

# 第30期-功能特征模块

2020年9月28日 11:20

ABP 中的特征系统用于在运行时启用、禁用、更改应用程序功能的行为，尤其是多租户多用户的场景下，对系统运行的行为可能有所不同。

## 功能模块包

Install-Package Volo.Abp.Features

[DependsOn(typeof(AbpFeaturesModule))]

## 定义功能

```
public class MyFeatureDefinitionProvider : FeatureDefinitionProvider
{
    public override void Define(IFeatureDefinitionContext context)
    {
        var myGroup = context.AddGroup("MyApp");

        myGroup.AddFeature("MyApp.PdfReporting", defaultValue: "false");
        myGroup.AddFeature("MyApp.MaxProductCount", defaultValue: "10");
    }
}
```

DisplayName: 可本地化的字符串，将用于在用户界面上显示功能名称。

Description: 较长的可本地化文字，用于描述功能。

ValueType: 值的类型。可以是实现的类IStringValueType。内置类型:

ToggleStringValueType: 用于定义true/false, on/off, enabled/disabled样式特征。UI上显示一个复选框。

FreeTextStringValueType: 用于定义自由文本值。UI上显示一个文本框。

SelectionStringValueType: 用于强制从列表中选择值。UI上显示一个下拉列表。

IsVisibleToClients (默认true值: ): 设置为false可向客户端 (浏览器) 隐藏此功能的值。与客户共享价值可帮助他们根据功能价值有条件地显示/隐藏/更改UI部件。

Properties: 字典，用于设置/获取与此功能相关的任意键值对。这可能是定制的重点。

```
public override void Define(IFeatureDefinitionContext context)
{
    var myGroup = context.AddGroup("MyApp");

    myGroup.AddFeature(
        "MyApp.PdfReporting",
```

```

        defaultValue: "false",
        displayName: LocalizableString.Create<TestResource>
("PdfReporting"),
        valueType: new ToggleStringValue()
    );

    myGroup.AddFeature(
        "MyApp.MaxProductCount",
        defaultValue: "10",
        displayName: LocalizableString.Create<TestResource>
("MaxProductCount"),
        valueType: new FreeTextStringValue(new NumericValueValidator(0,
100))
    );
}

```

定义子功能：当某个功能打开后，才能进行其它功能的设置，前置和依赖项。

```

public override void Define(IFeatureDefinitionContext context)
{
    var myGroup = context.AddGroup("MyApp");

    var reportingFeature = myGroup.AddFeature(
        "MyApp.Reporting",
        defaultValue: "false",
        displayName: LocalizableString.Create<TestResource>("Reporting"),
        valueType: new ToggleStringValue()
    );

    reportingFeature.CreateChild(
        "MyApp.PdfReporting",
        defaultValue: "false",
        displayName: LocalizableString.Create<TestResource>
("PdfReporting"),
        valueType: new ToggleStringValue()
    );

    reportingFeature.CreateChild(
        "MyApp.ExcelReporting",
        defaultValue: "false",
        displayName: LocalizableString.Create<TestResource>
("ExcelReporting"),
        valueType: new ToggleStringValue()
    );
}

```

## 更改现有模块的功能定义

```

var someGroup = context.GetGroupOrNull("SomeModule");
var feature = someGroup.Features.FirstOrDefault(f => f.Name == "SomeFeature");

```

```

if (feature != null)
{
    feature.Description = ...
    feature.CreateChild(...);
}

```

## 检查是否支持某功能

### 使用 [RequiresFeature] 标记

```

public class ReportingAppService : ApplicationService, IReportingAppService
{
    [RequiresFeature("MyApp.PdfReporting")]
    [RequiresFeature("Feature1", "Feature2")]
    [RequiresFeature("Feature1", "Feature2", RequiresAll = true)]
    public async Task<PdfReportResultDto> GetPdfReportAsync()
    {
        //TODO...
    }
}

```

[RequiresFeature] 使用拦截器实现，可标记类和方法，需要类实现接口，否则需要虚方法。

### 使用 IFeatureChecker 接口

```

bool result = await _featureChecker.IsEnabledAsync("MyApp.PdfReporting");

string maxProductCountLimit = await
    _featureChecker.GetOrNullAsync("MyApp.MaxProductCount");

int maxProductCountLimit = await _featureChecker.GetAsync<int>
    ("MyApp.MaxProductCount");

```

**ApplicationService 和 AbpController 基类已经注入 IFeatureChecker 实例。**

### 其它扩展方法

默认值: Task<T> GetAsync<T>(string name, T defaultValue = default)

抛异常: CheckEnabledAsync(string name)

<https://localhost:5001/api/app/reporting>

## 在客户端检查功能

需要设置 `IsVisibleToClients` 为 `True` 值。

```
var isEnabled = abp.features.values["MyApp.ExcelReporting"] === "true";
var count = abp.features.values["MyApp.MaxProductCount"];
```

## 功能值提供程序

任何派生自 `FeatureValueProvider`（或实现 `IFeatureValueProvider`）的类都可以对功能系统做出贡献。值提供者负责获取给定功能的当前值。

有三个预定义的值提供程序，它们按给定的顺序执行：

- `TenantFeatureValueProvider` 尝试获取是否为当前租户显式设置了功能值。
- `EditionFeatureValueProvider` 尝试获取当前版本的功能值。版本ID是从当前的主体标识（`ICurrentPrincipalAccessor`）和声明名称 `editionid`（定义为的常数 `AbpClaimTypes.EditionId`）获得的。租户管理模块未实现版本。您可以自己实施它，也可以考虑使用 ABP Commercial 的 SaaS 模块。
- `DefaultValueFeatureValueProvider` 获取功能的默认值。

```
public class SystemAdminFeatureValueProvider : FeatureValueProvider
{
    public override string Name => "SA";

    private readonly ICurrentPrincipalAccessor _currentPrincipalAccessor;

    public SystemAdminFeatureValueProvider(
        IFeatureStore featureStore,
        ICurrentPrincipalAccessor currentPrincipalAccessor)
        : base(featureStore)
    {
        _currentPrincipalAccessor = currentPrincipalAccessor;
    }

    public override Task<string> GetOrNullAsync(FeatureDefinition feature)
    {
        if (feature.ValueType is ToggleStringValue &&
            _currentPrincipalAccessor.Principal?.FindFirst("User_Type")
?.Value == "SystemAdmin")
        {
            return Task.FromResult("true");
        }

        return null;
    }
}
```

```
Configure<AbpFeatureOptions>(options =>
{
    options.ValueProviders.Add<SystemAdminFeatureValueProvider>();
});
```

## 实现原理分析

IFeatureDefinitionManager->\*IFeatureDefinitionProvider->\*FeatureGroupDefinition+\*FeatureDefinition

FeatureInterceptorRegistrar->FeatureInterceptor->  
IMethodInvocationFeatureCheckerService->  
IFeatureChecker->\*IFeatureValueProvider+IFeatureDefinitionManager

IFeatureValueProvide->IFeatureStore:NullFeatureStore