**功能类型**

有两个基本的功能类型。

**Boolean 功能**

可以是“true”或“false”，一个这种类型的功能可以是启用或禁用（为一个版本或一个租户）。

**Value 功能**

可以是任意值，它保存和获取一个字符串，数字也保存成字符串。

**定义功能**

在检查功能前，先要定义它，一个[模块](http://www.cnblogs.com/kid1412/p/5973274.html" \t "http://www.cnblogs.com/kid1412/p/_blank)可通过继承FeatureProvider类来定义自己的功能，此处，有一个非常简单的功能供应器定义了3个功能：

public class AppFeatureProvider : **FeatureProvider**

{

public override void SetFeatures(IFeatureDefinitionContext context)

{

var sampleBooleanFeature = context.**Create**("SampleBooleanFeature", defaultValue: "false");

sampleBooleanFeature.**CreateChildFeature**("SampleNumericFeature", defaultValue: "10");

context.**Create**("SampleSelectionFeature", defaultValue: "B");

}

}

在创建一个功能供应器之后，我们应当在我们模块的[PreInitialize](http://www.cnblogs.com/kid1412/p/5973274.html" \l "PreInitialize" \t "http://www.cnblogs.com/kid1412/p/_blank)方法里注册它，如下所示：

**Configuration.Features**.Providers.Add<AppFeatureProvider>();

**基本功能属性**

一个功能定义要求至少两个属性：

Name：一个唯一名称（字符串），这个功能的标志。

DefaultValue：一个默认值，当我们需要这个功能的值而又不能从当前租户取得时，我们需要一个默认值。

**其它功能属性**

虽然唯一名称和默认值属性是必须的，但也有一可选的属性，提供细节控制。

* Scope：一个FeatureScopes枚举值，它可以是Edition（如果这个功能只能设置版本级别），Tenant（如果这个功能只能设置租户级别）或All（如果这个功能可设置版本和租户，租户设置会覆盖版本设置）。默认值是All。
* DisplayName：一个本地化的字符串，为用户显示这个功能的名称。
* Description：一个本地化的字符串，为客户显示这个功能的细节描述。
* InputType：这个功能的一个UI输入类型，这个可被定义，当创建一个自动功能屏幕时可以使用它。
* Attribute：一个键值对的用户字典，关联这个功能。

public class AppFeatureProvider : **FeatureProvider**

{

public override void SetFeatures(IFeatureDefinitionContext context)

{

var sampleBooleanFeature = context.Create(

AppFeatures.SampleBooleanFeature,

defaultValue: "false",

displayName: L("Sample boolean feature"),

inputType: new CheckboxInputType()

);

sampleBooleanFeature.CreateChildFeature(

AppFeatures.SampleNumericFeature,

defaultValue: "10",

displayName: L("Sample numeric feature"),

inputType: new SingleLineStringInputType(new NumericValueValidator(1, 1000000))

);

context.Create(

AppFeatures.SampleSelectionFeature,

defaultValue: "B",

displayName: L("Sample selection feature"),

inputType: new ComboboxInputType(

new StaticLocalizableComboboxItemSource(

new LocalizableComboboxItem("A", L("Selection A")),

new LocalizableComboboxItem("B", L("Selection B")),

new LocalizableComboboxItem("C", L("Selection C"))

)

)

);

}

private static ILocalizableString L(string name)

{

return new LocalizableString(name, AbpZeroTemplateConsts.LocalizationSourceName);

}

}

让我们看一下上面那个功能更多的细节定义：**功能层次**

如上面所示的示例功能供应器，一个功能可以有子功能。一个父功能通常定义为Boolean功能，只有在父功能可用时，才能获取子功能。ABP不强制但建议这么做，应用应当小心处理它。

**检查功能**

我们定义一个功能来检查它的在应用里的值，从而为每个租户允许或阻止一些应用功能。有几种不同的检查方式。

**使用RequiresFeature特性**

我们可以为一个方法或类使用RequiredFeature，如下所示：

**[RequiresFeature("ExportToExcel")]**

public async Task<FileDto> GetReportToExcel(...)

{

...

}

只有当前租户（从[IAbpSession](http://www.cnblogs.com/kid1412/p/5986611.html" \t "http://www.cnblogs.com/kid1412/p/_blank)里获取）的“ExportToExcel”功能可用时，才能执行这个方法，如果不可用，就自动抛出一个AbpAuthorizationException。

当然，RequiresFeature特性应该用在Boolean类型的功能上，否则，你会收到异常。

**使用 IFeatureChecker**

我们可以注入IFeatureChecker，并使用它手动检查一个功能（它被自动注入到应用服务，Mvc和Web Api控制器，并被自动使用）。

**IsEnabled**

简单的检查一个给定的功能是否可用，如：

public async Task<FileDto> GetReportToExcel(...)

{

if (**await FeatureChecker.IsEnabledAsync("ExportToExcel")**)

{

throw new AbpAuthorizationException("You don't have this feature: ExportToExcel");

}

...

}

**GetValue**

获取一个值类型功能的当前值，例如：

var createdTaskCountInThisMonth = GetCreatedTaskCountInThisMonth();if (createdTaskCountInThisMonth >= **FeatureChecker.GetValue("MaxTaskCreationLimitPerMonth").To<int>())**

{

throw new AbpAuthorizationException("You exceed task creation limit for this month, sorry :(");

}

**客户端**

在客户端（Javascript），我们可以使用abp.features命名空间来获取功能的当前值。

**isEnabled**

var isEnabled = abp.features.isEnabled('SampleBooleanFeature');

**getValue**

var value = abp.features.getValue('SampleNumericFeature');