

Some problems of the KPI system

1. Do not use the "_Repositories" construct

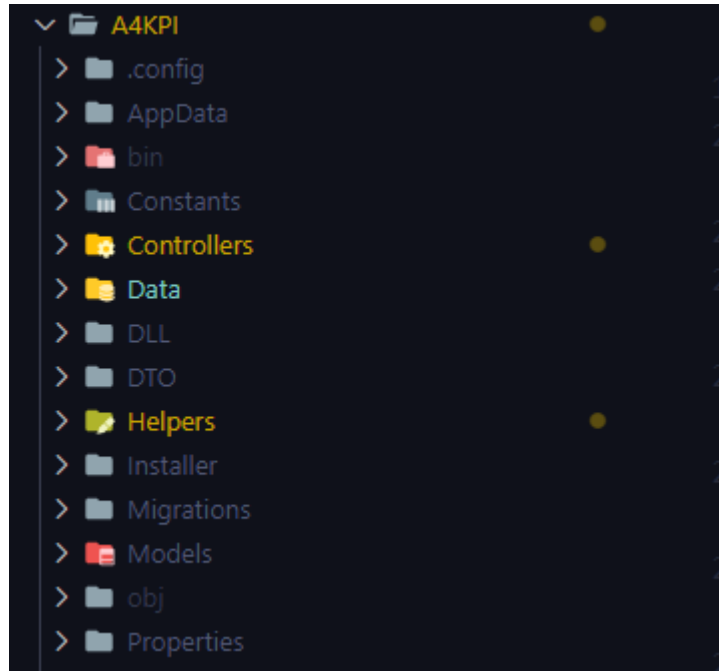


Photo 1: directory of system KPI

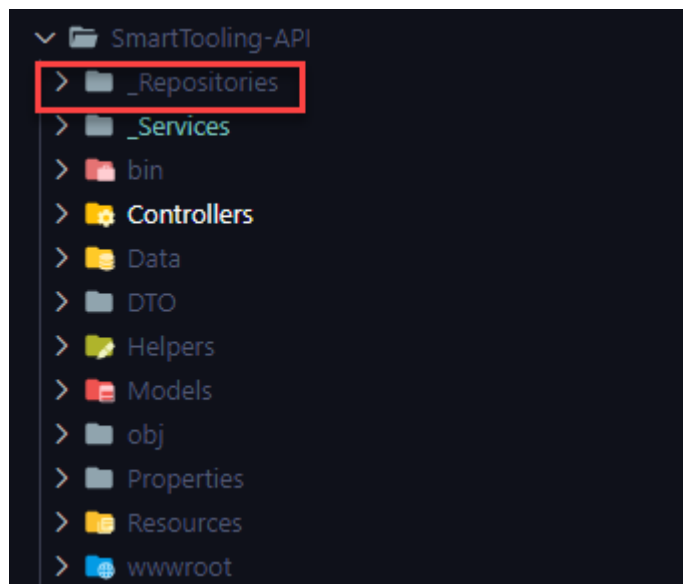


Photo 2: directory of project SD3 team

2. Interface and service are in the same file

```
Henry, a day ago | 1 author (Henry)
namespace A4KPI.Services
{
    6 references | Henry, a day ago | 1 author (Henry)
    public interface IAccountService : IServiceBase<Account, AccountDto>
    {
        1 reference
        Task<OperationResult> LockAsync(int id);
        1 reference
        Task<AccountDto> GetByUsername(string username);
        1 reference
        Task<OperationResult> ChangePasswordAsync(ChangePasswordRequest request);
        0 references
        Task<object> ChangePasswordAsync2(ChangePasswordRequest request);
        1 reference
        Task<object> GetAccounts();
    }
    1 reference | Henry, a day ago | 1 author (Henry)
    public class AccountService : ServiceBase<Account, AccountDto>, IAccountService
    {
        13 references
        private readonly IRepositoryBase<Account> _repo;
        4 references
        private readonly IRepositoryBase<OC> _repoOc;
        3 references
        private readonly IRepositoryBase<UserRole> _repoUserRole;
        3 references
        private readonly IRepositoryBase<Role> _repoRole;
        5 references
        private readonly IRepositoryBase<AccountGroupAccount> _repoAccountGroupAccount;
        9 references
        private readonly IUnitOfWork _unitOfWork;
    }
}
```

3. Use only one Repository (RepositoryBase)

```
Henry, a day ago | 1 author (Henry)
namespace A4KPI.Services
{
    4 references | Henry, a day ago | 1 author (Henry)
    public interface IAccountGroupAccountService: IServiceBase<AccountGroupAccount, AccountGroupAccountDto>
    {
    }
    1 reference | Henry, a day ago | 1 author (Henry)
    public class AccountGroupAccountService : ServiceBase<AccountGroupAccount, AccountGroupAccountDto>, IAccountGroupAccountService
    {
        1 reference
        private readonly IRepositoryBase<AccountGroupAccount> _repo;
        1 reference
        private readonly IUnitOfWork _unitOfWork;
        1 reference
        private readonly IMapper _mapper;
        1 reference
        private readonly MapperConfiguration _configMapper;
    }
    0 references
    public AccountGroupAccountService(
        IRepositoryBase<AccountGroupAccount> repo,
        IUnitOfWork unitOfWork,
        IMapper mapper,
        MapperConfiguration configMapper
    )
    {
        : base(repo, unitOfWork, mapper, configMapper)
        _repo = repo;
        _unitOfWork = unitOfWork;
        _mapper = mapper;
        _configMapper = configMapper;
    }
}
```

4. SD3 team don't use UnitOfWork

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace A4KPI.Data
{
    84 references
    public interface IUnitOfWork
    {
        76 references
        Task<int> SaveChangeAsync();
    }
    2 references
    public class UnitOfWork : IUnitOfWork
    {
        private DbFactory _dbFactory;

        0 references
        public UnitOfWork(DbFactory dbFactory)
        {
            _dbFactory = dbFactory;
        }

        76 references
        public Task<int> SaveChangeAsync()
        {
            return _dbFactory.DbContext.SaveChangesAsync();
        }
    }
}
```

5. SD3 team do not use DbFactory

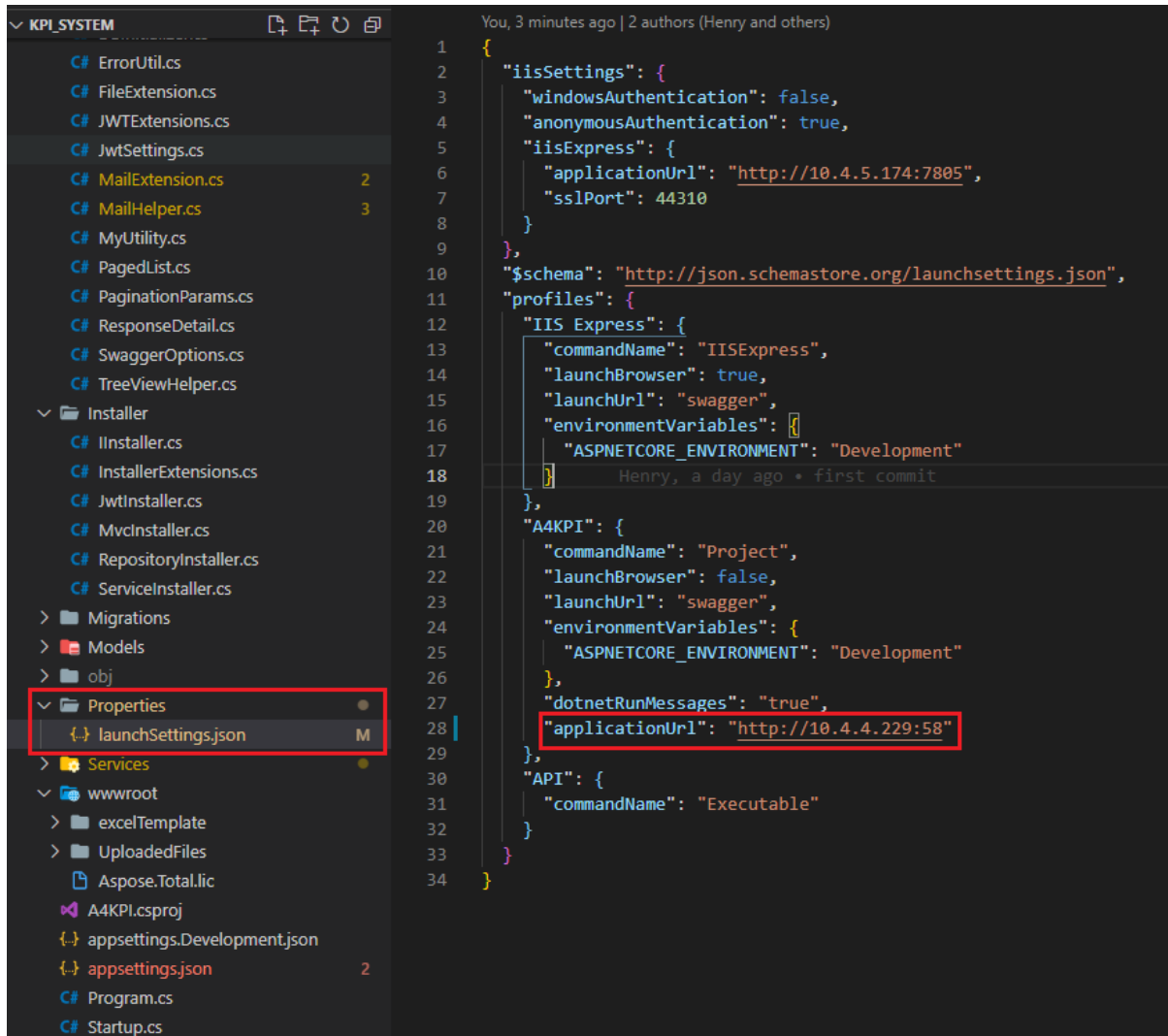
```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace A4KPI.Data
{
    4 references
    public class DbFactory : IDisposable
    {
        private bool _disposed;
        private Func<DataContext> _instanceFunc;
        private DbContext _dbContext;
        1 reference
        public DbContext DbContext => _dbContext ?? (_dbContext = _instanceFunc.Invoke());

        0 references
        public DbFactory(Func<DataContext> dbContextFactory)
        {
            _instanceFunc = dbContextFactory;
        }

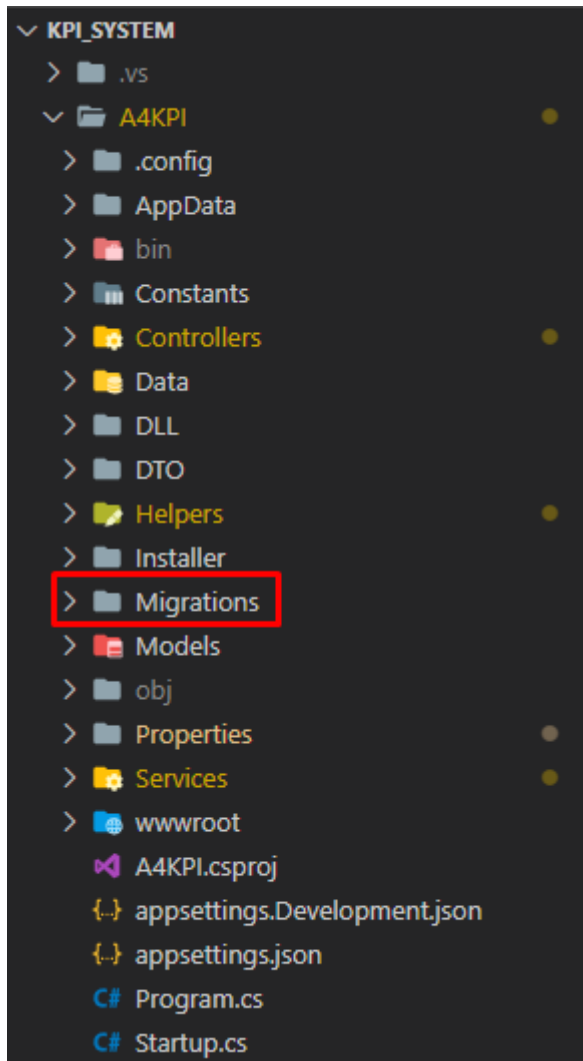
        0 references
        public void Dispose()
        {
            if (!_disposed && _dbContext != null)
            {
                _disposed = true;
                _dbContext.Dispose();
            }
        }
    }
}
```

6. Config launchSettings.json according to ip computer to start project in local, Database official setup in local computer



```
1 {
2   "iisSettings": {
3     "windowsAuthentication": false,
4     "anonymousAuthentication": true,
5     "iisExpress": {
6       "applicationUrl": "http://10.4.5.174:7805",
7       "sslPort": 44310
8     }
9   },
10  "$schema": "http://json.schemastore.org/launchsettings.json",
11  "profiles": {
12    "IIS Express": {
13      "commandName": "IISExpress",
14      "launchBrowser": true,
15      "launchUrl": "swagger",
16      "environmentVariables": {
17        "ASPNETCORE_ENVIRONMENT": "Development"
18      }
19    },
20    "A4KPI": {
21      "commandName": "Project",
22      "launchBrowser": false,
23      "launchUrl": "swagger",
24      "environmentVariables": {
25        "ASPNETCORE_ENVIRONMENT": "Development"
26      },
27      "dotnetRunMessages": "true",
28      "applicationUrl": "http://10.4.4.229:58"
29    },
30    "API": {
31      "commandName": "Executable"
32    }
33  }
34 }
```

7. Using MS Identity



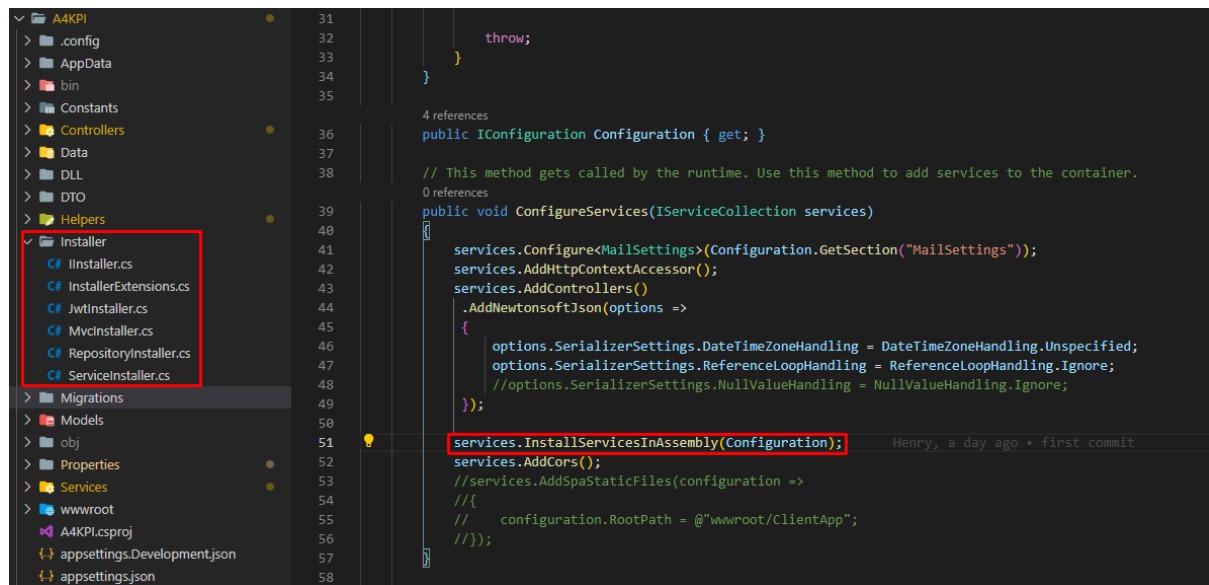
8. Differents setting startup

- SD3 team

```
72 // Repository
73
74 services.AddScoped<IPackingListRepository, PackingListRepository>();
75 services.AddScoped<ICodeIDDetailRepo, CodeIDDetailRepo>();
76 services.AddScoped<IRackLocationRepo, RackLocationRepo>();
77 services.AddScoped<IQRCodeMainRepository, QRCodeMainRepository>();
78 services.AddScoped<IPackingListDetailRepository, PackingListDetailRepository>();
79 services.AddScoped<IQRCodeDetailRepository, QRCodeDetailRepository>();
80 services.AddScoped<IHPVendorRepository, HPVendorRepository>();
81 services.AddScoped<IHPUploadTimeLogRepository, HPUploadTimeLogRepository>();
82 services.AddScoped<IMaterialPurchaseRepository, MaterialPurchaseRepository>();
83 services.AddScoped<IMaterialMissingRepository, MaterialMissingRepository>();
84 services.AddScoped<IMaterialViewRepository, MaterialViewRepository>();
85 services.AddScoped<ITransactionMainRepo, TransactionMainRepo>();
86 services.AddScoped<ITransactionDetailRepo, TransactionDetailRepo>();
87 services.AddScoped<IMaterialSheetSizeRepository, MaterialSheetSizeRepository>();
88 services.AddScoped<IUsersRepository, UsersRepository>();
89 services.AddScoped<IRolesRepository, RolesRepository>();
90 services.AddScoped<IRoleUserRepository, RoleUserRepository>();
91 services.AddScoped<IMesMoRepository, MesMoRepository>();
92 services.AddScoped<IViewMesMoRepository, ViewMesMoRepository>();
93 services.AddScoped<IViewPoRepository, ViewPoRepository>();
94 services.AddScoped<ISettingSupplierRepository, SettingSupplierRepository>();
95 services.AddScoped<ITransferFormRepository, TransferFormRepository>();
96 services.AddScoped<ICacheRepository, CacheRepository>();
97 services.AddScoped<IHPHolidaysi46Repository, HPHolidaysi46Repository>();
98 services.AddScoped<IHPHolidaysi46Repository, HPHolidaysi46Repository>();
99 services.AddScoped<ISettingReasonRepository, SettingReasonRepository>();
100 services.AddScoped<IMaterialOffsetRepository, MaterialOffsetRepository>();
101 services.AddScoped<IMesMoSizeRepository, MesMoSizeRepository>();
102 services.AddScoped<IMaterialsRepository, MaterialsRepository>();
103 services.AddScoped<IPoRootsRepository, PoRootsRepository>();
104 services.AddScoped<IPoMaterialsRepository, PoMaterialsRepository>();
105 services.AddScoped<IMesMoBasicRepository, MesMoBasicRepository>();
106 services.AddScoped<IReasonDetailRepository, ReasonDetailRepository>();
107 services.AddScoped<IMaterialPurchaseSplitRepository, MaterialPurchaseSplitRepository>();
108 services.AddScoped<ISettingT2SupplierRepository, SettingT2SupplierRepository>();
109 services.AddScoped<INoticeInforRepository, NoticeInforRepository>();
110 services.AddScoped<IReleaseDeliveryNoRepository, ReleaseDeliveryNoRepository>();
111 services.AddScoped<ISettingRTDRepository, SettingRTDRepository>();
112 services.AddScoped<ISettingClosingReasonRepository, SettingClosingReasonRepository>();
113 services.AddScoped<IHPP0j20Repository, HPP0j20Repository>();
```

```
114 // Service
115
116 services.AddScoped<IPackingListService, PackingListService>();
117 services.AddScoped<ICodeIDDetailService, CodeIDDetailService>();
118 services.AddScoped<IRackLocationService, RackLocationService>();
119 services.AddScoped<IQRCodeMainService, QRCodeMainService>();
120 services.AddScoped<IMaterialFormService, MaterialFormService>();
121 services.AddScoped<IReceivingService, ReceivingService>();
122 services.AddScoped<IInputService, InputService>();
123 services.AddScoped<ITransferLocationService, TransferLocationService>();
124 services.AddScoped<IOutputService, OutputService>();
125 services.AddScoped<IAuthService, AuthService>();
126 services.AddScoped<IKanbanService, KanbanService>();
127 services.AddScoped<IKanbanByRackService, KanbanByRackService>();
128 services.AddScoped<IKanBanByPoService, KanbanByPoService>();
129 services.AddScoped<IReportService, ReportService>();
130 services.AddScoped<IHistoryReportService, HistoryReportService>();
131 services.AddScoped<IReceivingEnoughService, ReceivingEnoughService>();
132 services.AddScoped<ISettingMailService, SettingMailService>();
133 services.AddScoped<ITransferFormService, TransferFormService>();
134 services.AddScoped<ICompareReportService, CompareReportService>();
135 services.AddScoped<IHPUploadService, HPUploadService>();
136 services.AddScoped<IMailUtility, MailUtility>();
137 services.AddScoped<IModifyStoreService, ModifyStoreService>();
138 services.AddScoped<ISettingReasonService, SettingReasonService>();
139 services.AddScoped<IMergeQrCodeService, MergeQrCodeService>();
140 services.AddScoped<ISettingT2SupplierService, SettingT2SupplierService>();
141 services.AddScoped<IReceivingMissingService, ReceivingMissingService>();
142 services.AddScoped<ISettingRTDService, SettingRTDService>();
143 services.AddScoped<IMaintainDeliveryDateService, MaintainDeliveryDateService>();
144 services.AddScoped<IClosingPurchaseService, ClosingPurchaseService>();
145
146 services.AddScoped<IReleaseDeliveryNoService, ReleaseDeliveryNoService>();
147 services.AddScoped<IModifyTransferFormService, ModifyTransferFormService>();
148 services.AddScoped<IGenerateNoticeNoService, GenerateNoticeNoService>();
149 services.AddSwaggerGen(c =>
150 {
151     c.SwaggerDoc("v1", new OpenApiInfo { Title = "Bottom API", Version = "v1" });
152 });
```


- KPI team



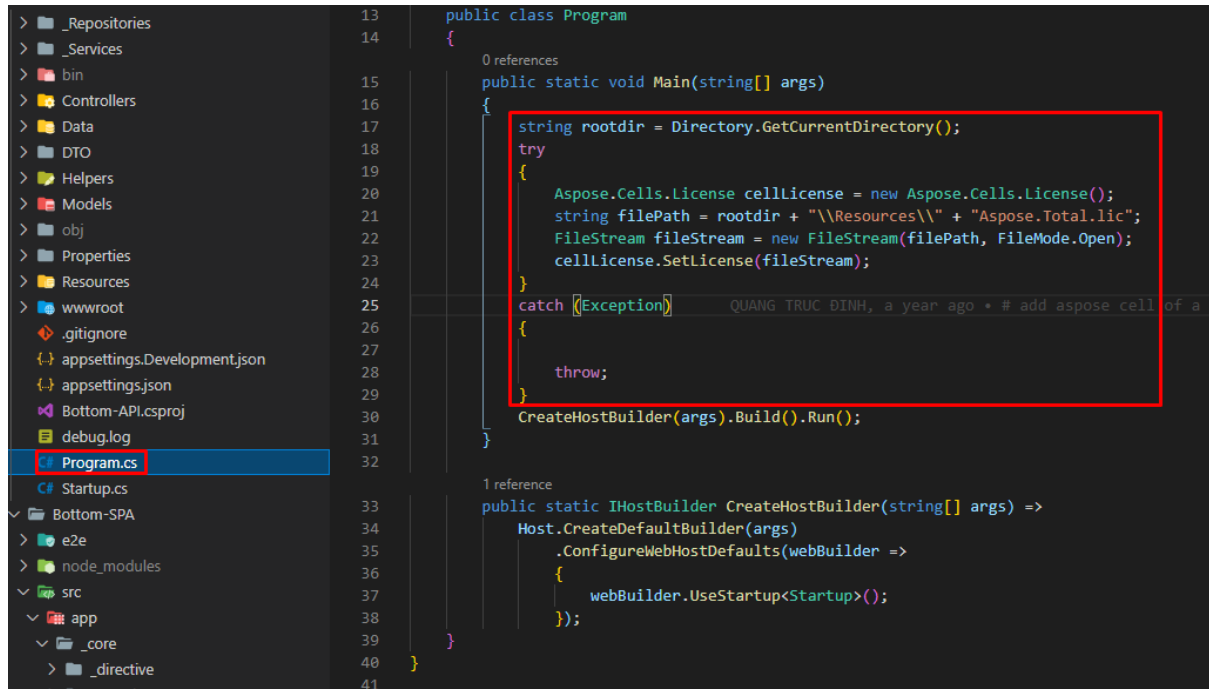
The image shows a Visual Studio interface. On the left, the 'Solution Explorer' displays a project structure for 'A4KPI'. The 'Installer' folder is expanded, and its contents are listed: 'IInstaller.cs', 'InstallerExtensions.cs', 'JwtInstaller.cs', 'MvcInstaller.cs', 'RepositoryInstaller.cs', and 'ServiceInstaller.cs'. The 'ServiceInstaller.cs' file is highlighted with a red box. On the right, the 'Code Editor' shows the 'ConfigureServices' method in 'Startup.cs'. The method is annotated with '4 references' and '0 references'. The code includes comments and calls to 'services.Configure', 'services.AddHttpContextAccessor', 'services.AddControllers', and 'services.AddNewtonsoftJson'. A red box highlights the line 'services.InstallServicesInAssembly(Configuration);'. To the right of this line, a commit message is visible: 'Henry, a day ago • first commit'.

```
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
throw;  
}  
}  
4 references  
public IConfiguration Configuration { get; }  
  
// This method gets called by the runtime. Use this method to add services to the container.  
0 references  
public void ConfigureServices(IServiceCollection services)  
{  
    services.Configure<MailSettings>(Configuration.GetSection("MailSettings"));  
    services.AddHttpContextAccessor();  
    services.AddControllers()  
        .AddNewtonsoftJson(options =>  
        {  
            options.SerializerSettings.DateTimeZoneHandling = DateTimeZoneHandling.Unspecified;  
            options.SerializerSettings.ReferenceLoopHandling = ReferenceLoopHandling.Ignore;  
            //options.SerializerSettings.NullValueHandling = NullValueHandling.Ignore;  
        });  
    services.InstallServicesInAssembly(Configuration);  
    services.AddCors();  
    //services.AddSpaStaticFiles(configuration =>  
    //{  
    //    configuration.RootPath = @"wwwroot/ClientApp";  
    //});  
}
```

Henry, a day ago • first commit

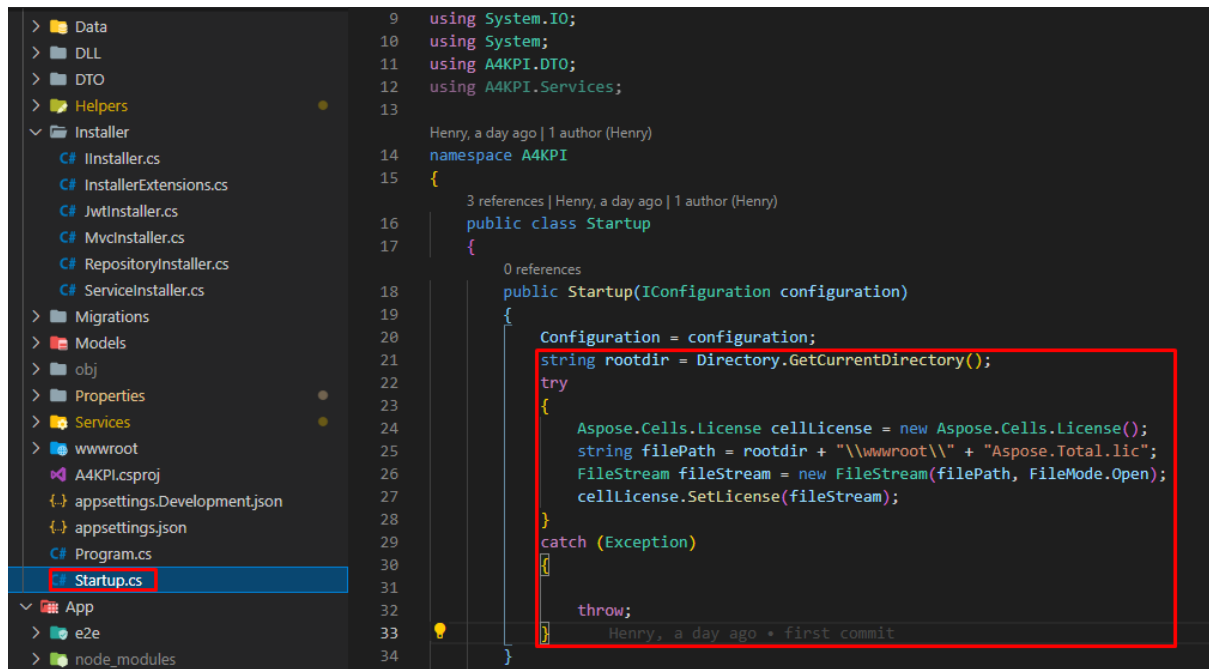
9. Differents setting Aspose cell License

- SD3 team



```
13 public class Program
14 {
15     0 references
16     public static void Main(string[] args)
17     {
18         string rootdir = Directory.GetCurrentDirectory();
19         try
20         {
21             Aspose.Cells.License celllicense = new Aspose.Cells.License();
22             string filePath = rootdir + "\\Resources\\" + "Aspose.Total.lic";
23             FileStream fileStream = new FileStream(filePath, FileMode.Open);
24             celllicense.SetLicense(fileStream);
25         }
26         catch (Exception)
27         {
28             throw;
29         }
30     }
31     CreateHostBuilder(args).Build().Run();
32 }
33
34 1 reference
35 public static IHostBuilder CreateHostBuilder(string[] args) =>
36     Host.CreateDefaultBuilder(args)
37         .ConfigureWebHostDefaults(webBuilder =>
38         {
39             webBuilder.UseStartup<Startup>();
40         });
41 }
```

- KPI team



```
9 using System.IO;
10 using System;
11 using A4KPI.DTO;
12 using A4KPI.Services;
13
14 Henry, a day ago | 1 author (Henry)
15 namespace A4KPI
16 {
17     3 references | Henry, a day ago | 1 author (Henry)
18     public class Startup
19     {
20         0 references
21         public Startup(IConfiguration configuration)
22         {
23             Configuration = configuration;
24         }
25         string rootdir = Directory.GetCurrentDirectory();
26         try
27         {
28             Aspose.Cells.License celllicense = new Aspose.Cells.License();
29             string filePath = rootdir + "\\wwwroot\\" + "Aspose.Total.lic";
30             FileStream fileStream = new FileStream(filePath, FileMode.Open);
31             celllicense.SetLicense(fileStream);
32         }
33         catch (Exception)
34         {
35             throw;
36         }
37     }
38 }
```

10. SPA use Syncfusion Components Library

```
@sweetalert2/ngx-sweetalert2": "^8.1.1",
"@syncfusion/ej2-angular-barcode-generator": "^19.1.56",
"@syncfusion/ej2-angular-base": "^19.1.58",
"@syncfusion/ej2-angular-buttons": "^19.1.54",
"@syncfusion/ej2-angular-calendars": "^19.1.56",
"@syncfusion/ej2-angular-charts": "^19.1.59",
"@syncfusion/ej2-angular-dropdowns": "^19.1.59",
"@syncfusion/ej2-angular-grids": "^19.1.58",
"@syncfusion/ej2-angular-inputs": "^19.1.59",
"@syncfusion/ej2-angular-navigations": "^19.1.59",
"@syncfusion/ej2-angular-popups": "^19.1.59",
"@syncfusion/ej2-angular-richtexteditor": "^19.1.59",
"@syncfusion/ej2-angular-schedule": "^19.1.59",
"@syncfusion/ej2-angular-spreadsheet": "^19.2.47",
"@syncfusion/ej2-angular-treegrid": "^19.1.59",
"@syncfusion/ej2-base": "^19.1.57",
```