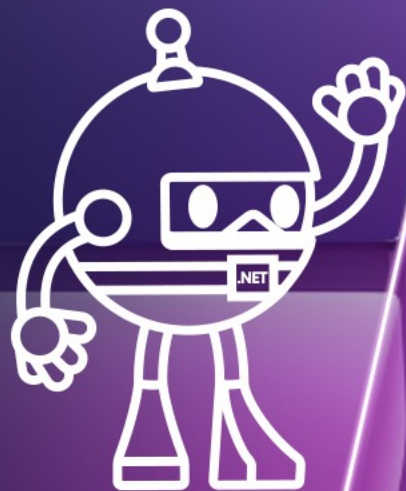


.NET Conf China 2023

2023/12/16
09:30 - 18:00

中国 · 北京



.NET Framework升级到 .NET 8 实践经验

骆姜斌 (SSW)

微软最有价值专家 (MVP)

FireUG技术社区组织者之一



 FireUG 技术社区

做些好玩的程序员下饭小视频





主要内容

- 升级前的准备工作
- 升级中常见问题以及解决方案
- 升级常用的工具和组件



项目背景

- 十年以上的旧项目, 历经多次升级
- 基于.NET Framework 4.7.2
- 包含了40+项目的单机应用
- 大量第三方服务引用





目标

- 升级过程中, 需要保证当前功能的稳定运行
- 升级过程中, 需要同时支持正常的功能迭代
- 最终, 需要删除所有的.NET Framework代码



准备工作

- 系统架构评估, 历史技术债务梳理
- 依赖项分析
 - 系统组件不再支持, 例如 System.Web
 - 第三方组件
- 修改所有项目为多目标框架 (.NET Framework和.NET8)
 - 在编译过程中暴露不支持.NET8的相关代码
 - 保证新增的逻辑能够同时支持.NET Framework和.NET8
- 为Web应用创建并行增量(side-by-side incremental)项目
 - 实质是创建一个ASP.NET Core Web应用
 - 通过反向代理, 实现请求在ASP.NET Web应用和ASP.NET Core Web应用间的分发





准备工作 - 多目标框架

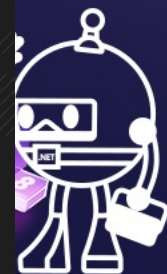
```

SSW.SugarLearn...esolver.csproj
↑ ↓ 15 changes -104 +20
Diff - Theirs vs. Ours

Theirs
- <?xml version="1.0" encoding="utf-8"?>
- <Project ToolsVersion="12.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/20
- <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists
  <PropertyGroup>
-   <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
-   <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
-   <ProjectGuid>{82FC9AC0-168C-4019-BA27-5A3EDCA5AAEE}</ProjectGuid>
-   <OutputType>Library</OutputType>
-   <AppDesignerFolder>Properties</AppDesignerFolder>
-   <RootNamespace>SSW.SugarLearning.DependencyResolver</RootNamespace>
-   <AssemblyName>SSW.SugarLearning.DependencyResolver</AssemblyName>
-   <TargetFrameworkVersion>v4.7.2</TargetFrameworkVersion>
-   <FileAlignment>512</FileAlignment>
-   <TargetFrameworkProfile />
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
-   <DebugSymbols>true</DebugSymbols>
-   <DebugType>full</DebugType>
-   <Optimize>>false</Optimize>
-   <OutputPath>bin\Debug</OutputPath>
-   <DefineConstants>DEBUG;TRACE</DefineConstants>
-   <ErrorReport>prompt</ErrorReport>
-   <WarningLevel>4</WarningLevel>
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
-   <DebugType>pdbonly</DebugType>
-   <Optimize>>true</Optimize>
-   <OutputPath>bin\Release</OutputPath>
-   <DefineConstants>TRACE</DefineConstants>
-   <ErrorReport>prompt</ErrorReport>
-   <WarningLevel>4</WarningLevel>
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'staging|AnyCPU' ">
-   <OutputPath>bin\staging</OutputPath>
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'production|AnyCPU' ">
-   <OutputPath>bin\production</OutputPath>
-   <DefineConstants>TRACE</DefineConstants>
-   <Optimize>>true</Optimize>
-   <DebugType>pdbonly</DebugType>
-   <PlatformTarget>AnyCPU</PlatformTarget>
-   <ErrorReport>prompt</ErrorReport>
-   <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'SSW.SugarLearning.Staging|AnyCPU' ">
-   <OutputPath>bin\SSW.SugarLearning.Staging</OutputPath>
  </PropertyGroup>
- <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'SSW.SugarLearning.Production|AnyCPU' ">
-   <OutputPath>bin\SSW.SugarLearning.Production</OutputPath>
  </PropertyGroup>
</Project>

Ours
+ <Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
+   <TargetFrameworks>net472</TargetFrameworks>
  </PropertyGroup>
  <OutputType>Library</OutputType>
+   <GenerateAssemblyInfo>false</GenerateAssemblyInfo>
+   <AssemblyTitle>SSW.SugarLearning.DependencyResolver</AssemblyTitle>
+   <Product>SSW.SugarLearning.DependencyResolver</Product>
+   <Copyright>Copyright © 2014</Copyright>
+   <AssemblyVersion>1.0.0.0</AssemblyVersion>
+   <FileVersion>1.0.0.0</FileVersion>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'staging|AnyCPU' ">
    <OutputPath>bin\staging</OutputPath>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'production|AnyCPU' ">
    <OutputPath>bin\production</OutputPath>
    <Optimize>>true</Optimize>
  </PropertyGroup>
  <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'SSW.SugarLearning.Staging|AnyCPU' ">
    <OutputPath>bin\SSW.SugarLearning.Staging</OutputPath>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'SSW.SugarLearning.Production|AnyCPU' ">
    <OutputPath>bin\SSW.SugarLearning.Production</OutputPath>
  </PropertyGroup>
</Project>

```



准备工作 - 多目标框架

- 修改项目文件(.csproj), 将目标框架设置为 .NET Framework 和 .NET8

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net472</TargetFramework>
  </PropertyGroup>
</Project>
```

→

```
1 <Project Sdk="Microsoft.NET.Sdk">
2   <PropertyGroup>
3+   <TargetFrameworks>net472;net8.0</TargetFrameworks>
4   </PropertyGroup>
</Project>
```

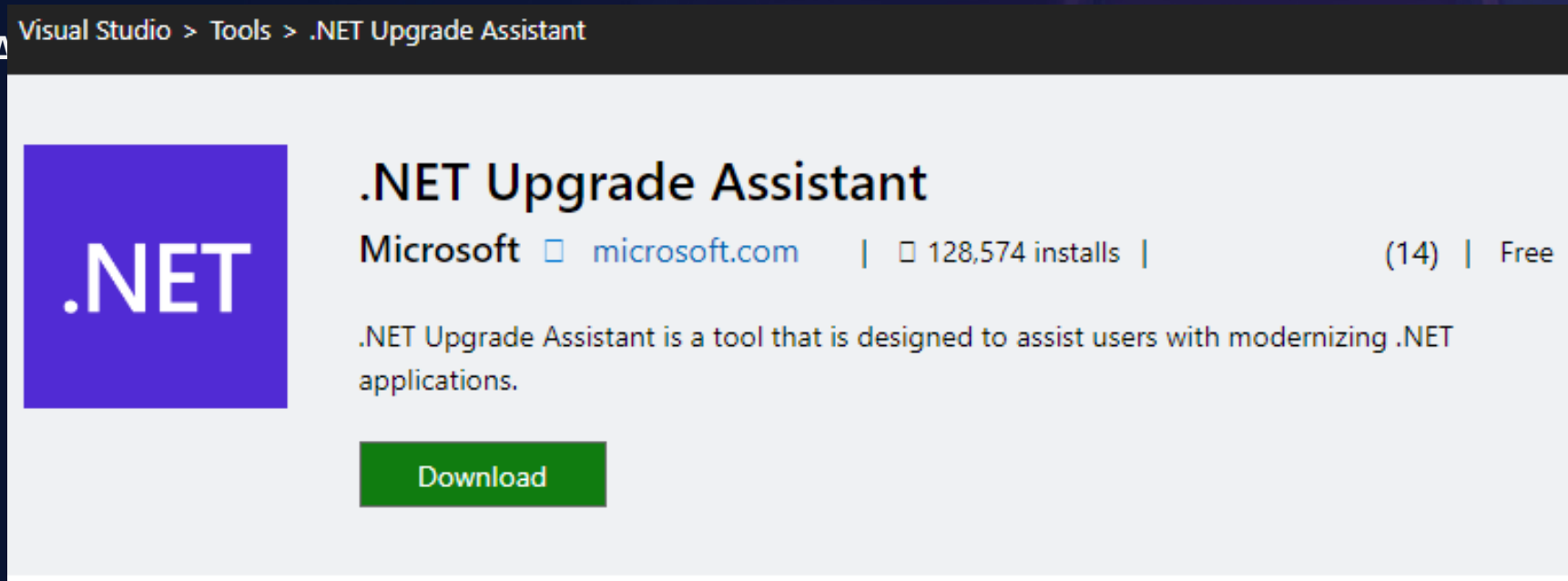
- 编译
- 编译失败
 - 如果失败问题非常简单, 直接修复后, 重新编译
 - 如果失败问题很多或很复杂, 新增PBI等待处理, 并在目标框架中删除.NET8



准备工作 - 并行增量项目

- .NET升级助手 (.NET Upgrade Assistant)
 - Visual Studio官方扩展
 - 帮助.NET开发人员将.NET Framework应用程序迁移到.NET5+

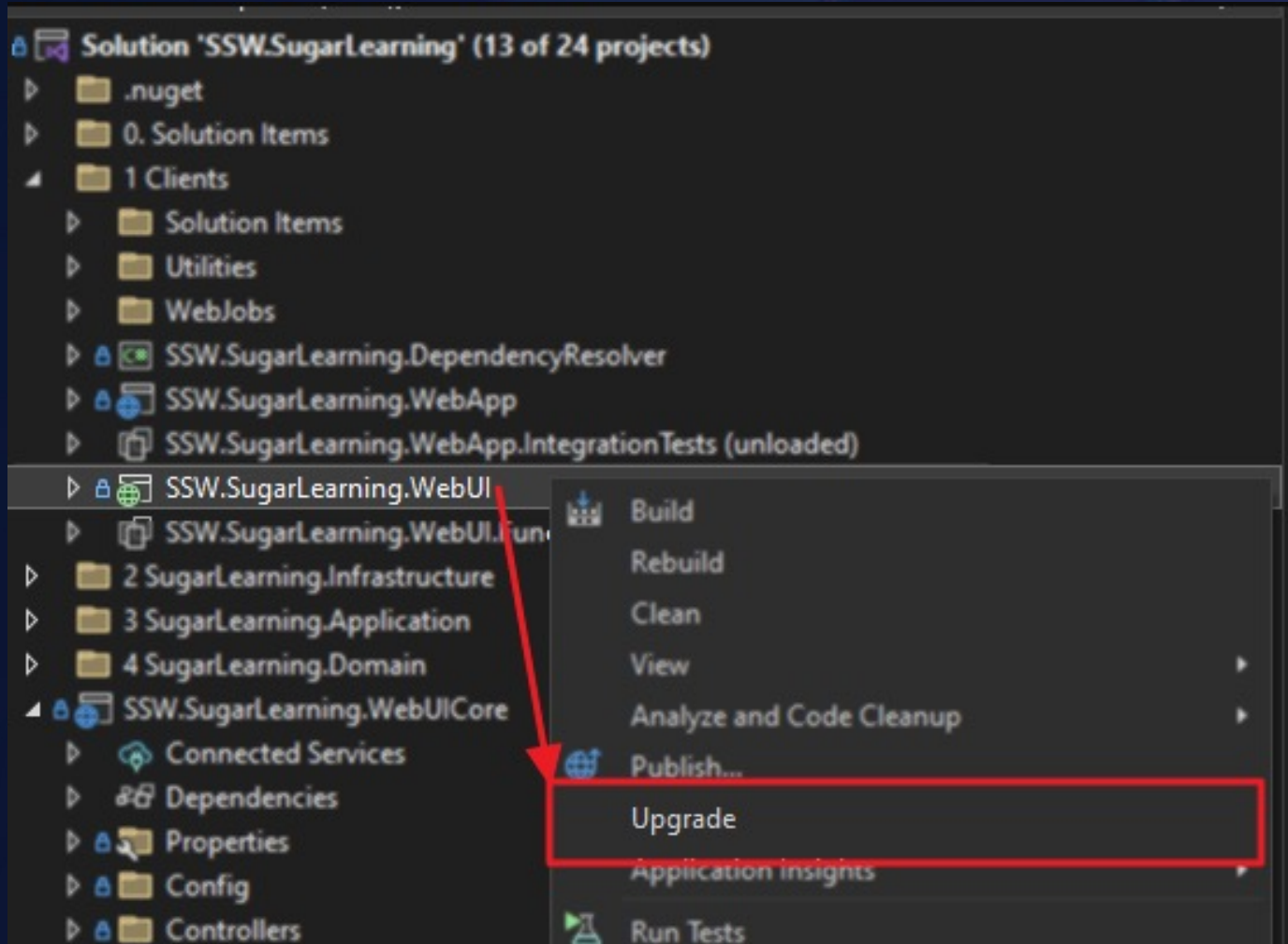
• YAA Visual Studio > Tools > .NET Upgrade Assistant



The screenshot shows the Visual Studio extension marketplace page for the ".NET Upgrade Assistant" extension. The breadcrumb navigation at the top reads "Visual Studio > Tools > .NET Upgrade Assistant". On the left is a purple square icon with ".NET" in white. The main title is ".NET Upgrade Assistant" by Microsoft, with a link to "microsoft.com". It shows "128,574 installs" and "(14)" reviews. The price is listed as "Free". A description states: ".NET Upgrade Assistant is a tool that is designed to assist users with modernizing .NET applications." A green "Download" button is visible at the bottom.



准备工作 - 并行增量项目



准备工作 - 并行增量项目



SSW.SugarLearning.WebUICore
..\SSW.SugarLearning.WebUICore\SSW.SugarLearning.WebUICore.csproj

Leave Feedback

Summary Endpoints explorer Upgrade More actions

Summary

SSW.SugarLearning.WebUI is now connected to SSW.SugarLearning.WebUICore via Yarp proxy.

To discover and upgrade individual components of your project use corresponding context menus on Solution Explorer items (controller, class view etc) or inside the editor. [Learn more](#)

Upgrade Controller

Upgrade Class

Upgrade View

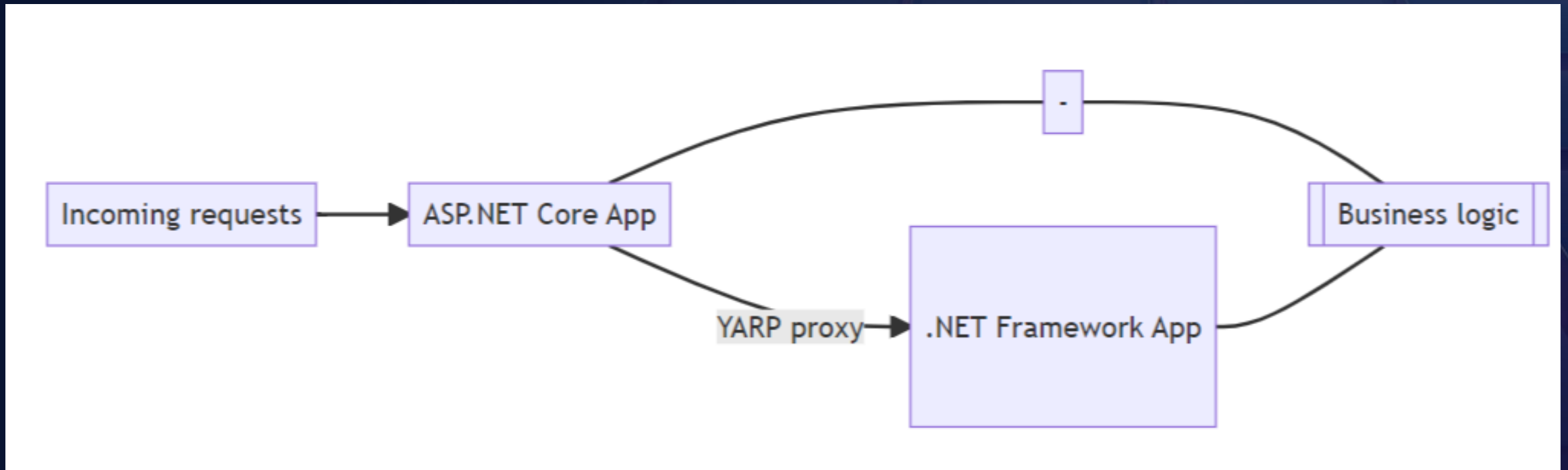
Endpoints



.NET Framework - 60 (55.56%)
.NET - 48 (44.44%)



准备工作 - 并行增量项目



准备工作 - 并行增量项目



SSW.SugarLearning.WebUICore
..\SSW.SugarLearning.WebUICore\SSW.SugarLearning.WebUICore.csproj

Leave Feedback

Summary Endpoints explorer Upgrade More actions

Summary

SSW.SugarLearning.WebUI is now connected to SSW.SugarLearning.WebUICore via Yarp proxy.

To discover and upgrade individual components of your project use corresponding context menus on Solution Explorer items (controller, class view etc) or inside the editor. [Learn more](#)

Upgrade Controller

Upgrade Class

Upgrade View

Endpoints



.NET Framework - 60 (55.56%)
.NET - 48 (44.44%)





Summary Endpoints explorer Upgrade More actions

Select components > Upgrade

Upgrade selected components

Complete: 4 succeeded, 0 failed, 69 skipped.

73/73

- Initialize project components ✔
- SSW.SugarLearning.Domain ✔
- SSW.SugarLearning.Application ✔
- SSW.SugarLearning.WebUI ✔
- System.Web ✔

Successfully applied all transformations corresponding to this component. See details in the Upgrade Assistant output pane.

- Finalize project dependencies ✔
- CompanyGroupController (SSW.SugarLearning.WebUI.Controllers.v2) ✔
- IMediator (MediatR) ✔
- ApiResponse (SSW.SugarLearning.Application.Common.Models) ✔
- DeleteCompanyGroup (SSW.SugarLearning.Application.Company.Commands) ✔
- SaveCompanyGroup (SSW.SugarLearning.Application.Company.Commands) ✔
- CompanyGroupModel (SSW.SugarLearning.Application.Company.Models) ✔
- CompanyUserModel (SSW.SugarLearning.Application.Company.Models) ✔
- GetCompanyGroup (SSW.SugarLearning.Application.Company.Queries) ✔
- GetCompanyGroupMembers (SSW.SugarLearning.Application.Company.Queries) ✔





.NET

准



SSW.SugarLearning.WebUICore
..\SSW.SugarLearning.WebUICore\SSW.SugarLearning.WebUICore.csproj

Leave Feedback

NET Conf China 2023

中国·北京

Summary Endpoints explorer Upgrade More actions

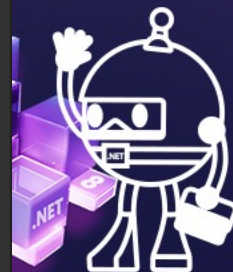
Endpoints explorer

SSW.SugarLearning.WebUI

- POST /api/v2/admin/modules/save, params (SSW.SugarLearning.ApplicationItem)
- POST /api/v2/admin/modules/search, params (SSW.SugarLearning.ApplicationItem)
- POST /api/v2/admin/modules/toggleReminder, params (int moduleId)
- POST /api/v2/backlog/{companyCode}/{userId}, params (string companyCode)
- POST /api/v2/backlog/{companyCode}/{userId}/search, params (string companyCode)
- GET /api/v2/companies/{companyCode}, params (string companyCode)
- GET /api/v2/companies/{companyCode}/admins, params (string companyCode)
- GET /api/v2/companies/{companyCode}/approvals, params (string companyCode)
- GET /api/v2/companies/{companyCode}/items, params (string companyCode)
- GET /api/v2/companies/{companyCode}/modules, params (string companyCode)
- GET /api/v2/companies/{companyCode}/users, params (string companyCode)
- GET /api/v2/companies/{companyCode}/users/count, params (string companyCode)
- DELETE /api/v2/company/{companyCode}/group/{groupId}/Delete
- GET /api/v2/company/{companyCode}/group/{groupId}/GetAssignedModules
- GET /api/v2/company/{companyCode}/group/{groupId}/GetDetails
- GET /api/v2/company/{companyCode}/group/{groupId}/searchUser
- GET /api/v2/company/{companyCode}/group/GetGroupMembers, params (string companyCode)
- POST /api/v2/company/{companyCode}/group/Post
- POST /api/v2/company/{companyCode}/modules/search, params (SSW.SugarLearning.ApplicationItem)

SSW.SugarLearning.WebUICore

- Routes
- Endpoints
 - POST /api/v2/activity/Create, params (SSW.SugarLearning.ApplicationItem)
 - GET /api/v2/admin/{companyCode}/GetPendingApprovals, params (string companyCode)
 - GET /api/v2/admin/{companyCode}/GetPendingApprovals/{Id}, params (string Id)
 - GET /api/v2/admin/{companyCode}/GetPendingApprovalsCountForItem/{Id}, params (string Id)
 - POST /api/v2/backlog/{companyCode}/{userId}, params (string companyCode)
 - POST /api/v2/backlog/{companyCode}/{userId}/search, params (string companyCode)
 - GET /api/v2/companies/{companyCode}, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/admins, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/approvals, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/items, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/modules, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/users, params (string companyCode)
 - GET /api/v2/companies/{companyCode}/users/count, params (string companyCode)
 - DELETE /api/v2/company/{companyCode}/group/{groupId}/Delete
 - GET /api/v2/company/{companyCode}/group/{groupId}/GetAssignedModules
 - GET /api/v2/company/{companyCode}/group/{groupId}/GetDetails
 - GET /api/v2/company/{companyCode}/group/{groupId}/searchUser



准备工作 - 最终结果

- 可以在兼容模式下正常运行的项目
- 技术债务相关PBI
- 兼容性相关PBI
- 升级相关PBI





开始升级 - 常见的依赖项问题

• 依赖

System.Web 适配器

命名空间 `Microsoft.AspNetCore.SystemWebAdapters` 是运行时帮助程序集合，用于在移动到 ASP.NET Core 时帮助使用针对 `System.Web` 编写的代码。可以通过几个包来使用这些适配器的功能：

• 依赖

- `Microsoft.AspNetCore.SystemWebAdapters`：此包用于支持库，并提供你可能已依赖的 `System.Web` API，例如 `HttpContext` 和其他 API。此包面向 .NET Standard 2.0、.NET 4.5+ 和 .NET 6+。
- `Microsoft.AspNetCore.SystemWebAdapters.FrameworkServices`：此包仅面向 .NET Framework，并旨在向可能需要提供增量迁移的 ASP.NET Framework 应用程序提供服务。这通常不应从库引用，而是从应用程序本身引用。
- `Microsoft.AspNetCore.SystemWebAdapters.CoreServices`：此包仅面向 .NET 6+，并旨在向 ASP.NET Core 应用程序提供服务，以配置 `System.Web` API 的行为以及选择使用任何行为来进行增量迁移。这通常不应从库引用，而是从应用程序本身引用。
- `Microsoft.AspNetCore.SystemWebAdapters.Abstractions`：此包是一个支持包，它为 ASP.NET Core 和 ASP.NET Framework 应用程序使用的服务提供抽象，例如会话状态序列化。

• 依赖

• 依赖

• 一些第三方组件





开始升级 - 如何管理多个目标框架的兼容性?

条件编译指令

```
public static class WebClientFactory
{
    public static IWebClient GetWebClient()
    {
#if NET472
        return new CustomWebClient();
#else
        return new CustomHttpClient();
#endif
    }
}
```

MSBuild conditions

```
<ItemGroup Condition="'$(TargetFramework)' == 'net472'">
    <Reference Include="System.Web" />
    <Reference Include="System.Web.Extensions" />
    <Reference Include="System.Web.ApplicationServices" />
</ItemGroup>
```



升级 – 如何升级EF EDMX到EF Core 3.1

- 为什么是EF Core3.1?
 - EF Core3.1是最后一个支持.NET Framework的版本
 - 所以就产生了一个鸡和蛋的问题
 - EF Core 8.0升级的前提是, 所有引用EF Core的项目先完成.NET 8的升级
 - 同时, 需要先完成EF Core 8.0的升级, 才可以顺利的完成.NET8的升级
- 完整的升级路径

EF EDMX → 升级到EF Core 3.1 → 完成.NET8升级 → 升级到EF Core 8.0



升级 – 如何升级EF EDMX到EF Core 3.1

- 1. 将ObjectContext/Entities类抽象成接口
- 2. 逆向工程, 生成EF Core 3.1相关类
 - 使用Visual Studio扩展: EF Core Power Tools
- 3. 基于EF Core 3.1相关类, 实现步骤1中的相关接口
 - 更新 DbContext.OnConfiguring方法
 - 用 DbSet<T>替代ObjectSet<T>
 - 使用贪婪加载(Eager Loading) 代替 懒加载(Lazy loading)
 - 使用.AsEnumerable()或SQL语句解决GroupBy的问题



升级 - 如何升级EF EDMX到EF Core 3.1

```
PrintSql(
    "Group by (no post count)",
    dbContext.Author
        .GroupBy(x => x.Country)
        .Select(x => new
            {
                Country = x.Key,
                Authors = x.Count()
            }));
```

✅ EF Core 3.1

```
PrintSql(
    "Group by",
    dbContext.Author
        .GroupBy(x => x.Country)
        .Select(x => new
            {
                Country = x.Key,
                Authors = x.Count(),
                Post = x.SelectMany(x => x.Posts).Count()
            }));
```

❌ EF Core 3.1
✅ EF
✅ EF Core 8.0

```
dbContext.Author
    .Include(x => x.Posts)
    .GroupBy(x => x.Country)
    .AsEnumerable()
    .Select(x => new
        {
            Country = x.Key,
            Authors = x.Count(),
            Post = x.SelectMany(x => x.Posts).Count()
        }));
```

✅ ❌ EF Core 3.1

```
dbContext.Author
    .Include(x => x.Posts)
    .AsEnumerable()
    .GroupBy(x => x.Country)
    .Select(x => new
        {
            Country = x.Key,
            Authors = x.Count(),
            Post = x.SelectMany(x => x.Posts).Count()
        }));
```

✅ EF Core 3.1



升级 – 如何升级EF EDMX到EF Core 3.1

- 4. 更新命名空间引用
 - 使用 `Microsoft.EntityFrameworkCore` 代替 `System.Data.Entity`
- 5. 更新依赖注入
 - 使用 `.AddDbContext()` 或 `.AddDbContextPool()`
- 6. 测试, 测试, 测试.. (重要的事情说3遍), 以下是升级后的常见问题:
 - 懒加载
 - 有些查询在EF Core 3.1中不支持
 - 部分复杂查询会出现性能问题
 - 部分查询结果会出现异常
- 7. 删除EDMX文件



升级 – 如何升级EF EDMX到EF Core 3.1

- 更多资料

- How to migrate from EDMX to EF Core - <https://www.ssw.com.au/rules/migrate-from-edmx-to-ef-core/>
- How to migrate to EF Core 3.1 video - <https://learn.microsoft.com/en-us/shows/on-net/migrating-edmx-projects-to-entity-framework-core>
- Official porting docs to EF Core 3.1 - <https://learn.microsoft.com/en-us/ef/efcore-and-ef6/porting/port-edmx>



升级 – System.Web

- 在ASP.NET Core中, System.Web已经被移除
 - 常见的问题是缺少 HttpContext.Current
- 方案一: 使用 IHttpContextAccessor
 - 只在.NET 8 中生效

```
public class SomeService
{
    public void DoSomething()
    {
        var httpContext = HttpContext.Current;
        // Rest of the code...
    }
}
```

```
public class SomeService
{
    private readonly IHttpContextAccessor _httpContextAccessor;

    public SomeService(IHttpContextAccessor httpContextAccessor)
    {
        _httpContextAccessor = httpContextAccessor;
    }

    public void DoSomething()
    {
        var httpContext = _httpContextAccessor.HttpContext;
        // Rest of the code...
    }
}
```





升级 – System.Web

- 方案二：抽象方法，分别用.NET Fra

```
public interface IRequestContext
{
    string GetCurrentUsername();
}
```

```
#if NETFRAMEWORK

// For .NET Framework
public sealed class LegacyHttpRequestContext : IRequestContext
{
    public string GetCurrentUsername()
        => HttpContext.Current?.User?.Identity?.Name;
}

#else // Or #if NET

// For .NET Core and .NET
public sealed class HttpRequestContext : IRequestContext
{
    private readonly IHttpContextAccessor _httpContextAccessor;

    public HttpRequestContext(IHttpContextAccessor httpContextAccessor)
    {
        _httpContextAccessor = httpContextAccessor;
    }

    public string GetCurrentUsername()
        => _httpContextAccessor.HttpContext.User.Identity?.Name;
}

#endif
```



升级 – System.Web

- 方案三: 在.NET Framework中, 实现HttpContextAccessor

```
<!-- .NET Framework reference for HttpContext -->
<ItemGroup Condition="'$(TargetFramework)' == 'net472'">
  <Reference Include="System.Web" />
</ItemGroup>

<!-- .NET reference for IHttpContextAccessor, already included if the project is WebApp -->
<ItemGroup Condition="'$(TargetFramework)' == 'net8.0'">
  <FrameworkReference Include="Microsoft.AspNetCore.App" />
</ItemGroup>

    public HttpContext HttpContext => HttpContext.Current;
  }
#endif
}
```





升级中 – System.Web

- 方案四: 使用System.Web适配器

System.Web 适配器

命名空间 `Microsoft.AspNetCore.SystemWebAdapters` 是运行时帮助程序集合，用于在移动到 ASP.NET Core 时帮助使用针对 `System.Web` 编写的代码。可以通过几个包来使用这些适配器的功能：

- `Microsoft.AspNetCore.SystemWebAdapters`：此包用于支持库，并提供你可能已依赖的 System.Web API，例如 `HttpContext` 和其他 API。此包面向 .NET Standard 2.0、.NET 4.5+ 和 .NET 6+。
- `Microsoft.AspNetCore.SystemWebAdapters.FrameworkServices`：此包仅面向 .NET Framework，并旨在向可能需要提供增量迁移的 ASP.NET Framework 应用程序提供服务。这通常不应从库引用，而是从应用程序本身引用。
- `Microsoft.AspNetCore.SystemWebAdapters.CoreServices`：此包仅面向 .NET 6+，并旨在向 ASP.NET Core 应用程序提供服务，以配置 `System.Web` API 的行为以及选择使用任何行为来进行增量迁移。这通常不应从库引用，而是从应用程序本身引用。
- `Microsoft.AspNetCore.SystemWebAdapters.Abstractions`：此包是一个支持包，它为 ASP.NET Core 和 ASP.NET Framework 应用程序使用的服务提供抽象，例如会话状态序列化。



升级 – Web.config

- Web.config一般包含 包引用配置, 服务相关配置以及配置项。
- 包的引用配置
 - 在ASP.NET Core中, 项目包含项列在项目的.CSPROJ文件中。需要检查应用程序的依赖项, 以确定它们是否仍然是必需的, 并且应该根据需要使用时使用NuGet包管理器添加它们。



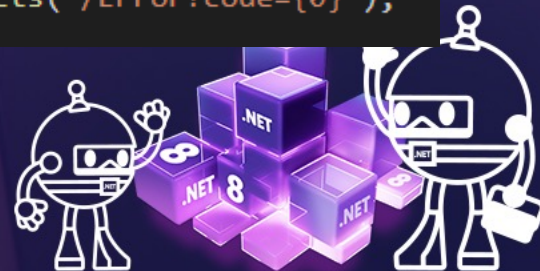
升级 – Web.config

- 服务相关配置

- <pages>/<namespaces>元素定义在程序集预编译期间使用的导入指令，使用用隐式导入功能代替
 - <https://learn.microsoft.com/en-us/dotnet/core/project-sdk/overview#implicit-using-directives>
- <httpHandlers> 负责映射URLs到IhttpHandler类，可以使用路由处理 (Routing)
 - <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-8.0>
- <httpModules> 设置了系统http相关的处理模块，需要检查并逐一代替
- 其他部分配置可以在Program.cs中实现，例如<customErrors>元素

```
<customErrors mode="RemoteOnly" defaultRedirect="~/Error">
  <error statusCode="403" redirect="~/Error?code=403" />
  <error statusCode="404" redirect="~/Error?code=404" />
</customErrors>
```

```
var app = builder.Build();
app.UseStatusCodePagesWithRedirects("/Error?code={0}");
```



升级 - Web.config

```
internal class ConfigurationService : IConfigurationService
{
    private readonly IConfiguration _configuration;

    public ConfigurationService(IConfiguration configuration)
    {
        _configuration = configuration;
    }

    public bool GetBooleanOrFalse(string key)
        => bool.TryParse(_configuration[key], out var value) && value;

    public string GetString(string key)
        => _configuration[key];

    public string GetConnectionString(string key)
        => _configuration.GetConnectionString(key);
}
```

```
internal class ConfigurationService : IConfigurationService
{
    public bool GetBooleanOrFalse(string key)
        => bool.TryParse(ConfigurationManager.AppSettings[key], out var value) && value;

    public string GetString(string key)
        => ConfigurationManager.AppSettings[key];

    public string GetConnectionString(string key)
        => ConfigurationManager.ConnectionStrings[key]?.ConnectionString;
}

namespace SSW.TimePRO.Infrastructure.NetFramework;

public class AutofacModule : Module
{
    protected override void Load(ContainerBuilder builder)
    {
        // .NET Framework specific DI
        builder.RegisterType<UserMembershipService>().As<IUserMembershipService>();
        builder.RegisterType<ConfigurationService>().As<IConfigurationService>();
        builder.RegisterType<AuthenticationFlowService>().As<IAuthenticationFlowService>();
    }
}
```

```
namespace SSW.TimePRO.Infrastructure.NetCore;

public class AutofacModule : Module
{
    protected override void Load(ContainerBuilder builder)
    {
        // .NET specific DI
        builder.RegisterType<UserMembershipService>().As<IUserMembershipService>();
        builder.RegisterType<ConfigurationService>().As<IConfigurationService>();
        builder.RegisterType<AuthenticationFlowService>().As<IAuthenticationFlowService>();
    }
}
```



升级 – Web.config

```
<ItemGroup Condition="'$(TargetFramework)' == 'net472'">
  <ProjectReference Include="..\SSW.TimePRO.Infrastructure.NetFramework\SSW.TimePRO.Infrastructure.NetFramework.csproj" />
</ItemGroup>

<ItemGroup Condition="'$(TargetFramework)' ≠ 'net472'">
  <ProjectReference Include="..\SSW.TimePRO.Infrastructure.NetCore\SSW.TimePRO.Infrastructure.NetCore.csproj" />
</ItemGroup>
```



升级 – Global.asax

- Global.asax规定了ASP.NET应用程序如何处理应用程序、会话和请求事件。
 - 升级的时候,这段代码需要重新构造。



升级 – Global.asax

- 应用程序事件

```
public class MvcApplication : HttpApplication
{
    protected void Application_Start() {
        Console.WriteLine("Start");
    }

    protected void Application_Init() {
        Console.WriteLine("Init");
    }

    protected void Application_Stopping() {
        Console.WriteLine("Stopping");
    }

    protected void Application_Stopped() {
        Console.WriteLine("Stopped");
    }
}
```

```
Console.WriteLine("Start");

var builder = WebApplication.CreateBuilder(args);
// ...
var app = builder.Build();
app.Lifetime.ApplicationStarted.Register(() => Console.WriteLine("Init"));
app.Lifetime.ApplicationStopping.Register(() => Console.WriteLine("Stopping"));
app.Lifetime.ApplicationStopped.Register(() => Console.WriteLine("Stopped"));
```



升级 – Global.asax

- 请求事件 `Application_BeginRequest` 和 `Application_EndRequest`
 - 一般采用中间件 (Middleware) 替换

```
public class MvcApplication : HttpApplication
{
    protected void Application_BeginRequest() {
        Console.WriteLine("Begin request");
    }

    protected void Application_EndRequest() {
        Console.WriteLine("End request");
    }
}
```

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.Use(async (context, next) =>
{
    Console.WriteLine("Begin request");
    await next.Invoke();
    Console.WriteLine("End request");
})
```



升级 – Global.asax

- 全局异常处理Application_Error
 - 一般采用中间件 (Middleware) 替换

```
public class MvcApplication : HttpApplication
{
    protected void Application_Error(object sender, EventArgs e) {
        var error = Server.GetLastError();
        Console.WriteLine("Error was: " + error.ToString());
    }
}
```

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.UseExceptionHandler(exceptionHandlerApp =>
{
    exceptionHandlerApp.Run(async context =>
    {
        var handlerPathFeature = context.Features.Get<IExceptionHandlerPathFeature>();
        var error = handlerPathFeature.Error;
        Console.WriteLine("Error was: " + error.ToString());

        // NOTE: context.Response allows you to set the returned status code
        // and response contents.
    })
});
```



升级 – Global.asax

- 会话事件

- `Session_Start()`，一般在检测到新用户会话时被调用，可以使用中间件替换。
<https://stackoverflow.com/questions/52533831/is-there-a-session-start-equivalent-in-net-core-mvc-2-1>
- `Session_End()`在用户会话结束时调用，通常是超时结束，在ASP.NET Core中没有等效替代。





.NET中文社区

升级 - OWIN

• OWIN (Open Web Interface for .NET)

- OWIN在.NET Web Server和托管环境中运行
- OWIN 是社区驱动
- Katana 是由微软支持
<https://github.com/dotnet/katana>

Learn / .NET / ASP.NET Core /



ASP.NET Core 中 .NET 的开放 Web 接口 (OWIN)

项目 • 2023/11/30 • 14 个参与者

反馈

本文内容

- 在 ASP.NET Core 管道中运行 OWIN 中间件
- 在基于 OWIN 的服务器上运行 ASP.NET Core 并使用其 WebSocket 支持
- OWIN 环境
- OWIN 键
- 其他资源

作者: [Steve Smith](#) 和 [Rick Anderson](#)

ASP.NET Core:

- 支持 .NET 的开放 Web 接口 (OWIN)。
- 具有与 .NET Core 兼容的 `Microsoft.Owin.*` (Katana) 库的替换项。

OWIN 允许 Web 应用从 Web 服务器分离。它定义了管道中使用中间件来处理请求和相关响应的标准方法。ASP.NET Core 应用程序和中间件可以与基于 OWIN 的应用程序、服务器和中间件进行互操作。

OWIN 提供了一个分离层，可一起使用具有不同对象模型的两个框架。`Microsoft.AspNetCore.Owin` 包提供了两个适配器实现：

- ASP.NET Core 到 OWIN
- OWIN 到 ASP.NET Core

此方法可将 ASP.NET Core 托管在兼容 OWIN 的服务器/主机上，或在 ASP.NET Core 上运行其他兼容 OWIN 的组件。

① 备注

使用这些适配器会带来性能成本。仅使用 ASP.NET Core 组件的应用不应使用 `Microsoft.AspNetCore.Owin` 包或适配器。

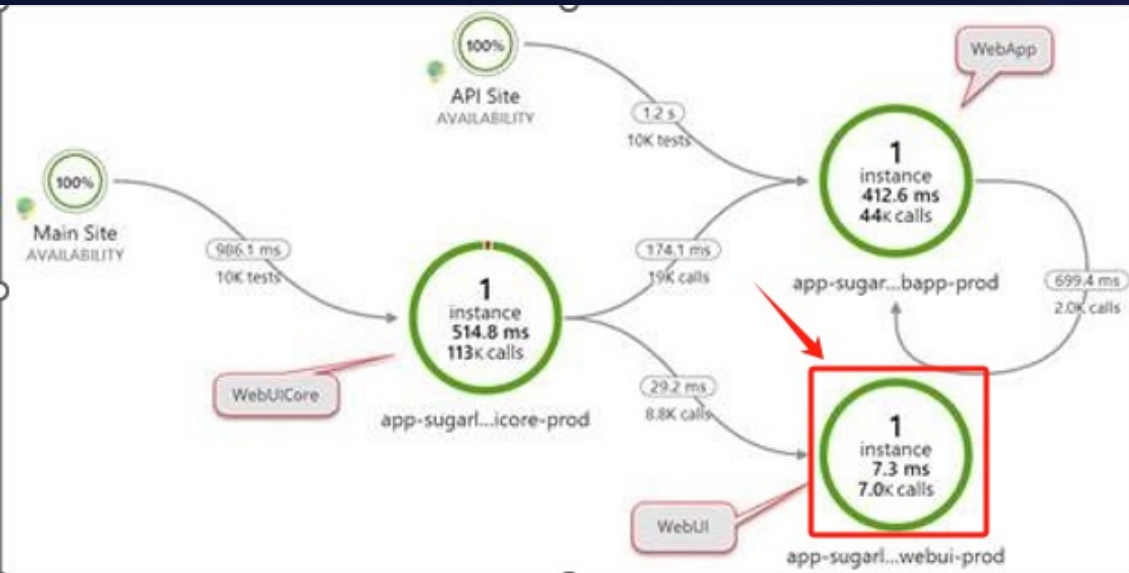
.NET Conf China 2023
中国 · 北京

OWIN的目标是用于解耦

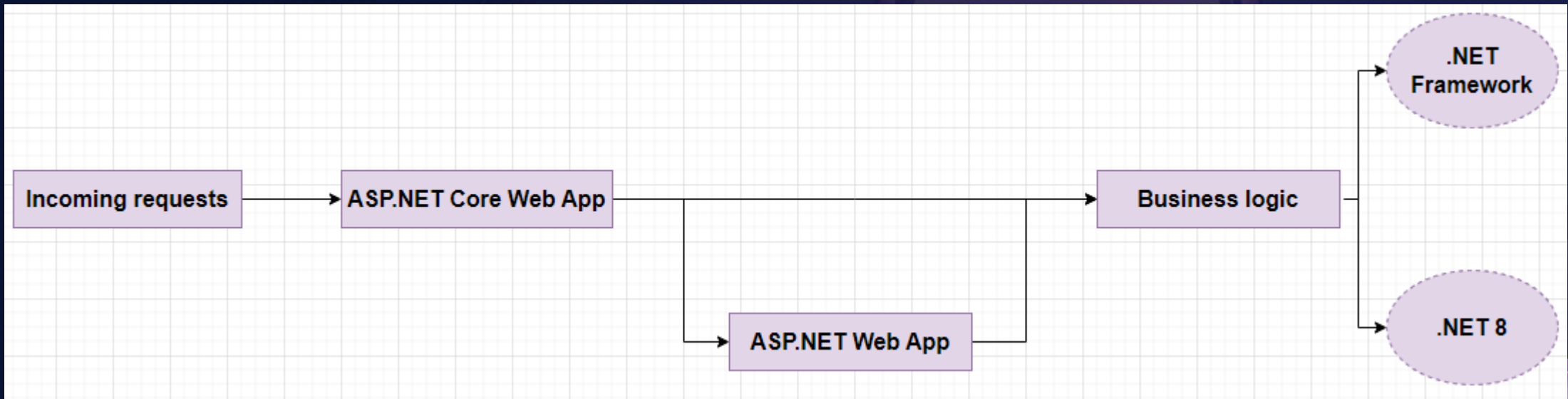
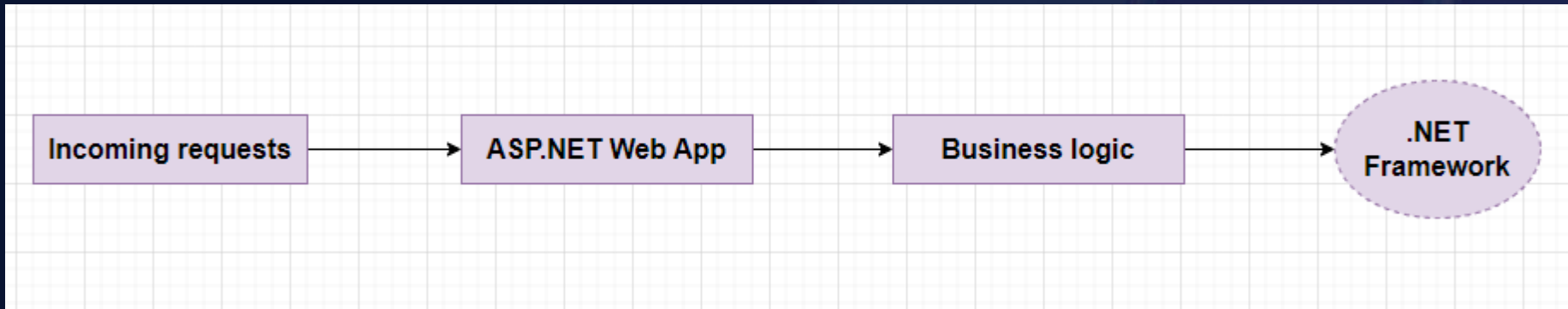


升级 - 监控

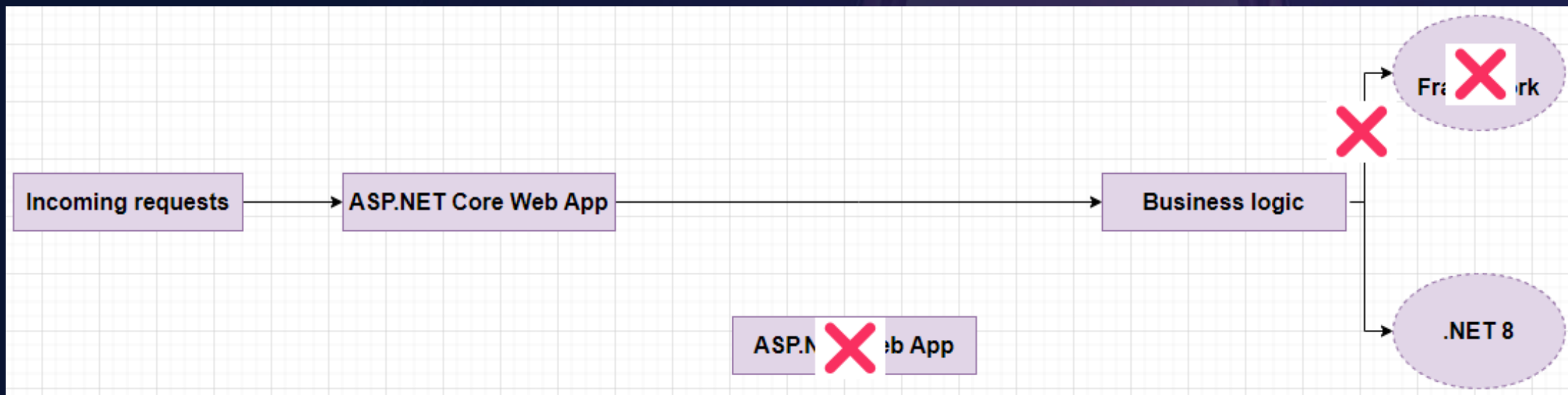
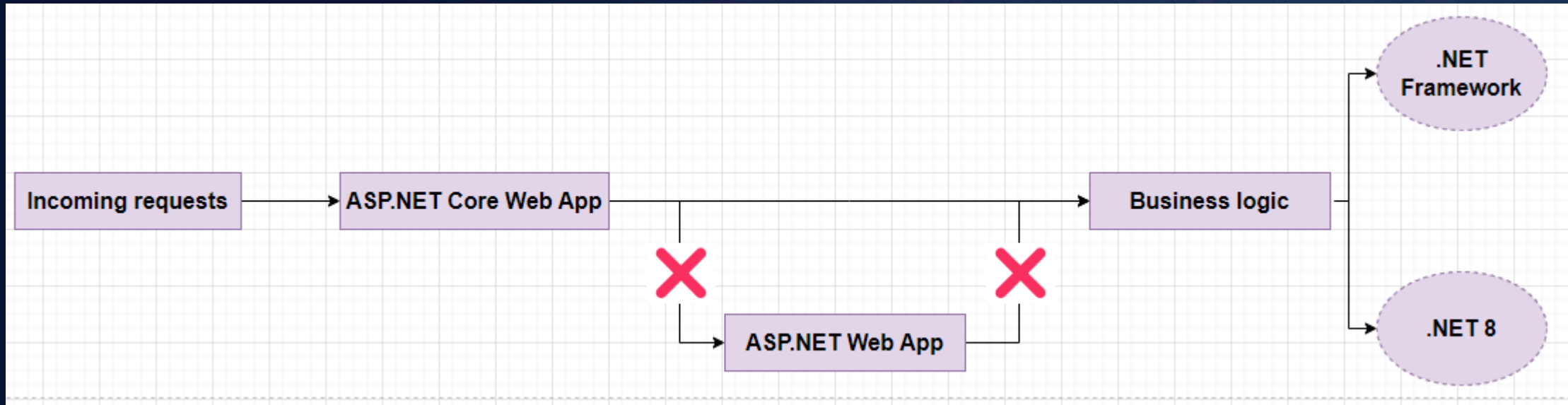
- 使用Azure Application Insights实现对应用的监控



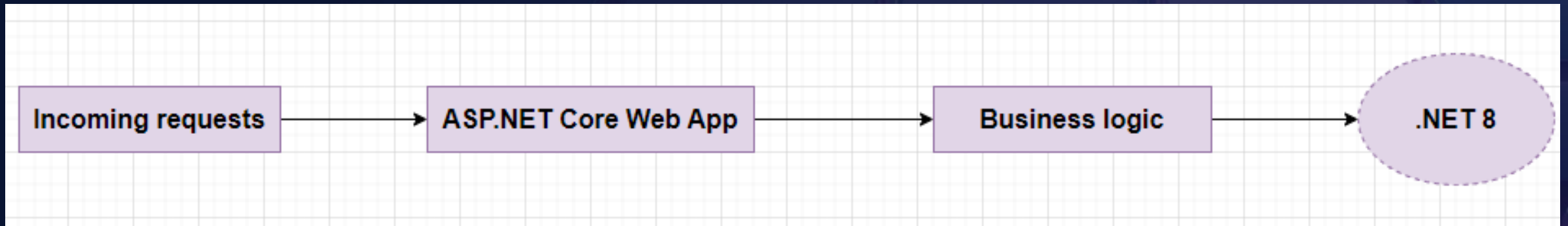
总结 - 升级流程



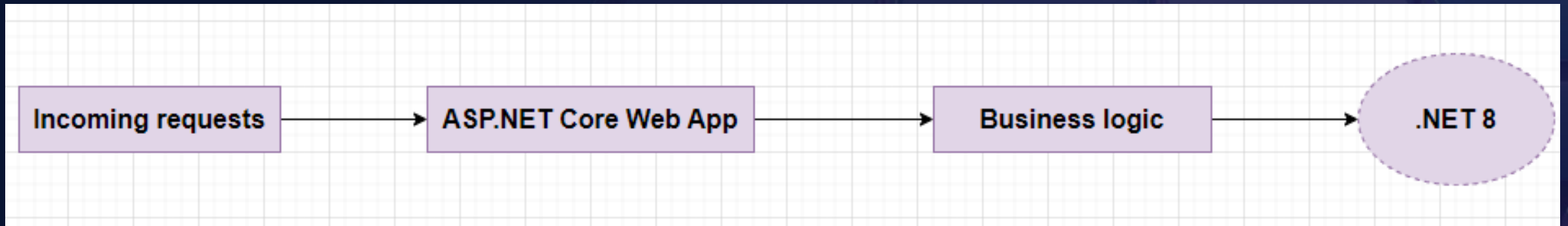
总结 - 升级流程



总结 - 升级流程



总结 - 升级流程



总结 - 工具

- try-convert组件, 转换项目文件(.csproj)为SDK格式
 - <https://github.com/dotnet/try-convert>
- .NET升级助手 (.NET Upgrade Assistant), 创建ASP.NET Core Web应用
 - <https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.upgradeassistant>
- YARP, 请求转发
 - <https://microsoft.github.io/reverse-proxy/>
- EF Core Power Tools, 逆向工程生成EF Core相关类
 - <https://marketplace.visualstudio.com/items?itemName=ErikEJ.EFCorePowerTools>
- Azure Application Insights, 监控系统运行情况



谢谢大家!



 FireUG 技术社区

做些好玩的程序员下饭小视频

