♥IdaoSource a EntityDaoFactory

Installery s EntitiyDaoFactory se volaji jen jednou pri startu aplikace. V boootstrapperu , nebo v nejakem entry pointu aplikace, se musi zavolat container.ResolveAll<IDaoSource>();

# Spravna verze EntityDaoFactory a IDaoSource



Pricing connectionInstaller posledni verze:

**Entity:**

container.Register(Component.For<EntityDaoFactory>().UsingFactoryMethod(() =>

new EntityDaoFactory(connectionStringsHolder.DefaultDbAlias, connectionStringsHolder.MediaDataConnectionString, new TransactionManager())

{

DaoAssemblies = new[] { "MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit" },

EnumTableAssemblies = new[] { "MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit" },

ServerTimeZoneId = container.Resolve<Configuration.Properties>().ServerTimeZone,

Interceptors = container.ResolveAll<IDbOperationInterceptor>()

}

).LifestyleSingleton().Named(ConnectionInstallerComponents.EntityDaoFactoryName));

**DependencyDaoSource:**

container.Register(Component.For<IDaoSource>().Forward<IMediaDataDaoSource>().ImplementedBy<DependencyDaoSource>()

.DependsOn(

Dependency.OnComponent(typeof(EntityDaoFactory), ConnectionInstallerComponents.EntityDaoFactoryName),

Property.ForKey("WorkableSchemas").Eq(new[] { "Membership", "dbo", "Media", "Creative" })));

# TransactionDbManager

Na daoTridach se tvori pomoci CreateDbContext. Nemuzu vytvorit noveho pomoci new protoze nebude mit spojeni na databazi. Da se udelat i z IDaoSource:

m\_daoSource.TransactionManager.CallInsideTransaction(Action, m\_daoSource.CreateCommonDbContext());

# K cemu je dobry Forward

Pouziva se, kdyz chci nekde resolvnout IDaoSource a mam jich vice (PrintStorageDaoSouce , MediaData, pooling). Timto je rozlisim. Misto IDaoSource si resolvnu proste IMediaDataDaoSouce. Pokud nepotrebuju daoSoruce, ale jen nejakou daoTridu, pak forward nepotrebuju.

# Resolvnuti EnumTableValues

private readonly IEnumTableValues<FastenType, byte> m\_fastenTypes;

public GetFastenTypesAction(IAuditableIdentityProvider auditableIdentityProvider, IEnumTableValues<FastenType, byte> fastenTypes)

: base(auditableIdentityProvider)

{

m\_fastenTypes = fastenTypes;

}

# TransactionManager

**Musi byt vzdy jedinecny. Pokud mam vice databazi, musim je odlisit napr. takto (Schedule) Toto nepouzivat (Mitroz tu tridu nechal zrusit). Misto toho tvorit managera pomoci new(). Kazde IDaoSource ma managera na sobe a ted jde uz transakce volat na jakemkoli dau, ktere je podedene od**

Do registrace EntityDaoFactory davame noveho TransactionManagera. Ten je totiz vzdy na IDaoSource a ted taky na vsech daoTridach muzu volat necoDao.CallInsideTransaction().

try

{

void Action()

{

m\_aggregator.Publish(createMessage());

}

m\_daoSource.TransactionManager.CallInsideTransaction(Action, m\_daoSource.CreateCommonDbContext());

}

catch (Exception)

{

m\_daoSource.TransactionManager.RollbackTransaction();

throw;

}

~~public class TransactionManagerInstaller : IWindsorInstaller~~

~~{~~

~~public static string TvStorageKey { get; }~~

~~public static string MediaDataExportKey { get; }~~

~~public static string MediaDataKey { get; }~~

~~public static string PoolingKey { get; }~~

~~static TransactionManagerInstaller()~~

~~{~~

~~TvStorageKey = Guid.NewGuid().ToString();~~

~~MediaDataKey = Guid.NewGuid().ToString();~~

~~MediaDataExportKey = Guid.NewGuid().ToString();~~

~~PoolingKey = Guid.NewGuid().ToString();~~

~~}~~

~~public void Install(IWindsorContainer container, IConfigurationStore store)~~

~~{~~

~~if (!container.Kernel.HasComponent(TvStorageKey))~~

~~container.Register(Component.For<ITransactionManager>().ImplementedBy<TransactionManager>().Named(TvStorageKey));~~

~~if (!container.Kernel.HasComponent(MediaDataExportKey))~~

~~container.Register(Component.For<ITransactionManager>().ImplementedBy<TransactionManager>().Named(MediaDataExportKey));~~

~~if (!container.Kernel.HasComponent(MediaDataKey))~~

~~container.Register(Component.For<ITransactionManager>().ImplementedBy<TransactionManager>().Named(MediaDataKey));~~

~~if (!container.Kernel.HasComponent(PoolingKey))~~

~~container.Register(Component.For<ITransactionManager>().ImplementedBy<TransactionManager>().Named(PoolingKey));~~

~~}~~

~~}~~

**~~Do registrace EntityDaoFactory pak predam managera s klicem, podle toho o jakou db se jedna.~~**

~~var transactionManager = container.Resolve<ITransactionManager>(TransactionManagerInstaller.TvStorageKey);~~

~~container.Register(Component.For<EntityDaoFactory>().UsingFactoryMethod(() =>~~

~~new EntityDaoFactory(stringHolder.TvStorageDbAlias, stringHolder.TvStorageConnectionString, transactionManager)~~

~~{~~

~~DaoAssemblies = new[] { "Mediaresearch.TvStorage2.Entities" },~~

~~EnumTableAssemblies = new[] { "Mediaresearch.TvStorage2.Entities" },~~

~~ServerTimeZoneId = stringHolder.ServerTimeZoneId,~~

~~Interceptors = container.ResolveAll<IDbOperationInterceptor>()~~

~~}~~

~~).LifestyleSingleton().Named(ConnectionStrings.TvStorageEntityDaoFactoryKey));~~

~~container.Register(Component.For<IDaoSource>().Forward<ITvStorageDaoSource>().ImplementedBy<DependencyDaoSource>().Named(DaoSourceKey)~~

~~.DependsOn(~~

~~Dependency.OnComponent(typeof(EntityDaoFactory), ConnectionStrings.TvStorageEntityDaoFactoryKey)));~~

# DaoFactory podle SimBiosu:

## Rozhovor s Filipem

[‎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;)

## Jak na to - V Kodovadle (parametrizace Kodovadla) ::

Mam tridu ConnectionInstaller : IwindsorInstaller. V ni je registrace EntityDaoFactory a IDaoSource. Protoze v K. pouzivam dve databaze – MediaData a PrintStorage, potrebuju dve instance Factory a DaoSource.

public class ConnectionInstaller : IWindsorInstaller

{

public void Install(IWindsorContainer container, IConfigurationStore store)

{

RegisterDaoSources(container);

}

private void RegisterDaoSources(IWindsorContainer container)

{

var config = DatabaseConfiguration.GetConfig(); // Vytvoreni appConfigu a sekci bude popsano v [DatabaseConfiguration, ConfigBase Sections AppConfig.docx](DatabaseConfiguration,%20ConfigBase%20Sections%20AppConfig.docx)

//Prvni Factory

container.Register(Component.For<EntityDaoFactory>().LifestyleSingleton().Named(FactoryInfo.MediaData3DaoFactoryName)

.DependsOn(

Property.ForKey("defaultDbAlias").Eq(config.MediaData3DaoSource.DbAlias), // DbAlias a dalsi property jsou definovane v Configu

Property.ForKey("defaultConnectionString").Eq(config.MediaData3DaoSource.ConnectionString),

Property.ForKey("DaoAssemblies").Eq(config.MediaData3DaoSource.DaoAssemblies.ToArray()),

Property.ForKey("ServerTimeZoneId").Eq(config.MediaData3DaoSource.ServerTimeZone),

Property.ForKey("EnumTableAssemblies").Eq(config.MediaData3DaoSource.EnumTableAssemblies.ToArray()),

Dependency.OnValue<ITransactionManager>(new TransactionManager())

));

//Druha Factory

container.Register(Component.For<EntityDaoFactory>().LifestyleSingleton().Named(FactoryInfo.PrintStorageFactoryName)

.DependsOn(

Property.ForKey("defaultDbAlias").Eq(config.PrintStorageDaoSource.DbAlias),

Property.ForKey("defaultConnectionString").Eq(config.PrintStorageDaoSource.ConnectionString),

Property.ForKey("DaoAssemblies").Eq(config.PrintStorageDaoSource.DaoAssemblies.ToArray()),

Property.ForKey("ServerTimeZoneId").Eq(config.PrintStorageDaoSource.ServerTimeZone),

Property.ForKey("EnumTableAssemblies").Eq(config.PrintStorageDaoSource.EnumTableAssemblies.ToArray()),

Dependency.OnValue<ITransactionManager>(new TransactionManager())

));

// IDaoSource pro MediaData,

container.Register(Component.For<IDaoSource>().Forward<IMediaDataDaoSource>().ImplementedBy<**DependencyDaoSource**>().Named(FactoryInfo.MediaData3DaoSource)

.DependsOn(

Dependency.OnComponent(typeof(EntityDaoFactory), FactoryInfo.MediaData3DaoFactoryName),

Property.ForKey("WorkableSchemas").Eq(config.MediaData3DaoSource.WorkableSchemas.ToArray())));

// IDaoSource pro PrintStorage

container.Register(Component.For<IDaoSource>().Forward<IPrintStorageDaoSource>().ImplementedBy<**DependencyDaoSource**>().Named(FactoryInfo.PrintStorageDaoSource)

.DependsOn(

Dependency.OnComponent(typeof(EntityDaoFactory), FactoryInfo.PrintStorageFactoryName)));

container.Register(Component.For<IParamsSource>().ImplementedBy<DatabaseParamsSource>().LifestyleSingleton());

// Daosourcy si musim nekde na zacatku aplikace resolvnout, aby mi zaregistrovaly vsechny daoTridy

container.Resolve<IPrintStorageDaoSource>();

container.Resolve<IMediaDataDaoSource>();

}

//Ve staticke tride je to proto, aby to bylo dostupne odkudkoli. Pojmenovani slouzi na identifikace Factory a DaoSource

public static class FactoryInfo

{

public const string MediaData3DaoFactoryName = "entityDaoFactory";

public const string MediaData3DaoSource = "mediaDataDaoSource";

public const string PrintStorageFactoryName = "printStorageDaoFactory";

public const string PrintStorageDaoSource = "printStorageDaoSource";

}

}

Ted uz nepotrebuju opicarny typu :

m\_creativeItemDao = (CreativeItemDao) entityDaoFactory.CreateDaoByEntityType<ICreativeItem, int>(defaultDbAlias);

m\_paramsDao = daoSource.GetDaoByEntityType<IParamsDao, Params, byte>();

Staci mi dat si do konstruktroru zavislost na jakekoli dao a container mi ho resolvne. Postara se o to DependencyDaoSource, ktere zaregistruje vsechny daoTridy pri prvnim resolvnuti.

## Jak na to podruhe: sluzba Schedule: (zatim nahrubo, nez se jednotlive property daji do app. configu do nejake sekce)

var transactionManager = container.Resolve<ITransactionManager>(TransactionManagerInstaller.MediaDataKey);

container.Register(Component.For<EntityDaoFactory>().UsingFactoryMethod(() =>

new EntityDaoFactory(stringHolder.DefaultDbAlias, stringHolder.MediaDataConnectionString, transactionManager)

{

DaoAssemblies = new[] {"MIR.Schedule.Service", "MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit"},

EnumTableAssemblies = new[] {"MIR.Schedule.Service", "MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit"},

ServerTimeZoneId = stringHolder.ServerTimeZoneId,

Interceptors = container.ResolveAll<IDbOperationInterceptor>()

}

).LifestyleSingleton().Named(ConnectionStrings.MediaDataEntityDaoFactoryKey));

container.Register(Component.For<IDaoSource>().Forward<IMediaDataDaoSource>().ImplementedBy<DependencyDaoSource>().Named(ConnectionStrings.MediaDataDaoSourceKey)

.DependsOn(

Dependency.OnComponent(typeof(EntityDaoFactory), ConnectionStrings.MediaDataEntityDaoFactoryKey),

Property.ForKey("WorkableSchemas").Eq(new[] { "Membership", "dbo", "Media", "Creative" })));

container.Resolve<IMediaDataDaoSource>();

# Dao tridu si muzeme z daoSource resolvnout takto :

public TvPricedItemWrapperLoader(IDaoSource daoSource)

{

if (daoSource == null)

{

throw new ArgumentNullException(nameof(daoSource));

}

**m\_tvMediaMessageDao = daoSource.GetDaoByEntityType<**TvMediaMessageDao**, ITvMediaMessage, int>();**

**m\_paramsDao = daoSource.GetDaoByEntityType<**IParamsDao**,** Params**, byte>();**

}

# Parametrizace Kodovadla

**Do konstruktoru pridame zavislost na daoSource**

, IDaoSource daoSource

## NormalKey

m\_motiveVersionDao = daoSource.GetDaoByEntityType<MotiveVersionDao, IMotiveVersion, int>();

## MultiKey:

m\_creativeToCreativeItemDao = (ICreativeToCreativeItemDao) entityDaoFactory.CreateMultiKeyDaoByEntityType<ICreativeToCreativeItem>(Container.Container.Current.Resolve<ConnectionStringsHolder>().DefaultDbAlias);

**se da predelat takto:**

m\_creativeToCreativeItemDao = daoSource.GetMultiKeyDaoByEntityType<ICreativeToCreativeItemDao, ICreativeToCreativeItem>();

# V Normovadle se paramsDao tvori takto:

m\_paramsDao = (ParamsDao)entityDaoFactory.CreateDaoByEntityType<Params, byte>();

# CreativeTranslationDao z daoSource (Normovadlo)

private readonly ICreativeTranslationDao m\_creativeTranslationDao;

var daoSource = Container.Container.Current.Resolve<IDaoSource>();

m\_creativeTranslationDao = daoSource.GetMultiKeyDaoByEntityType<CreativeTranslationDao, ICreativeTranslation>();

# MotivletDao (Admin)

m\_motivletDao = daoSource.GetDaoByEntityType<IMotivletDao, Motivlet, int>();

# CreativeToCreativeItemDao, MediaMessageDao (Kodovadlo)

private readonly ICreativeToCreativeItemDao m\_creativeToCreativeItemDao;

**m\_creativeToCreativeItemDao** = (ICreativeToCreativeItemDao)Container.Container.Current.Resolve<EntityDaoFactory>().CreateMultiKeyDaoByEntityType<ICreativeToCreativeItem>(Container.Container.Current.Resolve<ConnectionStringsHolder>().DefaultDbAlias);

nebo:

**m\_mediaMessageDao** = (MediaMessageDao) entityDaoFactory.CreateDaoByEntityType<IMediaMessage, int>();

string defaultDbAlias = Container.Container.Current.Resolve<ConnectionStringsHolder>().DefaultDbAlias;

m\_sponsoredProgrammeDao = (SponsoredProgrammeDao)EntityDaoFactory.CreateDaoByEntityType<SponsoredProgramme, short>(defaultDbAlias);

m\_creativeDescriptionDao = (CreativeDescriptionDao)entityDaoFactory.CreateDaoByEntityType<CreativeDescription, int>(defaultDbAlias);

m\_creativeDao = (CreativeDao)entityDaoFactory.CreateDaoByEntityType<ICreative, int>(defaultDbAlias);

m\_videoCreativeDao = (VideoCreativeDao)entityDaoFactory.CreateDaoByEntityType<IVideoCreative, int>(defaultDbAlias);

m\_creativeItemDao = (CreativeItemDao) entityDaoFactory.CreateDaoByEntityType<ICreativeItem, int>(defaultDbAlias);

m\_creativeToCreativeItemDao = (CreativeToCreativeItemDao) entityDaoFactory.CreateMultiKeyDaoByEntityType<ICreativeToCreativeItem>(defaultDbAlias);

# EntityDaoFactory Kodovadlo

var entityDaoFactory = Container.Container.Current.Resolve<EntityDaoFactory>();å

**Kodovadlo ConnectionInstaller (puvodni verze – koukni nahoru podle simbiosu, je tam I kodovadlo):**

container.Register(

Component.For<EntityDaoFactory>().Named("entityDaoFactory")

.DependsOn(

Property.ForKey("defaultDbAlias").Eq(container.Resolve<ConnectionStringsHolder>().DefaultDbAlias),

Property.ForKey("defaultConnectionString").Eq(container.Resolve<ConnectionStringsHolder>().MediaDataConnectionString),

Property.ForKey("DaoAssemblies").Eq(new[] {"MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit" }),

Property.ForKey("EnumTableAssemblies").Eq(new[] {"MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit" })

),

Component.For<EntityDaoFactory>().Named("printStorageDaoFactory")

.DependsOn(

Property.ForKey("defaultDbAlias").Eq(container.Resolve<ConnectionStringsHolder>().PrintStorageAlias),

Property.ForKey("defaultConnectionString").Eq(container.Resolve<ConnectionStringsHolder>().PrintStorageConnectionString),

Property.ForKey("DaoAssemblies").Eq(new[] { "MIR.Entities.PrintStorage", "Mediaresearch.Framework.Security.BLToolkit" }),

Property.ForKey("EnumTableAssemblies").Eq(new[] { "MIR.Entities.MediaData", "Mediaresearch.Framework.Security.BLToolkit" })

),

Component.For<IDaoSource>().ImplementedBy<EntityDaoFactoryDaoSource>().Named("printStorageDaoSource")

.DependsOn(

Property.ForKey("entityDaoFactory").Is("printStorageDaoFactory")

),

Component.For<IDaoSource>().ImplementedBy<EntityDaoFactoryDaoSource>().Named("mediaDataDaoSource")

.DependsOn(

Property.ForKey("entityDaoFactory").Is("entityDaoFactory")

).IsDefault()

);