编写自定义安全提供程序

1. 内容概述

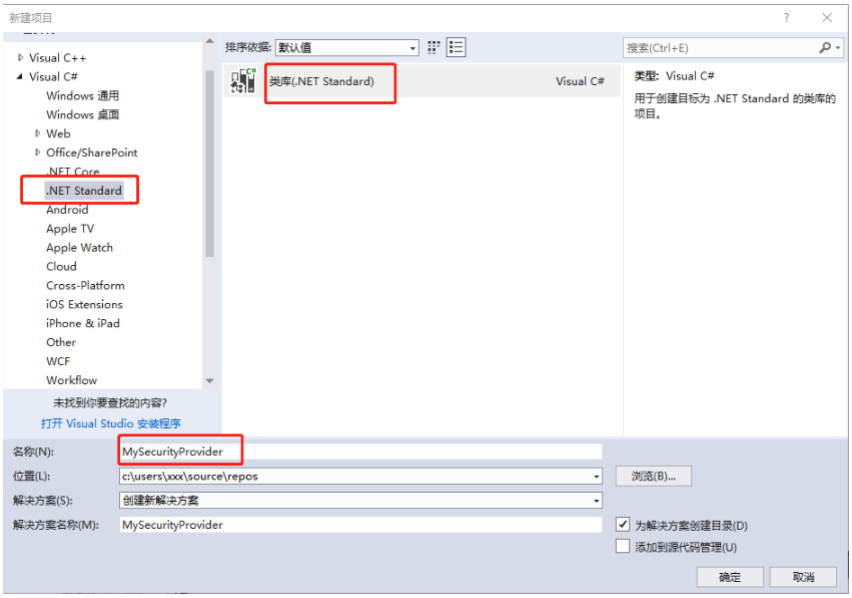
自定义安全提供程序是一个实现了Wyn规定接口的DLL程序。该程序的核心是一个验证用户账号和密码，通过验证后返回一个令牌。令牌是一个特殊的字符串，可用于访问Wyn的报表和仪表板等内容。

2. 操作步骤

编写一个自定义安全提供程序的步骤如下：

(1) 创建项目

使用Microsoft Visual Studio 2017（以下简称VS2017），创建一个新的项目，类型选为Visual C# - .NET Standard - 类库(.NET Standard)，输入项目名称，如：MySecurityProvider：



(2) 添加程序包依赖

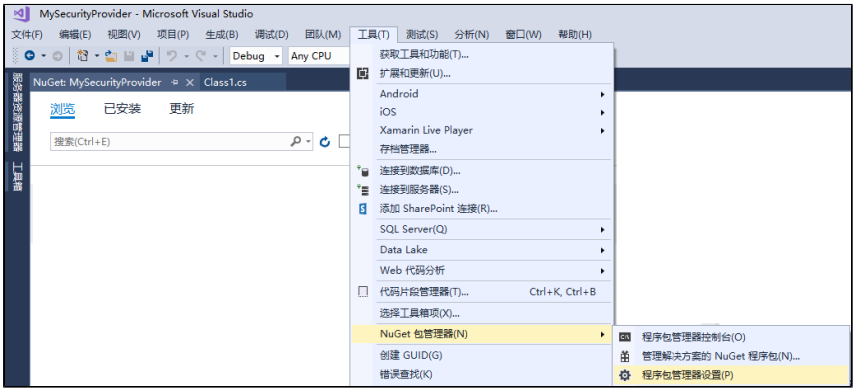
自定义安全提供程序所实现的接口是由几个程序包定义的，为此需要添加对这几个程序包的依赖。方法如下：

首先将下面这两个文件下载保存到本地硬盘，比如C:\Temp\pkg 文件夹下：

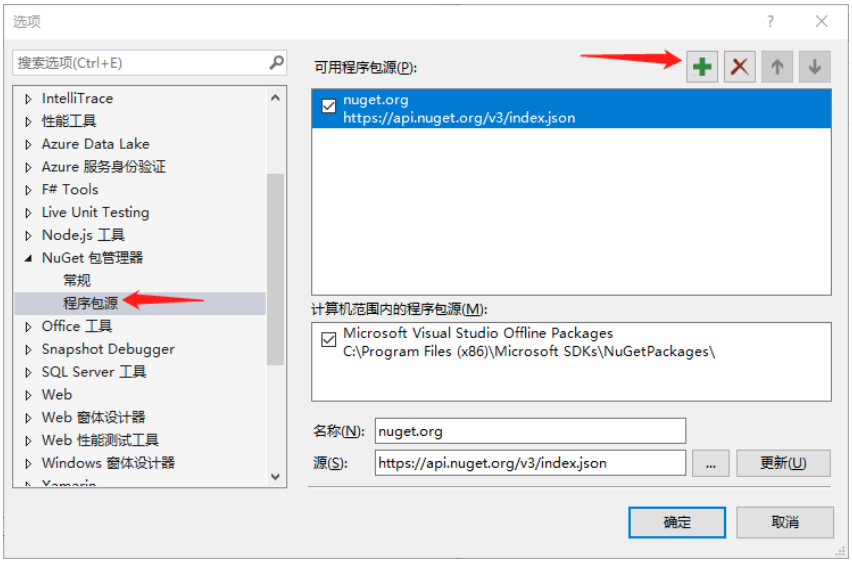
grapecity.enterprise.identity.externalidentityprovider.1.0.2.nupkg

GrapeCity.Enterprise.Identity.SecurityProvider.1.0.3.nupkg

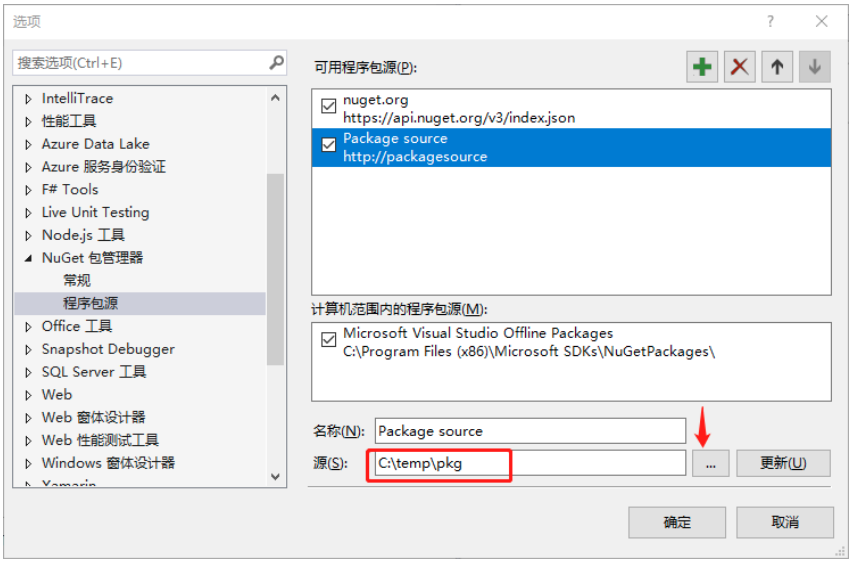
单击VS2017的“工具”菜单的“NuGet包管理器”>“程序包管理器设置”：



选中“程序包源”，再单击加号按钮：



单击【...】按钮，指定“源”的路径为nupkg文件所在的文件夹，如：C:\temp\pkg



单击“确定”按钮保存设置。

在右侧解决方案资源管理器窗格中，右键单击“依赖项”，点击“管理NuGet程序包”，再点击“浏览”，选中新添加的程序包源，将会列出两个需要依赖的程序包：

GrapeCity.Enterprise.Identity.ExternalIdentityProvider和GrapeCity.Enterprise.Identity.SecurityProvider，如下图：

逐个选中程序包，点击“安装”，即可添加本项目对这两个程序包的依赖。

(3) 实现接口

自定义安全提供程序需要实现两个接口：ISecurityProviderFactory和ISecurityProvider。

实现第一个接口的操作步骤：

添加一个新的类文件，如MySecurityProviderFactory.cs，以实现ISecurityProviderFactory接口。

public class MySecurityProviderFactory: ISecurityProviderFactory

该接口规定了两个属性和一个方法：

public string Description // 本安全提供程序的描述字串。

public IEnumerable<ConfigurationItem> SupportedSettings // 本安全提供程序支持的用户配置项。

这些用户配置项将出现在Wyn的管理画面中，允许系统管理员进行设置。典型的配置项是用户信息数据库的连接字串。通过提供这种配置项目，可以避免在安全提供程序中硬编码用户信息数据库连接字串的问题。

public Task<ISecurityProvider> CreateAsync(IEnumerable<ConfigurationItem> settings) // 本安全提供程序的实例创建方法。

这个方法的内容几乎是固定的，如：

```
public Task<ISecurityProvider>
CreateAsync(IEnumerable<ConfigurationItem>
settings)
{
 return
Task.FromResult<ISecurityProvider>(new
MySecurityProvider(settings));
}
```

实现ISecurityProvider接口

这个ISecurityProvider接口是安全提供程序的核心，其规定的属性和方法如下：

成员类型	名称	说明
属性	ProviderName	返回本安全提供程序的名称。
方法	GenerateTokenAsync	验证用户名和密码，通过时生成Wyn的访问令牌。
方法	GetUserContextAsync	返回用户的上下文信息，一般是根据用户名，从数据库查询得到用户的所属部门和其他业务数据。

方法	GetUserDescriptorAsync	返回用户的说明信息，该信息将用于门户页面的当前登录用户显示。
方法	GetUserOrganizationsAsync	返回用户的所属组织结构信息。
方法	GetUserRolesAsync	返回用户的角色信息，多个角色以字符串数组的形式返回。
方法	ValidateTokenAsync	验证令牌的合法性。在业务系统集成中，使用Token直接访问系统时，此方法用于检查传入Token的正确性。

除了上表所列成员，还有IExternalUserDescriptor，IExternalUserContext等接口，这些接口只是规定了实体类的属性，使用自定义类实现这些接口即可。

下面的文件附件是一个自定义安全提供程序的示例代码。

MySecurityProvider.zip

此示例代码中的解决方案（.sln）可在Visual Studio 2017中直接打开。示例代码文件夹\bin\debug中也包含Build产物DLL，可直接配置为Wyn的安全提供程序。示例的用户信息是保存在SQL Server数据库中的，请将本文件包中的db\MyUsers.bak文件恢复为SQL Server数据库。

有关接口的详细说明，请参考下面的接口介绍。

3. 接口介绍

ISecurityProviderFactory接口

定义

```
public interface ISecurityProviderFactory
{
 string ProviderName { get; }
 string Description { get; }
 IEnumerable<ConfigurationItem>
SupportedSettings { get; }
 Task<ISecurityProvider>
CreateAsync(IEnumerable<ConfigurationItem>
settings);
}
```

接口说明

属性和方法	说明
ProviderName	安全提供程序的名称，不能为空，不能和其它的安全提供程序重名。
Description	安全提供程序的描述文本，可以为空。
SupportedSettings	该安全提供程序加载和运行时所必须的配置项。比如安全提供程序需要访问数据库，那么数据库连接字符串即为一个必须的配置项，必须由管理员在安全提供程序管理页面配置好，该安全提供程序才能正常工作。可以没有任何必须的配置项，返回一个空列表即可。

CreateAsync	创建一个安全提供程序的实例。参数settings即为管理员已经配置好的配置项列表，用户可以在这里把配置项列表通过构造函数传入构建的安全提供程序实例。

ISecurityProvider接口

定义

```
public interface ISecurityProvider
{
 string ProviderName { get; }
 Task DisposeTokenAsync(string token);
 Task<string> GenerateTokenAsync(string
username, string password, object
customizedParam = null);
 Task<IExternalUserContext>
GetUserContextAsync(string token);
 Task<IExternalUserDescriptor>
GetUserDescriptorAsync(string token);
 Task<string[]>
GetUserOrganizationsAsync(string token);
 Task<string[]> GetUserRolesAsync(string
token);
 Task<bool> ValidateTokenAsync(string token);
}
```

接口说明

属性和方法	说明
ProviderName	安全提供程序的名称，不能为空，不能和其它的安全提供程序重名。
DesposeTokenAsync	使给定的token失效。
GenerateTokenAsync	判断给定的用户名和密码是否有效，如果有效，返回一个唯一的token；否则返回null或空字符串。注：该token可以是任何形式，比如用户的id，或这个用户信息加密后的字符串，只要确保安全提供程序可以根据这个token正确地返回这个用户的相关信息即可。
GetUserContextAsync	使用给定的token获取用户的上下文信息。用户的上下文信息包含哪些内容可以是随意的。
GetUserDescriptor	使用给定的token获取用户的基本信息。基本信息包括用户的id，用户名和安全提供程序的名称，都不能为空。
GetUserOrganizationsAsync	使用给定的token获取用户所属的部门信息。（该接口暂时没有使用）。
GetUserRolesAsync	使用给定的token获取用户的角色信息。返回用户所属角色的名称，这些角色的名称需要跟admin portal中列出的角色名完全匹配，否则会被忽略。
ValidateTokenAsync	验证给定的token是否是该安全提出程序提供的一个合法有效的token。

IExternalUserDescriptor接口

定义

```
public interface IExternalUserDescriptor
{
 string ExternalUserId { get; }
 string ExternalUserName { get; }
 string ExternalProvider { get; }
}
```

接口说明

参数	说明
ExternalUserId	用户的唯一标识符。
ExternalUserName	用户名。
ExternalProvider	用户的提供者，即为安全提供程序的名称。

IExternalUserContext接口

定义

```
public interface IExternalUserContext
{
 IEnumerable<string> Keys { get; }
 Task<string> GetValueAsync(string key);
 Task<IEnumerable<string>>
GetValuesAsync(string key);
}
```

接口说明

参数	说明
Keys	用户上下文信息所包含的项目。
GetValueAsync	对于给定的key，获取其对应的用户信息。
GetValuesAsync	对于给定的key，获取其对应的用户信息，适用于多值情况。

注意

- 在每个接口的实现函数中，必须有try-catch异常处理，在catch的异常处理部分，不要用throw语句再次抛出异常，而应返回Task对象，例如：return Task.FromResult<T>(null)；其中T为接口函数规定的某个类型。
- 用户上下文的key不要用以下字符串：sub，name，auth_time，idp，userid，email。

编写好的安全提供程序，通过构建，得到DLL文件。然后就可以配置到Wyn系统中了，具体步骤参见：

配置自定义安全提供程序