

OWIN & KATANA

최흥배

<https://github.com/jacking75/choiHeungbae>

이 문서는 마지막에 있는 '참고' 리스트에 있는 글들을 많이 참고하여 정리한 것입니다.

.NET 웹 프로그래밍...

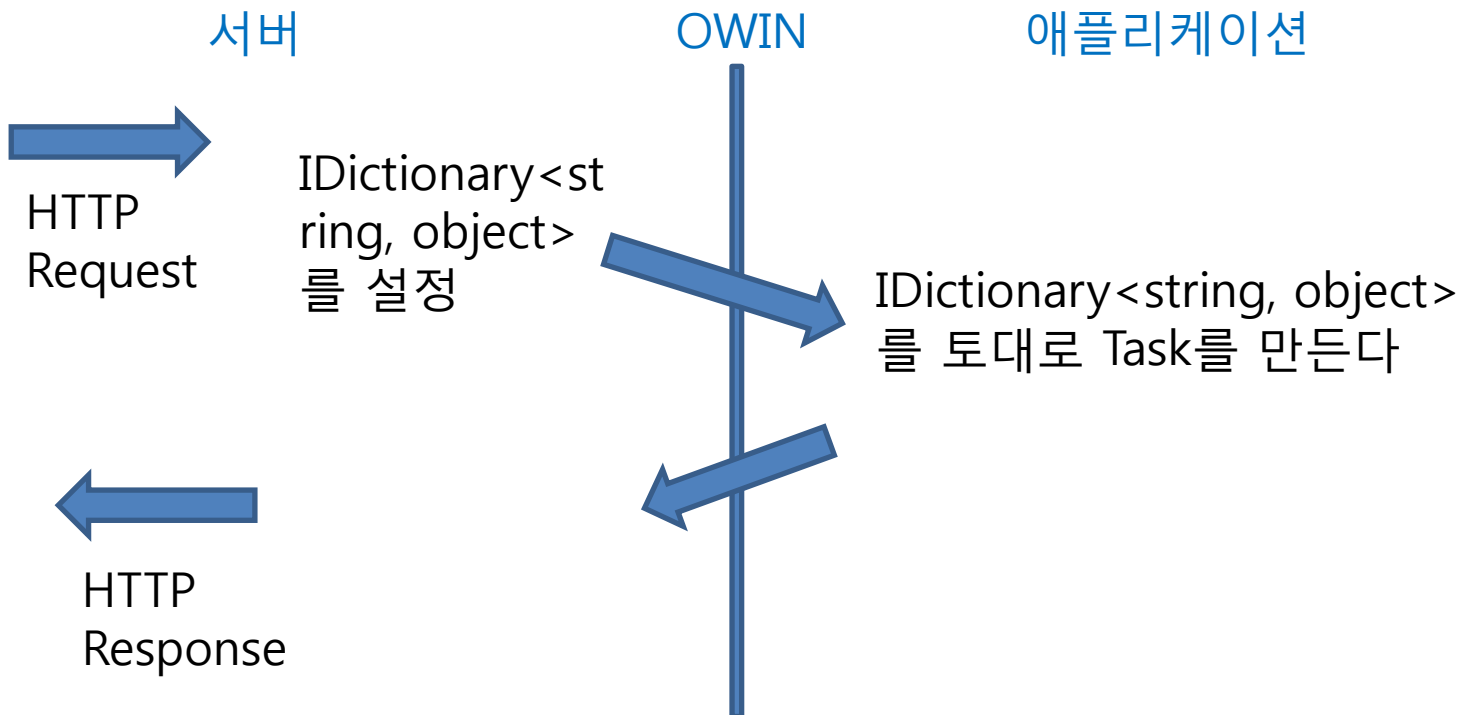
- .NET에서는 HTTP Server 프로그래밍 추상화 레이어는 ASP.NET 초기에 구축된 후 지금까지 거의 변화가 없었음.
- ASP.NET 초기의 주 목적은 'Classic ASP를 사용하고 있는 사람'과 'VB6 등에서 Windows로 업무용 앱을 개발하고 있는 사람'에게 Web 플랫폼 프로그래밍을 제공하는 것.
.NET Framework의 일부로 출시되기 위해 부족한 시간으로 만들었다.
- 이렇게 해서 나온 것이 기존의 VB6 앱에 익숙한 이벤트 모델을 베이스로 한 Web Forms 아키텍처와 논리적으로 다른 HTTP Object와 Web Forms Framework가 강하게 결합된 System.Web.dll
- ASP.NET은 초기 이후 기존 유산을 잘 활용은 했지만 새로운 기술 도입이 드뻔함. Ruby는 표준적인 Web 서버와 Web 앱을 연결하기 위해 수 많은 Web 서버와 프레임워크가 생기는 것을 해결하기 위해 Rack이 나옴.
- .NET은 처음부터 표준적인 Web 서버(IIS)와 Framework(System.Web.dll)이 있어서 혼란은 없었지만 반대로 자유로운 발전에 방해가 되었고, Ruby나 Python처럼 혼란은 있었지만 덕분에 뛰어난 표준(Rack/WSGI)가 나옴.

OWIN ???

- OWIN: Open Web Interface for .NET.
공식 사이트: <http://owin.org/>
- 서버와 애플리케이션 간의 인터페이스 사양.
 - HTTP Server 프로그래밍 추상화 레이어를 정의
 - 대부분이 사양서이며, IApplicationBuilder 인터페이스만 소스 코드
 - <https://github.com/owin/owin/tree/master/src>
 - NuGet을 통해 Owin.dll을 얻을 수 있다.
- MS가 주도하는 것이 아닌 커뮤니티에 의한 오픈 소스 프로젝트.
- node.js, Rack, WSGI에서 영향을 받음.
Rack: a Ruby Webserver Interface Feb 2007 <http://rack.github.io/>
- 서버와 애플리케이션을 분리하여 새로운 컴포넌트가 지금 보다 더 간단하게 개발되고, 더 쉽게 사용되도록 한다.
- 애플리케이션 이식을 쉽게 한다.
- 출시된 프레임워크. Web API, SignalR, Nancy

OWIN 사양

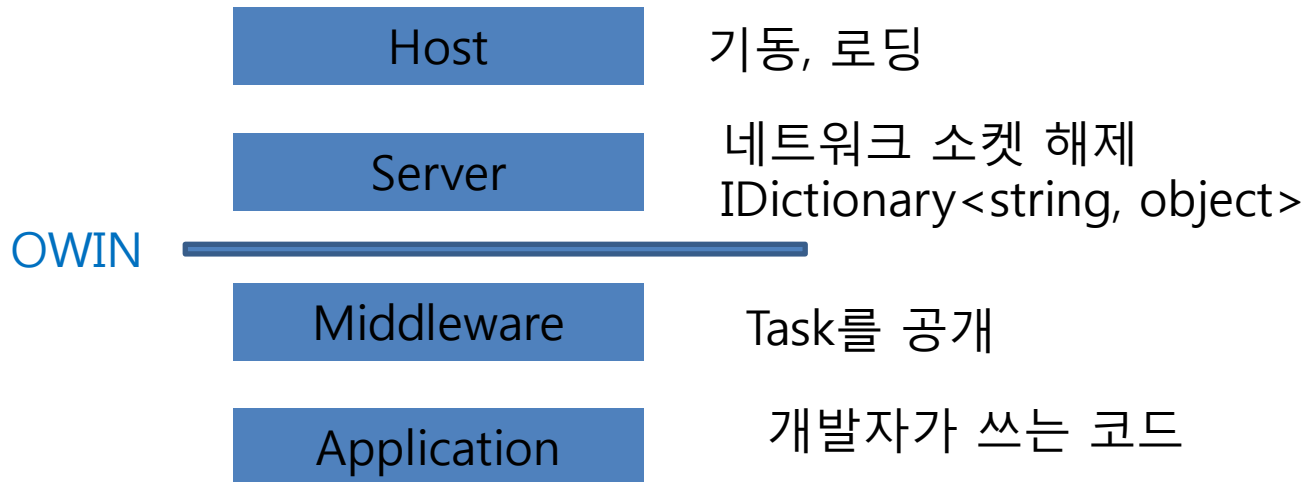
- `using AppFunc = Func<IDictionary<string, object>, task>;`
- `IDictionary<string, object>` "Environment Dictionary"
 - key와 값으로 리퀘스트/서버 등의 상태를 나타낸다.
- Task
 - 애플리케이션 이 담당하는 태스크. 비동기, 연결 가능.



KATANA

- OWIN 사양을 토대로 실제 구현한 것.
- MS에 의한 오픈 소스 프로젝트.
<http://katanaproject.codeplex.com/>
NuGet을 통해서 얻을 수 있음.
- OWIN 대응을 위해 컴포넌트/프레임워크 모음.
Microsoft.OWin 이름 공간에 라이브러리가 모여 있다.
- 지금까지의 ASP.NET은 IIS(System.Web)에 너무 의존.
IIS와 분리하여 ASP.NET의 웹 개발 진화에 대응
- 컴포넌트 재 이용성
인증, 캐시, 로깅 등
- 당근 Azure에도 대응.

KATANA - 구현



Host	IIS/ASP.NET, Custom 콘솔 App, OwnHost.exe ...
Server	IIS/ASP.NET (system.Web), Microsoft.Owin.Host.systemWeb, Microsoft.Owin.Host.HttpListener ...
Middleware	Nancy, Web API, SignalR, Microsoft.Owin.Security.Google ...
Application	우리 코드

검은 글자색의 컴포넌트는 Katana에 포함된 것.

KATANA - 호스팅 라이브러리

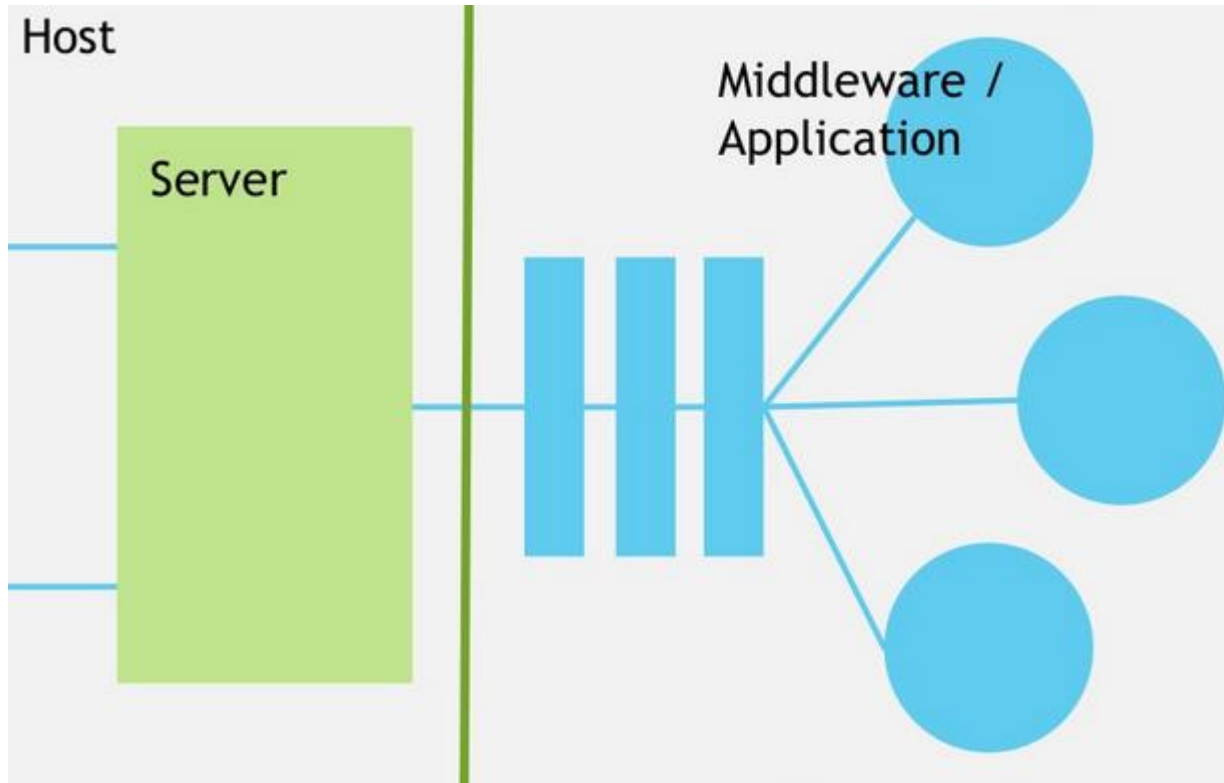
- Microsoft.Owin.Host.SystemWeb
OWIN을 ASP.NET 상에서 사용하기 위한 라이브러리
- Microsoft.Owin.Host.HttpListener
OWIN을 HttpListener 상에서 사용하기 위한 라이브러리
- Microsoft.Owin.Host.IIS
OWIN을 IIS 상에서 사용하기 위한 라이브러리

KATANA - 목표

- Portable
컴포넌트는 치환 할 수 있다.
- Modular / Flexible
기능은 작고, 컴팩트, 빼내기 편하게
- Lightweight / Performance / Scalable

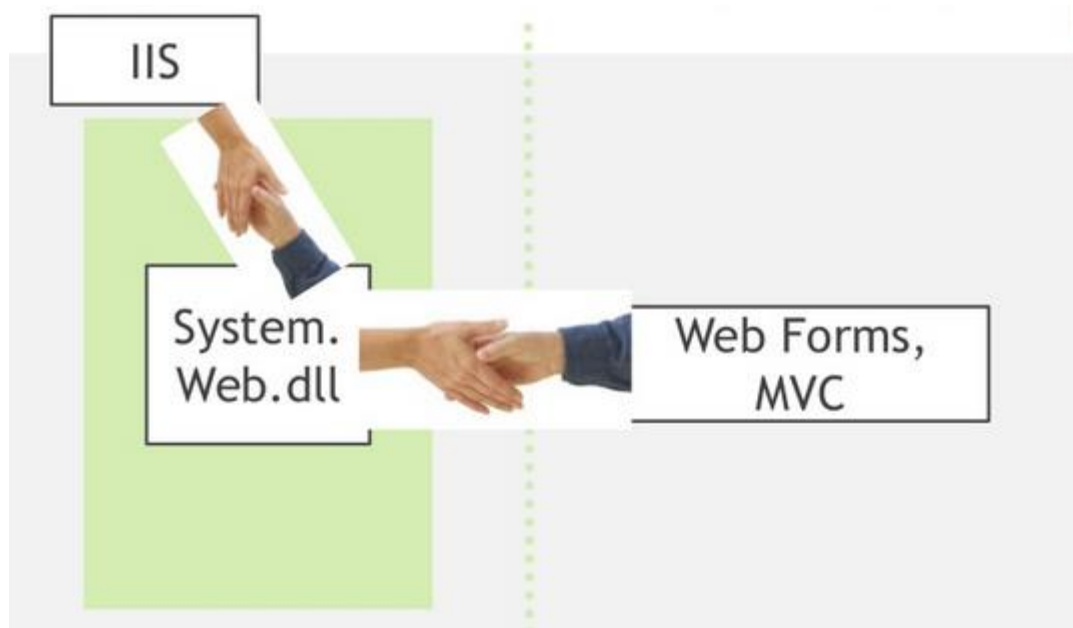
KATANA

OWIN



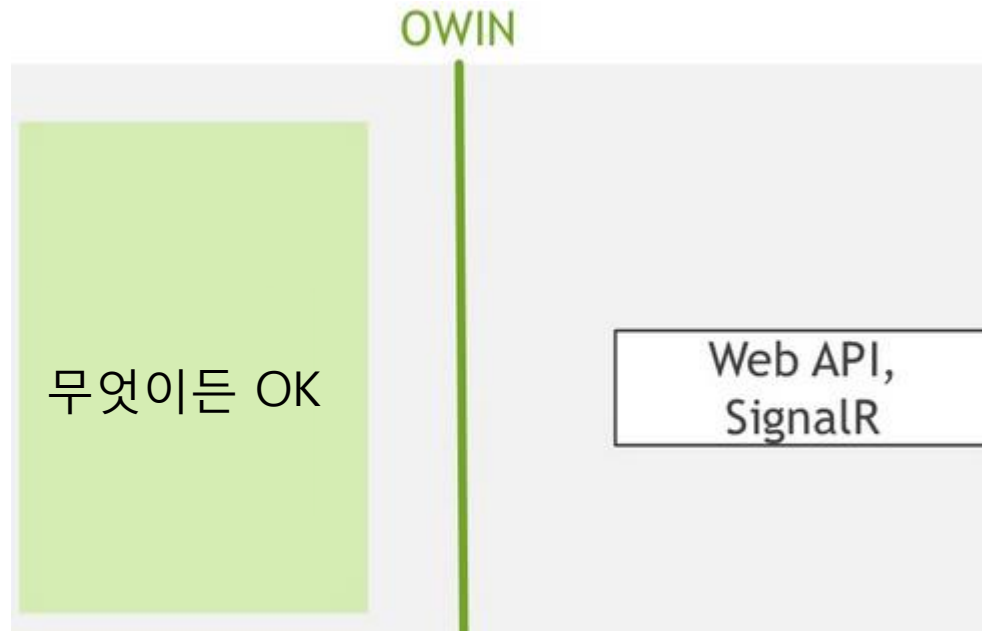
OWIN&KATANA - ASP.NET

- 현재(2014.02)는 ASP.NET Web Forms, MVC는 OWIN 비 대응.



OWIN&KATANA - ASP.NET

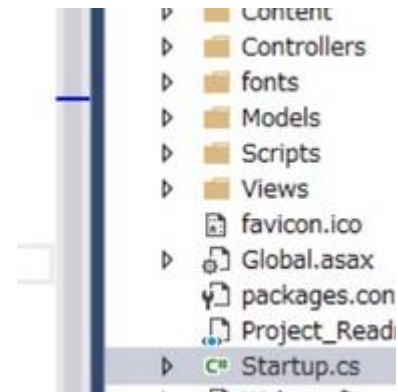
- 현재(2014.02)는 ASP.NET Web API, SignalR은 OWIN 대응.



OWIN&KATANA - ASP.NET

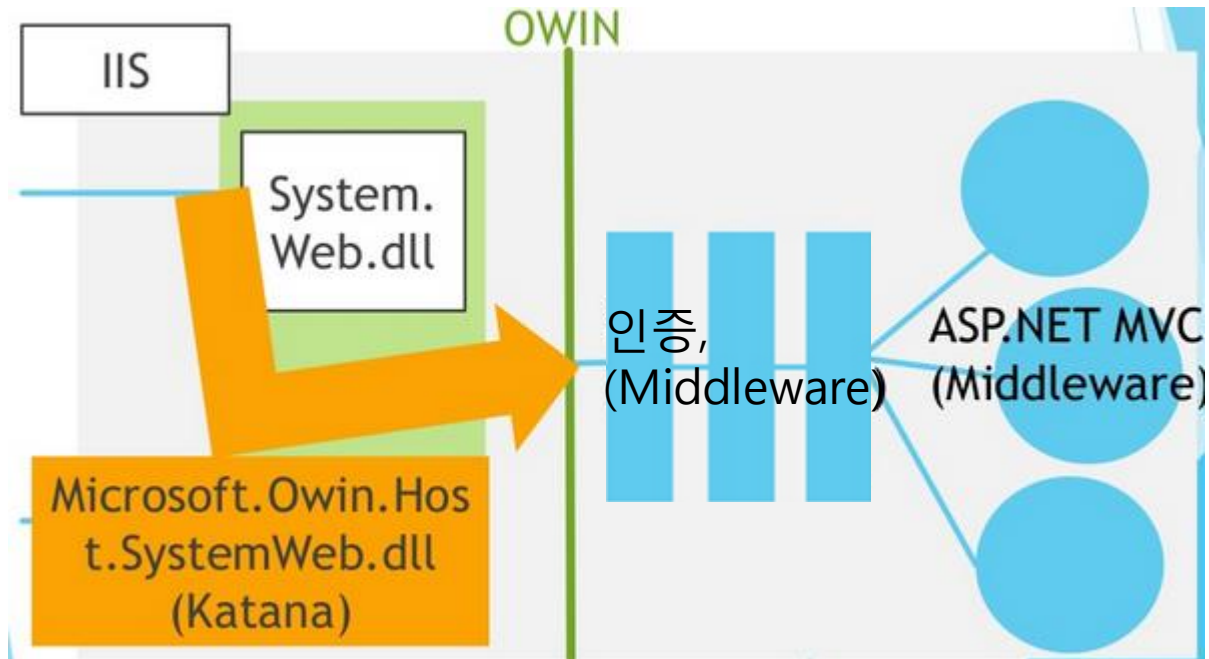
- 그렇지만 ASP.NET MVC 5의 코드를 보면 OWIN 관련 코드가 있음!

```
using Microsoft.Owin;  
using Microsoft.Owin.Security.Cookies;  
using Owin;  
  
namespace WebApplication29  
{  
    public partial class Startup  
    {  
        // 認証設定の詳細については、http://go.microsoft.com/fwlink  
        public void ConfigureAuth(IAppBuilder app)  
        {  
            // アプリケーションが Cookie を使用して、サインインした  
            app.UseCookieAuthentication(new CookieAuthenticati  
            {  
                AuthenticationType = DefaultAuthenticationType  
                LoginPath = new PathString("/Account/Login")  
            });  
            // Use a cookie to temporarily store information a
```



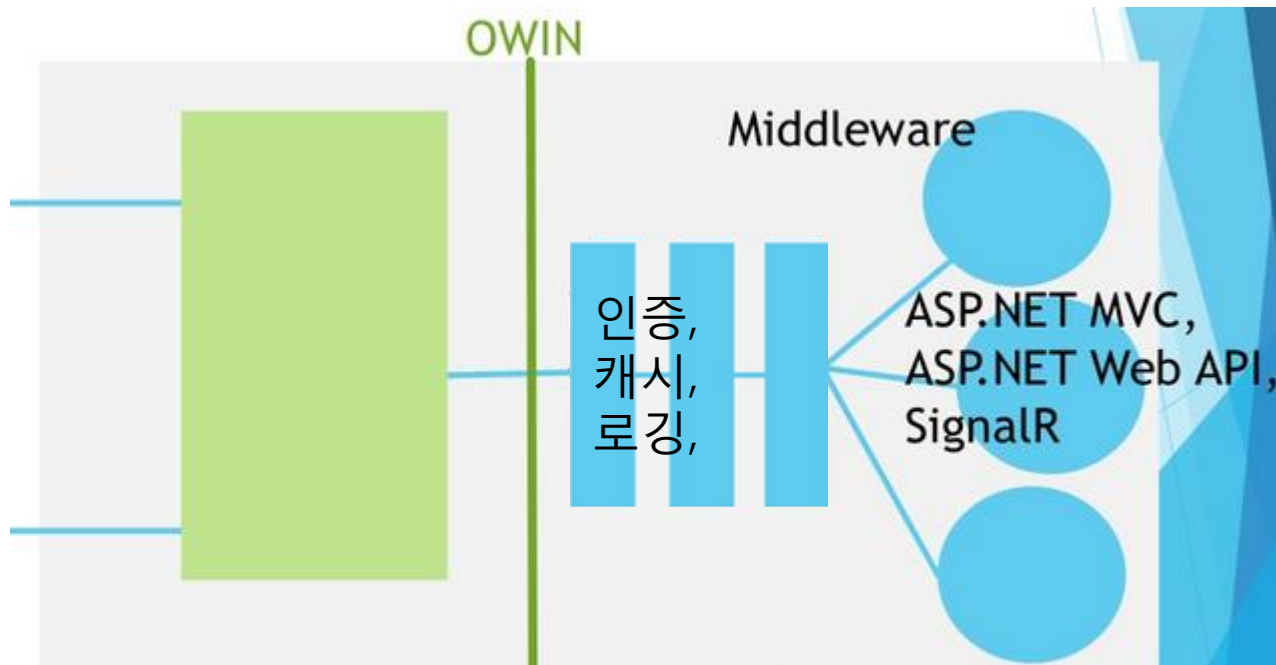
OWIN&KATANA - ASP.NET

- IIS에 호스트 되어도 인증 등의 Middleware을 적용하기 위해



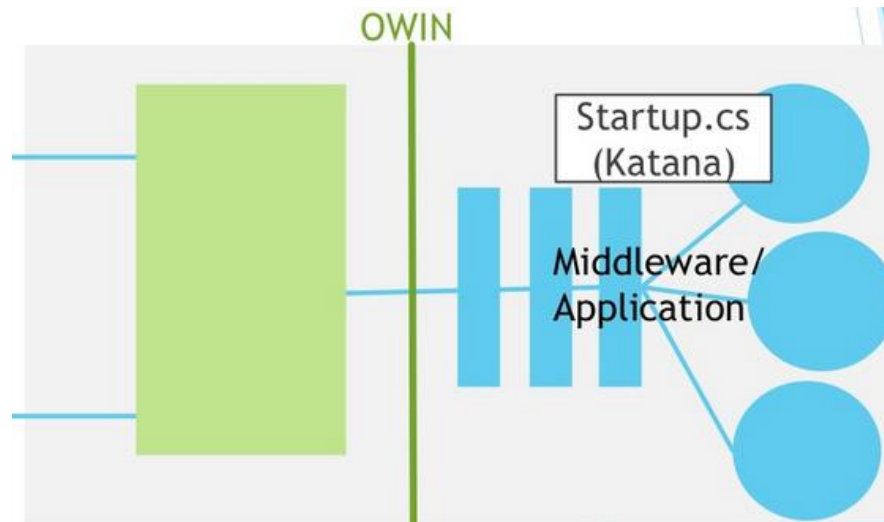
KATANA - Middleware

- 모든 처리 공통의 Middleware와 개별 프레임워크의 Middleware가 있다.



KATANA - Middleware: Startup.cs

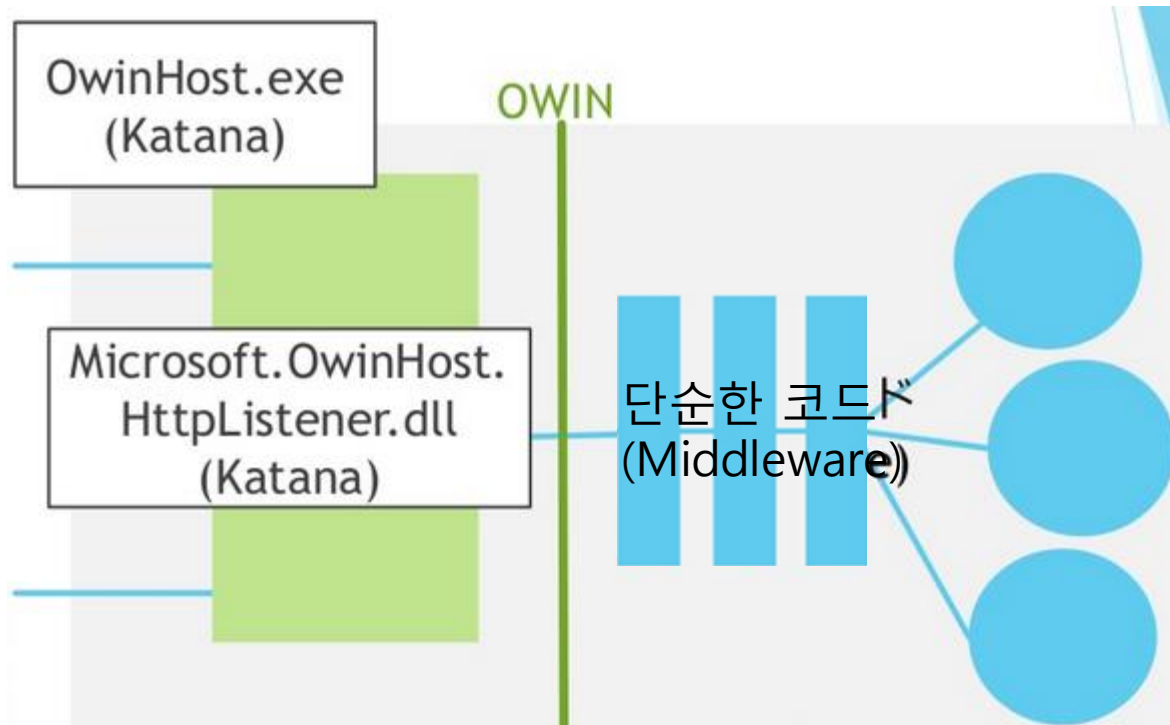
- OWIN의 처리에 Middleware를 등록하기 위해 Katana에 의한 설정 클래스



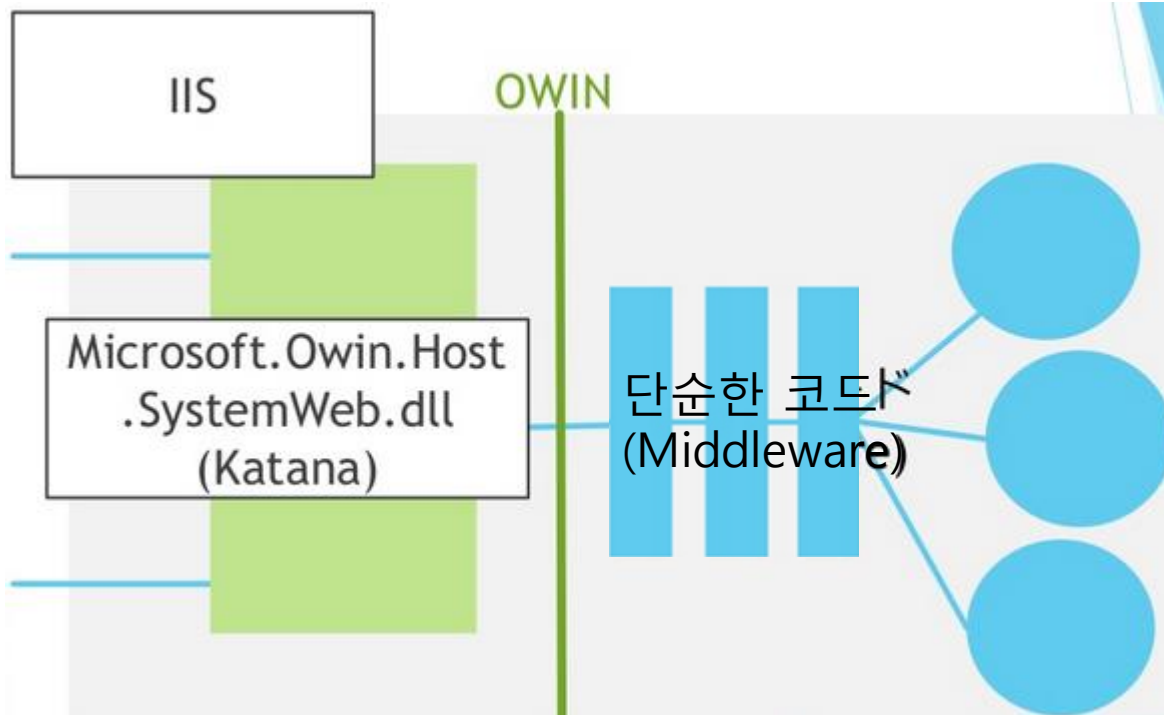
- Startup.cs에서 Use...() 메소드에 의해 Middleware를 등록한다.
- 복수 개 연결 가능.
- 애플리케이션 코드를 훼손하지 않고 개선, 교환 할 수 있다.
- 인증, 캐시, 로깅 등 Web 프레임워크 공통의 처리를 컴포넌트로서 등록 할 수 있다.

앞으로 ASP.NET에서는 Web 앱 공통 처리를 OWIN의 Middleware로 변경/개선할 예정

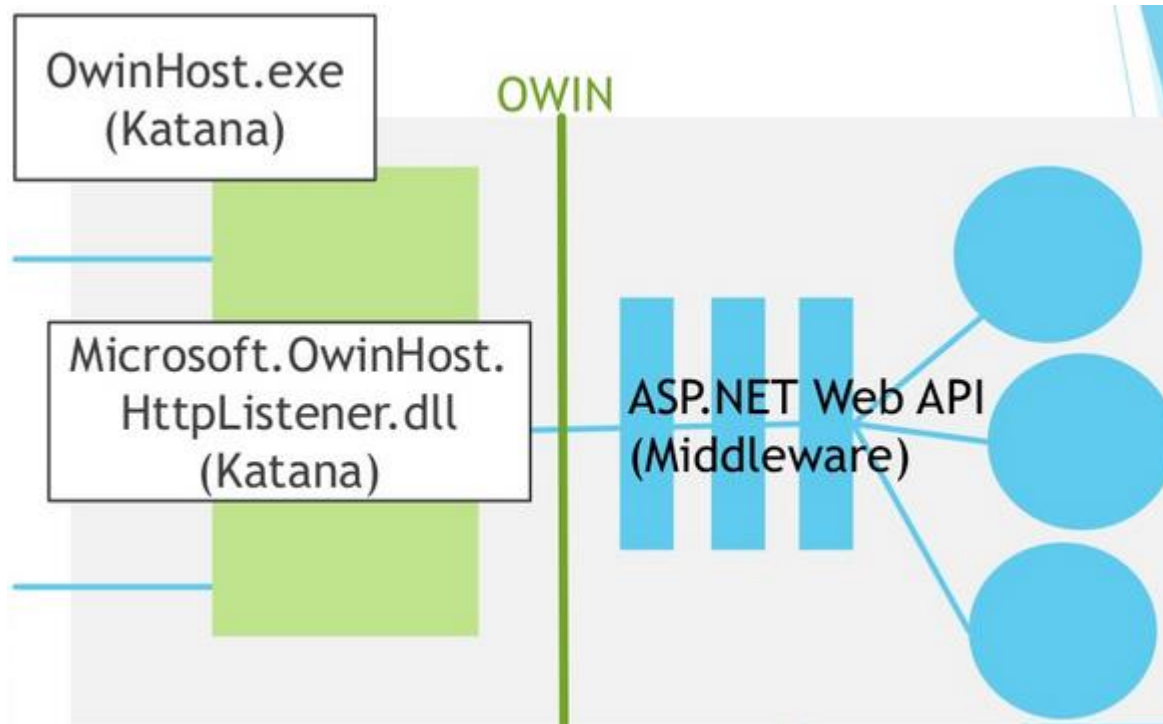
KATANA - 사례 1



KATANA - 사례 2



KATANA - 사례 3



KATANA - 사례 4

- ASP.NET Empty 프로젝트 생성
- 솔루션 탐색기에서 추가->새로운 항목->Owin Startup 클래스 선택
- NuGet에서 'Owin', 'Microsoft.Owin' 추가

```
using System;
using System.Threading.Tasks;
using Microsoft.Owin;
using Owin;

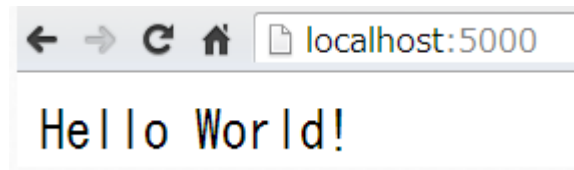
[assembly: OwinStartup(typeof(WebApplication40.Startup1))]

namespace WebApplication40
{
    public class Startup1
    {
        public void Configuration(IAppBuilder app)
        {
            app.Run(context =>
            {
                context.Response.ContentType = "text/plain";
                return context.Response.WriteAsync("Hello World!");
            });
        }
    }
}
```

KATANA - 사례 5

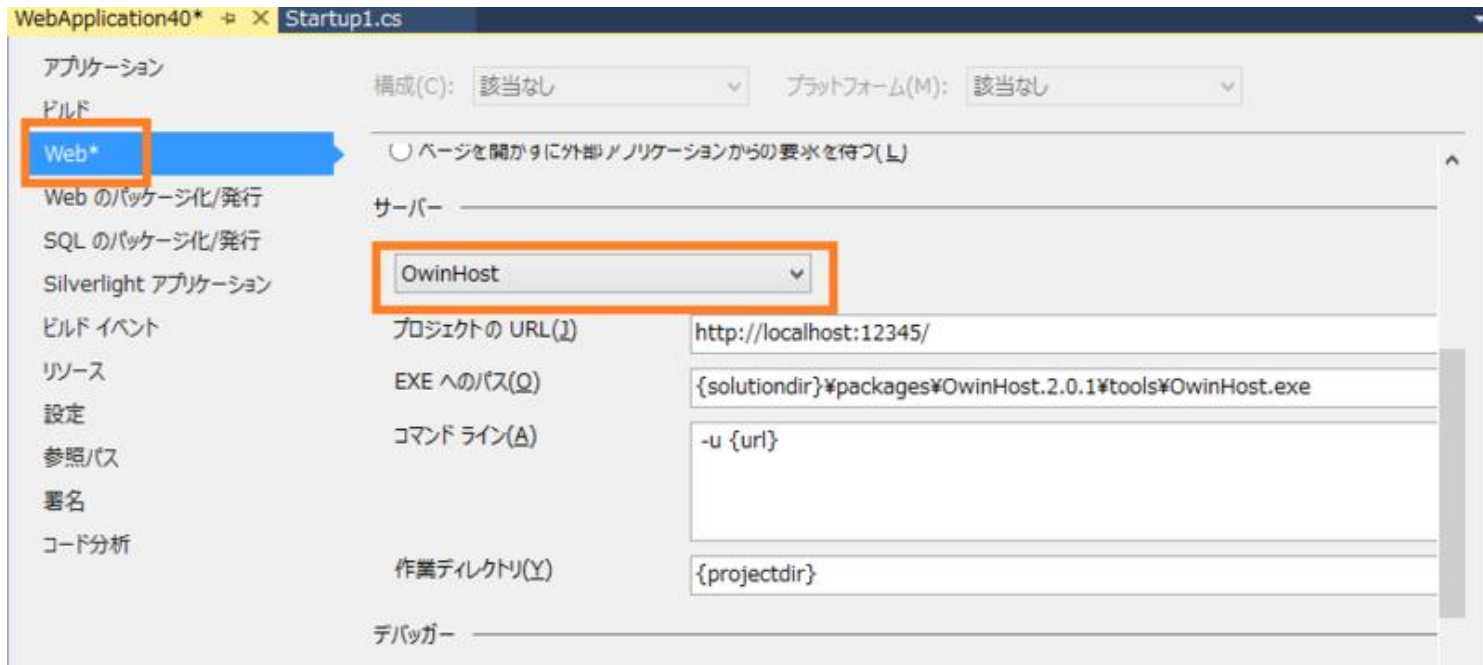
- NuGet에서 'OwinHost' 추가.
- '..\packages\OwinHost.2.0.1\tools\OwinHost'에서 OwinHost.exe 실행.
- OwinHost는 실행되면 프로젝트 내에 있는 Startup.cs를 찾는다.
 - web.config 파일에 key="owin:AppStartup"이 설정된 appSetting가 있는 로더는 이 설정 값을 사용한다.
 - 어셈블리에 속성 [assembly: OwinStartup(typeof(MyStartup))] 가 포함된 경우 로더는 속성 값에 지정된 형을 사용한다.
 - 위의 둘다 실패한 경우 로더는 읽어들이는 어셈블리를 스캔하여 시그네처가 void Configure(IAppBuilder app)에 일치한 메소드가 설정된 Startup이라는 이름의 형을 찾는다.

```
Starting with the default port: 5000 (http://localhost:5000/)
Started successfully
Press Enter to exit
```



KATANA - 사례 5

- OwinHost.exe를 VS 2013의 F5 디버그로 실행하기



IIS에서 호스트 하려면 NuGet으로 'Microsoft.Owin.Host.System.Web'을 설치한다.

KATANA - 사례 6

- Middleware로 커스텀 로그를 만든다.

```
public class MyLoggerMiddleware : OwinMiddleware
{
    public MyLoggerMiddleware(OwinMiddleware next)
        : base(next)
    {
    }

    public override async Task Invoke(IOwinContext context)
    {
        Debug.WriteLine("★ My Logger Start");
        await this.Next.Invoke(context);
        Debug.WriteLine("★ My Logger End");
    }
}

public class Startup1
{
    public void Configuration(IAppBuilder app)
    {
        app.Use<MyLoggerMiddleware>(); // ←추가
        app.Run(context =>
        {
            Debug.WriteLine("-- My Hello World Code"); //←추가
            context.Response.ContentType = "text/plain";
            return context.Response.WriteAsync("Hello World!");
        });
    }
}
```

KATANA - 사례 6

```
C:\Users\    \Documents\Visual Studio 2013\Projects\WebApplicat
ion40>..\packages\OwinHost.2.0.1\tools\OwinHost
Starting with the default port: 5000 (http://localhost:5000/)
Started successfully
Press Enter to exit
★ My Logger Start
-- My Hello World Code
★ My Logger End
```

Startup.cs에서 커스텀 Middleware 적
용 코드를 여러번 쓰면

```
app.Use<MyLoggerMiddleware>();
app.Use<MyLoggerMiddleware>();
app.Use<MyLoggerMiddleware>();
app.Use<MyLoggerMiddleware>();
```

```
★ My Logger Start
★ My Logger Start
★ My Logger Start
★ My Logger Start
-- My Hello World Code
★ My Logger End
★ My Logger End
★ My Logger End
★ My Logger End
```


OWIN - 사례 1



NANCY

Install

```
PM> Install-Package Nancy
```

Write

```
public class SampleModule : Nancy.NancyModule
{
    public SampleModule()
    {
        Get["/"] = _ => "Hello World!";
    }
}
```

Go!

Packages
Source Code
Documentation
MVM Program
Forum / Chat
Contributors
Swag^[EU] / Swag^[US]

<http://nancyfx.org/>

OWIN - 사례 2

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Text;
using System.Threading.Tasks;

namespace OwinHelloWorld
{
    public class Startup
    {
        public Task Invoke(IDictionary<string, object> env)
        {
            env["owin.ResponseStatusCode"] = 200;
            ((IDictionary<string, string[]>)env["owin.ResponseHeaders"]).Add(
                "Content-Type", new string[] { "text/html" });
            StreamWriter w = new StreamWriter((Stream)env["owin.ResponseBody"]);
            w.Write("Hello, from C#. Time on server is " + DateTime.Now.ToString());
            w.Flush();

            return Task.FromResult<object>(null);
        }
    }
}
```

<http://tomasz.janczuk.org/2013/02/hosting-net-code-in-nodejs-applications.html>

OWIN - 사례 3

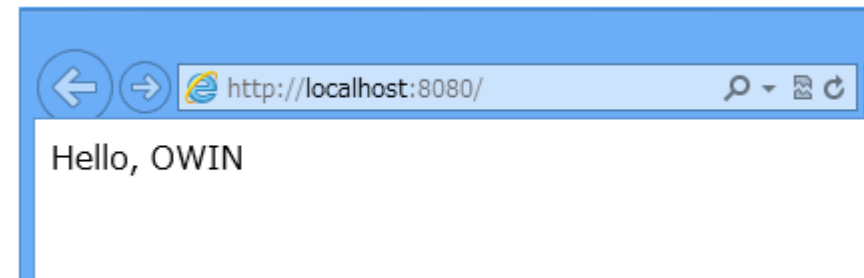
```
class Program
{
    static void Main(string[] args)
    {
        using (WebApplication.Start<Startup>("http://localhost:8080/"))
        {
            Console.ReadKey();
        }
    }
}

public class Startup
{
    public void Configuration(IAppBuilder app)
    {
        app.Use(typeof(Handler));
    }
}

public class Handler
{
    public Handler(Func<IDictionary<string, object>, Task> next)
    {
    }

    public Task Invoke(IDictionary<string, object> environment)
    {
        environment["owin.ResponseStatusCode"] = 200;

        using (var writer = new StreamWriter((Stream)environment["owin.ResponseBody"]))
        {
            return writer.WriteAsync("Hello, OWIN");
        }
    }
}
```



OWIN - 사례 4

- Hosting .NET code in node.js applications using OWIN
<http://tomasz.janczuk.org/2013/02/hosting-net-code-in-nodejs-applications.html>
- CPU bound workers for node.js applications using in-process .NET and OWIN
<http://tomasz.janczuk.org/2013/02/cpu-bound-workers-for-nodejs.html>
- Access MS SQL from a node.js application using OWIN, in-process CLR hosting, .NET, and ADO.NET
<http://tomasz.janczuk.org/2013/02/access-ms-sql-from-nodejs-application.html>
- EDGE.JS
<http://tjanczuk.github.io/edge/#>
GitHub <https://github.com/tjanczuk/edge>

OWIN - Helios

- 2013년 MS MVP Global Summit에서 발표.
- Checking out the Helios IIS Owin Web Server Host
<http://weblog.west-wind.com/posts/2013/Nov/23/Checking-out-the-Helios-IIS-Owin-Web-Server-Host>
- OWIN을 IIS에서 동작하는 경량 라이브러리.
NuGet <https://www.nuget.org/packages/Microsoft.Owin.Host.IIS/>
- Microsoft.Owin.Host.IIS를 사용하여 ASP.NET의 파이프라인을 완전하게 우회하여 IIS의 네이티브 API 위에서 아주 얇은 OWIN으로 변환한 층을 제공.
- 기존에 있던 관리 모듈은 사용할 수 없음. 기존의 폼 인증이 FormsAuthenticationModule는 OWIN+Helios에서는 OWIN Security Middleware로 치환.
NuGet Microsoft.Owin.Security
<https://www.nuget.org/packages?q=Microsoft.Owin.Security>

OWIN - Helios

- Helios 상의 Web API의 성능이 HttpHandler 보다 더 좋다.
- 현재(2014.02)는 정식 판이 아니고, Helios가 비 관리 코드와 상호 운용하는 부분이 많아서 메모리 릭이나 메모리 침범의 위험성도 있다.
- 장래에는 System.Web을 대체할 수도 있음

Apache Bench Test Results

Requests per seconds

Helios	4,554.44
HeliosWebApiJson	4,474.79
HttpHandler	4,047.98
HandlerJson	3,973.99
Mvc	3,919.87
WebApiJson	3,712.28
WebApi	3,366.36
MvcJson	3,177.74

참고

One asp.net owin&katana

http://sssslide.com/www.slideshare.net/miso_desu/one-aspnet-owinkatana

(일어)OWIN - Open Web Interface for .NET를 사용하자

http://kyrt.in/2013/12/17/owin_azure_cache_session_middleware.html

OWIN

<http://owin.org/>

OWIN Github

<https://github.com/owin/owin>

KATANA

<http://katanaproject.codeplex.com/>

<http://www.asp.net/aspnet/overview/owin-and-katana>

Getting Started with the Katana Project

<http://msdn.microsoft.com/en-us/magazine/dn451439.aspx>

(일어)Helios를 Azure Cloud Service에서 사용

http://kyrt.in/2013/11/28/using_helios_on_azure_cloud_service.html

(일어)Katana에서 Hello, world

<http://miso-soup3.hateblo.jp/entry/2013/11/11/014258>

(일어)Ubuntu 12.04.3 LTS에 Katana(OwinHost.exe)을 인스톨

<http://espresso3389.hatenablog.com/entry/2014/01/10/021441>

참고

(일어)OWIN 파이프 라인과 Middleware 작성 가이드

http://neue.cc/2014/01/06_442.html

(일어)LightNode - Owin으로 구축하는 Micro RPC/REST Framework

http://neue.cc/2013/12/23_439.html

Owin.RedisSession

<https://github.com/neuecc/Owin.RedisSession>