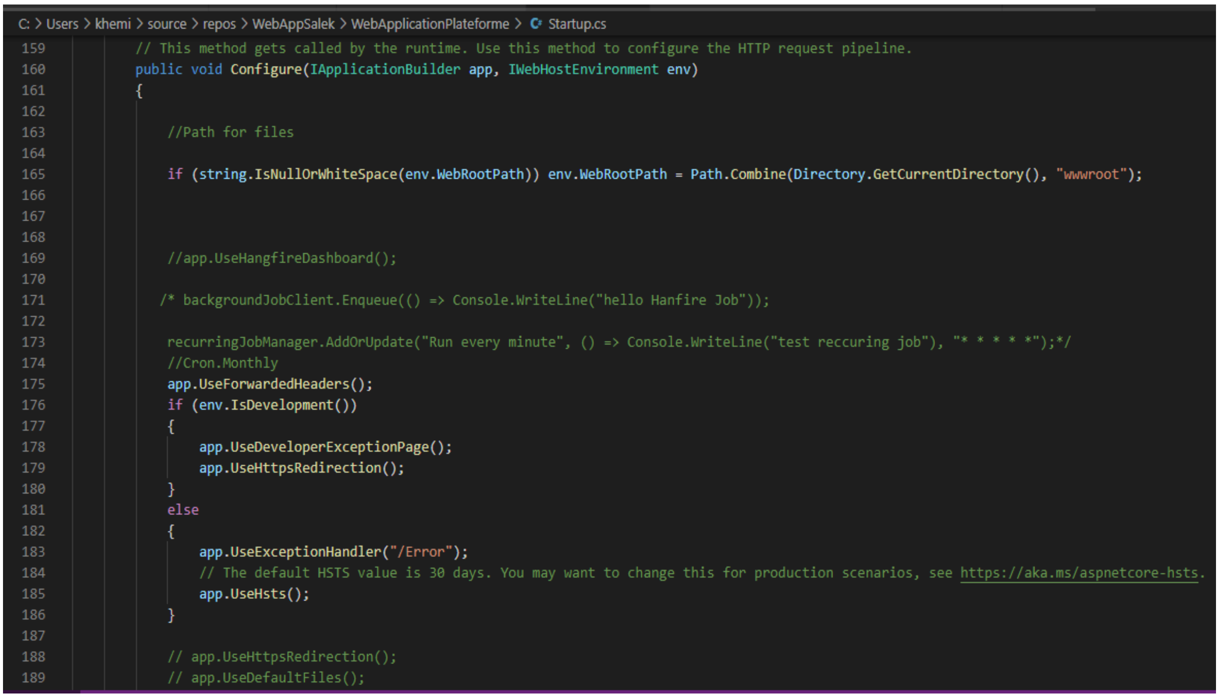
```
using WebApplicationPlateforme.Model.AdministrativeCommunication.Emise;

namespace WebApplicationPlateforme.Controllers.CommunicationAdministrative.Emise
{
    [Route("api/[controller]")]
    [ApiController]
    public class LiaisonEsController : ControllerBase
    {
        private readonly FinanceContext _context;

        public LiaisonEsController(FinanceContext context)
        {
            _context = context;
        }

        // GET: api/LiaisonEs
        [HttpGet]
        public async Task<ActionResult<IEnumerable<LiaisonE>>> GetliaisonsE()
        {
            return await _context.liaisonsE.ToListAsync();
        }

        // GET: api/LiaisonEs/5
        [HttpGet("{id}")]
        public async Task<ActionResult<LiaisonE>> GetLiaisonE(int id)
        {
            var liaisonE = await _context.liaisonsE.FindAsync(id);

            if (liaisonE == null)
            {
                return NotFound();
```

4. Startup:

The Startup class provides the entry point for an application, and is required for all applications.The place where we set up configuration and winnig up services.

Our Startup class defines a Configure method, and also define a ConfigureService method, which will be called when the application is started.

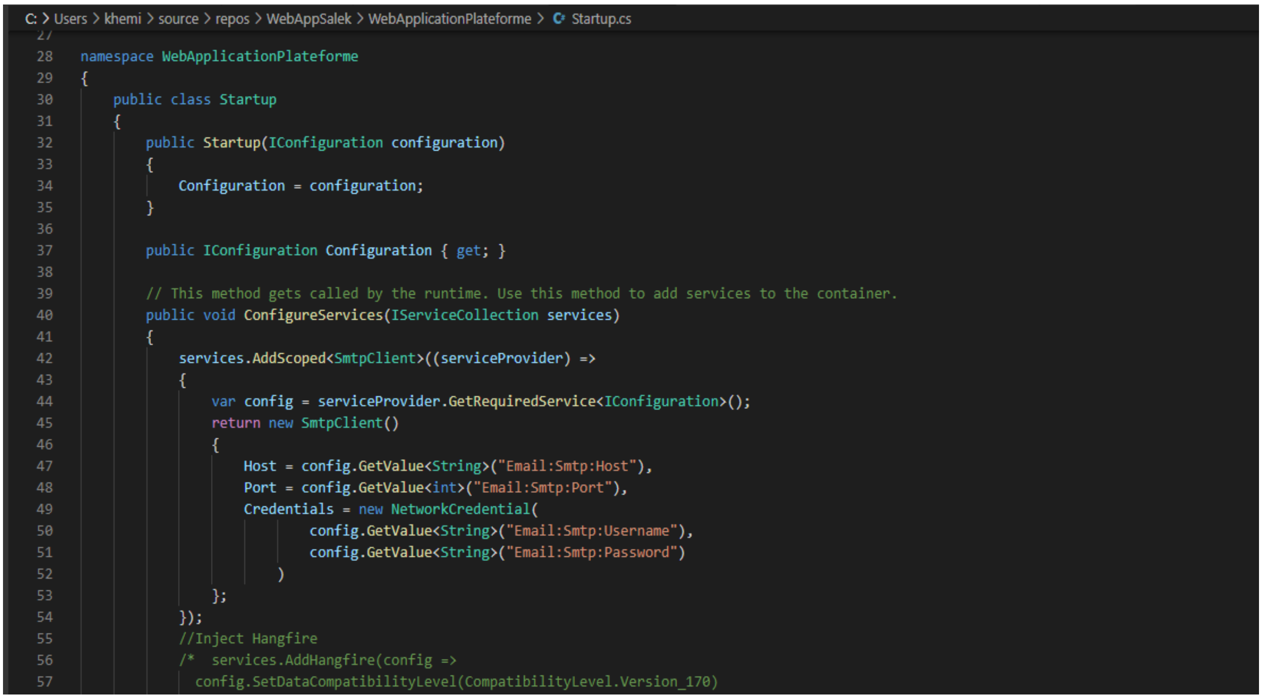
Configure Method :



C: > Users > khemi > source > repos > WebAppSalek > WebApplicationPlateforme > Startup.cs

```
        // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {

            //Path for files

            if (string.IsNullOrWhiteSpace(env.WebRootPath)) env.WebRootPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot");



            //app.UseHangfireDashboard();

           /* backgroundJobClient.Enqueue(() => Console.WriteLine("hello Hanfire Job"));

            recurringJobManager.AddOrUpdate("Run every minute", () => Console.WriteLine("test reccuring job"), "* * * * *");*/
            //Cron.Monthly
            app.UseForwardedHeaders();
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
                app.UseHttpsRedirection();
            }
            else
            {
                app.UseExceptionHandler("/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.UseDefaultFiles();
```

ConfigureServices Method :

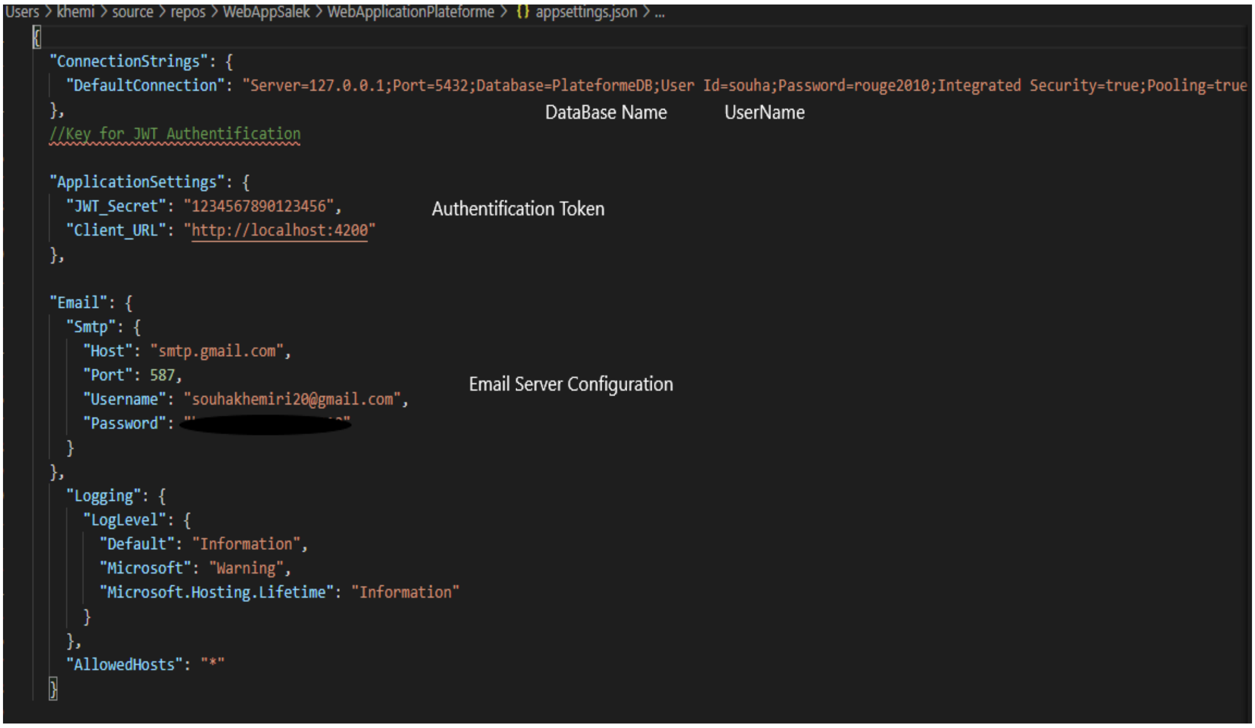


C: > Users > khemi > source > repos > WebAppSalek > WebApplicationPlateforme > Startup.cs

```
namespace WebApplicationPlateforme
{
    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)
        {
            services.AddScoped<SmtpClient>((serviceProvider) =>
            {
                var config = serviceProvider.GetRequiredService<IConfiguration>();
                return new SmtpClient()
                {
                    Host = config.GetValue<String>("Email:Smtp:Host"),
                    Port = config.GetValue<int>("Email:Smtp:Port"),
                    Credentials = new NetworkCredential(
                            config.GetValue<String>("Email:Smtp:Username"),
                            config.GetValue<String>("Email:Smtp:Password")
                        )
                };
            });
            //Inject Hangfire
            /*  services.AddHangfire(config =>
              config.SetDataCompatibilityLevel(CompatibilityLevel.Version_170)
```

5. AppSettings:

Gets current application's default configuration.



Users > khemi > source > repos > WebAppSalek > WebApplicationPlateforme > {} appsettings.json > ...
DataBase Name
UserName
Authentification Token
Email Server Configuration

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=127.0.0.1;Port=5432;Database=PlateformeDB;User Id=souha;Password=rouge2010;Integrated Security=true;Pooling=true
  },
  //Key for JWT Authentification

  "ApplicationSettings": {
    "JWT_Secret": "1234567890123456",
    "Client_URL": "http://localhost:4200"
  },

  "Email": {
    "Smtp": {
      "Host": "smtp.gmail.com",
      "Port": 587,
      "Username": "souhakhemiri20@gmail.com",
      "Password": [redacted]
    }
  },
    "Logging": {
      "LogLevel": {
        "Default": "Information",
        "Microsoft": "Warning",
        "Microsoft.Hosting.Lifetime": "Information"
      }
    },
    "AllowedHosts": "*"
  }
```