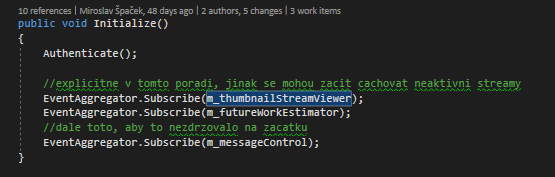
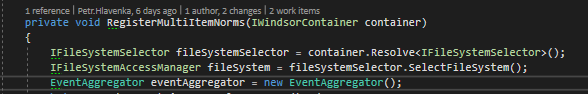
# Jak nastavit poradi handleru jednoho eventu (Zachytavadlo - ActualStreamChanged)

Mam dve tridy ktere obe handluji stejny event. Tyto tridy si resolvnu v shellViewModelu v metode Initialize() a oba tam subscribnu. Podle poradi subscribu budou vyvolany handlery.

Funguje to ovsem jen v ramci jednoho threadu..



# IEventAggregator - (pokud nefunguje, muze to byt tim, ze je tam vice aggregatoru. Zaregistrovan muze byt jen jeden, ale da se vytvorit I pomoci new:



**)**

Je soucasti CaliburnMicro. Pouzito ve zmenovadle.

Podstatou je poslat “message” z jednoho viewModelu do jineho. Message muze byt jakykoli objekt (string, image, cokoli..) Navod na pouziti:

<https://caliburnmicro.codeplex.com/wikipage?title=The%20Event%20Aggregator>

* Nejprve se musi zaregistrovat singleton instance ve Windsoru:

container.Register(Component.For<IEventAggregator>().ImplementedBy<EventAggregator>().LifestyleSingleton());

* Trida ktera chce event (message) **posilat**, musi chtit v konstruktoru IEventAggregator:

private readonly IEventAggregator m\_eventAggregator;

public FilterPartViewModel(IEventAggregator eventAggregator)

{

m\_eventAggregator = eventAggregator;

}

* Trida ktera chce message **prijmout** (byt posluchacem ) , musi implementovat rozhrani

public class DataViewModel : Screen, **IHandle<object>**

{

private BindableCollection<MotivletWrapper> m\_motives;

public DataViewModel(IClientToServicePublisher clientToServicePublisher, **IEventAggregator eventAggregator**)

{

**eventAggregator.Subscribe(this);**

}

* Ve tride kde chceme event poslat a v miste kde ho chceme poslat typicky v obsluzne metode tlacitka rekneme:

private void DoSearch()

{

m\_eventAggregator.**Publish**(" Poslana message");

}

* Aby aggregator vedel ktera message patri kteremu posluchaci, musim ji zabalit do nejakeho objektu . Objekty jsou pro kazdou message ruzne , to je princip rozliseni posluchacu.

[‎12/‎20/‎2017 1:47 PM] Filip Čálek:

co se tyka notifikovani jineho viewmodelu, tak to je klasicky priklad, kdy je dobre pouzit event aggregator

[‎12/‎20/‎2017 1:47 PM] Filip Čálek:

caliburn ma na to primo podporu

https://caliburnmicro.codeplex.com/wikipage?title=The%20Event%20Aggregator

<https://caliburnmicro.codeplex.com/wikipage?title=The%20Event%20Aggregator>

Tuto konverzaci jsme uložili. Brzo ji najdete na kartě Konverzace ve Skypu pro firmy a ve složce Historie konverzací v Outlooku.

[‎12/‎20/‎2017 3:27 PM] Peter Hlavenka:

message jsem si uspesne poslal, co nechapu je, jak vi aggregator kterou message ma predat kteremu posluchaci.

je singleton

[‎12/‎20/‎2017 3:28 PM] Filip Čálek:

pozna to podle typu predaneho generickeho argumentu

[‎12/‎20/‎2017 3:29 PM] Peter Hlavenka:

takze se da v projektu poslat napr string jen jednou? potom musim pouzit jiny objekt?  uz mi to asi doslo takze message zabalim do ruznych objektu a to je zpusob jakym se message odlisuji

[‎12/‎20/‎2017 3:30 PM] Filip Čálek:

presne tak

idealne, kdyz si vzdy vytvoris nejakou obalku

# Event aggregator z jineho threadu:

V Adminovi - zalozka repricing nefungoval aggregator. Pomohlo tohle: ble nepomohlo..

~~Aggregator.PublicationThreadMarshaller.Invoke(()=> Aggregator.Publish(true));~~ Vytvoren novy bug : <https://tfs.mediaresearch.cz/AdIntel/AdIntel/_workitems/edit/50385>

# Pouziti wrapperu

public class IsBusyWrapper

{

public IsBusyWrapper(bool isBusy)

{

IsBusy = isBusy;

}

public bool IsBusy {get; }

}

---------------------------------------------------------

try

{

Aggregator.Publish(new IsBusyWrapper(true));

}

catch (Exception ex)

{

if (m\_log.IsErrorEnabled)

m\_log.Error("Failed to refresh Pricing queue status!", ex);

InformationService.InvokeInformationOccured(this, Localisation.RefreshingStatusFailed, InformationSeverity.Error);

}

finally

{

Aggregator.Publish(new IsBusyWrapper(false));

}

--------------------------------------------------------------------------

public class MessageRepricingViewModel : ScreenWithIcon, IHandle<IsBusyWrapper>

public void Handle(IsBusyWrapper message)

{

IsBusy = message.IsBusy;

}