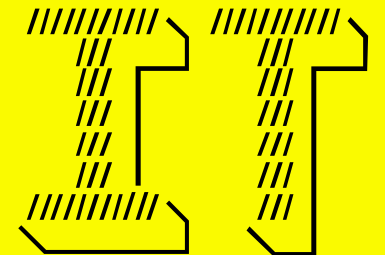


# Практика построения сервисов на основе ViennaNET

Сергей Огородников,  
старший разработчик

**Райффайзен**



# О докладчике



## **Сергей Огородников**

Профессионально разрабатывает на C# с 2005 года, пришёл в .NET за пару месяцев до того, как подвезли generics. Сейчас работает старшим разработчиком в Райффайзенбанке в команде, занимающейся разработкой продуктов для HR. Интересуется DDD, software architecture, разработкой анализаторов кода, немного ФП.

# Содержание

1. Сложности разработки микросервисов
2. Цели создания фреймворка ViennaNET
3. Обзор возможностей ViennaNET
4. Пример WebAPI сервиса
5. Добавим CRQS, Domain
6. Кратко о других возможностях – очереди, HttpClient, саги, валидация...

# Сложности разработки микросервисов

- Один сервис – один репозиторий
- Реализация стандартов
- Enterprise – интеграции, специфическое ПО
- Повторение кода
- Экспертиза концентрируется по командам

# История ViennaNET

- Разные проекты в разных репозиториях
- Выделение общего инфраструктурного кода в отдельный репо
- Доработка InnerSource, использование сообществом

Константин Густов «Как вырастить open source в банке»

[https://www.youtube.com/watch?v=\\_mKyZDS9v4k](https://www.youtube.com/watch?v=_mKyZDS9v4k)

# Основные цели ViennaNET

- Разработчик пишет бизнес, а не инфру
- Инфра должна соответствовать внутренним стандартам
- Инфра должна быть гибкой и развиваемой
- Аккумуляция экспертизы сообщества

# Основные возможности ViennaNET



Архитектура и  
тактические паттерны  
DDD



Конфигурируемость



Модульность



Логирование и аудит



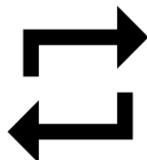
Наблюдаемость и  
метрики



Авторизация



Транзакционность




Адаптеры к внешним  
источникам





CQRS & event  
sourcing


# OpenSource


<https://github.com/Raiffeisen-DGTL/ViennaNET>


 122 commits

 2 branches

 0 packages

 1 release

 5 contributors

 MIT

C# 99.9%

JavaScript 0.1%

Branch: master ▾

New pull request

Find file

Clone or download ▾

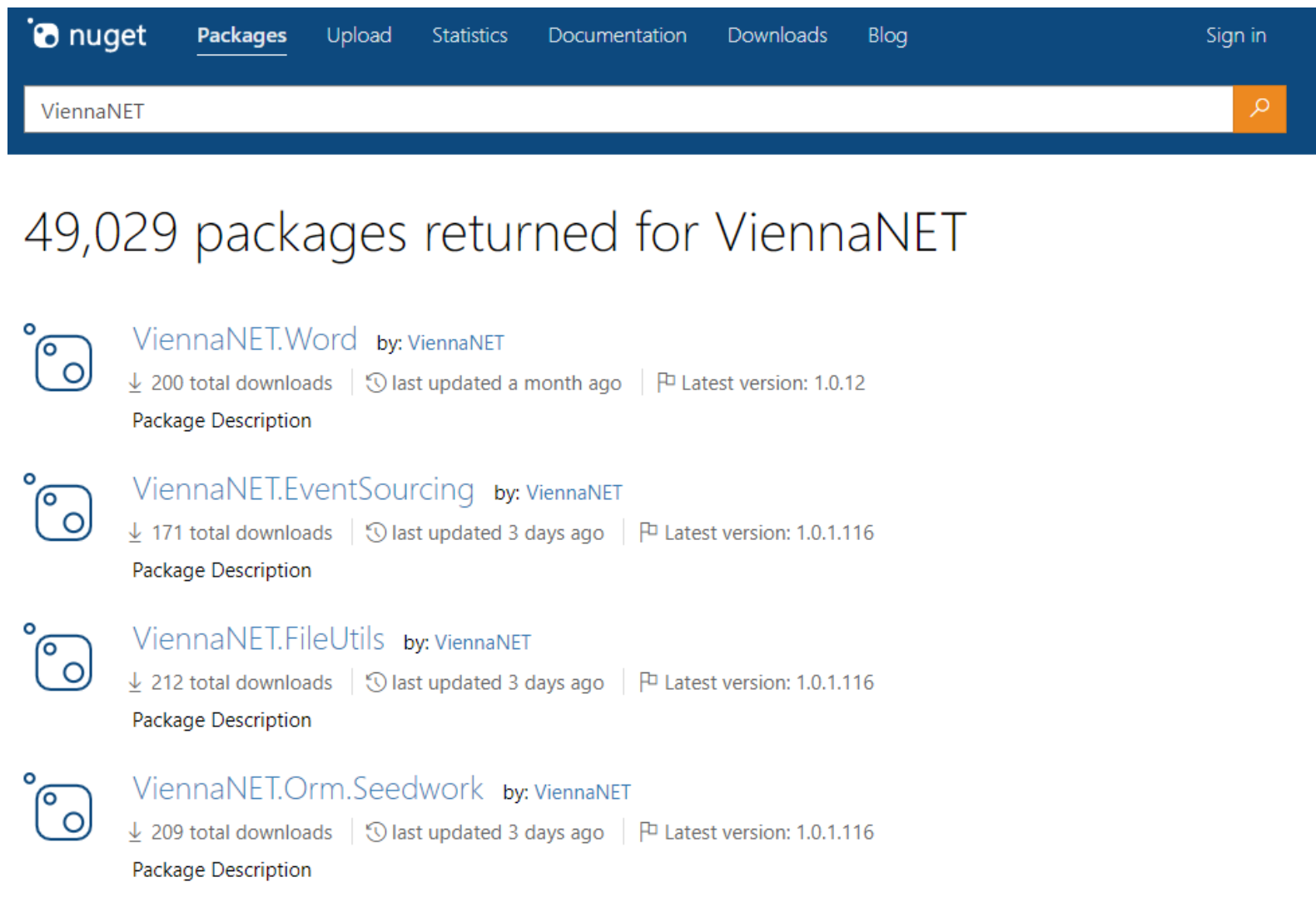
 gorrka84 Update publish.yml ...

✓ Latest commit 68783a6 3 days ago



# OpenSource

<https://www.nuget.org/packages?q=ViennaNET>



The screenshot shows the NuGet.org website with a search for 'ViennaNET'. The search bar at the top contains 'ViennaNET' and a magnifying glass icon. Below the search bar, the text '49,029 packages returned for ViennaNET' is displayed. Four packages are listed, each with a small icon, the package name, the author 'by: ViennaNET', and statistics for total downloads, last updated time, and latest version. Each entry also has a 'Package Description' link.

Package Name	Author	Total Downloads	Last Updated	Latest Version
ViennaNET.Word	ViennaNET	200	last updated a month ago	1.0.12
ViennaNET.EventSourcing	ViennaNET	171	last updated 3 days ago	1.0.1.116
ViennaNET.FileUtils	ViennaNET	212	last updated 3 days ago	1.0.1.116
ViennaNET.Orm.Seedwork	ViennaNET	209	last updated 3 days ago	1.0.1.116

# ViennaNET.WebApi

Слои:

## 1. Базовые библиотеки

(**ViennaNET.WebApi**, ViennaNET.Logging, ViennaNET.CallContext, ViennaNET.Security.Jwt, ...)

## 2. Конфигураторы

(ViennaNET.WebApi.Configurators.Diagnostic, ViennaNET.WebApi.Configurators.HttpClients.Jwt)

## 3. Runners

(ViennaNET.WebApi.Runners.BaseKestrel, ViennaNET.WebApi.Runners.BaseHttpSys)

# ViennaNET.WebApi

```
public interface ICompanyHostBuilder
{
    /// <summary> Собирает сервис согласно зарегистрированным конфигураторам
    7 references | ATINK Dmitry, 110 days ago | 1 author, 1 change
    IWebHost BuildWebHost(string[] args);

    /// <summary> Позволяет дополнительно сконфигурировать MVC Builder
    5 references | ATINK Dmitry, 110 days ago | 1 author, 1 change
    ICompanyHostBuilder AddMvcBuilderConfiguration(Action<IMvcCoreBuilder, IConfiguration> configAct

    /// <summary> Позволяет сконфигурировать создание MVC Builder
    1 reference | ATINK Dmitry, 110 days ago | 1 author, 1 change
    ICompanyHostBuilder AddMvcOptionsConfiguration(Action<MvcOptions> configureMvcOptions);

    /// <summary> Конфигурирование внутреннего билдера приложения, здесь добавляются ...
    12 references | ATINK Dmitry, 110 days ago | 1 author, 1 change
    ICompanyHostBuilder ConfigureApp(
        Action<IApplicationBuilder, IConfiguration, IHostEnvironment, object> appConfigurationAction,

    /// <summary> Регистрирует стандартное middleware, без привязки к стороннему кон ...
    1 reference | ATINK Dmitry, 110 days ago | 1 author, 1 change
    ICompanyHostBuilder AddMiddleware<T>() where T : IMiddleware;
```

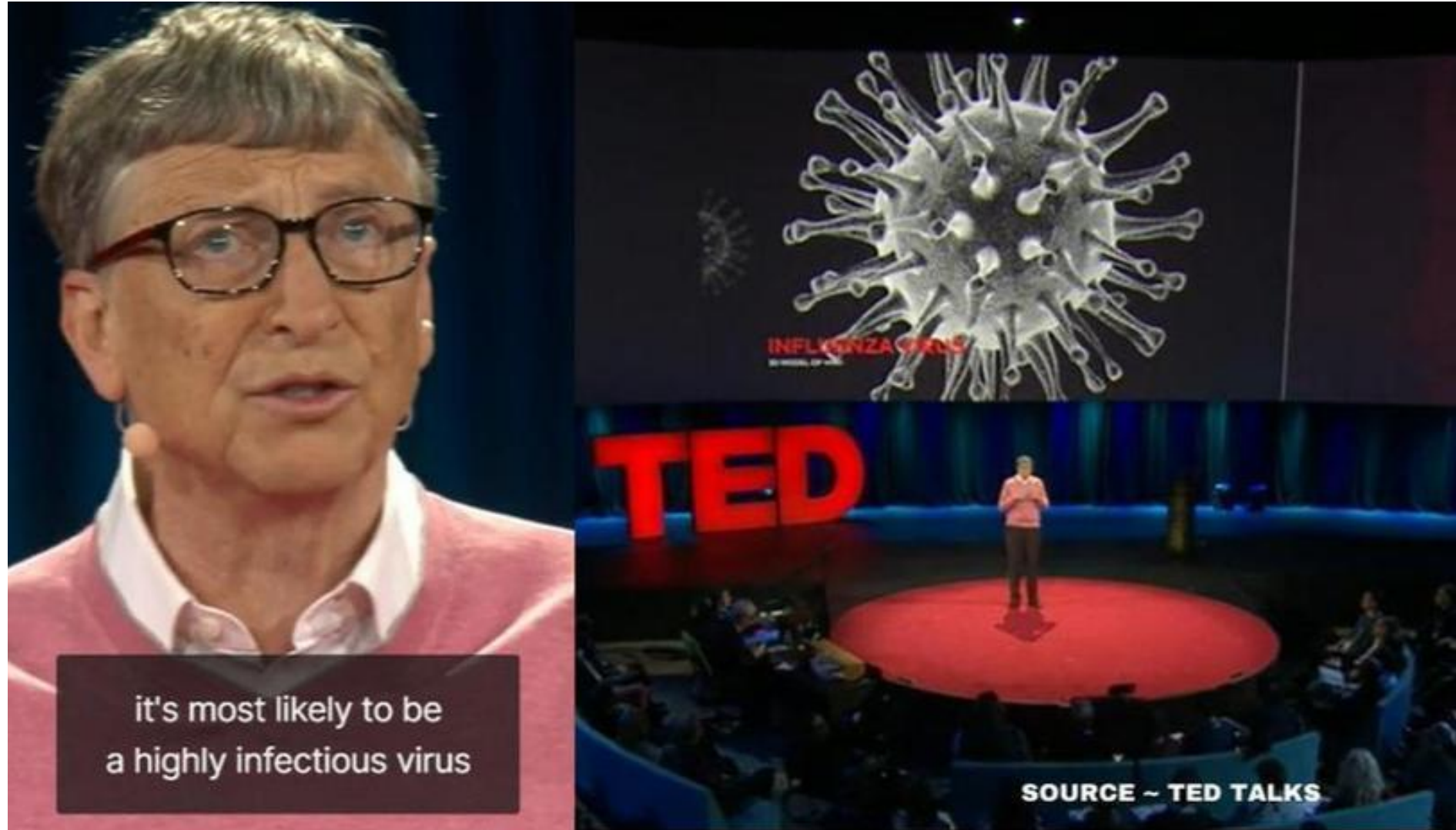
# ViennaNET.WebApi.Configurators.\*

```
namespace ViennaNET.WebApi.Configurators.CallContext
{
    /// <summary> Включает базовые сервисы и middleware
    0 references | ATINK Dmitry, 63 days ago | 1 author, 3 changes
    public static class CallContextConfigurator
    {
        /// <summary> Включает базовые сервисы и middleware Необходимо подключать сразу ...
        2 references | ATINK Dmitry, 70 days ago | 1 author, 2 changes
        public static ICompanyHostBuilder UseCallContext(this ICompanyHostBuilder companyHostBuilder)
        {
            return companyHostBuilder.ConfigureApp(ConfigureCallContextMiddleware)
                .RegisterServices(RegisterCallContextServices); // ICompanyHostBuilder
        }
    }
}
```

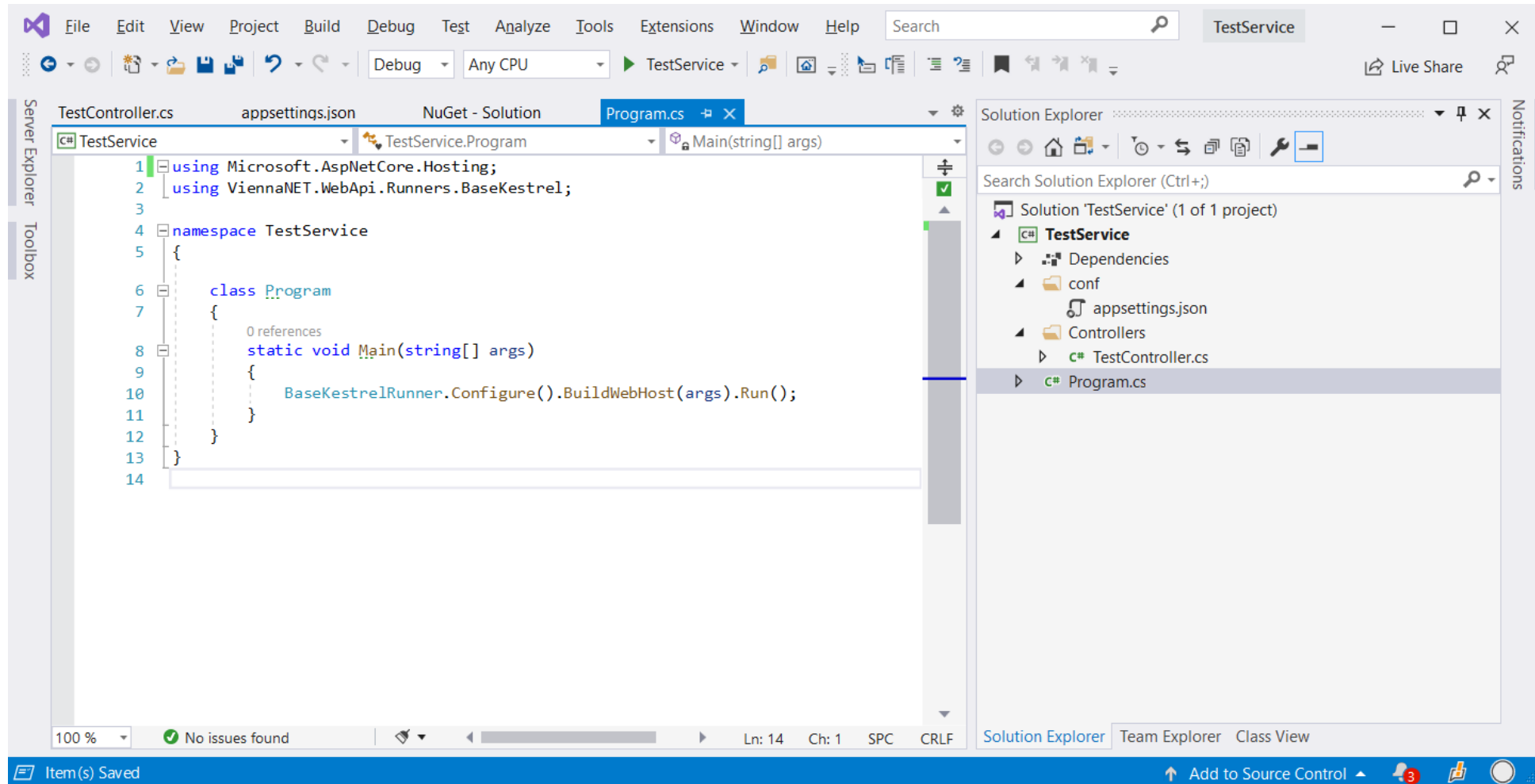
# ViennaNET.WebApi.Runners.\*

```
namespace ViennaNET.WebApi.Runners.BaseKestrel
{
    /// <summary> Класс для создания и конфигурирования базового сервиса на Kestrel
    4 references | ATINK Dmitry, 82 days ago | 1 author, 2 changes
    public static class BaseKestrelRunner
    {
        4 references | ATINK Dmitry, 82 days ago | 1 author, 2 changes
        public static ICompanyHostBuilder Configure()
        {
            return CompanyHostBuilder.Create() // CompanyHostBuilder
                .UseKestrel()
                .UseJwtAuth()
                .UseCallContext()
                .UseCommonModules()
                .UseSimpleInjector()
                .UseSwaggerWithJwtAuth()
                .UseDiagnosing()
                .UseJwtHttpClient(); // ICompanyHostBuilder
        }
    }
}
```

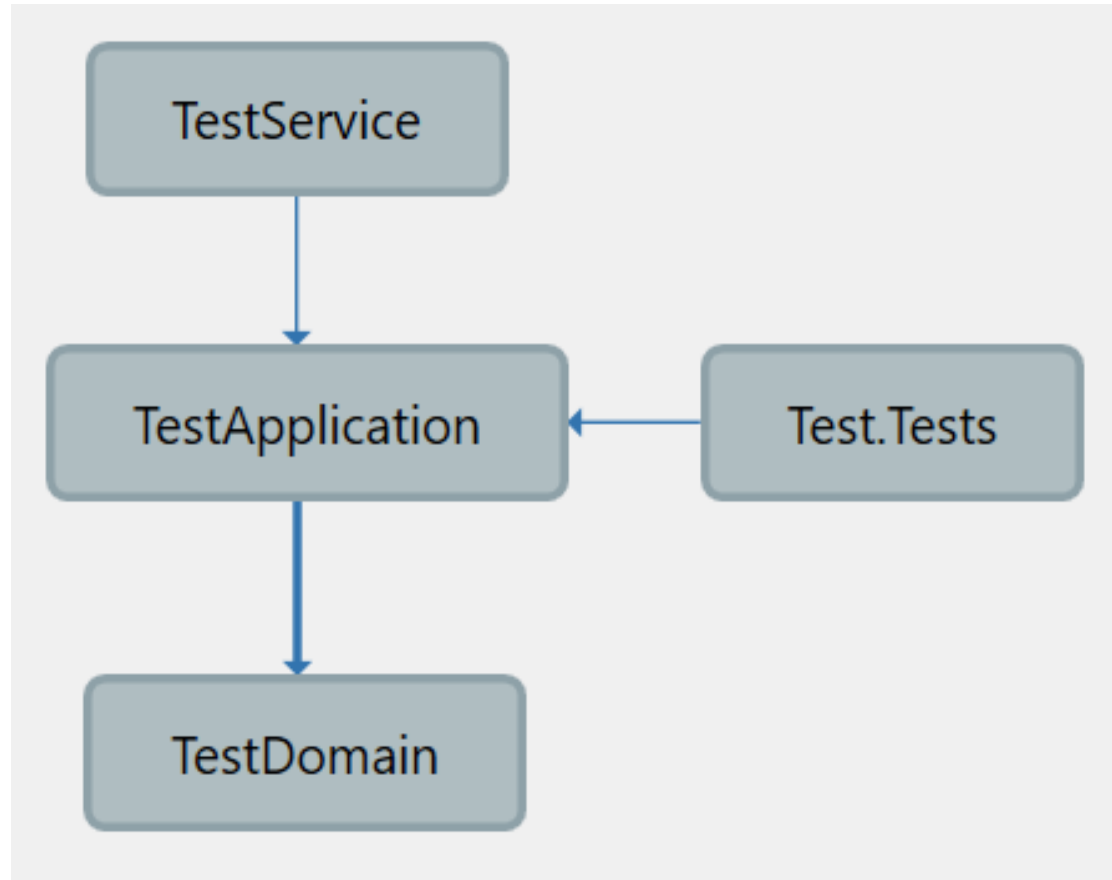
# We need new microservice!



# Live: BaseKestrelRunner



# Onion architecture





# ViennaNET.Mediator

```
public interface IMediator
{
    /// <summary> Асинхронно отправляет команды или события. Данная отправка не треб ...
    12 references | ATINK Dmitry, 110 days ago | 1 author, 1 change
    Task SendMessageAsync<TMessage>(TMessage message, CancellationToken cancellationToken = default)
        where TMessage : class, IMessage;
}

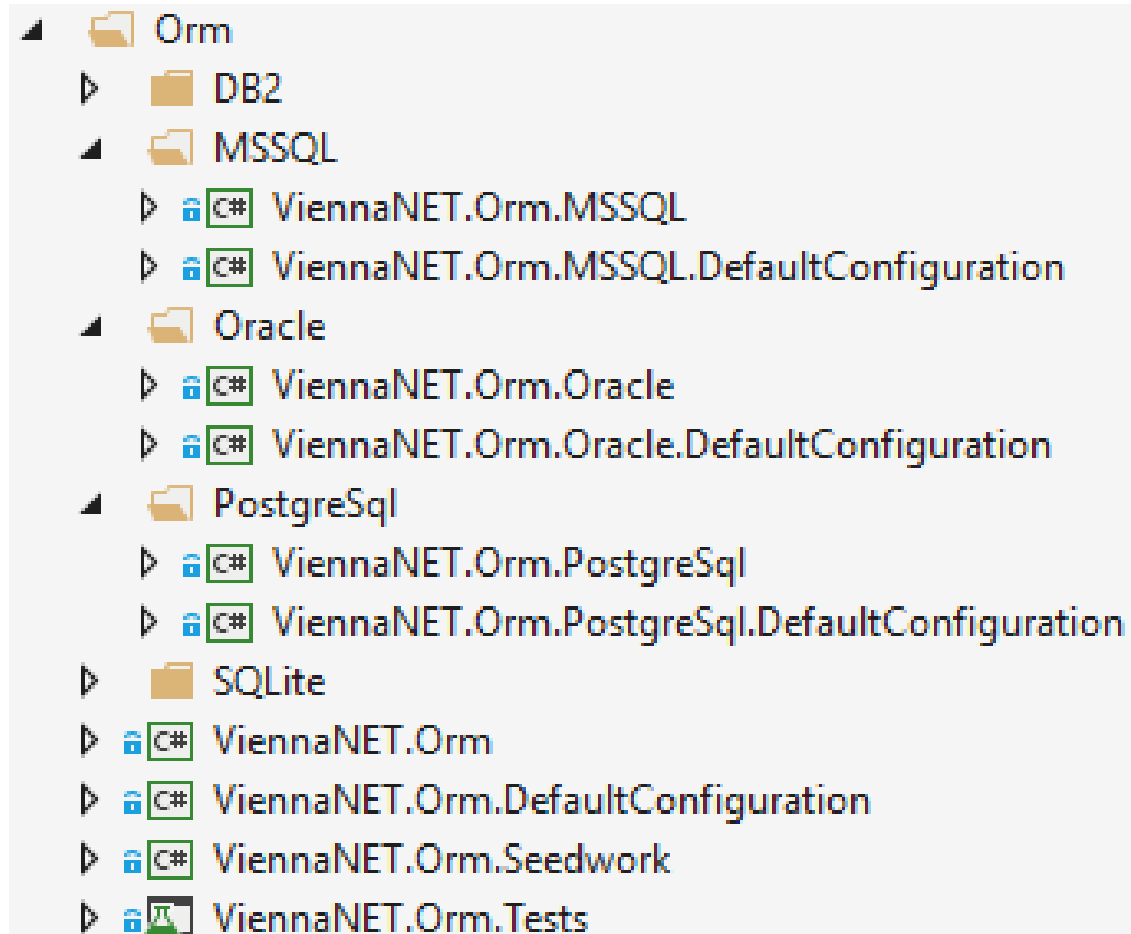
[HttpPost(template: "card")]
0 references | 0 changes | 0 authors, 0 changes
public ActionResult CloseCard(int cardId)
{
    _mediator.SendMessage(new CloseCardCommand(cardId));
    return Ok();
}

public interface IMessageHandler<in TMessage, TResponse> : IMessageHandler
{
    where TMessage : class, IMessage
    /// <summary> Синхронно обрабатывает сообщение, возвращая результат.
    4 references | ATINK Dmitry, 110 days ago | 1 author, 1 change
    TResponse Handle(TMessage message);
}
```

# ViennaNET.Orm

1. Поддержка DDD
2. Паттерн Repository
3. Паттерн UnitOfWork
4. Под капотом – NHibernate, есть возможность добавить реализацию на EF Core
5. Наличие 5 драйверов разных БД
6. Тестируемость

# ViennaNET.Orm



# Live: IEntityRepository

The screenshot displays the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Toolbar:** Includes icons for file operations, a search bar, and a dropdown menu showing 'Debug', 'Any CPU', and 'TestService'.
- Code Editor:** Shows the file `GetRegisteredVaccineCountQueryHandler.cs` with the following code:

```
7 | using ViennaNET.Orm.Seedwork;
8 |
9 | namespace TestApplication.Queries.GetRegisteredVaccineCount
10 | {
11 |     1 reference
12 |     internal class GetRegisteredVaccineCountQueryHandler : IMessageHandlerAsync<GetRegisteredVaccineCountQuery, int>
13 |     {
14 |         private readonly IEntityRepositoryFactory _entityRepositoryFactory;
15 |
16 |         0 references
17 |         public GetRegisteredVaccineCountQueryHandler(IEntityRepositoryFactory entityRepositoryFactory)
18 |         {
19 |             _entityRepositoryFactory = entityRepositoryFactory;
20 |         }
21 |
22 |         1 reference
23 |         public async Task<int> HandleAsync(GetRegisteredVaccineCountQuery message, CancellationToken cancellationToken)
24 |         {
25 |             var allManufacturers = await _entityRepositoryFactory.Create<Manufacturer>() // IEntityRepository<Manufacturer>
26 |                 .Query() // IQueryable<Manufacturer>
27 |                 .FetchMany(relatedObjectSelector: m : Manufacturer => m.TradeMarks) // INhFetchRequest<Manufacturer, TradeMark>
28 |                 .Where(m : Manufacturer => m.HasMedicalLicense) // IQueryable<Manufacturer>
29 |                 .ToListAsync(cancellationToken); // Task<List<>>
30 |
31 |             return allManufacturers.SelectMany(m : Manufacturer => m.TradeMarks) // IEnumerable<TradeMark>
32 |                 .Count(predicate: tm : TradeMark => tm.IsActiveVaccine()); // int
33 |         }
34 |     }
35 | }
```

- Solution Explorer:** Shows the project structure for 'TestService' (3 of 3 projects). The 'TestApplication' project is expanded, showing folders for 'Dependencies', 'Queries', and 'GetRegisteredVaccineCount'. The file `GetRegisteredVaccineCountQueryHandler.cs` is selected.
- Toolbox:** Located on the left side of the code editor.
- Status Bar:** At the bottom, it shows '100 %', 'No issues found', and 'Ln: 33 Ch: 1 SPC CRLF'.

# ViennaNET.Messaging

```
public class TestMessageProcessor: IMessageProcessor
{
    11 references | ATINK Dmitry, 304 days ago | 1 author, 1 change
    public bool Process(BaseMessage message)
    {
        var deserializer = new PlainTextSerializer();
        var msg = deserializer.Deserialize(message);
        Logger.LogDebug(msg);
        return true;
    }
}
```

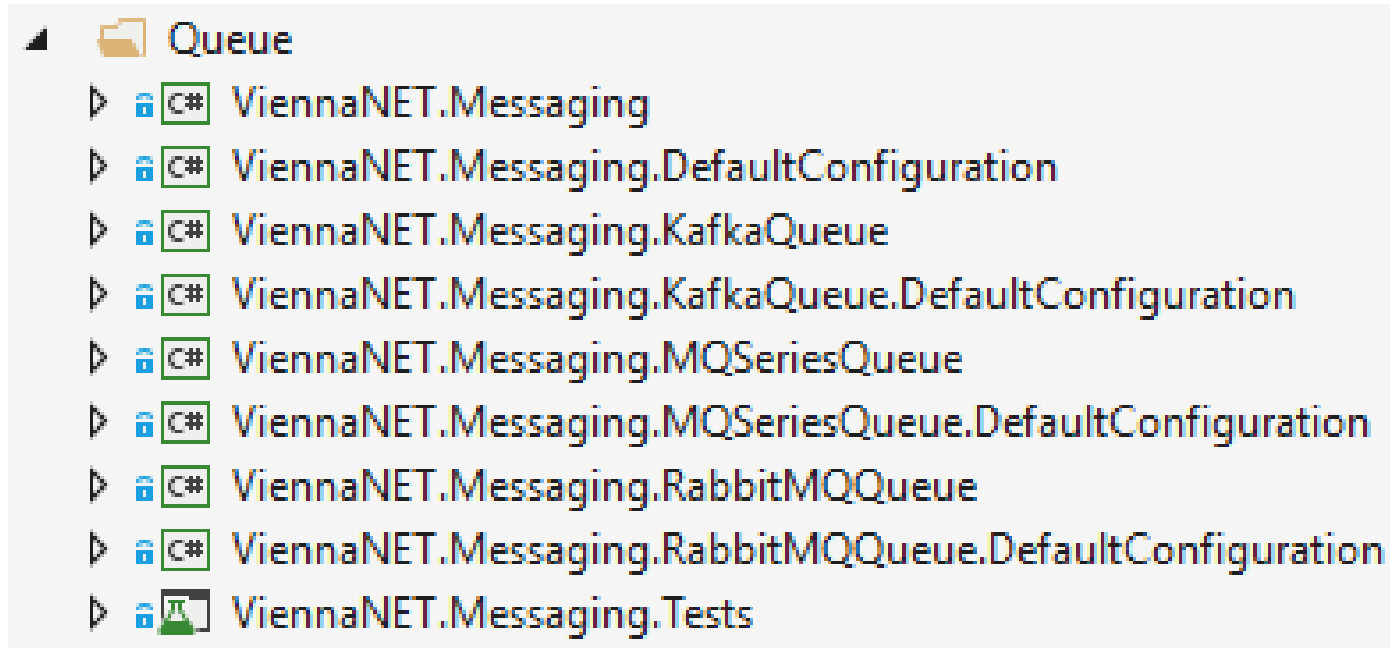
```
/// <summary> Предоставляет функционал для прослушки очередей
16 references | ATINK Dmitry, 118 days ago | 1 author, 1 change
public interface IQueueReactor: IDisposable
{
    /// <summary> Запускает процесс прослушивания
    11 references | ATINK Dmitry, 118 days ago | 1 author, 1 change
    bool StartProcessing();

    /// <summary> Останавливает процесс прослушивания
    12 references | ATINK Dmitry, 118 days ago | 1 author, 1 change
    void Stop();
}
```

```
public interface IQueueReactorFactory
{
    /// <summary> Регистрация обработчика IMessageProcessor сообщений BaseMessage
    12 references | 0 changes | 0 authors, 0 changes
    IMessageProcessorRegister Register<T>(string queueId) where T : class;

    /// <summary> Создает экземпляр IQueueReactor
    11 references | Konstantin Gustov, 114 days ago | 1 author, 1 change
    IQueueReactor CreateQueueReactor(string queueId);
}
```

# ViennaNET.Messaging



# Другие возможности ViennaNET

- Валидация
- EventSourcing, Саги
- Enterprise фичи
- Кэш (Redis)
- Утилиты

# ViennaNET ждёт вашего фидбэка

- <https://github.com/sergeyo/ViennaNET.Example>
- <https://habr.com/ru/company/raiffeisenbank/blog/494830/>
- Проект с историей, в проде в нескольких проектах
- Используется около 60% .NET команд Райффайзенбанка
- Готов к фидбэку сообщества, ждём pull request



# Спасибо за внимание!

Буду рад ответить на Ваши вопросы!



Sergey.Ogorodnikov@raiffeisen.ru  
[raiffeisen.ru](https://raiffeisen.ru)

**Райффайзен**

