ASP.NET MVC 5 Fundamentals
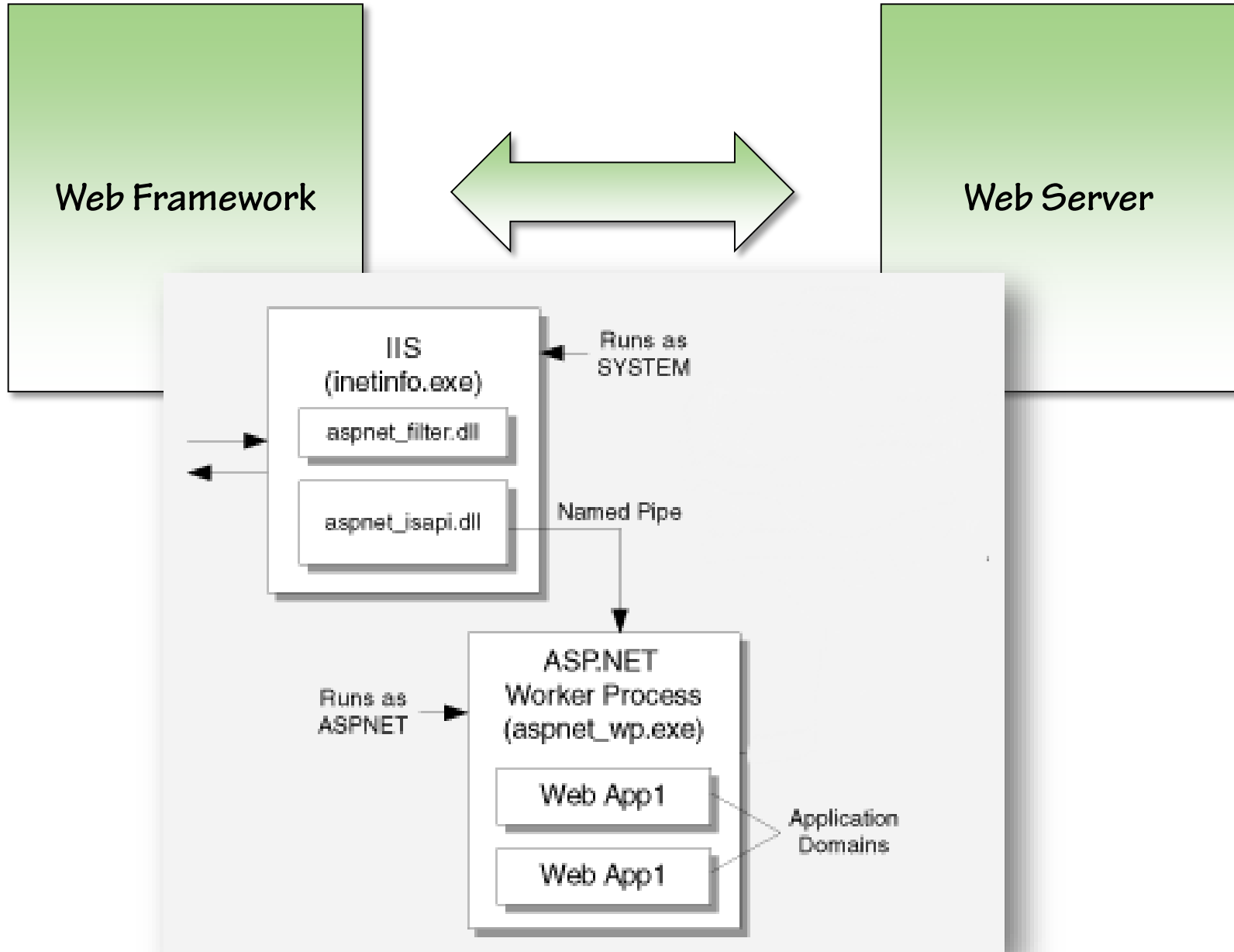
OWIN & Katana

K. Scott Allen
odetocode.com
@OdeToCode
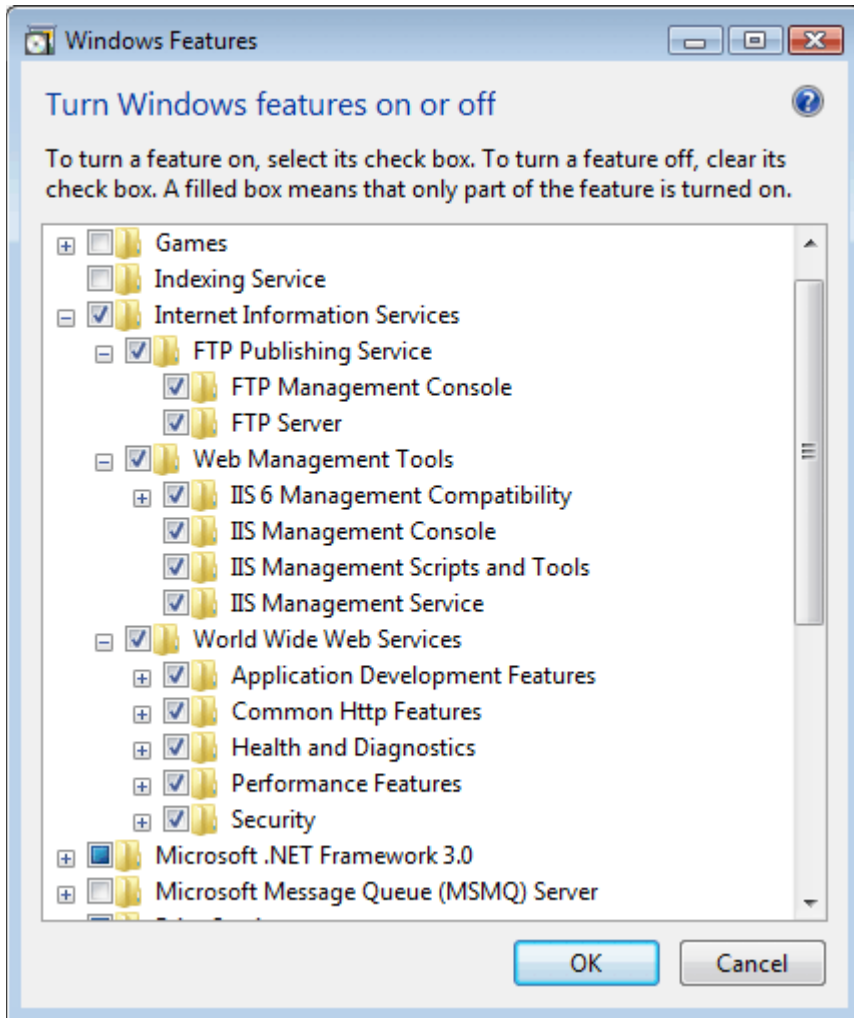
pluralsight
hardcore developer training

Architecture

Web Framework

Web Server

IIS
(inetinfo.exe)

Runs as
SYSTEM

aspnet_filter.dll

aspnet_isapi.dll

Named Pipe

ASP.NET
Worker Process
(aspnet_wp.exe)

Runs as
ASPNET

Web App1

Web App1

Application
Domains

IIS and ASP.NET

Session State

Caching

Membership

Dynamic Data

Modules

Handlers

Configuration

Request Tracing

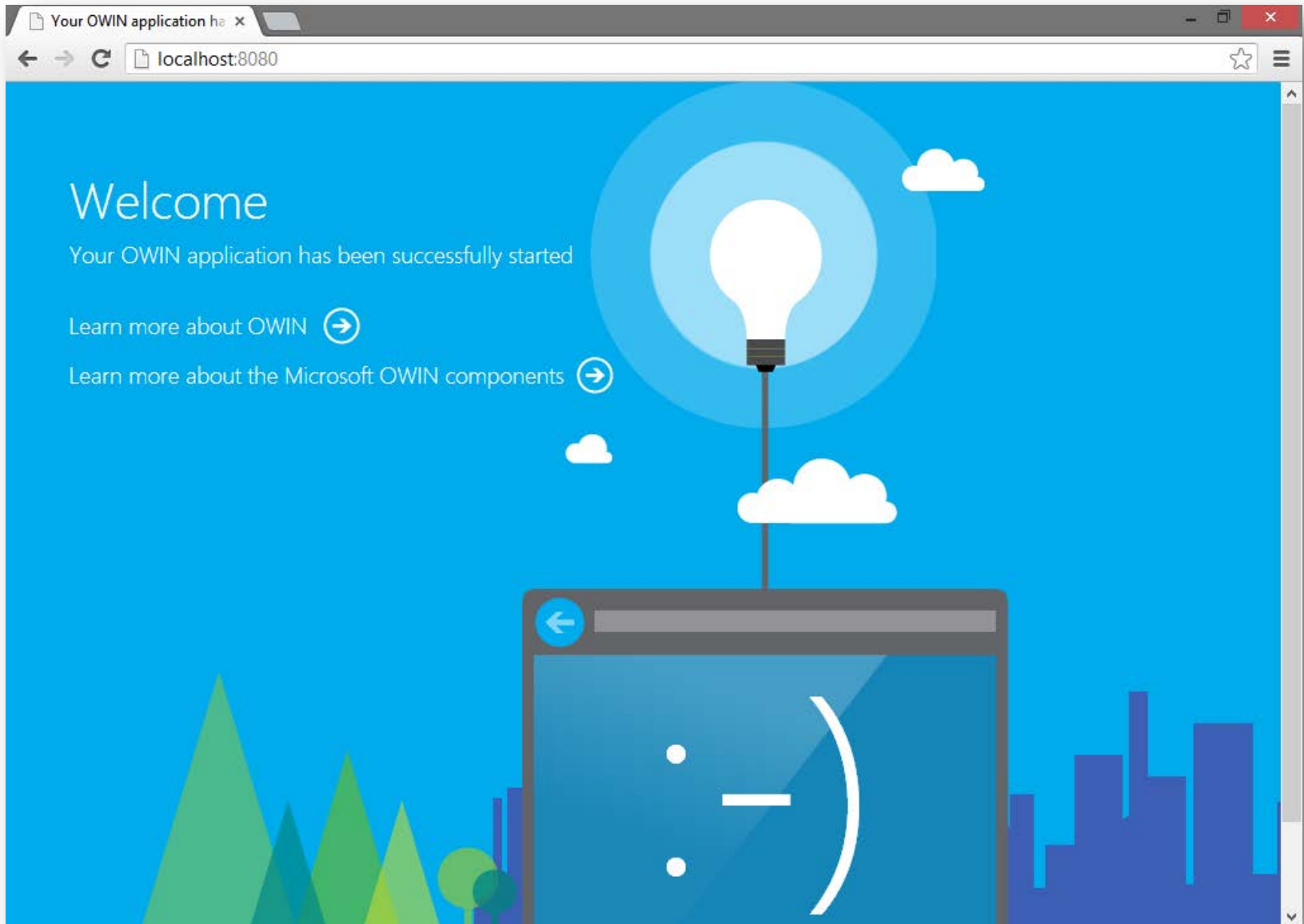# Exclusive: How LinkedIn used Node.js and HTML5 to build a better, faster app
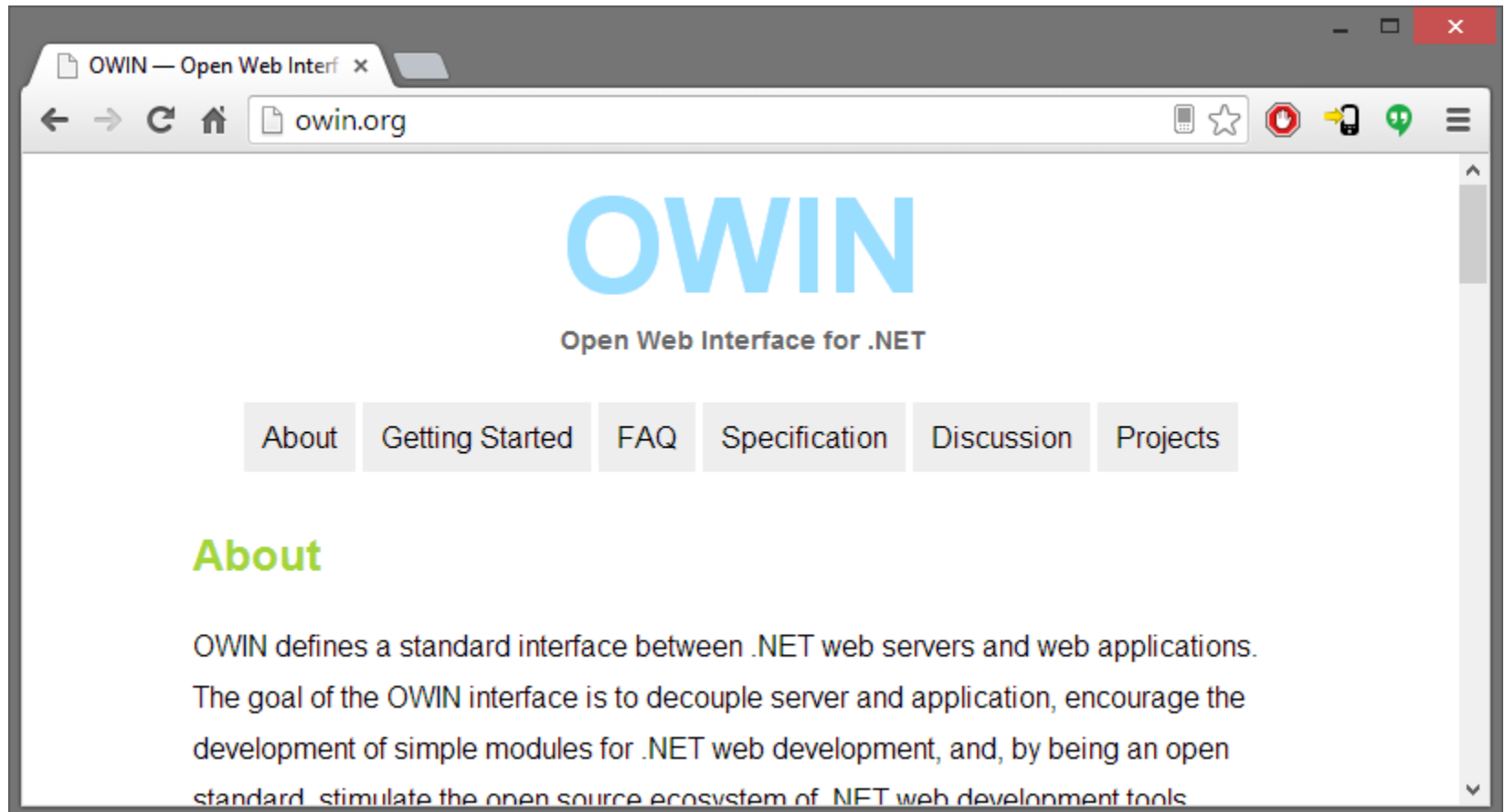

August 16, 2011 6:01 AM
Jolie O'Dell

The app is two to 10 times faster on the client side than its predecessor, and on the server side, it's using a fraction of the resources, thanks to a switch from Ruby on Rails to Node.js, a server-side JavaScript development technology that's barely a year old but already rapidly gaining traction.

Katana

OWIN

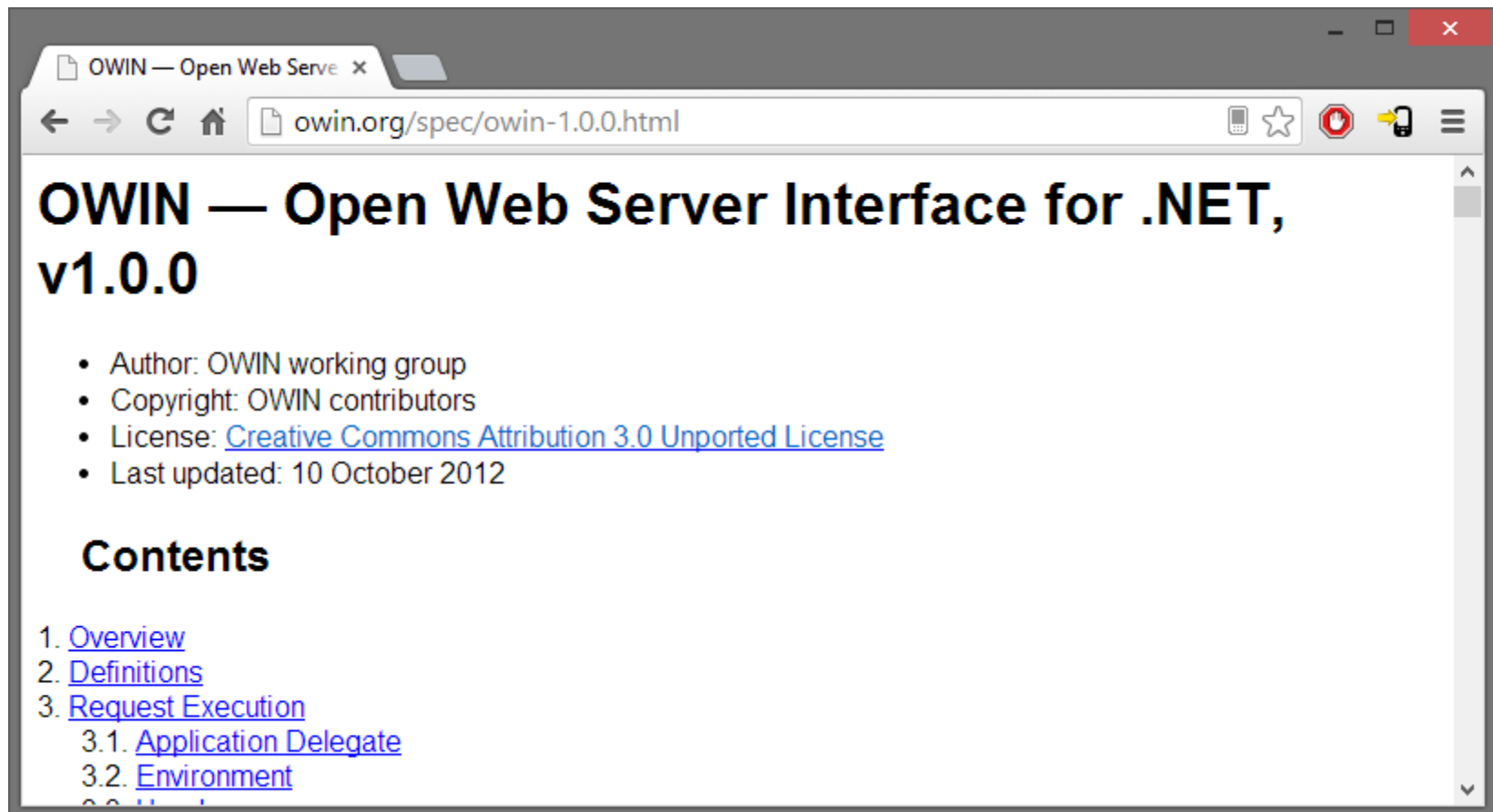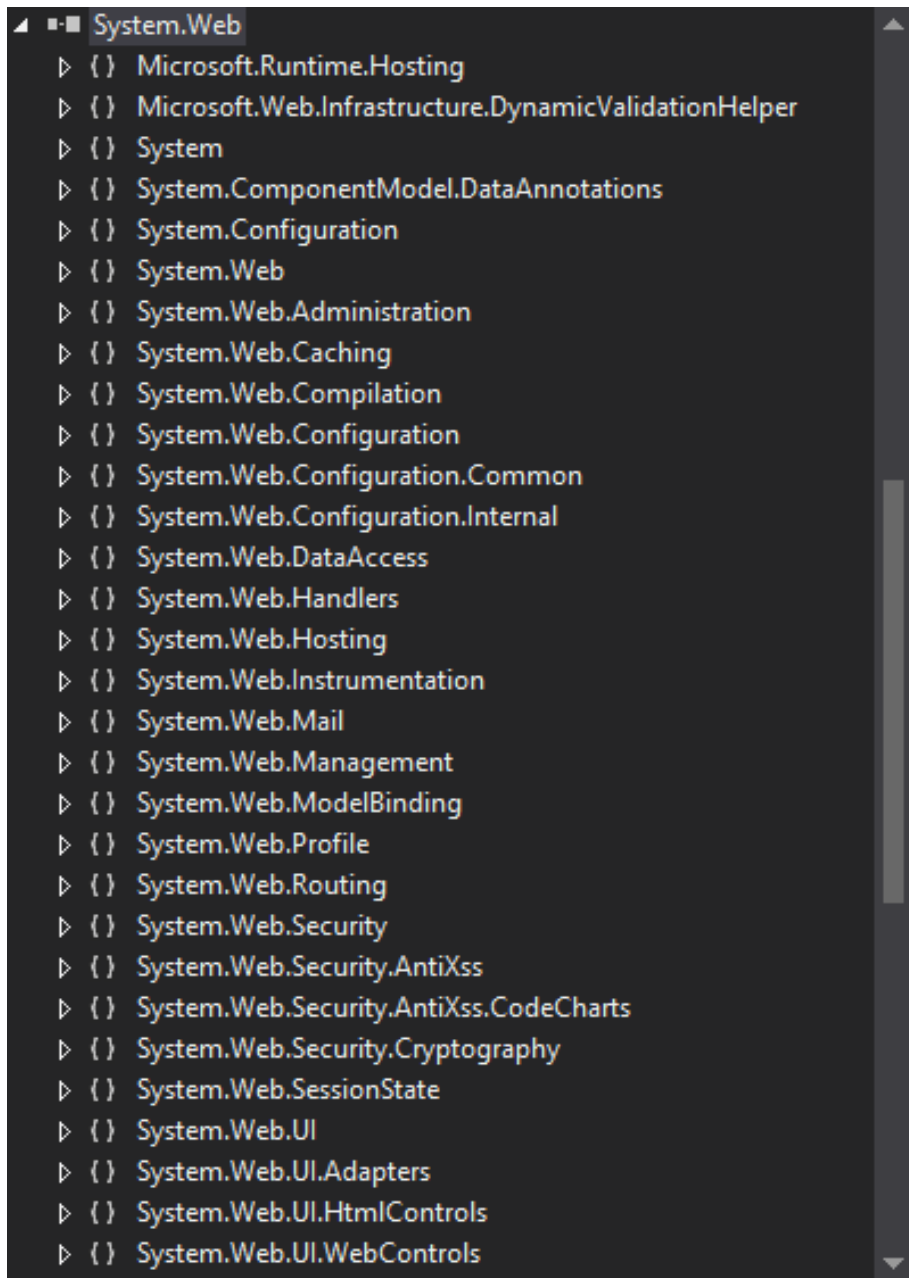- **Open Web Interface for .NET**

OWIN — Open Web Server Interface for .NET, v1.0.0

- Author: OWIN working group
- Copyright: OWIN contributors
- License: Creative Commons Attribution 3.0 Unported License
- Last updated: 10 October 2012

Contents

System.Web
- Microsoft.Runtime.Hosting
- Microsoft.Web.Infrastructure.DynamicValidationHelper
- System
- System.ComponentModel.DataAnnotations
- System.Configuration
- System.Web
- System.Web.Administration
- System.Web.Caching
- System.Web.Compilation
- System.Web.Configuration
- System.Web.Configuration.Common
- System.Web.Configuration.Internal
- System.Web.DataAccess
- System.Web.Handlers
- System.Web.Hosting
- System.Web.Instrumentation
- System.Web.Mail
- System.Web.Management
- System.Web.ModelBinding
- System.Web.Profile
- System.Web.Routing
- System.Web.Security
- System.Web.Security.AntiXss
- System.Web.Security.AntiXss.CodeCharts
- System.Web.Security.Cryptography
- System.Web.SessionState
- System.Web.UI
- System.Web.UI.Adapters
- System.Web.UI.HtmlControls
- System.Web.UI.WebControls

versus

OWIN + Katana

Portable

Modular

Lightweight

The AppFunc

```csharp
Func<IDictionary<string, object>, Task>;
```

Environment

All Async

```csharp
public async Task Invoke(IDictionary<string, object> environment)
{
    // processing

    await _nextComponent(environment);

    // processing
}
```

| C! | → next → | MyConponent | → next → | C2 |

OWIN Components

- **Also Known As Middleware**

```csharp
public class HelloWorldComponent
{
    AppFunc _next;
    public HelloWorldComponent(AppFunc next)
    {
        _next = next;
    }

    public Task Invoke(IDictionary<string, object> environment)
    {
        var response = environment["owin.ResponseBody"] as Stream;
        using (var writer = new StreamWriter(response))
        {
            return writer.WriteAsync("Hello!!");
        }

    }
}
```

```csharp
app.Run(environment =>
    {
        return environment.Response.WriteAsync("Hello!!");
    });
```
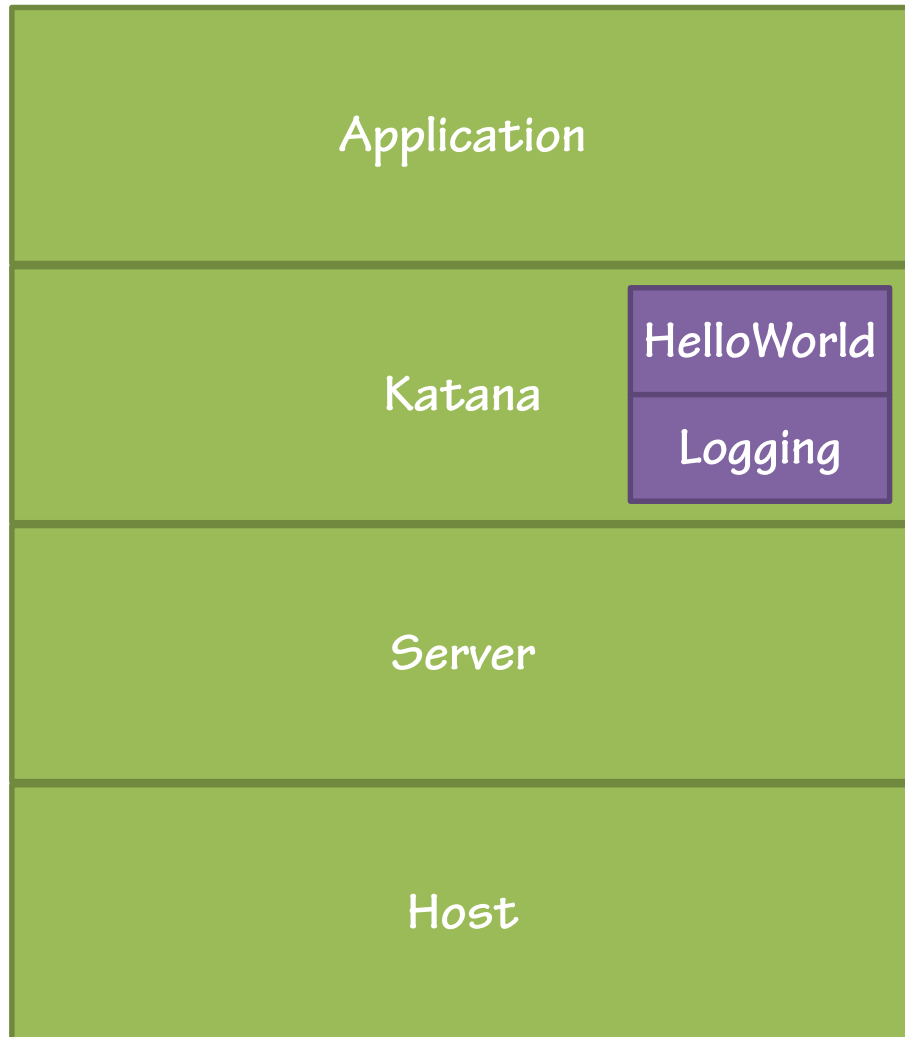
The AppFunc Request Environment

3.2.1 Request Data

Required	Key Name	Value Description
Yes	`"owin.RequestBody"`	A `Stream` with the request body, if any. `Stream.Null` MAY be used as a placeholder if there is no request body. See Request Body.
Yes	`"owin.RequestHeaders"`	An `IDictionary<string, string[]>` of request headers. See Headers.
Yes	`"owin.RequestMethod"`	A `string` containing the HTTP request method of the request (e.g., `"GET"`, `"POST"`).
Yes	`"owin.RequestPath"`	A `string` containing the request path. The path MUST be relative to the "root" of the application delegate; see Paths.
Yes	`"owin.RequestPathBase"`	A `string` containing the portion of the request path corresponding to the "root" of the application delegate; see Paths.
Yes	`"owin.RequestProtocol"`	A `string` containing the protocol name and version (e.g. `"HTTP/1.0"` or `"HTTP/1.1"`).
Yes	`"owin.RequestQueryString"`	A `string` containing the query string component of the HTTP request URI, without the leading "?" (e.g., `"foo=bar&baz=quux"`). The value may be an empty string.
Yes	`"owin.RequestScheme"`	A `string` containing the URI scheme used for the request (e.g., `"http"`, `"https"`); see URI Scheme.

AppFunc Response Environment

3.2.2 Response Data

Required	Key Name	Value Description
Yes	`"owin.ResponseBody"`	A `Stream` used to write out the response body, if any. See Response Body.
Yes	`"owin.ResponseHeaders"`	An IDictionary<string, string[]> of response headers. See Headers.
No	`"owin.ResponseStatusCode"`	An optional `int` containing the HTTP response status code as defined in RFC 2616 section 6.1.1. The default is 200.
No	`"owin.ResponseReasonPhrase"`	An optional `string` containing the reason phrase associated the given status code. If none is provided then the server SHOULD provide a default as described in RFC 2616 section 6.1.1
No	`"owin.ResponseProtocol"`	An optional `string` containing the protocol name and version (e.g. `"HTTP/1.0"` or `"HTTP/1.1"`). If none is provided then the `"owin.RequestProtocol"` key's value is the default.

Architecture

Summary

Func<IDictionary<string, object>, Task>;

Application
- WebAPI
- MVC

Katana
- HelloWorld
- Logging

Server

Host