

# Анализаторы кода и их польза в сельском хозяйстве

**Мамай Игорь**

mamay.igor@gmail.com

<https://github.com/BigBabay>



# О чем поговорим

- Зачем нам анализаторы
- Примеры задач из жизни нашей команды
- Советы по выбору анализатора

# Останется за кадром

- Общая теория анализаторов
- Код использования анализаторов

# Зачем нам анализаторы?

- Большая команда (20 бекэнд-разработчиков)
- Большая кодовая база (1 000 000 строк кода)
- Shared Code Ownership
- Единый стиль кодирования

# Анализаторов много

## Поиск багов

Fxcop  
Clocksharp  
PRQA QA·C#  
PVS-Studio  
CAT.NET  
Spec#  
...

## Оценка качества кода

NDepend  
Nitriq  
RSM Squared  
C# Metrics  
SourceMonitor  
Code Metrics  
Vil

## Поиск дублирующегося кода

Simian  
CloneDR  
Clone Detective  
Atomiq

## Общий стиль

StyleCop  
Agent Smith

# Анализаторы с программным API

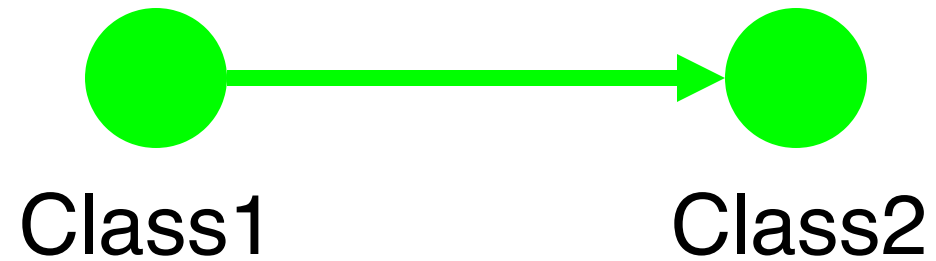


# Классы задач

- Анализ
- Миграции
- Рефакторинг
- Собственные инструменты

# Анализ

## Граф вызовов





# Граф вызовов RPC



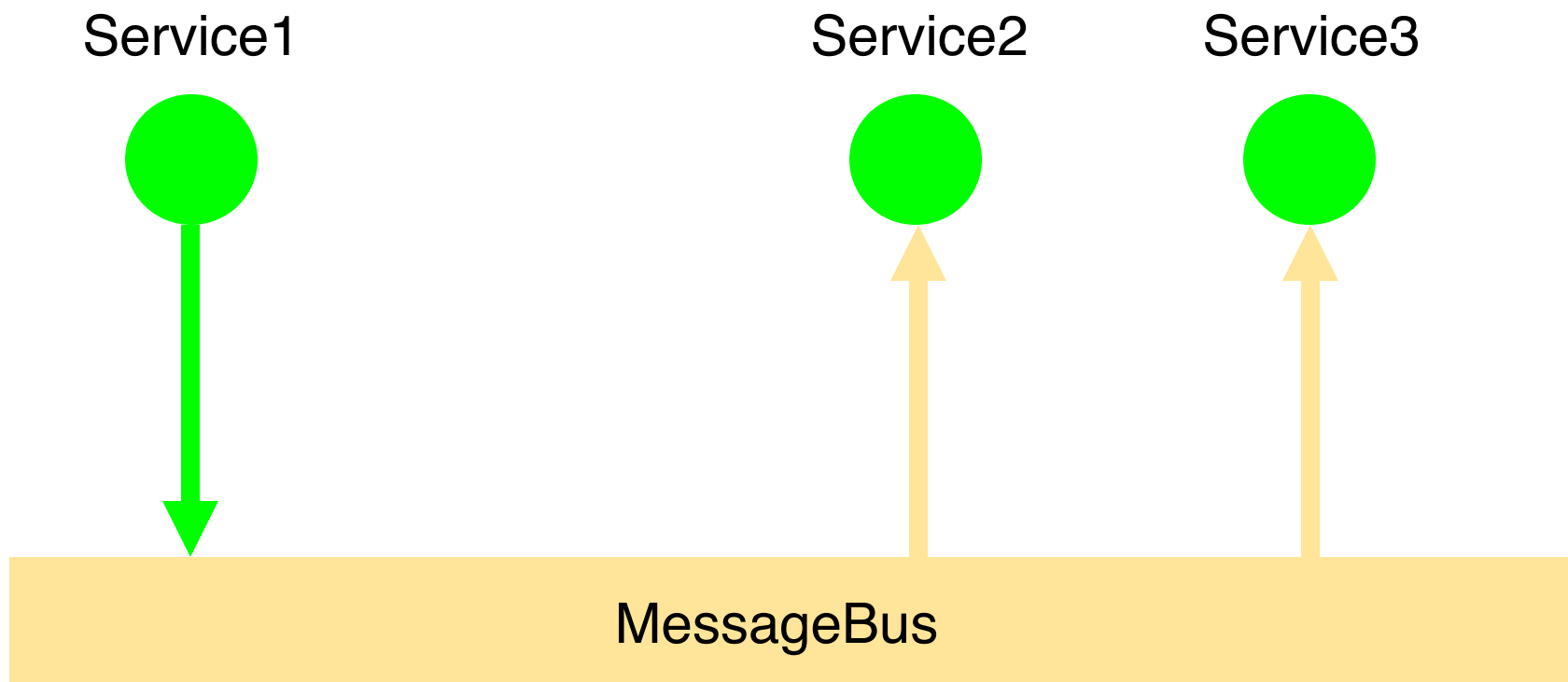
```
var request = Request.Get("MethodName");  
return client.Send<int>(request);
```

```
priceService.CalculateAsync(calculationContext);
```

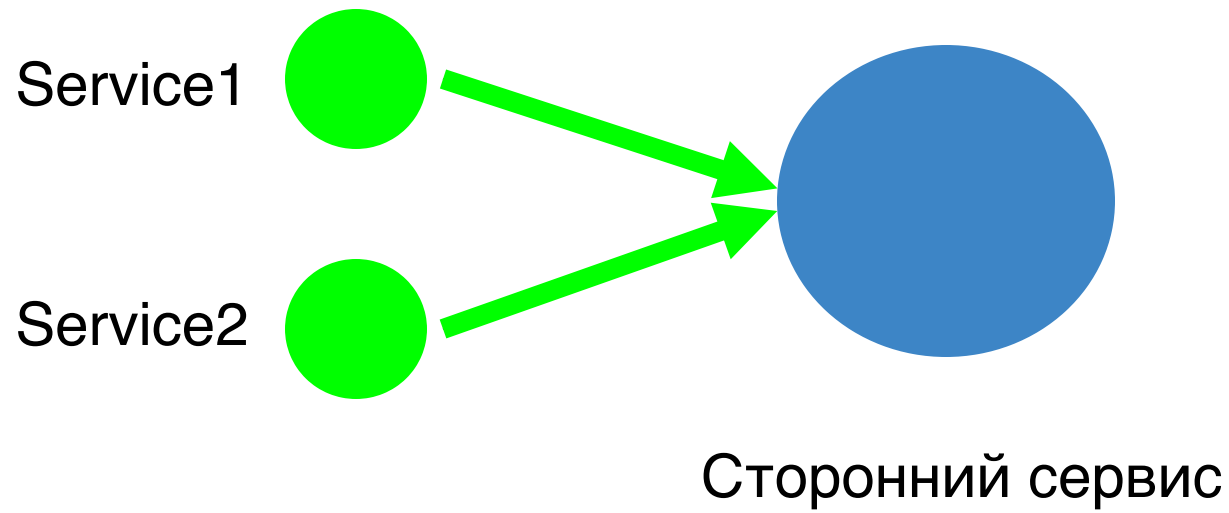
```
[NetworkService]
```

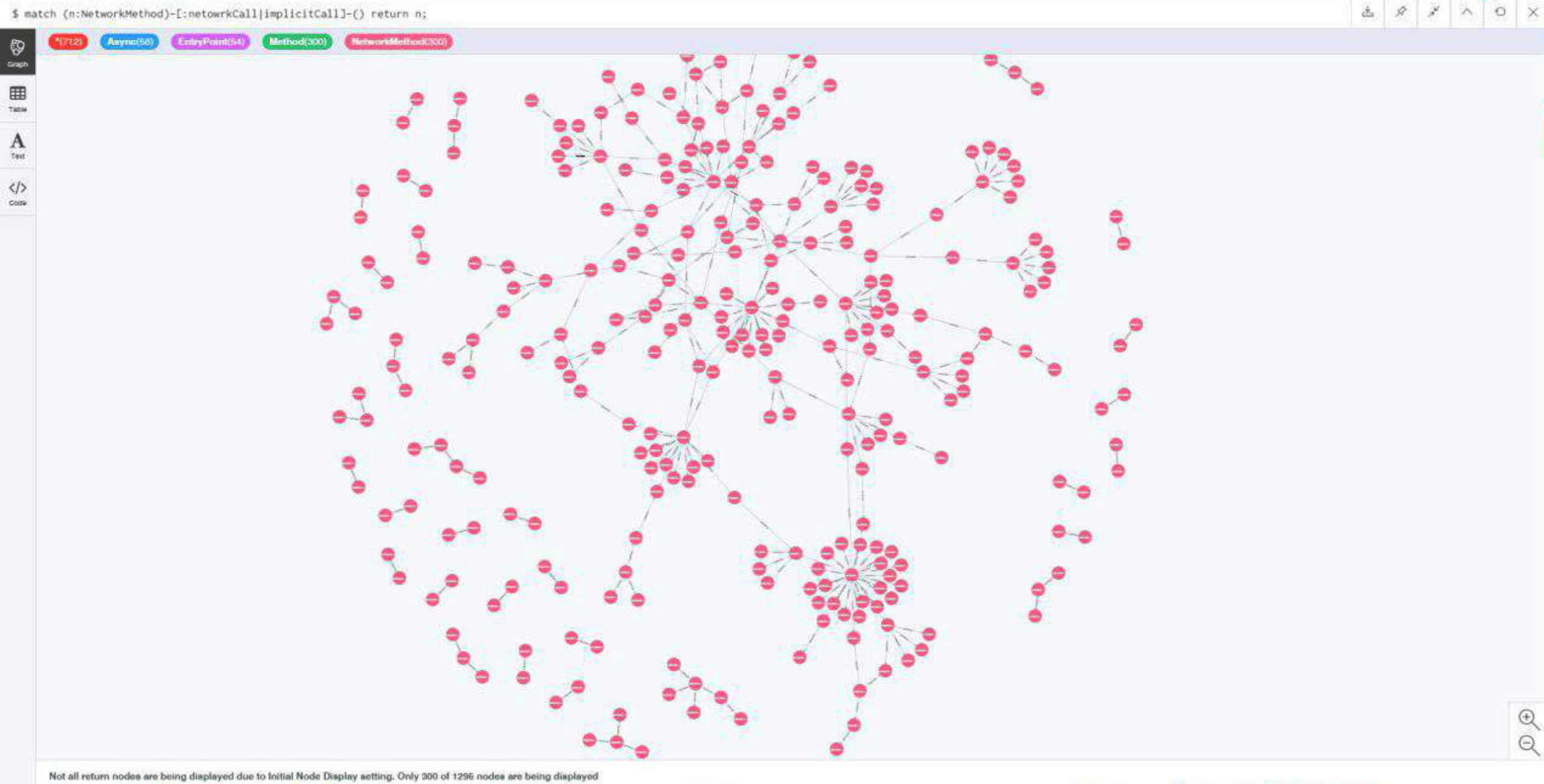
```
public interface IPriceService
```

# Граф вызовов с очередью



# Граф вызовов на сторонние сервисы





# Миграция или обновление технологии

# unit 2.x to 3.x

```
[Test]
[ExpectedException(typeof(Exception)
    , ExpectedMessage = "Message")]
public void Test()
{
    //prepare
    //.....

    testingClass.TestMethod();
}
```

```
[Test]
public void Test()
{
    //prepare
    //.....

    Assert.Throws<Exception>(
        () => testingClass.TestMethod()
        , "Message");
}
```

# unit 2.x to 3.x

- Найти методы, размеченные ExpectedException
- Найти тестируемый метод
- Завернуть в обертку, подставив данные из ExpectedException

# Рефакторинг

- Программисты любят рефакторить
- Никто не любит делать одно и тоже больше 1 раза
- Иногда нужно изменить концепт, который используется по всему коду



# Способы решения

- Посмотреть в IDE
- Сделать руками (посадить стажера)
- Нажать Ctrl+N и попробовать заменить
- Написать RegEx
- Использовать ReplacePattern
- Написать утилиту, которая все

```

private readonly IFactory factory;
7
public FooManager(IFactory factory)
    => this.factory = factory;

public int GetCount()
{
    using (var queryExecutor = factory.Create())
    {
        return queryExecutor.Execute<IFooStorage, int>(
            foo => foo.GetCount());
2
    }
1
}

```

```

private readonly IFooStorage fooStorage;
4
public FooManager(IFooStorage fooStorage)
    => this.fooStorage = fooStorage;
3
5

8|[Scope]
public int GetCount()
    => fooStorage.GetCount();
6

```

# Написание собственных инструментов



# Возможности

- Интегрируются в повседневную работу
- Автоматизируют ваши стилистические договоренности
- Встраиваются в CI
- Помогают при переходе к новым парадигмам


# Переход к асинхронному взаимодействию *Async/Await*

Бережем ThreadPool

# Плагин AsyncConverter – помощник по написанию асинхронного кода

# AsyncConverter

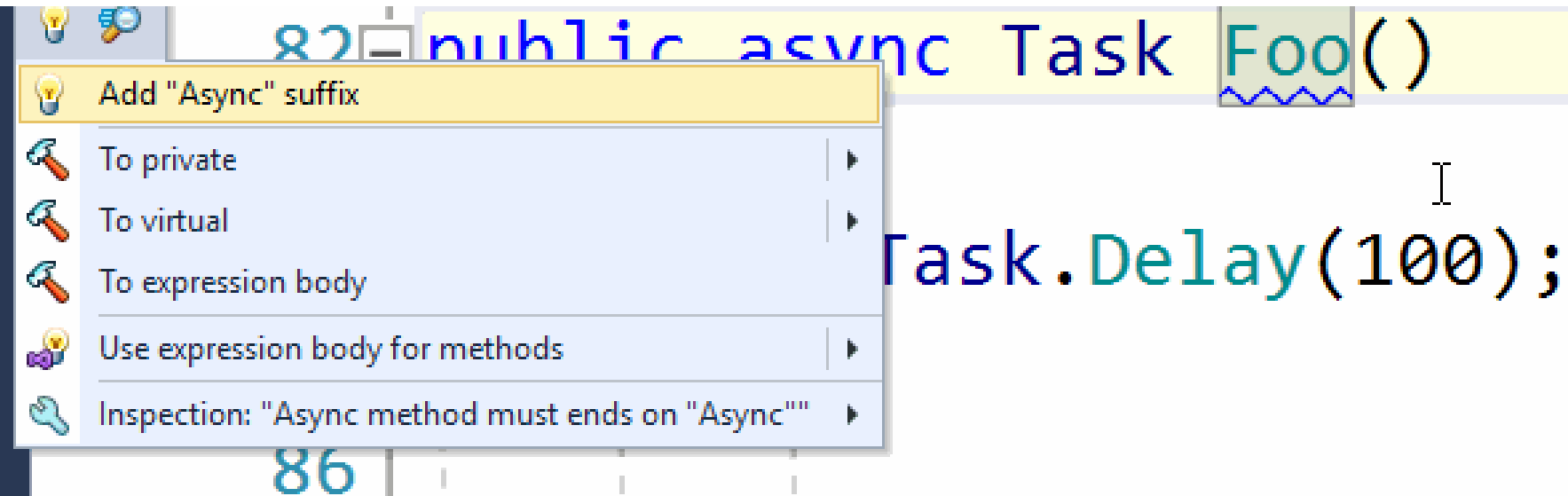
## Потенциальная ошибка



```
79 public Task<object> FooAsync()  
80 {  
81     return null;  
82 }  
83
```

# AsyncConverter

## Суффикс Async





# AsyncConverter

## Обязательный ConfigureAwait

```
81 public async Task FooAsync()  
82 {  
83     await Task.Delay(100);  
84 }  
85
```

# AsyncConverter

## Использование неправильного метода

```
82 public async Task BarAsync()  
83 {  
84     Foo();  
85 }  
86 public void Foo() ...  
89 public async Task FooAsync() ...
```







# AsyncConverter

Возможность убрать async/await

```
81 public async Task BarAsync()  
82 {  
83     await FooAsync().ConfigureAwait(false);  
84 }  
85  
86
```

# AsyncConverter

## Конвертация из синхронных в асинхронные

```
81  public async Task CallBarAsync()  
82 {  
83     Bar();  
84 }  
85  public void Bar()  
86 {  
87     Foo();  
88     FooAsync().Wait();  
89 }  
90  public void Foo()   
93  public async Task FooAsync() 
```

# AsyncConverter

## Конвертация из синхронных в асинхронные

- Заменить возвращаемый тип
- Добавить суффикс Async
- Добавить пространство имен
- Посмотреть все вызовы, если есть асинхронные аналоги, то заменить
- Заменить вызовы `.Result` и `.Wait()` на `await`
- Во всех точках вызовов либо вызвать `await`, либо `.Result`, либо `.Wait()`

# AsyncConverter

Sort by: name ▾

 AsyncConverter

INSPECTION

69 ★★★★★

one week ago

 Atlassian Connector

TOOLS INTEGRATION

467,681 ★★★★★

2 years ago

 Atom File Icons IDEA

UI

16,107 ★★★★★

one month ago

 AtSoundtrack

FUN STUFF

7,201 ★★★★★

one year ago

 AureliaStorm

FRAMEWORK INTEGRATION

4,523 ★★★★★

6 months ago

 auto generate log

CODE EDITING

727 ★★★★★

9 months ago

INSPECTION

AsyncConverter

 Install

★★★★★ 69 downloads

Updated 07/04/2018 v1.1.8.8

Plugin for converting code to async.

**Change Notes**

fix issue 25. Don't suggest "Async" suffix on methods marked "override"

**Vendor**

i.mamay

<https://github.com/BigBabay/AsyncConverter>

**Size**

24.5 K

Updated (1)



AsyncConverter 1.1.8.8

Update

Uninstall

Skip

Plugin for resharper, for converting code to async. Currently installed AsyncConverter 1.1.8.8 [More info...](#)

© i.mamay # async

# Автоматизируй это!

- Если повторяете что-то – может есть алгоритм
- Если есть алгоритм – можно автоматизировать
- Если автоматизируете – идите от простого
- Получайте удовольствие от автоматизации

	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf



	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf

	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf

	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf

	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf

	Roslyn	ReSharper
Написание внешней утилиты	✓	✗
Поддержка IDE	VisualStudio Rider(2018.1 EAP)	VisualStudio+ReSharper Rider
Удобство написания	Шаблон от MS Syntax Visualizer	Тестовый фреймворк PSI Viewer
Способ распространения	Локальная установка Референс в проект	Локальная установка Настройки TeamSettings
Интеграция с CI	Встраивается в билд	ReSharper CLT
Поддерживаемые технологии	C#, VB.NET	JetBrains products(dotPeek, dotMemory, dotTrace, dotCover) Языки C#, VB.NET, C++, HTML, ASP.NET, JavaScript, TypeScript, Angular 2 HTML, CSS, JSON, XML, XAML, MSBuild, Protobuf

<https://github.com/BigBabay/AsyncConverter>



**Мамай Игорь**  
mamay.igor@gmail.com  
<https://github.com/BigBabay><sup>38</sup>