# Rozšíření FW Bltoolkit DataAccess

Ahoj všem,

Rozšířil jsem předky DAO o možnost volání Insert/Upade/Delete linq queries.

Jsou to například tyto:

db

        .Into(db.Employee)

            .Value(e => e.FirstName, "John")

            .Value(e => e.LastName,  "Shepard")

            .Value(e => e.Title,     () => "Spectre")

            .Value(e => e.HireDate,  () => Sql.CurrentTimestamp)

        .InsertWithIdentity();

db.Employee

    .Where(e => e.Title == "Spectre")

    .Set(e => e.Title, "Commander")

    .Update();

b.Employee

    .Where(e => e.Title == "Spectre")

    .Delete();

Určitě je používáte, ale ve stávající implementaci bylo potřeba myslet na to, že pokud je použijete, nevolají se automaticky interceptory atd. Narazil jsem na to při použití interceptoru pro auditní stopu a do update metod, která třeba updatovala pouze jeden sloupec buď musel jak pitomec neustále předávat datum a uzivatele, který záznam modifikoval, nebo si vzpomenout a zavolat metody před a po update ručně. Po pravdě nevím, proč mě to nenapadlo dříve, ale  přidal jsem následující metody do všech předku DAO:

void CallUpdateQuery(TEntity entity, IUpdatable<TEntity> updateQuery);

void CallInsertQuery(TEntity entity, IValueInsertable<TEntity> insertQuery);

void CallDeleteQuery(TEntity entity, IQueryable<TEntity> deleteQuery);

k nim samozřejmě ještě přetížené verze které mají jako argument již existující DbManager.

Jejich použití je snadné:

var insertQuery = db.Into(table).Value(d => d.Id, entity.Id).Value(d => d.Name, entity.Name);

CallInsertQuery(entity, insertQuery);

var updateQuery = table.Where(d => d.Id == entity.Id).Set(d => d.Name, entity.Name);

CallUpdateQuery(entity, updateQuery);

Metody zajistí správné volání interceptorů atd. Identity insert samozřejmě funguje také i v případě linq query budete mít v objektu po volání Id vyplněno.

Implementace je v od verze "Mediaresearch.Framework.DataAccess.BLToolkit" version="2.3.3.0"

**Petr Mitrofan** | Senior developer   
  
[petr.mitrofan@admosphere.eu](mailto:petr.mitrofan@admosphere.eu) | mobile: +420 608 579 850   
Nielsen Admosphere, a.s. | Ceskobratrska 2778/1 | 130 00 Prague 3 | Czech Republic | tel.: +420 222 717 763 | <www.nielsen-admosphere.eu>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| http://www.mediaresearch.cz/directmail/sign/png/na_nieadm.png | [http://www.mediaresearch.cz/directmail/sign/png/na_fb.png](https://www.facebook.com/pages/MEDIARESEARCH/191775633735?ref=hl) | [http://www.mediaresearch.cz/directmail/sign/png/na_yt.png](https://www.youtube.com/channel/UCpHAgO4ug-AG3ry_08_JyMg) | [http://www.mediaresearch.cz/directmail/sign/png/na_ss.png](http://www.slideshare.net/MEDIARESEARCH) | [http://www.mediaresearch.cz/directmail/sign/png/na_in.png](https://www.linkedin.com/company/211833?trk=vsrp_companies_cluster_name&trkInfo=VSRPsearchId:1025904261431603663332,VSRPtargetId:211833,VSRPcmpt:companies_cluster) |
|  | | | | |

**The company is registered in the Commercial Register at the Municipal Court of Prague, File reference: section B, insertion 7364.**  
Disclaimer: The only body authorised to represent Nielsen Admosphere, a.s. is in all cases solely its statutory body; to a limited extent, certain persons authorised or empowered in writing in the name of the company are also entitled to act, a list of which is published at <www.nielsen-admosphere.cz/na_list>; if not explicitly stated otherwise, these persons are not authorised to represent the company in legal matters verbally or via the means of electronic communication. This e-mail does not represent any offer to contract or its adoption, it is not a manifestation of the will to establish, change or end any legal relationship, to confirm, guarantee or affirm any liability and/or debt, if not specifically stated otherwise in the e-mail or its attachments. Confidentiality: This e-mail and its attachments are confidential and can contain communication protected by law. If you are not an authorised or intended recipient, you are notified that any disclosure, copying, distribution or other use of this e-mail is strictly prohibited. If you have received this e-mail by mistake, please announce this fact immediately to the sender and then delete it. Additional notifications related to this e-mail are available on <www.nielsen-admosphere.cz/na_disclaimer>.

# AuditableProvider

1. **Do configu musime pridat ServerTimeZone a propertu do rozhrani**

<serverTimeZone>Central Europe Standard Time</serverTimeZone> Ceska republika

<serverTimeZone>E. Europe Standard Time</serverTimeZone> Bulharsko

<components>

<component id="MediaDataDbConfiguration" type="MIR.Media.Changing2.Shell.Installers.MediaDataDbConfiguration, MIR.Media.Changing2.Shell.BG.Production">

<parameters>

<serverTimeZone>#{serverTimeZone}</serverTimeZone>

</parameters>

</component>

public string ServerTimeZone { get; }

1. **SimpleAuditableEntityProvider a jeho zavislost TimeZoneTimeProvider je zaregisrovany v ChangingInstalleru do Castlu**

container.Register(Component.For<ITimeProvider>().UsingFactoryMethod(() => new TimeZoneTimeProvider(configuration.ServerTimeZone)).LifestyleSingleton());

container.Register(Component.For<IAuditableIdentityProvider>().ImplementedBy<SimpleAuditableEntityProvider>().LifestyleSingleton());

container.Register(Component.For<IUserSource>().ImplementedBy<MD3DomainUserSource>().LifestyleSingleton());

MediaDataDbConfiguration configuration = container.Resolve<MediaDataDbConfiguration>();

**(**V normovadle jsem si vytvoril tridu MD3DomainUserSource a rozhrani UserSource, **)**

1. **Nastaveni Usera**

V Bootstrapperu je metoda ConfigureUser která nastavuje providerovi usera

private void ConfigureUser()

{

var userSource = m\_globalContainer.Resolve<IUserSource>(); // resolvne MD3DomainUserSource

var auditableIdentityProvider = m\_globalContainer.Resolve<IAuditableIdentityProvider>(); // resolvneme providera

var login = WindowsIdentity.GetCurrent().Name; // string

if (userSource.Login(login)) // podiva se do Security.User jestli tam je takovyto login, vraci return m\_logedUser != null;

{

var userId = userSource.GetUserId(); // z nactene entity si vezme Id

auditableIdentityProvider.SetAuditableIdentity(new AuditableIdentity(userId, login)); // nastavi Id a login providerovi

}

else

{

IsUserLogged = false;

}

}

# 6.3.2018

[‎3/‎6/‎2018 4:48 PM] Peter Hlavenka:

servisni akce dedi od AuditableServiceActionBase ktere predava providera.  Co se s nim deje dal tomu moc nerozumim . Co musi delat moje bazovka abych tohle mel poreseny ?

[‎3/‎6/‎2018 4:50 PM] Filip Čálek:

to vubec nepotrebujes nikam prehazovat

v servisni akci proste zavolas metodu (metody) z nejake dalsi tridy a auditable bude fungovat dal

[‎3/‎6/‎2018 4:50 PM] Peter Hlavenka:

super

# Popis pouziti providera pri insertu je v InsertQuery.docx

# Ziskani instance providera:

Abych mohl dat nejakemu konstruktoru dependency na providera, musim v containeru zaregistrovat :

container.Register(Component.For<IAuditableIdentityProvider>().ImplementedBy<SimpleAuditableEntityProvider>().LifestyleSingleton());

SimpleProvider je v namespace Mediaresearch.Framework.DataAccess.Core.Auditable

musi tam byt nainstalovany nuget

[‎5/‎25/‎2018 9:00 AM] Karel Honzl:

to mam, to je v pohode

ma ale dependency na ITimeProvider , to jsi taky registroval?

[‎5/‎25/‎2018 9:02 AM] Peter Hlavenka:

container.Register(Component.For<ITimeProvider>().UsingFactoryMethod(() => new TimeZoneTimeProvider(configuration.ServerTimeZone)).LifestyleSingleton());

[‎5/‎25/‎2018 9:05 AM] Karel Honzl:

configuration.ServerTimeZone - tohle je nekde v configu, jo?

[‎5/‎25/‎2018 9:05 AM] Peter Hlavenka:

 <components>

    <component id="MediaDataDbConfiguration"

               type="MIR.Media.Changing2.Shell.Installers.MediaDataDbConfiguration, MIR.Media.Changing2.Shell">

      <parameters>

        <mediaDataDbAlias>#{mediaDataDbAlias}</mediaDataDbAlias>

        <mediaDataConnectionString>#{mediaDataConnectionString}</mediaDataConnectionString>

        <serverTimeZone>#{serverTimeZone}</serverTimeZone>

# Pridani IauditableIdentityProvidera do Kodovadla a Normovadla:

1. **Musim pridat tridy :**

## MD3DomainUserSource : IUserSource

using System;

using MIR.Entities.MediaData.Dao.Security;

using MIR.Entities.MediaData.Security;

namespace MIR.Media.Norming.Auditable

{

public class MD3DomainUserSource : IUserSource

{

private readonly IUserDao m\_userDao;

private IUser m\_logedUser;

public MD3DomainUserSource(IUserDao userDao)

{

m\_userDao = userDao;

}

public bool Login(string login)

{

m\_logedUser = m\_userDao.SelectCurrentUser();

return m\_logedUser != null;

}

public byte GetUserId()

{

if (m\_logedUser == null)

{

throw new InvalidOperationException("No user is logged in");

}

return m\_logedUser.Id;

}

}

}

## public interface IUserSource

namespace MIR.Media.Norming.Auditable

{

public interface IUserSource

{

bool Login(string login);

byte GetUserId();

}

}

1. **Registrace :**

Component.For<ITimeProvider>().UsingFactoryMethod(() => new TimeZoneTimeProvider(configuration.ServerTimeZone)).LifestyleSingleton

Component.For<IAuditableIdentityProvider>().ImplementedBy<SimpleAuditableEntityProvider>().LifestyleSingleton(),

Component.For<IUserSource>().ImplementedBy<MD3DomainUserSource>().LifestyleSingleton()

(pripadne : Component.For<ITimeProvider>().UsingFactoryMethod(() => new TimeZoneTimeProvider(container.Resolve<Properties>().ServerTimeZone)).LifestyleSingleton(),)

**2.5) Do vsech configu vlozim**

<serverTimeZone>E. Europe Standard Time</serverTimeZone> BG

Nebo

<serverTimeZone>Central Europe Standard Time</serverTimeZone> CZ

a

<ServerTimeZone>#{serverTimeZone}</ServerTimeZone> Do properties

1. **Do properties:**

public string ServerTimeZone { get; set; }

1. **Bootstrapper**

private void **ConfigureUser()**

{

var userSource = Container.Container.Current.Resolve<IUserSource>();

var auditableIdentityProvider = Container.Container.Current.Resolve<IAuditableIdentityProvider>();

var login = WindowsIdentity.GetCurrent().Name;

if (userSource.Login(login))

{

var userId = userSource.GetUserId();

auditableIdentityProvider.SetAuditableIdentity(new AuditableIdentity(userId, login));

}

else

{

IsUserLogged = false;

}

}

**private bool IsUserLogged { get; set; } = true;**

private void DisplaySplash()

{

IWindowManager manager = Container.Container.Current.Resolve<IWindowManager>();

SplashScreenViewModel splashScreen = Container.Container.Current.Resolve<SplashScreenViewModel>();

splashScreen.DoWork = () =>

{

LoadPermissions();

InstallComponents();

ConfigureDeploymentService();

**ConfigureUser();**

Container.Container.Current.Resolve<IShell>().Initialize();

};

manager.ShowDialog(splashScreen);

**if (IsUserLogged)**

{

DisplayShell();

}

else

{

MessageBox.Show(LanguageLocalisation.UserDoesNotHavePermissions, string.Empty, MessageBoxButton.OK);

}

Application.Shutdown();

}

**V kodovadle problem Md3 nemohla resolvnout UserSource**

m\_userDao = daoSource.GetDaoByEntityType<IUserDao, IUser, byte>();