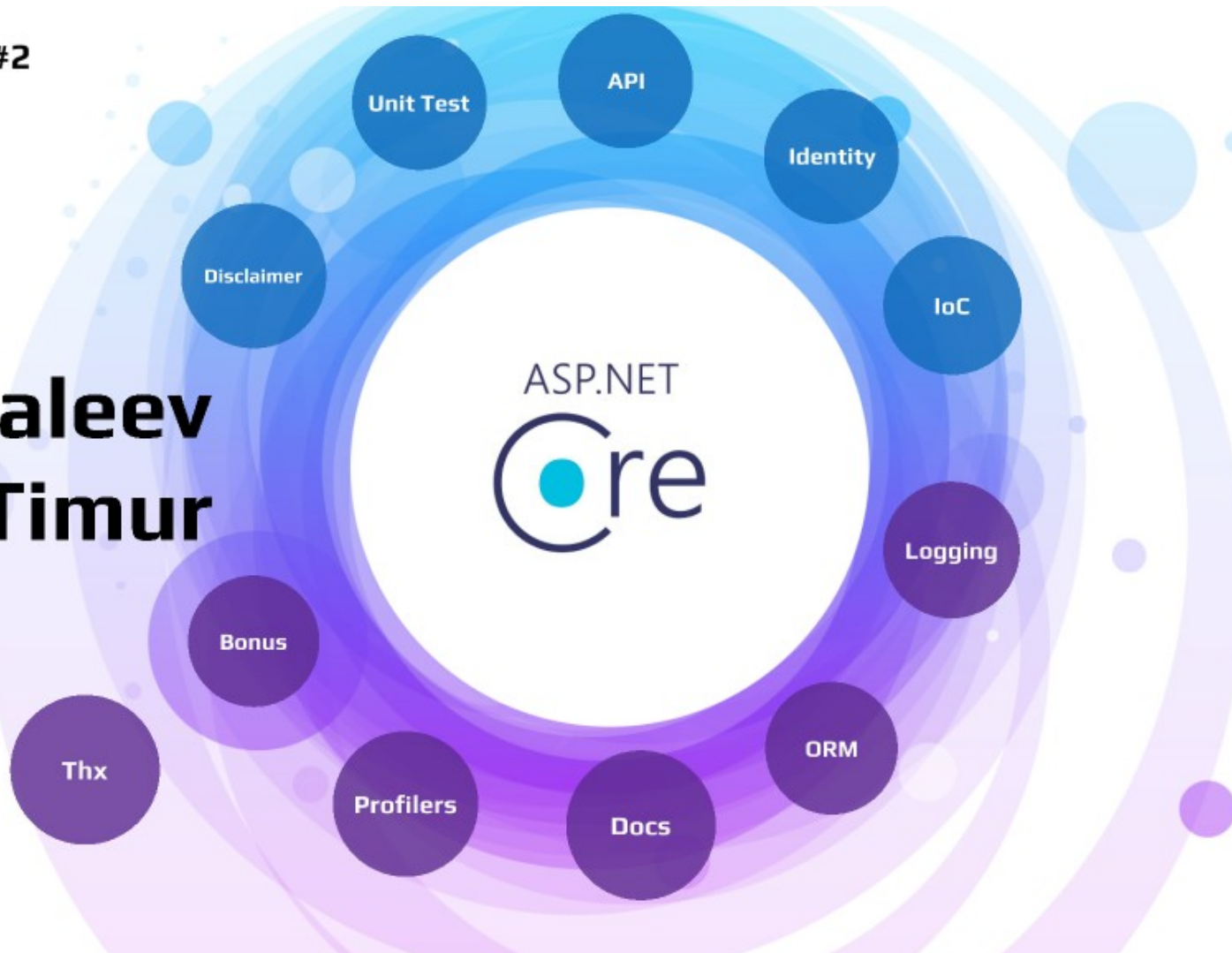


**Zamaleev  
Timur**



## Disclaimer



**РОСКОМНАДЗОР**

.NET Core

# .NET Core

- 7 мая конференция Microsoft Build
- анонс preview 2.1
- анонс 3.0



# Unit Test

- На данный момент основные игроки адаптировались к .NET Core
- Проблемы с тест ранерами устранены
- У nUnit не работает Code Coverage и Live Unit Testing (состояние на 16 февраля 2018 года)
- TypeMock не поддерживает тестирование .NET Core / .NET Standart проектов

AutoFixture

Fluent  
Assertions

# AutoFixture



- Mark Seemann aka ploeh прекратил работу над проектом
- Изменилось пространство имен
- Не все пакеты поддерживают .NET Core 2.0

# Fluent Assertions



- `everyone.Should().Like(FluentAssertio  
ns, "because everything is so damn  
cool and readable");`
- удобная библиотека для  
формирования Assert'ов



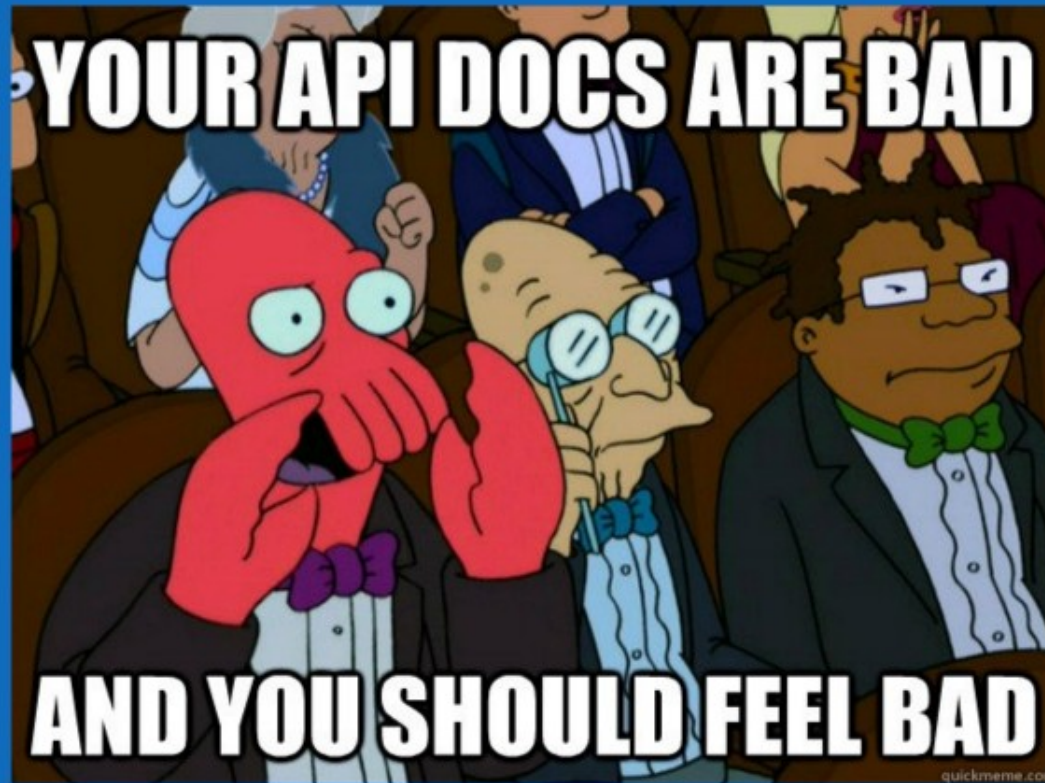
# API

- А что эта «ручка» возвращает?
- Сервер не работает...
- А точно работает? Ну перезагрузите пул пожалуйста...
- Какой content type?
- Вы @#\$!#% изменили формат ответа и не предупредили меня?!

Frontend  
Team

Swagger

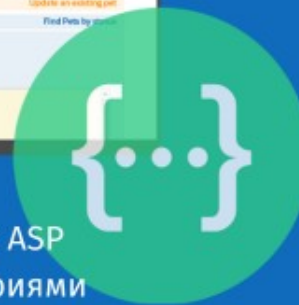
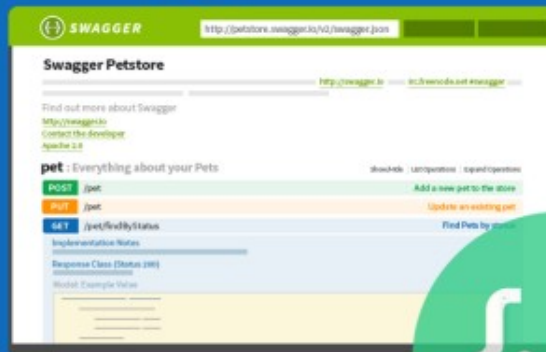
Postman



This.

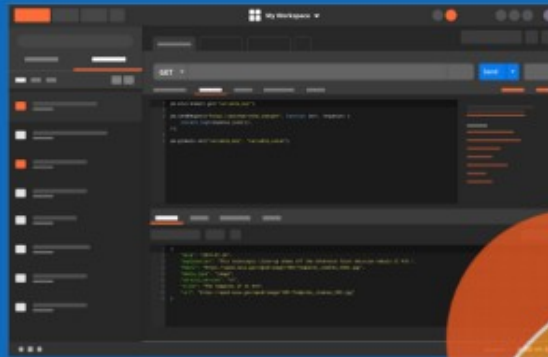


# Swagger



- Бесплатно
- Готовое решение для ASP
- Работает с комментариями
- Не поддерживает множественные сценарии авторизации
- Пришлось «форкнуть»

# Postman



- Условно бесплатно
- Публикация API коллекции доступна в платной версии
- Тест сценарии, переменные среды

# Identity

- Раковая опухоль ASP
- Кровавый Enterprise
- Ломает scaffolding
- AspNetUsers, AspNetRoles, AspNetUserClaims, AspNetUserLogins, AspNetUserRoles
- Username & NormalizedUsername, Email & NormalizedEmail

JWT

Identity  
Server

# JWT Bearer

- Стандарт RFC 7519, OAuth 2.0
- Позволяет создать собственную реализацию
- Быстро и легко добавить множественные сценарии авторизации
- Задавать claim'ы через стандартные атрибуты
- Бесплатная книга <https://auth0.com/resources/ebooks/jwt-handbook>

## Альтернативное решение



**identity**  
**SERVER**

- Бесплатно, Apache 2
- Открытый код
- Open ID
- Для больших проектов

# IoC



- Да, действительно появился встроенный
- Работает из коробки, критичных проблем не обнаружено
- Подходит для большинства задач
- Не самый быстрый, но и не самый медленный

Battle

Basic  
features

Advanced  
features

Prepare

Dry IoC



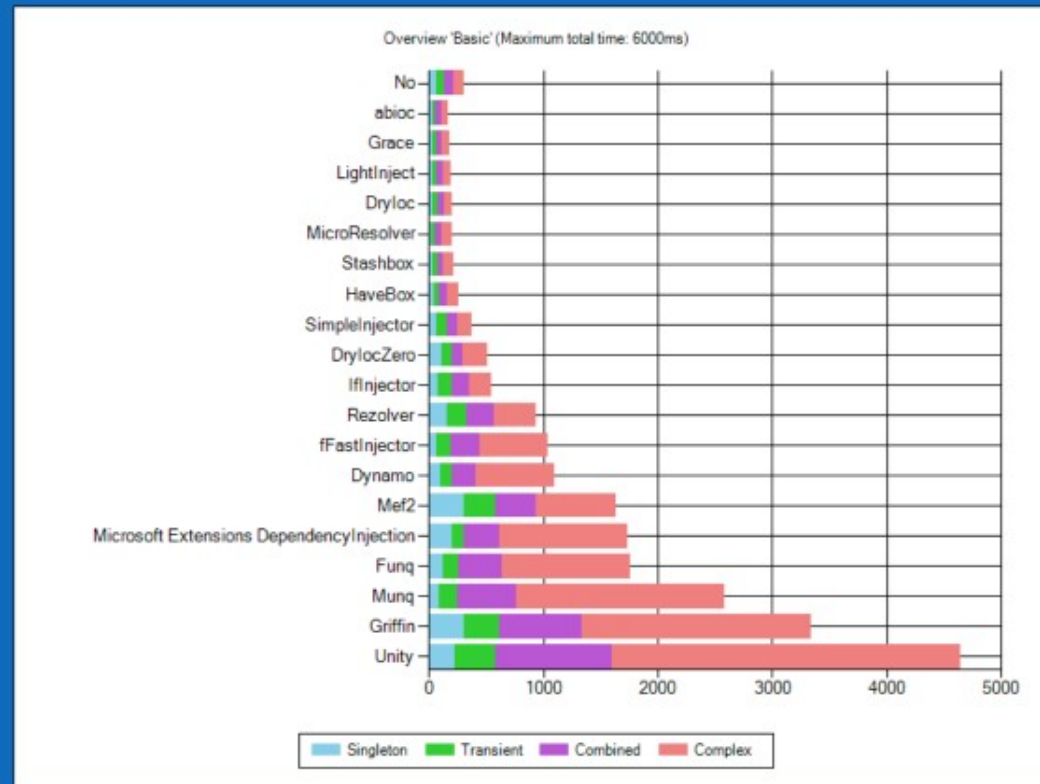
# IoC Battle

- <http://www.palmmedia.de/Blog/2011/8/30/ioc-container-benchmark-performance-comparison>
- Дата обновления 04.03.2018
- The benchmarks: Singleton, Transient, Property, Generics, IEnumerable, Conditional, Child Container, Interception With Proxy и другие

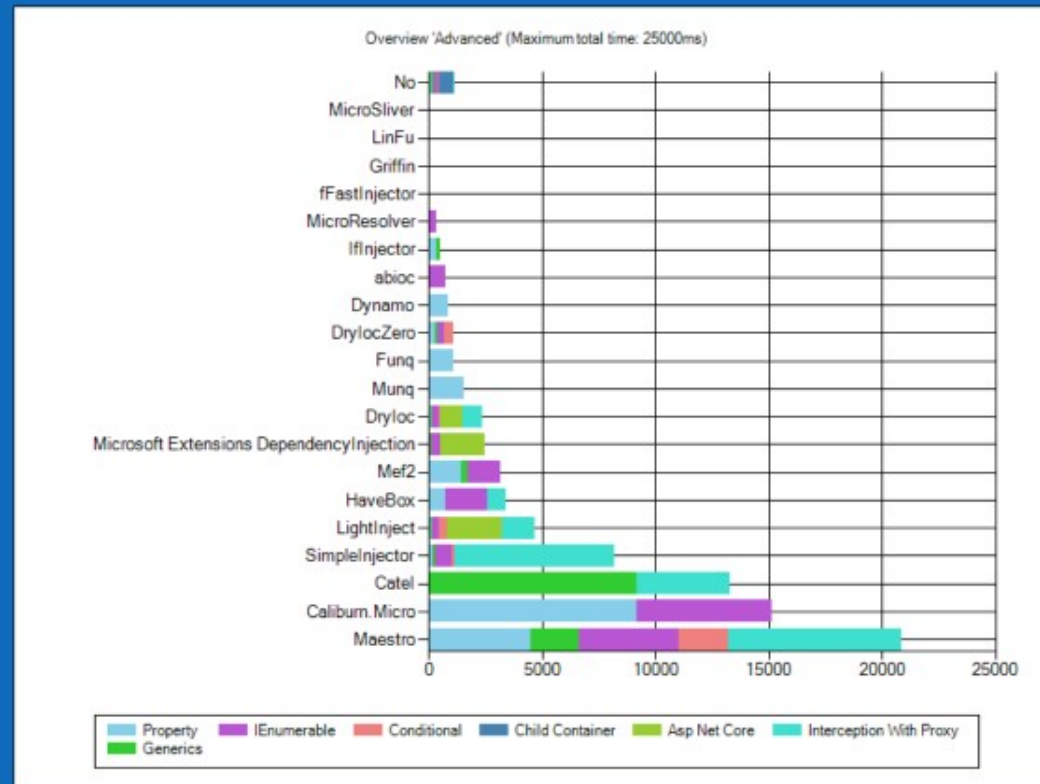


Daniel Palme

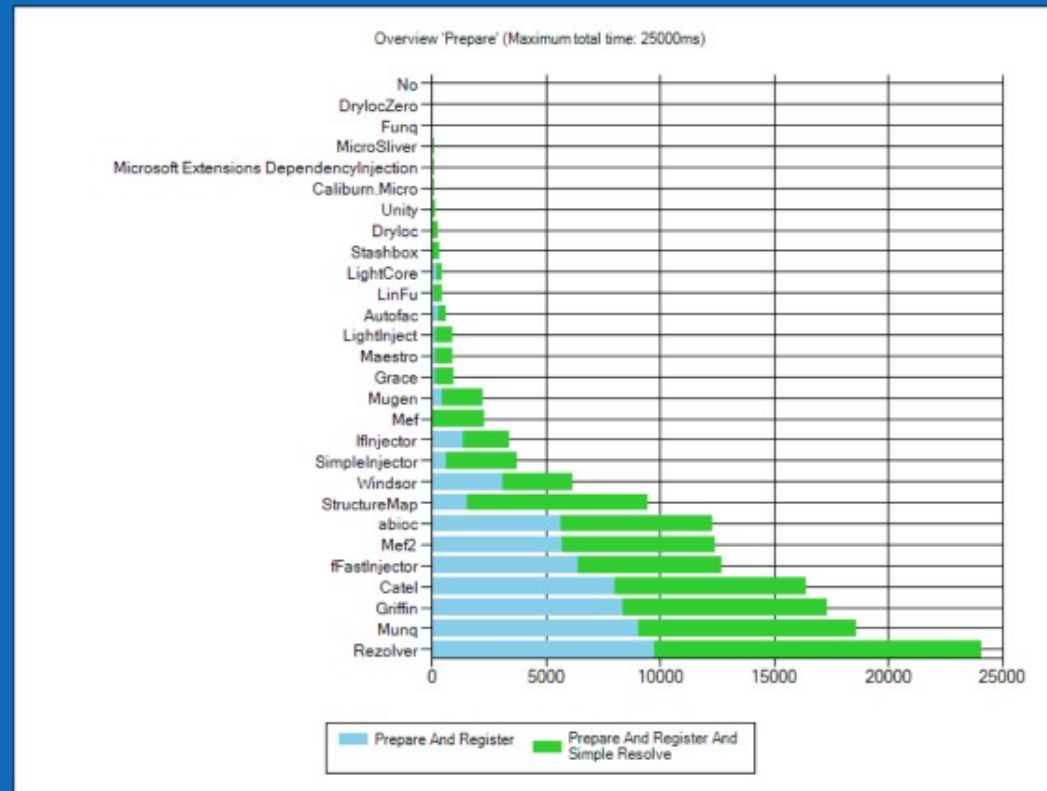
.NET consultant from Germany  
<https://twitter.com/danielpalme>



# Basic features



# Advanced features



# Prepare

# Dry IoC

- <https://bitbucket.org/dadhi/dryioc>
- Открытый код
- Быстрый, быстрее чем new()
- Внедрение через MEF атрибуты
- Предоставляет много возможностей:

```
this.Container.RegisterDelegate(x =>  
x.Resolve<ILoggerProvider>().Logger);  
this.Container.RegisterMapping<ILoggerProvider,  
ILoggerFacade>();
```



Maksim Volkau

Passionate .NET Software Developer  
<https://twitter.com/dadhi>

# Logging

- Имеется готовое решение Microsoft.Extensions.Logging и Microsoft.Extensions.Logging.Abstractions
- Все еще нет нормального фасада, известное решение <https://github.com/net-commons/common-logging> имеет много недостатков

nLog

serilog



## nLog

- Проверенный годами инструмент
- Широкие возможности конфигурации
- Поддерживает «структурное логирование» aka <https://messagetemplates.org/>
- Позволяет вести записи в файлы, системный журнал, базы данных, консоль
- Имеется поддержка буферизации и асинхронной записи



# serilog

- Быстрая и легкая конфигурация
- Низкий порог вхождения
- Поддерживает «структурное логирование» aka <https://messagetemplates.org/>
- Предоставляет огромное количество провайдеров записи aka sinks, например akka, Amazon, различные БД, glimpse



## EF Core

- В сравнение с EF 6 не весь функционал до сих пор поддерживается
- <https://docs.microsoft.com/en-us/ef/efcore-and-ef6/features>
- Полностью переписан с нуля, упор делался на возможности расширения
- Ждем 2.1
- Субъективно: обновлять существующие проекты не сложно, зависит от качества кода

nHibernate

dapper

# nHibernate

- Поддерживает .NET Core с декабря 2017 года
- Большое количество issue
- Зачем?



# dapper

- Микро ORM, object mapper
- Минимальный порог освоения
- Транзакции
- Платная библиотека  
Entity Framework Extensions



## Doc FX

- Авто генерация документации на основании исходного кода и комментариев
- Кроссплатформенный
- Использование из VS либо CI сервера
- Поддерживает swagger
- Огромные возможности по расширению, настройке
- Особенно полезно если надо передать проект другой команде, отдать заказчику

Sandcastle



# Sandcastle

- Поддерживает .NET Core
- Меньше функционала, CHM
- Поддерживает nDoc
- Зачем?

# Glimpse

- Разработка для .Net Core ведется в другом репозитории
- На данный момент с 25 февраля ожидает слияния pull request для работы с .NET Core 2.0
- Проект выглядит заброшенным, сайт проекта не работает :(



## Знакомо?

```
public static void ForEach<T>  
    (this IEnumerable<T> source,  
     Action<T> action) {  
    foreach (T element in source)  
        action(element);  
}
```

**One nuget package to rule them all**



R

S

D

N

# RSDN Code Jam

- Набор готовых и наиболее востребованных расширений
- Имплементации алгоритмов
- Все то, что вы обычно ищите на SO
- С хорошим покрытием тестами



# RSDN Code Jam

- CodeJam.Arithmetic
- CodeJam.Collections
- CodeJam.Dates
- CodeJam.Expressions
- CodeJam.IO



# RSDN Code Jam

- CodeJam.Mapping
- CodeJam.Metadata
- CodeJam.Ranges
- CodeJam.Reflection
- CodeJam.Services





# RSDN Code Jam

- CodeJam.Strings
- CodeJam.TableData
- CodeJam.Threading
- CodeJam.Xml



# Спасибо

Для связи:

- ra.public@in-tech.org
- <https://twitter.com/radevops>

