DEV-B209

ASP.NET 5 开发攻略

多奇数位创意有限公司

技术总监 黄保翕 (Will 保哥)

部落格：http://blog.miniasp.com/

介绍 ASP.NET 5

什么是 ASP.NET 5

- 跨平台
- 使用 .NET 技术
- 开放原始码 (GitHub)
- 可建构云端应用程序
- 可部署到云端或本地环境
- 使用模组化的 CoreFX 框架
- 一套全新且重新打造的 ASP.NET 框架

为什么要建构 ASP.NET 5

- 历史包袱
  - ASP.NET 1.0 是 15 年前打造的框架
  - ASP.NET 5 不再使用 System.Web.dll 程序集 (assembly)
  - 整合 MVC, Web API 与 WebForms 框架，去芜存菁
- 云端狂潮
  - 优化模块加载与使用，并以最小资源执行 Web 应用程序
  - ASP.NET 5 实现轻量且模块化的 HTTP 要求管线
  - ASP.NET 5 使用 NuGet 加载必要的 CoreFX
  - ASP.NET 5 可执行在任意 self-host 环境 (IIS, Kestrel, ...)
- 现有 Web 开发主流
  - 开放原始码与跨平台
  - 明显区分前端开发与后端开发
  - 使用相依性注入 (Dependency injection)

安装 ASP.NET 5 Beta 8

安装必要工具 (Windows)

- ASP.NET and Web Tools 2015 (Beta8)
  - 安装 beta8 版本
    http://go.microsoft.com/fwlink/?LinkId=690242
    - 下载并安装 .NET Version Manager (DNVM) 并下指令安装 runtime
      DotNetVersionManager-x64.msi
      - dnvm install 1.0.0-beta8 -arch x86 (已默认安装)
      - dnvm install 1.0.0-beta8 -arch x64
      - dnvm install 1.0.0-beta8 -arch x86 -r coreclr
      - dnvm install 1.0.0-beta8 -arch x64 -r coreclr
    - 若为 Visual Studio 2015 (Community, Professional, Enterprise)
      - 下载并安装 **WebToolsExtensionsVS14.msi**
      - 下载并安装 **WebToolsExtensionsVSLP14_chs.msi** (语言包)
    - 若为 Visual Studio 2015 Express for Web
      - 下载并安装 **WebToolsExtensionsVWD14.msi**
      - 下载并安装 **WebToolsExtensionsVWDLP14_chs.msi** (语言包)

安装必要工具 (Mac OS X)

- Installing ASP.NET 5 On Mac OS X
  - 安装 1.0.0-beta8 版本
  - http://docs.asp.net/en/1.0.0-beta8/getting-started/installing-on-mac.html

    - 下载并安装 .NET Version Manager (DNVM)

```
curl -sSL
https://raw.githubusercontent.com/aspnet/Home/dev/dnvminstall.sh |
DNX_BRANCH=dev sh && source ~/.dnx/dnvm/dnvm.sh
```

    - 下载并安装 .NET Execution Environment (DNX) (无须 Mono 相依)
      - 安装 DNX for .NET Core

```
brew update
brew install icu4c
dnvm upgrade -r coreclr
```

      - 安装 DNX for Mono

```
brew update
brew install mono
dnvm upgrade -r mono
```

安装必要工具 (Linux)

- Installing ASP.NET 5 On Linux
  - 安装 1.0.0-beta8 版本
  - http://docs.asp.net/en/1.0.0-beta8/getting-started/installing-on-linux.html
    - 下载并安装 .NET Version Manager (DNVM) (与 Mac 相同指令)
    - 下载并安装以下套件
      - **libunwind8 gettext libssl-dev libcurl3-dev zlib1g**
      - **libicu-dev** (因为 Kestrel 会用到)
    - 下载并安装 .NET Execution Environment (DNX) (无须 Mono 相依)
      - 安装 DNX for .NET Core
        dnvm upgrade -r coreclr
      - 安装 DNX for .NET Core
        dnvm upgrade -r coreclr

选择正确的 .NET 版本

- 重要观念
  - ASP.NET 5 基于 .NET Execution Environment (DNX) 打造
  - DNX 是一个微软官方支持的跨平台支持 .NET 执行环境 (Win, Mac, Linux)
- 可选版本
  - .NET Framework (CLR)
    - 拥有完整的 程序集 (assembly) 与 大量的 API 与 较慢的发行周期
    - .NET 4.6 可在在线浏览完整原始码
    - 只能跑在 Windows 操作系统
  - .NET Core (CoreCLR)
    - 为 .NET 4.6 的子集合，拥有较少的 API 支持，但支持跨平台执行，也可执行在资源受限的执行环境中（例如 Windows Server Nano）
    - .NET Core 包含一组函式库称为 "CoreFX"，并统一由 NuGet 发布
    - 目前 Windows 平台已有完整功能实作，而 Linux, OS X 还在持续开发中
    - 现有的 .NET 项目若要跑在 CoreCLR 上，必须重新编译
  - Mono
    - 支持跨平台的完整 .NET Framework 实作，开源原始码，但微软官方不支持

关于 ASP.NET 5 Beta 8

- 从这版开始不再「主动」新增功能
  - 专注于让 ASP.NET 5 进入稳定状态、工具支持
  - 除非有人提出重要的建议，各位可以到这里提出任何问题：https://github.com/aspnet/Home/issues
  - **ASP.NET 5 Beta 8 已于 2015/10/15 释出！**
- 开发服务器
  - 已移除 **Helios** 项目
    - **Microsoft.AspNet.Server.IIS** **(Windows)**
  - 未来将会专注在以下两项项目
    - **Microsoft.AspNet.Server.WebListener** **(Windows) (Self-hosted) (HTTP.sys)**
    - **Microsoft.AspNet.Server.Kestrel** **(跨平台) (Self-hosted) (libuv)**

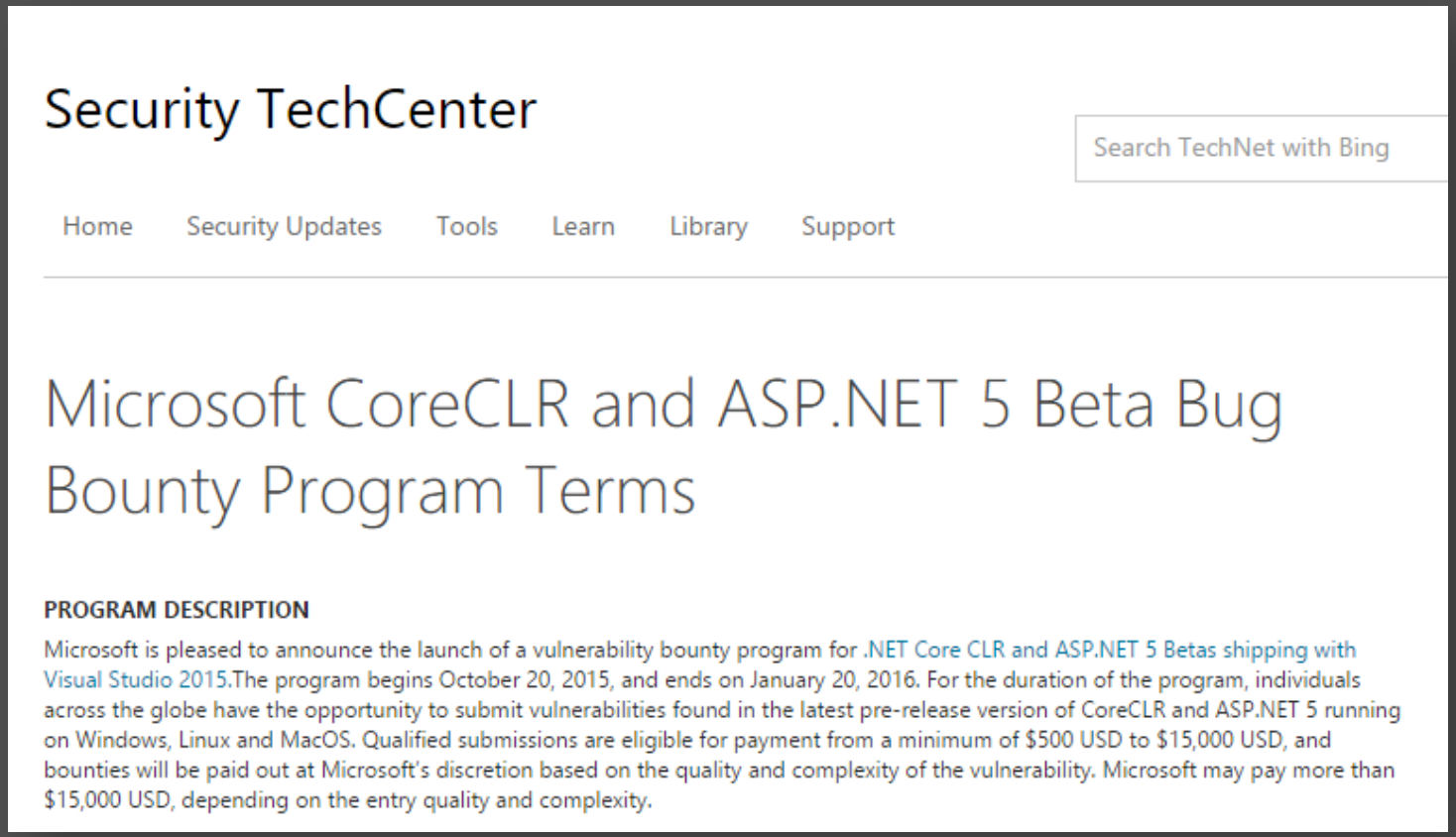
关于 HttpPlatformHandler 模块 (IIS 8+)

- HttpPlatformHandler
  - 一个支援 IIS 8 与 IIS 8.5 的原生模块 (native module)
  - 主要使用于 Microsoft Azure Websites 网站服务中
  - 用途是让第三方程序可以轻松整合到 IIS 之中
  - ASP.NET 5 Beta8 版本也正式宣告改采 HttpPlatformHandler 模块搭配 Kestrel HTTP Server 执行 ASP.NET 5 应用程序

- 介绍文章
  - 介绍 IIS 8 全新的 HttpPlatformHandler 模块与 ASP.NET 5 Beta8 重大变更

CoreCLR 或 ASP.NET 5 臭虫奖励计划

- 微软推出为期 3 个月的臭虫奖励计划
  - 只要发现 CoreCLR 或 ASP.NET 5 Beta 产品有任何安全性弱点，即可获得多达**美金 $15,000** 的高额奖金，你准备好了吗？



Security TechCenter

Search TechNet with Bing

Home Security Updates Tools Learn Library Support

Microsoft CoreCLR and ASP.NET 5 Beta Bug Bounty Program Terms

PROGRAM DESCRIPTION

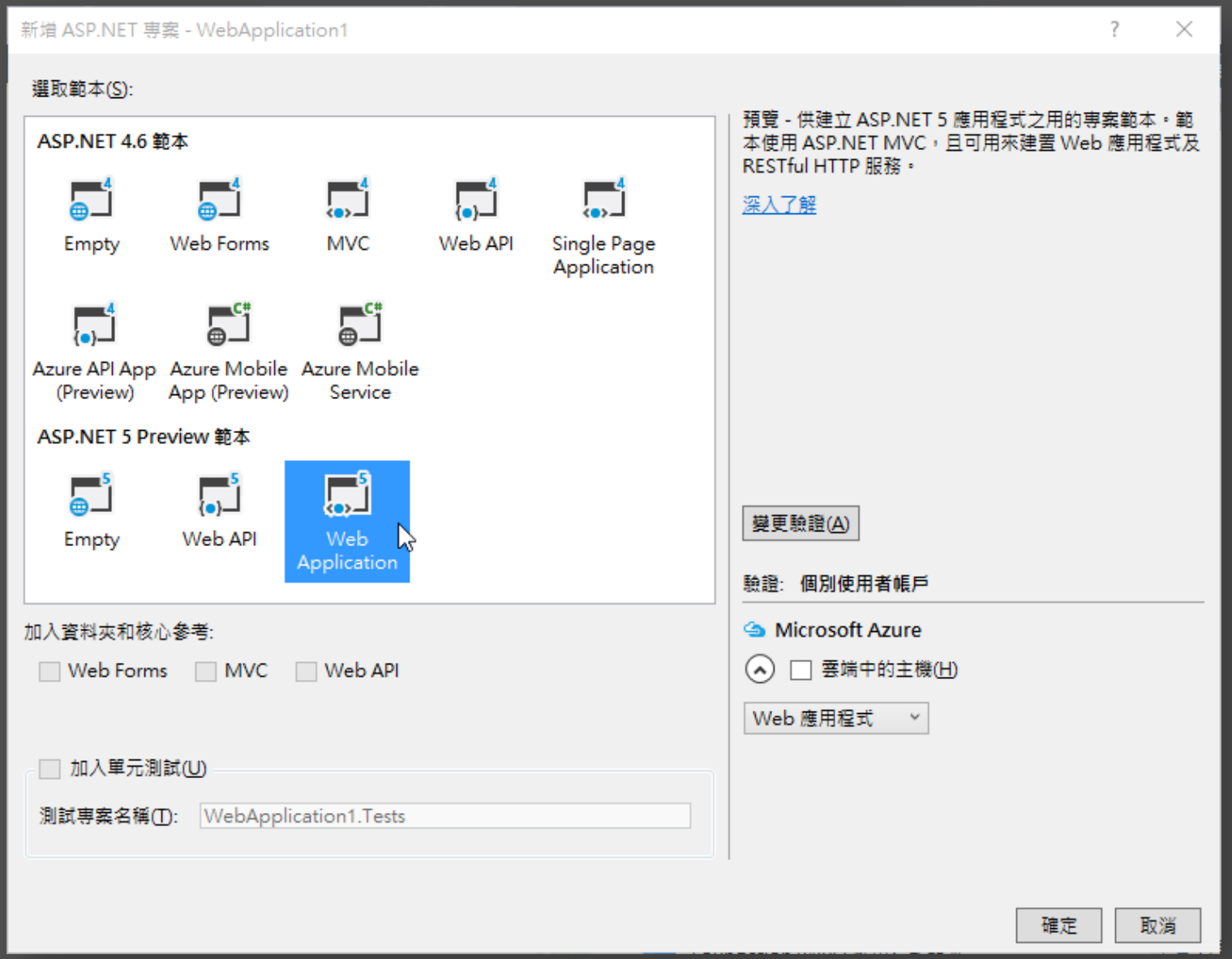
Microsoft is pleased to announce the launch of a vulnerability bounty program for .NET Core CLR and ASP.NET 5 Betas shipping with Visual Studio 2015.The program begins October 20, 2015, and ends on January 20, 2016. For the duration of the program, individuals across the globe have the opportunity to submit vulnerabilities found in the latest pre-release version of CoreCLR and ASP.NET 5 running on Windows, Linux and MacOS. Qualified submissions are eligible for payment from a minimum of $500 USD to $15,000 USD, and bounties will be paid out at Microsoft's discretion based on the quality and complexity of the vulnerability. Microsoft may pay more than $15,000 USD, depending on the entry quality and complexity.

创建 ASP.NET 5 项目

创建项目（使用 yo 工具）

- 安装 Node.js
- `npm install -g yo`
- `npm install -g generator-aspnet`
- `yo aspnet`
- `cd WebApplication`
- `dnu restore`
- `dnu build`　　（非必要步骤）
- `dnx web`

创建项目（使用 VS2015 工具）

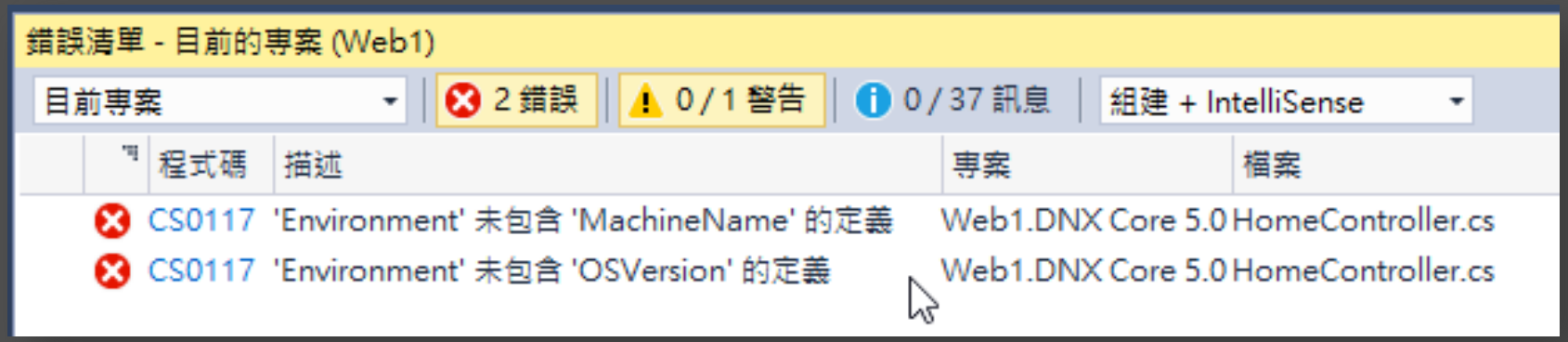


全新的 .NET 执行环境与开发环境

- 修改任何档案，不再需要事先生成
  - 优点：更流畅的开发体验
  - 缺点：会有不知道何时编译完成的不确定感
- 前端与后端的分工与整合
  - ASP.NET 5 从纯后端技术，转向整合更多前端工程
  - ASP.NET 5 整合许多常见的前端技术与前端工具
    - 技术：Less, Sass, Font Awesome, TypeScript, RWD, …
    - 框架：Bootstrap, Knockout.js, AngularJS, ReactJS, …
    - 工具：Gulp, Grunt, npm, Bower, Yeoman, …

修改 Controller 与编译项目

- HomeController.About()
  - ViewBag.EnvName = Environment.MachineName;
  - ViewBag.Software = Environment.OSVersion.ToString();
- 编译项目（F6）



- 修正错误
  - 移除 project.json 里面的 frameworks 里面的 **dnxcore50**

项目内容介绍

ASP.NET 5 方案设定

- 一样有 *.sln 方案檔
  - WebApplication1.sln
- 项目的 NuGet 定义档
  - NuGet.Config
- 方案定义
  - global.json
  - 定义项目目录
    - 会自动导入 *.sln 方案檔
  - 定义 SDK 版本

```
{
  "projects": [ "src", "test" ],
  "sdk": {
    "version": "1.0.0-beta8"
  }
}
```

ASP.NET 5 项目设定

- 改用 *.xproj 项目档
  - ASP.NET 5 取消 `*.csproj` 定义项目 (使用正向表列档案)
  - ASP.NET 5 改用 `*.xproj` 定义项目 (改用负向表列档案)
  - 所有项都改采「自动扫描」的方式导入
    - 优点：不再需要手动加入项目，减少档案忘记加入版控的问题
    - 缺点：需要不断扫描是否有新档案加入项目
- 许多项目设定都移到 project.json 配置文件
  - https://github.com/aspnet/Home/wiki/Project.json-file

project.json

- ASP.NET 5 项目配置文件
- ASP.NET 5 项目配置文件
  - 摘要说明
    - ASP.NET 5 已改用 project.json 定义项目 (现在改用负向表列项目文件)
    - 此档案是 JSON 格式，其内容会与 Gulp 共享 (透过 Node.js 载入)
  - 重要属性
    - webroot
      - 定义网站根目录位置 (所有静态档案)
    - userSecretsId
      - 定义网站的 "使用者秘密ID"
      - 同主机每个网站应用程序都不能重复
    - version
      - 定义目前 ASP.NET 5 应用程序版本
    - commands
      - 设定 dnx 的自定义命令
    - dependencies
      - 设定项目要加载哪些套件
    - frameworks
      - 设定特定执行框架下 (Runtime Framework) 要加载哪些套件
    - exclude
      - 哪些文件夹或档案要排除编译范围
    - publishExclude
      - 哪些文件夹在发布时要排除在外
    - scripts
      - 设定 dnu 的事件动作

appsettings.json

- ASP.NET 5 参数设定檔
  - 摘要说明
    - 类似之前 web.config 中的 appSettings 设定
    - 这里定义着应用程序执行过程所需的参数，例如**连接字符串**
    - 由于档案是 JSON 格式，因此可以自由添加任何所需参数
    - 这里的 JSON 参数设定完全没有限制，不一定要按照预设的下去设定

Startup.cs

- ASP.NET 5 项目的进入点
  - Startup() 建构式
    - 注入对象
      - IHostingEnvironment env
        - 宿主环境变量
        - 提供环境名称(EnvironmentName)、网站根目录位置(WebRootPath)等信息
      - IApplicationEnvironment appEnv
        - 应用程序环境变量
        - 提供应用程序名称、路径、版本、执行框架等信息
    - 定义配置文件来源
      - AddJsonFile("config.json")
      - AddCommandLine(args)
      - AddEnvironmentVariables()
      - AddIniFile("appsettings.ini")
      - AddInMemoryCollection()
      - AddUserSecrets()
  - 公开属性：Configuration
    - 用于共享设定信息于类中

Startup.cs

- ASP.NET 5 项目的进入点
  - ConfigureServices() 方法
    - 注入对象
      - `IServiceCollection services`
        - 用来记录那些服务要被加入到 ASP.NET 5 容器内的重要对象
        - 因为一个 ASP.NET 5 应用程序会包含多个「服务」并在此被注册进去
    - 注册服务
      - Add*()　　增加服务到集合里
      - Configure<T>()　　设定特定服务选项
      - AddTransient<T1, T2>()　　注册自定义的应用程序服务对象
    - 重要观念
      - 「服务」是 ASP.NET 5 的模块化组件 (component)

Startup.cs

- ASP.NET 5 项目的进入点
  - Configure() 方法
    - 注入对象
      - `IApplicationBuilder app`
        - 用来创建应用程序 **执行管线** (Execution Pipeline) 的重要对象
      - `IHostingEnvironment env`
        - 宿主环境变量
        - 提供环境名称(EnvironmentName)、网站根目录位置(WebRootPath)等信息
      - `ILoggerFactory loggerFactory`
        - ASP.NET 5 内置的 Logger 对象，用来设定 Logger 的纪录方式
    - 使用服务
      - `Use*()` 使用服务
      - `UseMvc()` 使用 ASP.NET MVC 6 服务，路由 (Routing) 也改从这里宣告
    - 注意事项
      - 这个方法会决定 ASP.NET 5 执行时期的 **执行管线** (Execution Pipeline) 的顺序
      - 这里执行顺序很重要，必须想清楚你要如何处理每一个 HTTP 要求

前端工程相关档案

- package.json
  - npm 配置文档
- bower.json
  - Bower 配置文档
- gulpfile.js
  - Gulp 工作定义档

重要观念解析

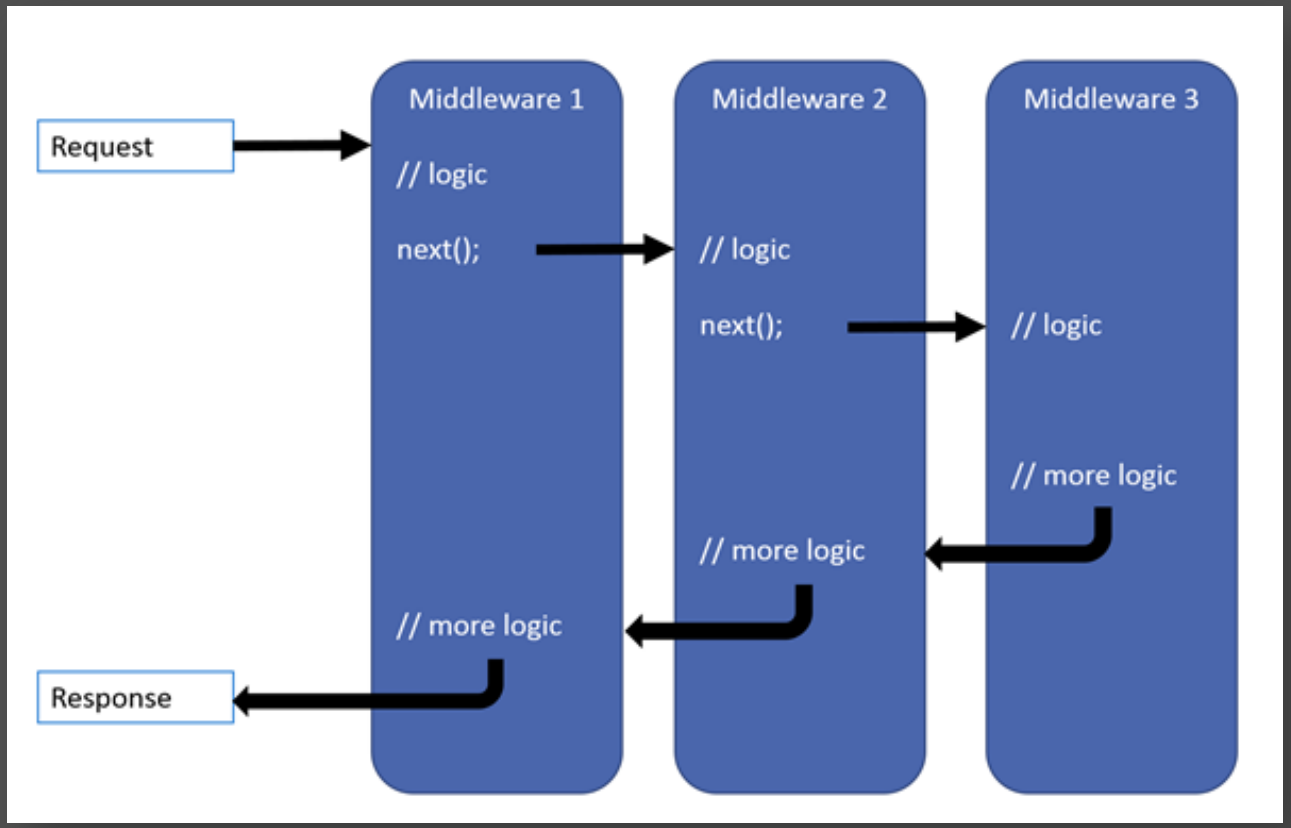
服务组件 (service components)

- ASP.NET 5 的模块化以「服务」为中心
  - 要在 ASP.NET 5 中使用这些组件 (component)，必须靠 DI (相依性注入)
  - ASP.NET 5 内置 IoC (控制反转) 容器，也可抽换成其他的 IoC
  - ASP.NET 5 默认可让类建构式注入服务组件 (component)

- 服务区分三种类型
  - **Singleton** 每次 Web 应用程序启动会只会创建一个实体
    - `services.AddSingleton<IMyApp, MyApp>();`
  - **Scoped** 每次 HTTP 要求都会创建一次实体
    - `services.AddScoped<IMember, Member>();`
  - **Transient** 每次执行注入都会创建全新实体
    - `services.AddTransient<IEmailSender, AuthMessageSender>();`

中间件 (Middleware)

- ASP.NET 5 支援 Middleware 架构
  - 可将组件依序套用到 HTTP 要求管线 (顺序很重要)
  - 预设定义在 Startup 类中的 Configure() 方法



思考题：套用 Middleware 顺序

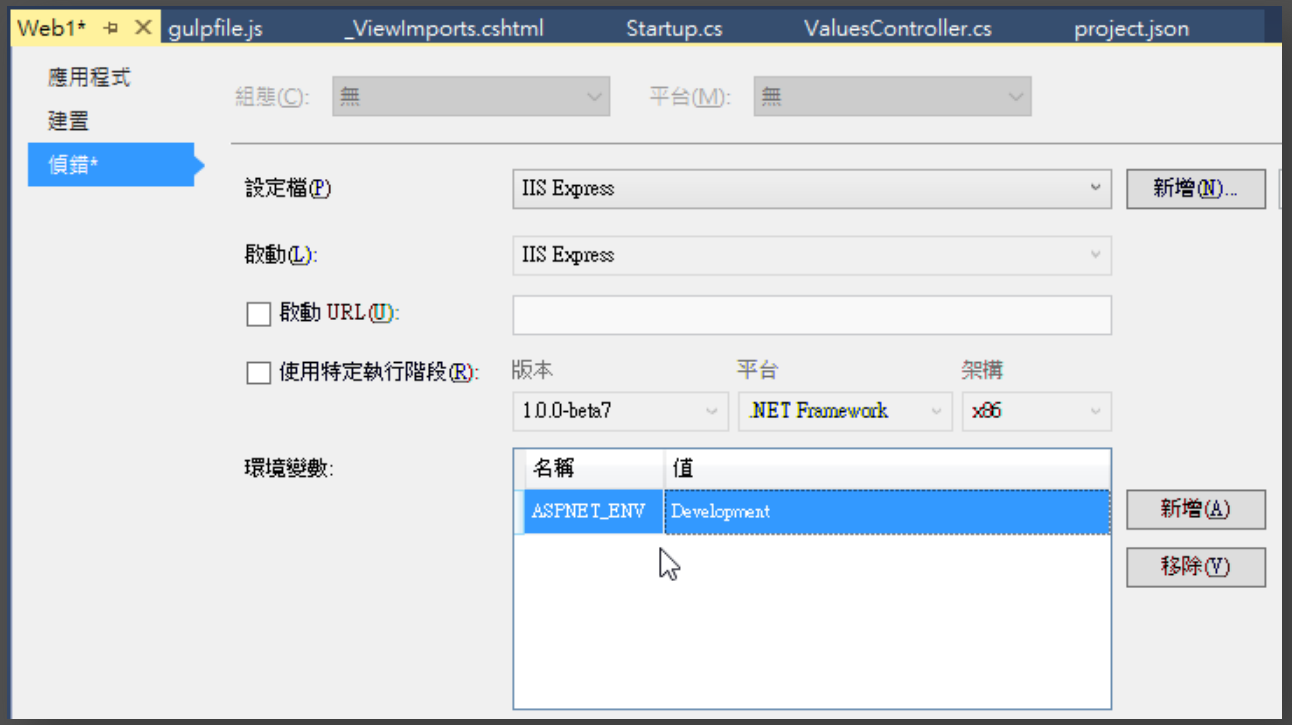
- 看一下 Startup.cs 的 Configure() 方法
  - 为何 loggerFactory.AddConsole(); 要摆在第一顺位?
  - 为何 app.UseStaticFiles(); 要设定在app.UseIdentity(); 之前？
  - 如果顺序调换会怎样？
  - 如果加入 app.UseDirectoryBrowser(); 应该摆在哪里？
  - 每个 Middleware 到底在做甚么？做了什么事？有没有文檔?

ASP.NET 5 内置的 Middleware

- Authentication
  - 提供身分验证支持
- CORS
  - 设定 Cross-Origin Resource Sharing (CORS)
- Diagnostics
  - 支持错误页面显示与执行时期信息
- Routing
  - 定义路由与限制
- Session
  - 提供 Session 支援
- Static Files
  - 提供 wwwroot (网站根目录) 下的静态档案

多重执行环境

- ASP.NET 5 可依据不同环境套用不同设定
- 指定 **ASPNET_ENV** 环境变量
  - Development
  - Staging
  - Production



- 项目设定
  - launchSettings.json

自定义环境与 Middleware 范例

- **SET ASPNET_ENV=LogInline**
- **dnx web** http://localhost:5000

```
1   public void ConfigureLogInline(IApplicationBuilder app, ILoggerFactory loggerfactory)
2   {
3       loggerfactory.AddConsole(minLevel: LogLevel.Information);
4       var logger = loggerfactory.CreateLogger(_environment);
5       app.Use(async (context, next) =>
6       {
7           logger.LogInformation("Handling request.");
8           await next.Invoke();
9           logger.LogInformation("Finished handling request.");
10      });
11
12      app.Run(async context =>
13      {
14          await context.Response.WriteAsync("Hello from " + _environment);
15      });
16  }
```

课后提醒

分会场课程视频观看及课件下载

大会结束后一个月内，敬请访问微软 Channel 9 官网观看更多课程视频及下载课件

https://channel9.msdn.com/Events/Ignite/Microsoft-Ignite-China-2015

或

http://aka.ms/IgniteChina2015

Channel 9

请您填写分会场课程反馈表

您的反馈将帮助我们改进

- 填写表格后，请离场时交给本会场的现场工作人员
- 谢谢！

您对本节课程的满意度：
(每行只勾选一项，请注意：**9 分代表非常满意**，分数越少，满意度则越小，1 分代表非常不满意)

本节课程整体满意度评价	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
您对讲师的专业水平是否满意？									
您对讲师的表达技巧与能力是否满意？									
您对 Demo 的解说及效果是否满意？ (如本节课没有 Demo 请留白)									
您对课程内容的安排是否满意？									

如您还有其他意见与建议：

谢谢！

Microsoft