



ASP.NET MVC: OWIN & KATANA

2017

MICROSOFT WEB HISTORY

- 1996: ASP — VBscript на странице. никакого [SOC](#).
- 2002: ASP.NET
 - Архитектурное разделение:
 - Presentation (ASPX)
 - Business logic (C#/VB.NET)
 - Сниженный порог вхождения для WinForms-разработчиков
 - Полное сохранение состояния между запросами
 - Монолитный framework (никакой модульности)
 - Привязан к релизам .NET Framework

MICROSOFT WEB HISTORY

- 2009: ASP.NET MVC
 - Архитектурный подход **Model-View-Controller**, SOC!
 - Не входит в базовую поставку **.NET Framework** → Независимый
 - Ускоренный релизный цикл (По сравнению с ASP.NET)
 - [Open Source!](#)
 - Сильная связанность с **System.Web.dll** → зависит от IIS

MICROSOFT WEB HISTORY

- 2012: ASP.NET Web Api
 - *“Уход от динамических, генерируемых сервером, веб-страниц к страницам со статической базовой разметке с динамическими секциями, генерируемыми на стороне клиента при помощи AJAX-запросов к Web Api”*
 - Поставляется через NuGet
 - Нет зависимостей от **System.Web.dll** → больше модульности
 - Custom host! Разработчики могут использовать легковесные решения для хостинга своих сервисов → **нет привязки к IIS**

MICROSOFT WEB HISTORY

- 2013: ASP.NET SignalR
 - [Вики](#): “ASP.NET SignalR is a library for [ASP.NET](#) developers to add real-time web functionality to their applications. Real-time web functionality is the ability to have server-side code push content to the connected clients as it happens, in real-time.”
 - Не имеет привязки к конкретному хосту, так же как ASP.NET Web Api

WHY NOT HOSTING ON IIS?

- Использование маленьких, легковесных веб-серверов
- Нет привязки к серверу → нет прямой привязки к платформе
- Нет нужды использовать впечатляющую, но громоздкую функциональность IIS
- В определенных случаях, IIS может быть недоступен

HOW?

Если вы хотите разработать приложение, которое можно будет запустить, используя разные хосты и веб-серверы — вы не можете использовать компоненты конкретных реализаций для сервера (`System.Web.dll`).

Здесь на помощь приходит OWIN.

OWIN

- OWIN расшифровывается как: **The Open Web Interface for .NET**
- На создание был вдохновлен [Rack](#) из Ruby
- OWIN описывает абстракцию между веб-сервером и компонентами приложения/фреймворка
- Не имеет никакой привязки к конкретному фреймворку/платформе
- Имеет всего два базовых элемента:
 - **Environment dictionary**
 - **Application delegate**

OWIN. ENVIRONMENT DICTIONARY

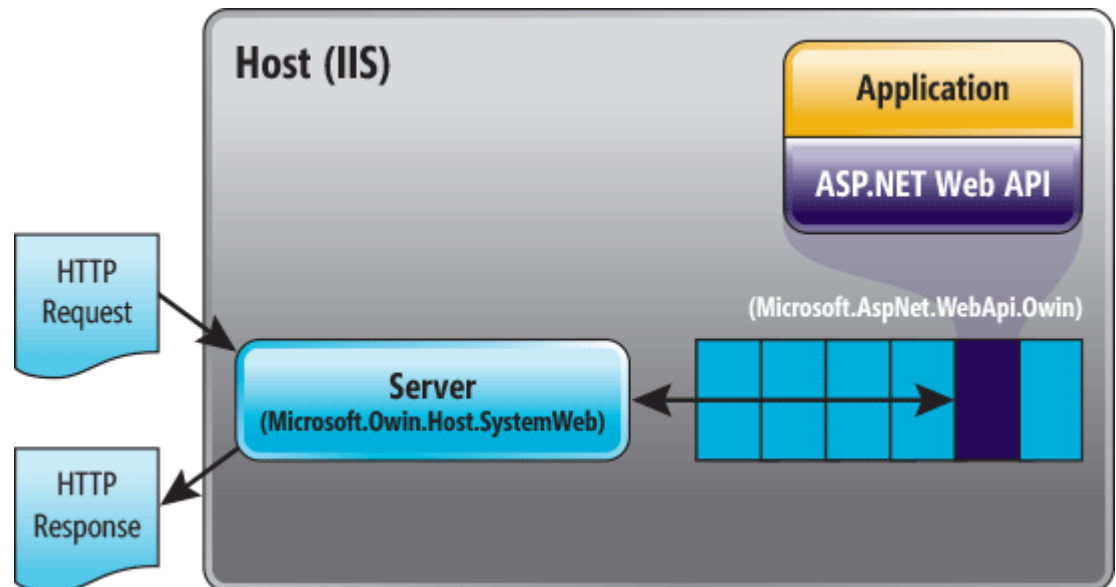
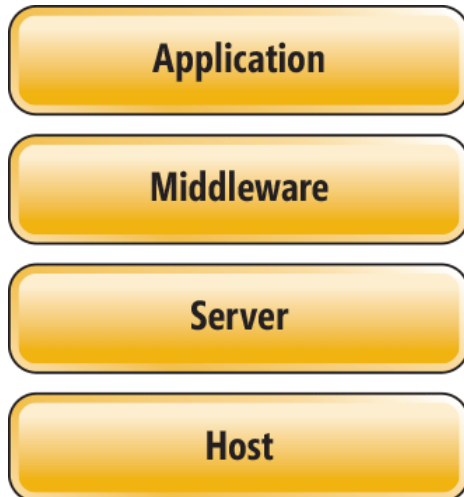
- `IDictionary<string, object>`
- “Любой OWIN-совместимый веб-сервер ответственен за **наполнение environment dictionary** данными, такими как входной поток (*request body stream*) и коллекция заголовков для HTTP запросов и ответов. Далее следует обязанность приложения или компонентов фреймворка **заполнить или обновить environment dictionary** дополнительными значениями и записать результат в выходной поток (*response body stream*)”

OWIN. APPLICATION DELEGATE

- `Func<IDictionary<string, object>, Task>`
- “*Application delegate* — реализация *функционального делегата*, который принимает *environment dictionary* на вход и возвращает *Task*”
- Асинхронная модель позволяет абстракции быть более эффективной в использовании **вычислительных ресурсов**, в частности, при выполнении операций **ввода/вывода**
- Поскольку `application delegate` является атомарной единицей исполнения и поскольку `environment dictionary` передается, как входной параметр делегата, компоненты OWIN могут быть очень легко построены цепочкой для создания комплексных путей обработки HTTP запросов

OWIN. LAYERS AND COMPONENTS

- **Host:** процесс, который запускает приложение (IIS/custom).
- **Server:** Отвечает за привязку к TCP порту, формирование environment dictionary и обработку запросов через OWIN-pipeline.
- **Middleware:** имя, которое дается всем компонентам, которые участвуют в обработке запроса через OWIN pipeline.
 - Попросту: компонент, предоставляющий application delegate.
- **Application:** Ваш код!

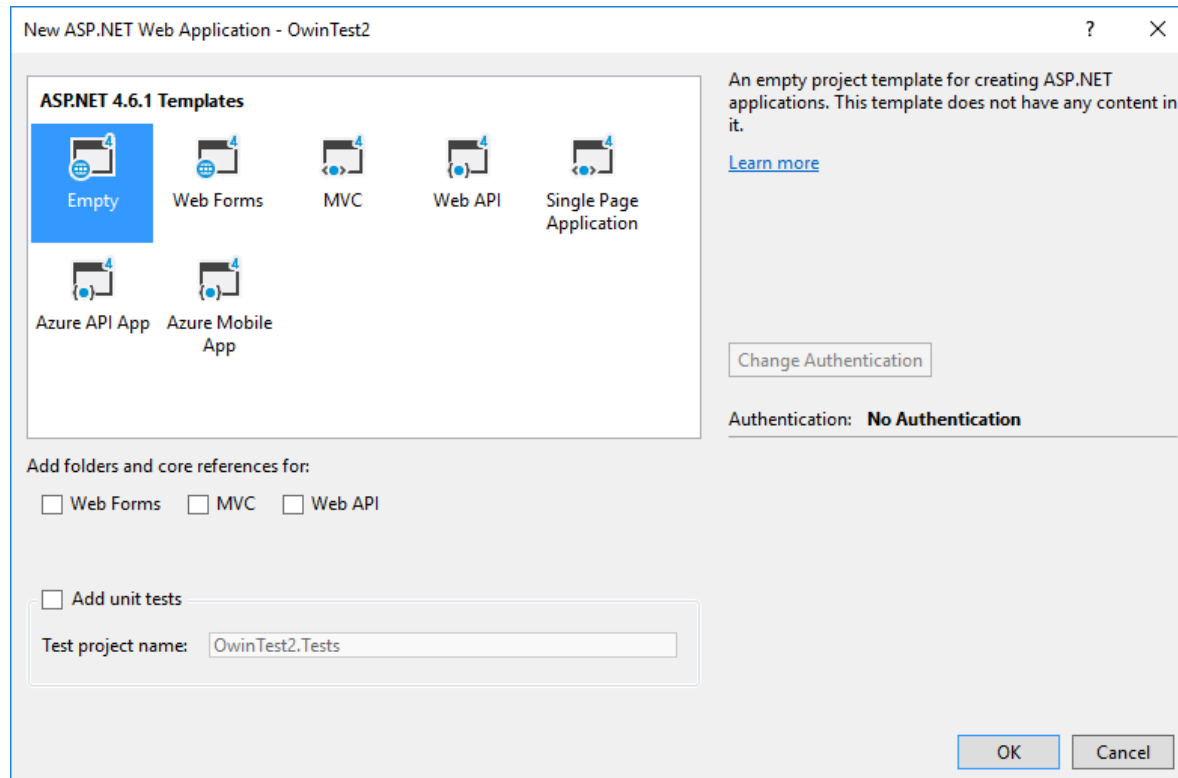


OWIN. IMPLEMENTATIONS & IMPLEMENTORS

Serves & Hosts	Frameworks
<u>Katana</u>	<u>Jasper</u>
<u>Nowin</u>	<u>Nancy</u>
<u>Suave</u>	<u>SignalR</u>
<u>Freya</u>	<u>WebApi</u>
<u>ASP.NET vNext</u>	<u>WebSharper</u>
	<u>DuoVia.Http</u>
	<u>Simplify.Web</u>

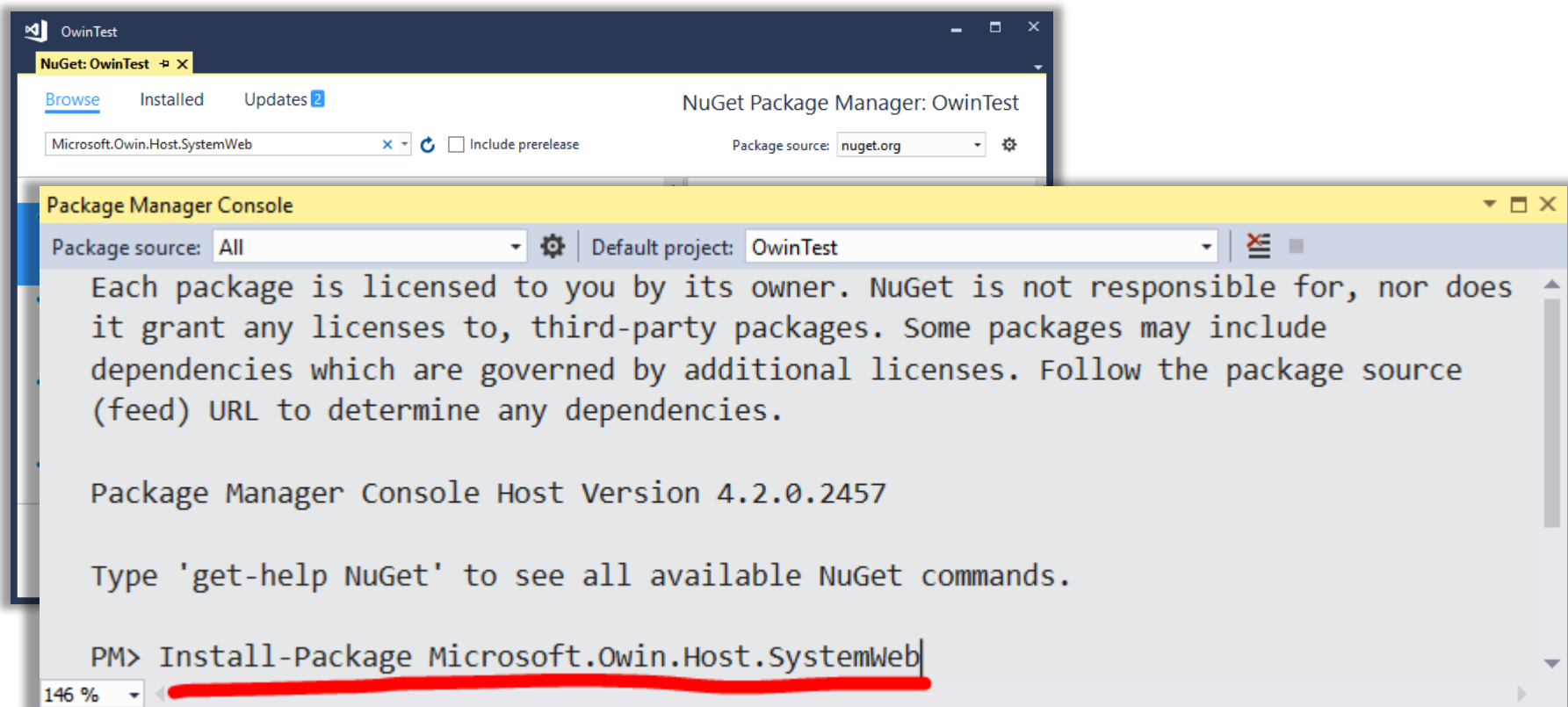
OWIN TUTORIAL. IIS HOST

- Создайте новое веб-приложение, используя шаблон **Empty**



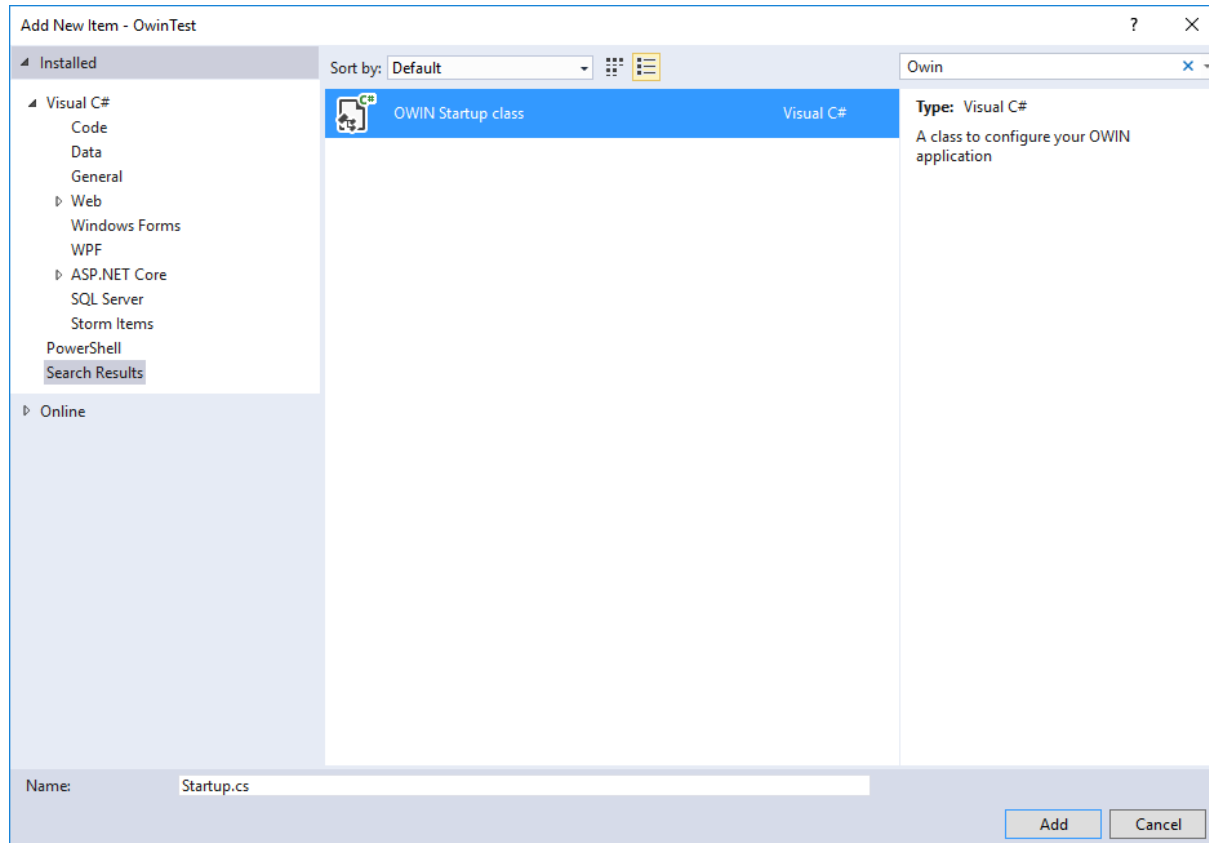
OWIN TUTORIAL. IIS HOST

- Создайте новое веб-приложение, используя шаблон Empty
- Подключите NuGet-пакет «Microsoft.Owin.Host.SystemWeb»



OWIN TUTORIAL. IIS HOST

- Создайте новое веб-приложение, используя шаблон **Empty**
- Подключите NuGet-пакет «**Microsoft.Owin.Host.SystemWeb**»
- Добавьте **OWIN Startup Class** в проект



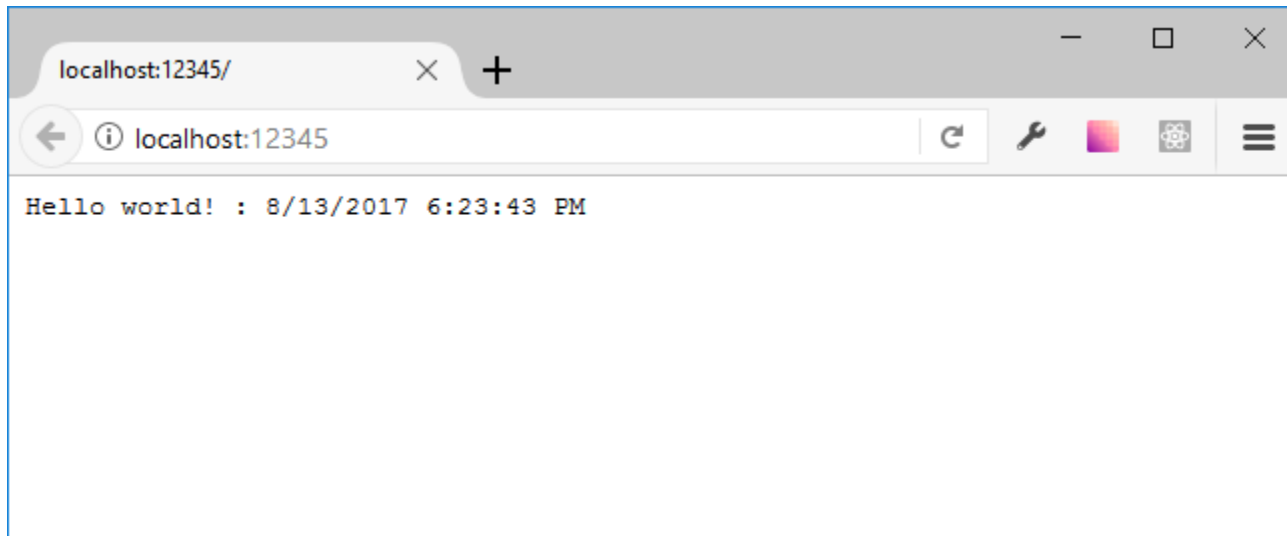
OWIN TUTORIAL. IIS HOST

- Создайте новое веб-приложение, используя шаблон **Empty**
- Подключите NuGet-пакет «**Microsoft.Owin.Host.SystemWeb**»
- Добавьте **OWIN Startup Class** в проект
- Замените метод `Configuration()` следующей реализацией

```
public void Configuration(IAppBuilder app)
{
    app.Run(context =>
    {
        context.Response.ContentType = "text/plain";
        return context.Response.WriteAsync($"Hello world! : {DateTime.UtcNow}");
    });
}
```


OWIN TUTORIAL. IIS HOST

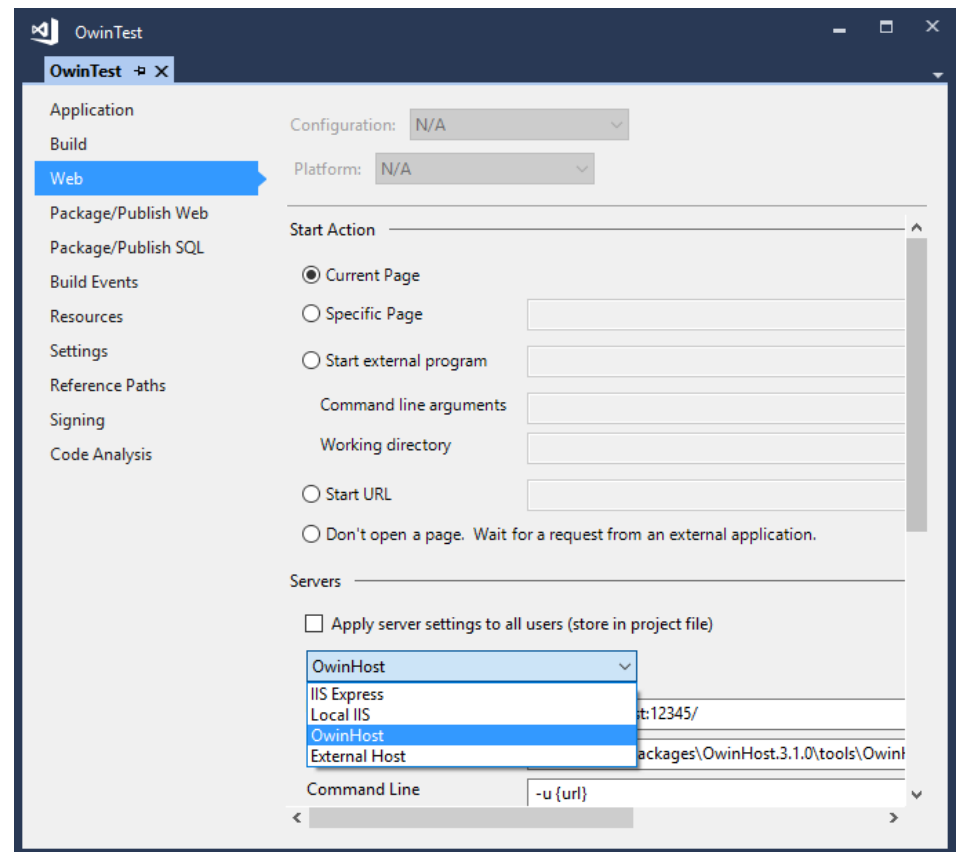
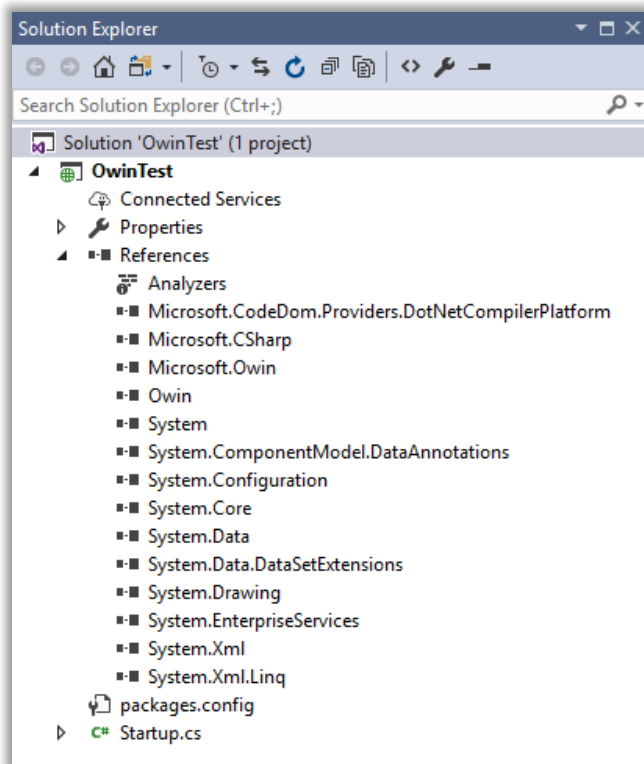
- Нажмите F5



- Вы восхитительны!

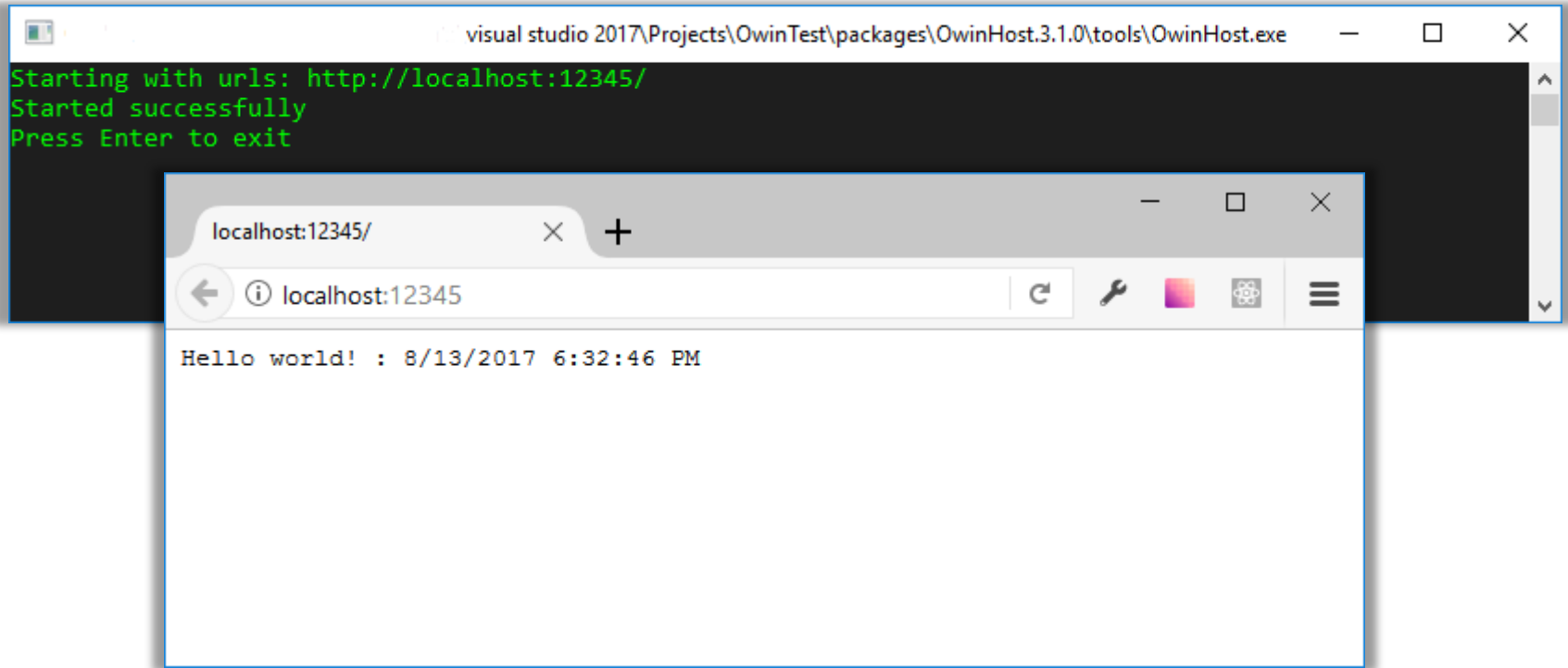
OWIN TUTORIAL. OWINHOST

- Подключите NuGet-пакет «OwinHost». Он загрузит OwinHost.exe в папку *Packages*.
- Удалите все упоминания о «Microsoft.Owin.Host.SystemWeb» и «System.Web»
- Смените сервер в настройках проекта



OWIN TUTORIAL. OWINHOST

- Нажмите Ctrl-F5 (Debug режим невозможен без app.config/web.config)



- Наслаждайтесь скоростью запуска и формирования ответа!

ANY QUESTIONS?



**THANKS FOR YOUR
ATTENTION!**