Interceptory

**Interceptor je neco jako trigger, ktery se spusti pri nejake udalosti. Napr IdbOperationInterceptor ma mimo jine tyto metody:**

namespace Mediaresearch.Framework.DataAccess.BLToolkit.Interceptors

{

public interface IDbOperationInterceptor

{

int Order { get; }

void BeforeInsert(object obj);

void AfterInsert(object obj);

void BeforeUpdate(object obj);

void AfterUpdate(object obj);

**V Kodovadle jsem presouval interceptory z configu do installeru. V Pricing.configu vypadala puvodni registrace takto:**

<component id="focusUsePriceOnPricingServiceCalculationMethodAspect"

service="Castle.DynamicProxy.IInterceptor, Castle.Core"

type="MIR.Media.Coding.Core.WorkFlow.Aspects.Focus.SetFocusAspect, MIR.Media.Coding.Core">

<parameters>

<elementPath>UsePricingService</elementPath>

</parameters>

</component>

<component id="calculatePricepricingServiceCalculationMethodInterceptingMapping"

type="MIR.Media.Coding.Core.WorkFlow.Aspects.MethodNameRegexToInterceptorsMapping, MIR.Media.Coding.Core">

<parameters>

<MethodNameRegex>CalculatePrice</MethodNameRegex>

<Interceptors>

<list>

<item>${focusUsePriceOnPricingServiceCalculationMethodAspect}</item>

</list>

</Interceptors>

</parameters>

</component>

<component id="pricingServiceCalculationMethodInterceptorSelector"

service="Castle.DynamicProxy.IInterceptorSelector, Castle.Core"

type="MIR.Media.Coding.Core.WorkFlow.Aspects.ByMethodNameRegexInterceptorSelector, MIR.Media.Coding.Core">

<parameters>

<mappings>

<array>

<item>${calculatePricepricingServiceCalculationMethodInterceptingMapping}</item>

</array>

</mappings>

</parameters>

</component>

<component id="pricingServiceCalculationMethod"

service="MIR.Media.Coding.Core.Utilities.Dialogs.PriceCalculationDialog.IPriceCalculationMethod, MIR.Media.Coding.Core"

type="MIR.Media.Coding.Core.Utilities.Dialogs.PriceCalculationDialog.PricingService.PricingServiceMethodViewModel, MIR.Media.Coding.Core"

lifestyle="transient">

<parameters>

<MessagingService>${messagingServicePricingDialog}</MessagingService>

<priceListsIdHolder>${priceListsIdHolder}</priceListsIdHolder>

</parameters>

<interceptors selector="${pricingServiceCalculationMethodInterceptorSelector}">

<interceptor>${focusUsePriceOnPricingServiceCalculationMethodAspect}</interceptor>

</interceptors>

</component>

**To same presunute do PricingInstalleru pak takto:**

container.Register(

Component.For<IInterceptor>().ImplementedBy<SetFocusAspect>().Named(PricingComponents.FocusUsePriceOnPricingServiceCalculationMethodAspect)

.DependsOn(

Dependency.OnValue(typeof(string), "UsePricingService")

));

container.Register(Component.For<MethodNameRegexToInterceptorsMapping>().Named(PricingComponents.CalculatePricepricingServiceCalculationMethodInterceptingMapping)

.DependsOn(

Property.ForKey("MethodNameRegex").Eq("CalculatePrice"),

Dependency.OnComponentCollection(typeof(List<IInterceptor>), PricingComponents.FocusUsePriceOnPricingServiceCalculationMethodAspect)

));

container.Register(Component.For<IInterceptorSelector>().ImplementedBy<ByMethodNameRegexInterceptorSelector>().Named(PricingComponents.PricingServiceCalculationMethodInterceptorSelector)

.DependsOn(

Dependency.OnComponentCollection(typeof(IEnumerable<MethodNameRegexToInterceptorsMapping>), PricingComponents.CalculatePricepricingServiceCalculationMethodInterceptingMapping)

));

container.Register(Component.For<IPriceCalculationMethod>().ImplementedBy<PricingServiceMethodViewModel>().Named(PricingComponents.PricingServiceCalculationMethod).LifestyleTransient()

.Interceptors(

InterceptorReference.ForKey(PricingComponents.FocusUsePriceOnPricingServiceCalculationMethodAspect)

)

.SelectedWith(container.Resolve<IInterceptorSelector>(PricingComponents.PricingServiceCalculationMethodInterceptorSelector))

.Anywhere);

public static class PricingComponents

{

public const string FocusUsePriceOnPricingServiceCalculationMethodAspect = "focusUsePriceOnPricingServiceCalculationMethodAspect";

public const string PricingServiceCalculationMethodInterceptorSelector = "pricingServiceCalculationMethodInterceptorSelector";

public const string PricingServiceCalculationMethod = "pricingServiceCalculationMethod";

public const string ManualCalculationMethod = "manualCalculationMethod";

public const string CalculatePricepricingServiceCalculationMethodInterceptingMapping = "calculatePricepricingServiceCalculationMethodInterceptingMapping";

}