

public class SimpleContainer

{

private static volatile Container \_Container = null;

private static void Init()

{

var container = new Container();

container.Options.DefaultScopedLifestyle = new AsyncScopedLifestyle();

container.Register<IGenerateDbContext, GenerateAppDbContext>(Lifestyle.Transient);

InicializarServiciosExternos(container);

container.Register<IValidator<ListarEstadosQuery>, ListarEstadosQueryValidator>();

var assemblies = new[] { typeof(ValidateNada<>).Assembly };

container.Collection.Register(typeof(IValidator<>), assemblies);

container.RegisterSingleton(() => GetMapper(container));

container.BuildMediator(typeof(ListarEtapasQueryHandler).GetTypeInfo().Assembly);

container.Verify();

\_Container = container;

}

public static Container Instance()

{

if (\_Container == null)

{

lock (typeof(Container))

{

if (\_Container == null)

{

Init();

}

}

}

return \_Container;

}

public static Container Get()

{

return Instance();

}

protected static AutoMapper.IMapper GetMapper(Container container)

{

var mp = container.GetInstance<MapperProvider>();

return mp.GetMapper();

}

protected static void InicializarServiciosExternos(Container container)

{

container.Register<IExportadorExcel, ExportadorExcel>(Lifestyle.Scoped);

container.Register<IApiRecup02, ApiRecup02>(Lifestyle.Singleton);

container.Register<IApiEquifax, ApiEquifaxMock>(Lifestyle.Singleton);

}

}

public class MapperProvider

{

private readonly Container \_container;

public MapperProvider(Container container)

{

\_container = container;

}

public IMapper GetMapper()

{

var config = new MapperConfiguration(cfg =>

{

cfg.AddProfile<MappingProfile>();

cfg.ConstructServicesUsing(\_container.GetInstance);

});

IMapper mapper = new Mapper(config);

return mapper;

}

}

public class SomeProfile : Profile

{

public SomeProfile()

{

}

}