DatabaseConfiguration, ConfigBase, Sections, Elements,

<https://msdn.microsoft.com/en-us/library/2tw134k3.aspx>

# Rozhovor s Filipem a zaklady vytvoreni sekci v AppConfigu

[‎05.‎09.‎2018 10:01]  Peter Hlavenka:

Ahoj, zacal jsem na kodovadle. Chtel jsem zacit presunem komponent z ContainerConfigu do CommonInstalleru.  V tomto configu musi zustat definovany ExtensionInstaller, abychom rozlisili prostredi, dal bych tu nechal bezne zalezitosti jako connnectionStringHolder a connectionStrings. Nejjednodussi mi prislo zacit s presunem tridy Fraction, ktera ma jen dve property. Pokud to dobre chapu, chceme presunout komponenty, ale jejich property nechat v configu. Kdyz chci setnout propertu v installeru z xml-ka, musim ji mit na nejakem jinem objektu typicky Properties. V Installeru par reknu neco jako : Property.ForKey("colorDictionary").Eq(container.Resolve<Properties>().PlausibilityColors).  Rikam to dobre, nebo chceme odstranit z configu i Properties ?

[‎05.‎09.‎2018 10:02]  Filip Čálek:

postupne chceme z container configu odstranit vsechno. pravdepodobne

property chceme mit v App.configu. ten bude pro kazdou verzi jiny viz environments

co ma presah do jinych softu, tak patri do params

[‎05.‎09.‎2018 10:03]  Filip Čálek:

a toto  Property.ForKey("colorDictionary").Eq(container.Resolve<Properties>().PlausibilityColors) nechci nikde videt

[‎05.‎09.‎2018 10:03]  Filip Čálek:

v installeru nema byt zadnej resolve

jenom registrace

hele neboj se toho. proste zacni a ja ti to budu postupne revidovat

[‎05.‎09.‎2018 10:05]  Peter Hlavenka:

kde je definovane, ze se ma nejake prostredi podivat do AppConfigu ? Ten je jen jeden, takze jak se rozlisi prostredi, kdyz nebudu mit container.config?

[‎05.‎09.‎2018 10:05]  Filip Čálek:

stejne jako se resi container.config. pri deployi se nakopiruje spravnej config do binu

[‎05.‎09.‎2018 10:07]  Peter Hlavenka:

jak teda zaregistruju tridu ktera ma dve property definovane v AppConfigu aniz bych si je vytahl z nejakeho objektu ? Musim je pri registraci asi rovnou nainicializovat ..

[‎05.‎09.‎2018 10:09]  Peter Hlavenka:

appConfigu teda bude taky vic a budou v kazdem prostredi ?

[‎05.‎09.‎2018 10:10]  Filip Čálek:

dobre jsou uricte vyjimky, jako treba provider na appConfig

v mr frameworku mame IConfigurationProvider

pouziti pak vypada takto configProvider.GetConfig<EmailConfiguration>()

[‎05.‎09.‎2018 10:11]  Peter Hlavenka:

to uz zni lepe

[‎05.‎09.‎2018 10:11]  Filip Čálek:

bude jich vic. ano

[‎05.‎09.‎2018 10:15]  Filip Čálek:

az budes resit connection string, tak rekni. na to taky uz neco mame

ale to si nechame na potom

tam bude potreba udelat nejaky zmeny

[‎05.‎09.‎2018 10:19]  Peter Hlavenka:

kdyz chci appConfig dostat od IConfigurationProvidera, musim k appConfigu vytvorit .Net tridu podobne jako se to dela s Properties, ktere bude dedit od ConfigBase ?

[‎05.‎09.‎2018 10:22]  Filip Čálek:

pro jednotlivy sekce v app configu. ano

https://msdn.microsoft.com/en-us/library/2tw134k3.aspx

[‎05.‎09.‎2018 10:23]  Filip Čálek:

dedi z configBase. presne tak

jenom nechci jednu super tridu pro vsechno, ale mit to rozdeleny do sekci. DatabaseConfiguration, DeploymenConfiguration, EmailConfiguration atd

[‎05.‎09.‎2018 10:24]  Peter Hlavenka:

j pokusim se ..

* Pokud chci definovat v sekci tridu, musi to byt ConfigurationElement public class ColorWrapper : ConfigurationElement (musi byt uvnitr jineho elementu -> napr. CodingConfiguration)
* Pokud chci definovat jen propertu, musi by uvnitr nejakeho ConfigurationElementu. MinimalNumerator je uvnitr CodingConfiguration : ConfigBase : ConfigurationSection : ConfigurationElement
* Pokud chci definovat kolekci, musi to byt ConfigurationCollection uvnitr elementu.

1. **Vytvorim tridu CodingConfiguration**

public class CodingConfiguration : ConfigBase

{

//codeReview: uplne zbytecny radek

public static CodingConfiguration CodingConfigurationSettings => ConfigurationManager.GetSection("CodingSection") as CodingConfiguration;

public override string GetConfigName()

{

return nameof(CodingConfiguration);

}

[ConfigurationProperty(nameof(MinimalFractionNumerator), IsRequired = true)]

public int MinimalFractionNumerator => (int)this["MinimalFractionNumerator"];

[ConfigurationProperty(nameof(MinimalFractionDenominator), IsRequired = true)]

public int MinimalFractionDenominator => (int)this["MinimalFractionDenominator"];

1. **V AppConfigu pridam configSections a definuju svoji sekci**

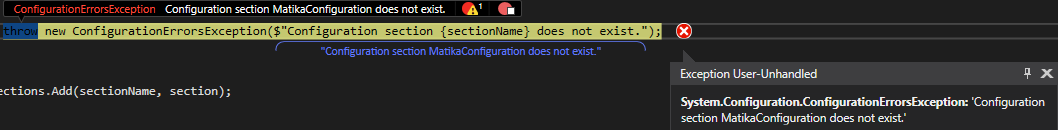
<configuration>

<configSections>

<section name="CodingSection" type="MIR.Media.Coding.Configurations.CodingConfiguration, MIR.Media.Coding" />

</configSections>

Mozny problem:



Reseni: V app.configu se musi sekce jmenovat tak jak je definovano v metode public override string GetConfigName() Napr.:

private readonly string m\_name = "MatikaSection";

public override string GetConfigName() => m\_name;

1. **V AppConfigu otevru svoji sekci a pridam hodnoty pro property**

<CodingConfig MinimalFractionNumerator="1" MinimalFractionDenominator="256"/>

1. **Zaregistruju komponentu a IConfigurationProvidera v installeru.**

**Abych mohl v Installeru pouzivat IConfigurationProvidera, resolvnu si ho tady do property. (V Installeru by zadne resolvy byt nemely ale tohle je vyjimka, protoze trida Fraction je v Coding.Core ale moje CodingConfiguration je v assembly Coding, kam Core nevidi)**

using Mediaresearch.Framework.Utilities.Configuration;

container.Register(Component.For<IConfigurationProvider>().ImplementedBy<ConfigurationProvider>().LifestyleSingleton());

var configurationProvider = container.Resolve<IConfigurationProvider>();

Trida Fraction ma konstruktor ve kterem chce dve property numerator a denominator.

container.Register(Component.For<Fraction>()

.DependsOn(

Property.ForKey("numerator").Eq(configurationProvider.GetConfig<CodingConfiguration>().MinimalFractionNumerator),

Property.ForKey("denominator").Eq(configurationProvider.GetConfig<CodingConfiguration>().MinimalFractionDenominator)

));

1. **K properte se pak dostanu takto: (zde resolvnuto pres container.Current)**

var configurationProvider = Core.Container.Container.Current.Resolve<IConfigurationProvider>();

var **testProperty** = **configurationProvider.GetConfig<CodingConfiguration>().TestProperty;**

Druha moznost je si resolvnout tridu Fraction pomoci Castlu v konstruktoru, nebo public properte

Tim, ze mame statickou propertu muzeme rict (cutting): (ale jen v assembly ktera na tridu vidi)

TimeSpan.FromMinutes(CuttingConfiguration.CuttingConfigurationSettings.UpdateShutDownTimeOutInMinutes)

# ConfigurationElementCollection + jak naplnit z AppConfigu string[] => AresConfiguration

<https://www.codeproject.com/Tips/1037705/Make-Your-ConfigurationElementCollection-Type-Expl>

* **Pridame propertu do tridy podedene od ConfigBase (ConfigurationSection)**:

public class AresConfiguration : ConfigBase //(ConfigurationSection)

{

[ConfigurationProperty(nameof(SuffixesToRemove), IsDefaultCollection = false)]

[ConfigurationCollection(typeof(SuffixiesToRemoveCollection))]

public SuffixiesToRemoveCollection SuffixesToRemove => (SuffixiesToRemoveCollection) base["SuffixesToRemove"];

public override string GetConfigName()

{

return nameof(AresConfiguration);

}

}

* **Musim mit Elementy, ktere budou v kolekci**

public class SuffixToRemove : ConfigString //(ConfigurationElement)

{

[ConfigurationProperty(nameof(Name), IsRequired = true)]

public new string Name => (string) this["Name"];

}

* **Musim mit kolekci do ktere dam elementy**

public class SuffixiesToRemoveCollection : ConfigStringCollection //(ConfigurationElementCollection)

{

protected override ConfigurationElement CreateNewElement()

{

return new SuffixToRemove();

}

protected override object GetElementKey(ConfigurationElement element)

{

return ((SuffixToRemove) element).Name;

}

}

* **Musim mit tridu ve ktere bude properta, typu moji kolekce, ktera se bude inicializovat hodnotami z configu**

public class MyClass

{

private MyCollection Neco { get; set; }

public MyClass(MyCollection neco)

{

Neco = neco;

}

}

* **Pokud tuto MyClass budu chtit od Castlu, musim ji zaregistrovat**

container.Register(Component.For<MyClass>()

.DependsOn(

Property.ForKey("neco").Eq(configurationProvider.GetConfig<AresConfiguration>().SuffixesToRemove.**ToArray()**)));

* **V** **AppConfigu udelame section a v inicializaci pridame property do kolekce**

<configSections>

<section name="AresSection" type="MIR.Media.Coding.Configurations.AresConfiguration, MIR.Media.Coding"/>

</configSections>

<AresSection>

<SuffixesToRemove>

<add Name="ahoj"/>

<add Name="cao"/>

</SuffixesToRemove>

</AresSection>

## Jak naplnit string[]

Prakticky priklad je Kodovadlo, AresSearcher

Postup je skoro stejny jako je popsan nahore, s tim rozdilem, ze kolekci nededime od ConfigurationElementCollection ale od ConfigStringCollection, a Element se nededi od ConfigurationElement, ale od ConfigString.

Obe tyto tridy jsou z Frameworku. A dedi od puvodnich trid.

public class MyCollection : ConfigStringCollection

public class MyElement : ConfigString

**ConfigString:**

public class ConfigString : ConfigurationElement

{

[ConfigurationProperty("Name", IsKey = true, IsRequired = true)]

public string Name

{

get { return (string) base["Name"]; }

set

{

base["Name"] = value;

}

}

}

**ConfigStringCollection**

public class ConfigStringCollection : ConfigurationElementCollection

{

protected override ConfigurationElement CreateNewElement()

{

return new ConfigString();

}

protected override object GetElementKey(ConfigurationElement element)

{

return ((ConfigString)element).Name;

}

public string[] **ToArray()**

{

return this.Cast<ConfigString>().Select(x => x.Name).ToArray();

}

public List<string> **ToList()**

{

return this.Cast<ConfigString>().Select(x => x.Name).ToList();

}

}

**Registrace tridy do containeru se zavislosti na pole stringu:**

container.Register(Component.For<AresSearcher>()

.DependsOn(

Property.ForKey("SuffixesToRemove").Eq(configurationProvider.GetConfig<AresConfiguration>().SuffixesToRemove.ToArray())).LifestyleSingleton());

# Vice string[] v ramci jedne sekce => EntityDaoFactoryConfiguration

using System.Configuration;

using Mediaresearch.Framework.Utilities.Configuration;

using Mediaresearch.SimAdmin.Shared.Configuration;

namespace MIR.Media.Coding.Configurations

{

public class EntityDaoFactoryConfiguration : ConfigBase

{

public override string GetConfigName()

{

return nameof(EntityDaoFactoryConfiguration);

}

[ConfigurationProperty(nameof(MediaDataDaoAssemblies), IsDefaultCollection = false)]

[ConfigurationCollection(typeof(DaoAssemblyCollection),

AddItemName = "add",

ClearItemsName = "clear",

RemoveItemName = "remove")]

public DaoAssemblyCollection MediaDataDaoAssemblies => (DaoAssemblyCollection)base["MediaDataDaoAssemblies"];

[ConfigurationProperty(nameof(PrintStorageDaoAssemblies), IsDefaultCollection = false)]

[ConfigurationCollection(typeof(DaoAssemblyCollection),

AddItemName = "add",

ClearItemsName = "clear",

RemoveItemName = "remove")]

public DaoAssemblyCollection PrintStorageDaoAssemblies => (DaoAssemblyCollection)base["PrintStorageDaoAssemblies"];

[ConfigurationProperty(nameof(EnumTableAssemblies), IsDefaultCollection = false)]

[ConfigurationCollection(typeof(DaoAssemblyCollection),

AddItemName = "add",

ClearItemsName = "clear",

RemoveItemName = "remove")]

public DaoAssemblyCollection EnumTableAssemblies => (DaoAssemblyCollection)base["EnumTableAssemblies"];

}

public class DaoAssembly : ConfigString

{

[ConfigurationProperty(nameof(Name), IsRequired = true)]

public new string Name => (string)this["Name"];

}

public class DaoAssemblyCollection : ConfigStringCollection

{

protected override ConfigurationElement CreateNewElement()

{

return new DaoAssembly();

}

protected override object GetElementKey(ConfigurationElement element)

{

return ((DaoAssembly)element).Name;

}

}

}

<configSections>

<section name="EntityDaoFactoryConfiguration" type="MIR.Media.Coding.Configurations.EntityDaoFactoryConfiguration, MIR.Media.Coding"/>

</configSections>

<EntityDaoFactoryConfiguration>

<MediaDataDaoAssemblies>

<add Name="MIR.Entities.MediaData"/>

<add Name="Mediaresearch.Framework.Security.BLToolkit"/>

</MediaDataDaoAssemblies>

<PrintStorageDaoAssemblies>

<add Name="MIR.Entities.PrintStorage"/>

<add Name="Mediaresearch.Framework.Security.BLToolkit"/>

</PrintStorageDaoAssemblies>

<EnumTableAssemblies>

<add Name="MIR.Entities.MediaData"/>

<add Name="Mediaresearch.Framework.Security.BLToolkit"/>

</EnumTableAssemblies>

</EntityDaoFactoryConfiguration>

# Presunuti vice instanci jedne tridy z ContainerConfigu do appConfigu neboli jak dostat do appConfigu cely typ (tridu)

1. **Mam v Kodovadle v containerConfigu definovane tri instance jedne tridy:**

<component id="untouchedPlausibilityColor"

type="MIR.Media.Coding.Core.Utilities.ColorWrapper, MIR.Media.Coding.Core">

<parameters>

<r>186</r>

<g>212</g>

<b>255</b>

</parameters>

</component>

<component id="surePlausibilityColor"

type="MIR.Media.Coding.Core.Utilities.ColorWrapper, MIR.Media.Coding.Core">

<parameters>

<r>201</r>

<g>224</g>

<b>211</b>

</parameters>

</component>

<component id ="otherPlausibilityColor"

type="MIR.Media.Coding.Core.Utilities.ColorWrapper, MIR.Media.Coding.Core">

<parameters>

<r>255</r>

<g>200</g>

<b>150</b>

</parameters>

</component>

1. **Vrazil bych to do uz existujici tridy CodingConfiguration, at tech sekci neni milion. Zatim tam jsou definovane jen int typy, tady ale chci definovat tridu. Trida se jmenuje ColorWrapper.**

[ConfigurationProperty(nameof(UntouchedInetSingleCreativeColor), IsRequired = true)]

public ColorWrapper UntouchedInetSingleCreativeColor => (ColorWrapper)this["UntouchedInetSingleCreativeColor"];

1. **Tridu ColorWrapper upravim tak, ze pridam configuration atributy nad property:**

public class ColorWrapper : ConfigurationElement

{

[ConfigurationProperty(nameof(Color), IsRequired = false)]

public Color Color => new Color {R = R, B = B, G = G, A = 255};

[ConfigurationProperty(nameof(R), IsRequired = true)]

private byte R => (byte) this["R"];

[ConfigurationProperty(nameof(G), IsRequired = true)]

private byte G => (byte) this["G"];

[ConfigurationProperty(nameof(B), IsRequired = true)]

private byte B => (byte) this["B"];

}

1. **V appConfigu ted musim v sekci CodingConfiguration pridat Element**

<CodingConfiguration

MinimalFractionNumerator="1"

MinimalFractionDenominator="256">

<UntouchedInetSingleCreativeColor R="209" G="238" B="168"/>

</CodingConfiguration>

1. **Nyni ze sve sekce dostanu propertu typu ColorWrapper, ktera uz bude mit setnute property R, G, B.**

var test = Core.Container.Container.Current.Resolve<IConfigurationProvider>().GetConfig<CodingConfiguration>().**UntouchedInetSingleCreativeColor**;

nebo:

container.Register(Component.For<IConfigurationProvider>().ImplementedBy<ConfigurationProvider>().LifestyleSingleton());

var configurationProvider = container.Resolve<IConfigurationProvider>();

neco.DependsOn(

Property.ForKey("untouchedInetSingleCreativeColor").Eq(configurationProvider.GetConfig<CodingConfiguration>().**UntouchedInetSingleCreativeColor**)

# Dictionary

<http://salmontech.blogspot.com/2009/05/custom-dictionary-sections-in-net.html>

Vyborny clanek

## Jak precastit HashTable na Dictionary

var hashtable = ConfigurationManager.GetSection("PlausibilityColorsConfiguration") as Hashtable;

//precastim to na dictionary

var configDictionary = hashtable?.Cast<DictionaryEntry>().ToDictionary(n => n.Key.ToString(), n => n.Value.ToString());

# Byte [] v ConfigurationSection

* Vse co je v configu je string. Kdyz chci v configurationSection stornout byte[] , budu muset Cast –it.

Jinak je vse opet klasika. Potrebuju tridu podedenou od ConfigBase (ConfigurationSection) a v ni bude ConfigurationProperty typu kolekce.

public class PropertiesConfiguration : ConfigBase

{

public override string GetConfigName()

{

return nameof(PropertiesConfiguration);

}

[ConfigurationProperty(nameof(**OnlyPrimaryRoleIds**), IsDefaultCollection = false)]

[ConfigurationCollection(typeof(**OnlyPrimaryRoleIdsCollection**))]

public OnlyPrimaryRoleIdsCollection OnlyPrimaryRoleIds => (OnlyPrimaryRoleIdsCollection)base[nameof(**OnlyPrimaryRoleIds)**];

}

* Dale potrebujeme jinou tridu, odvozenou od ConfigurationElement

public class OnlyPrimaryRoleIds : ConfigurationElement

{

[ConfigurationProperty(nameof(Name), IsRequired = true)]

public string Name => (string)this[nameof(Name)];

}

* A nakonec potrebujeme tridu odvozenou od ConfigurationElementCollection

public class OnlyPrimaryRoleIdsCollection : ConfigurationElementCollection

{

protected override ConfigurationElement CreateNewElement()

{

return new OnlyPrimaryRoleIds();

}

protected override object GetElementKey(ConfigurationElement element)

{

return ((OnlyPrimaryRoleIds)element).Name;

}

To dulezite je zde – pokud chci ze sekce dostat byte[] potrebuju metodu ktera mi prekonverti stringy

public byte[] ToArray()

{

return this.Cast<OnlyPrimaryRoleIds>().Select(d => Convert.ToByte(d.Name)).ToArray();

}

}

* AppConfig klasika, definuju sekci a jeji prvky,

<configSections>

<section name="PropertiesConfiguration" type="MIR.Media.Coding.Configurations.PropertiesConfiguration, MIR.Media.Coding"/>

</configSections>

<PropertiesConfiguration>

<OnlyPrimaryRoleIds>

<add Name="6"/>

<add Name="7"/>

</OnlyPrimaryRoleIds>

</PropertiesConfiguration>

* V miste, kde chci kolekci ziskat ji dostanu klasickym zpusobem pomoci IConfigurationProvider.GetConfig<> a pomoci metody ToArray() dostanu pole bytu.

var propertiesConfiguration = Core.Container.Container.Current.Resolve<IConfigurationProvider>().GetConfig<PropertiesConfiguration>();

byte[] onlyPrimaryRoleIds = propertiesConfiguration.OnlyPrimaryRoleIds.ToArray() ;

# DatabaseConfiguration podle SimBiosu

[‎12.‎09.‎2018 10:12]  Filip Čálek:

udelej si clone https://tfs.mediaresearch.cz/NielsenAtm/\_git/SimBios

[‎12.‎09.‎2018 10:14]  Filip Čálek:

registraci DaoSource a EntityDaoFactory mas spatne, ale chapu, ze ti to nikdy nikdo nevysvetlil

stahni si to repo. otevri si simadmina a ja ti vysvetlim jak a proc to ma bejt;)

 [‎12.‎09.‎2018 10:45]  Filip Čálek:

otevri si Mediaresearch.SimAdmin.Service.Host.Installers.EntitiesInstaller

  zajima te cast v podmince      if (config.DifferentDatabases)

1) je treba takhle dalet rozdilne instance TransactionManager

2) IDaoSource musi implementovat DependencyDaoSource

DependencyDaoSource  ma prave na svedomi to automaticke registrovani dao objektu do containeru

ten forward je taky dulezitej

zadefinujes tam IPrintStorageDaoSource a IMediaDataDaoSource

otevri si v codovadle PressCreativeListLoader

v ctoru je IDaoSource printStorageDaoSource, IDaoSource mediaDaoSource

coz je opicarna

pak musi bejt naka definice v installeru...

[‎12.‎09.‎2018 10:56]  Peter Hlavenka:

jo na to jsem narazil

[‎12.‎09.‎2018 10:57]  Filip Čálek:

public PressCreativeListLoader(IEnumerable<MessageType> supportedMessageTypes, IPrintStorageDaoSource printStorageDaoSource, IMediaDataDaoSource mediaDaoSource)

a nemusis pak v installeru nic nefinovat

[‎12.‎09.‎2018 10:58]  Filip Čálek:

nicmene to bylo jen ciste hypoteticky... v tomto pripade to vlastne neni vubec potreba. vlastne bys skoro nikdy nemel resolvovat DaoSource

v tomto pripade by v **ctoru melo byt rovnou IPageDao, IPressPageDao a ICreativeItemDao**

[‎12.‎09.‎2018 10:59]  Peter Hlavenka:

**jak potom dostanu daoTridu ?**

aha

[‎12.‎09.‎2018 10:59]  Filip Čálek:

protoze to to uz zaregistruje DependencyDaoSource

az to predelas;)

**DatabaseConfiguration je trida ve Frameworku**

public class DatabaseConfiguration : ConfigBase

{

private const string ConfigName = "DatabaseConfiguration";

public override string GetConfigName() { return ConfigName; }

public static DatabaseConfiguration GetConfig()

{

var section = ConfigurationManager.GetSection(ConfigName) as DatabaseConfiguration;

return section ?? new DatabaseConfiguration();

}

//V appConfigu bude section pojmenovana MediaData3DaoSource

[ConfigurationProperty("MediaData3DaoSource")]

public DaoSourceConfig MediaData3DaoSource

{

get

{

object obj = this["MediaData3DaoSource"];

return obj as DaoSourceConfig;

}

}

//V appConfigu bude section pojmenovana PrintStorageDaoSource

[ConfigurationProperty("PrintStorageDaoSource")]

public DaoSourceConfig PrintStorageDaoSource

{

get

{

object obj = this["PrintStorageDaoSource"];

return obj as DaoSourceConfig;

}

}

**appConfig**

<configSections>

<section name="DatabaseConfiguration" type="MIR.Media.Coding.Common.Configuration.DatabaseConfiguration, MIR.Media.Coding.Common"/>

</configSections>

<DatabaseConfiguration

DeclaredTvDirectoryPath="\\192.168.0.67\ads\_admin\ads\_mediaone\test"

DeclaredTvImportedDirectoryPath="\\192.168.0.67\ads\_admin\ads\_mediaone\test\old"

PricingServiceAddress="net.tcp://AdsApp1:13893/PricingService">

<MediaData3DaoSource DbAlias="MediaData3" ConnectionString="Data Source=Stoupa;Initial Catalog=MediaData3Auto;User ID=MediaDataCoding2User;Password=Ws2kNS-e2e;" ServerTimeZone="Central Europe Standard Time">

<WorkableSchemas> // jsou schemata v databazi – z MediaData3Databaze kodovadlo pouziva tyto schemata (existuje jeste napr History atd)

<Item Name="Membership"/>

<Item Name="dbo"/>

<Item Name="Media"/>

<Item Name="Creative"/>

</WorkableSchemas>

<DaoAssemblies> // jsou assembly ve kterych jsou daoTridy kazda assembly muze byt jen jednou v ramci konfigurace

<Item Name="MIR.Entities.MediaData"/>

<Item Name="Mediaresearch.Framework.Security.BLToolkit"/>

</DaoAssemblies>

<EnumTableAssemblies> // jsou assembly ve kterych jsou enumy

<Item Name="MIR.Entities.MediaData"/>

<Item Name="Mediaresearch.Framework.Security.BLToolkit"/>

</EnumTableAssemblies>

</MediaData3DaoSource>

<PrintStorageDaoSource DbAlias="PrintStorage" ConnectionString="Data Source=Stoupa;Initial Catalog=PrintStorageAuto;User ID=MediaDataCoding2User;Password=Ws2kNS-e2e;" ServerTimeZone="Central Europe Standard Time">

<WorkableSchemas> // jsou schemata v databazi na PrintStorage db je jedine schema a to dbo.

<Item Name="dbo"/>

</WorkableSchemas>

<DaoAssemblies>

<Item Name="MIR.Entities.PrintStorage"/>

</DaoAssemblies>

<EnumTableAssemblies>

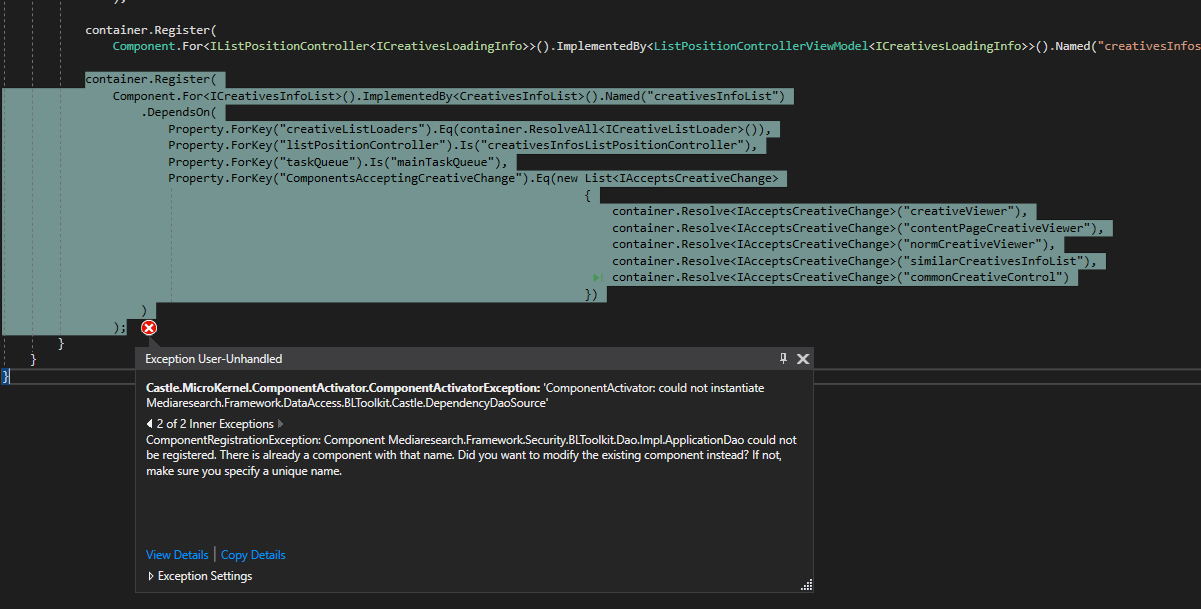
<Item Name="MIR.Entities.PrintStorage"/>

</EnumTableAssemblies>

</PrintStorageDaoSource>

</DatabaseConfiguration>

**Problem**:



**Reseni:**

[‎12.‎09.‎2018 15:45]  Peter Hlavenka:

mam v appconfigu pro obe databaze tohle :  <Item Name="Mediaresearch.Framework.Security.BLToolkit"/>

[‎12.‎09.‎2018 15:45]  Filip Čálek:

nj ale proc to pada pri tyhle registraci

jooo ale to bude ta chyba

[‎12.‎09.‎2018 15:46]  Filip Čálek:

md3DaoSource by melo mit na starosti jine Dao objekty nez PSDaoSource. jinak dojde ke konfliktu

[‎12.‎09.‎2018 15:48]  Peter Hlavenka:

hm a co s tim ? kdyz jsem odebral BLToolkit item z Printstorage tak to spadlo v DependencyDaoSource :  Component.For(genericIfaceType).Named($"enum\_{enumType.FullName}")

                    .UsingFactoryMethod(k => EntityDaoFactory.GetEnumTableValues(DefaultDbAlias, enumType, genericArguementKey))

[‎12.‎09.‎2018 15:50]  Filip Čálek:

jde o to z jaky databaze se berou membership tabulky

tam musi byt security.bltoolkit a membership schema

zkus to zjistit