Pouziti sluzeb podle Mitroze ( PdfImageSource atd..)

Ve skenovadle v comboBoxu source je mozne zadat jako zdroj skenovani Pdf, Soubory , velky a maly skener. Mitroz docela zajimavym zpusobem resil volani akci ktere se maji stat podle toho jaky zdroj skenovani je zvoleny.

Mame bazovku FilesImageSourceBase ktera implementuje rozhrani IImageSource**.**

Bazovku implementuji tridy jako : FilesImageSource a PdfImageSource.

Tridy dedi od bazovky a ta implementuje rozhrani. Trida se tedy muze schovat za rohrani. Kazda trida ma svoje property ktere se pouzivaji na nastavovani ruznych zalezitosti ktere se maji stat kdyz je zvoleny tento typ zdroje

public override string Description => GuiResources.Files;

public override bool Multiselect => true;

protected override string Filter => "Image files (\*.jpg) | \*.jpg|All files (\*.\*)|\*.\*";

protected override string DefaultExt => ".jpg";

a zaroven ma property typu ScanResponse a IscanSettings

Odzacatku:

ComboBox ma nastaven jako ItemsSource kolekci rozhrani: BindableCollection<IImageSource> ImageSources

ItemsSource="{Binding ImageSources}"

A selectedItem :

SelectedItem="{Binding SelectedImageSource, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"

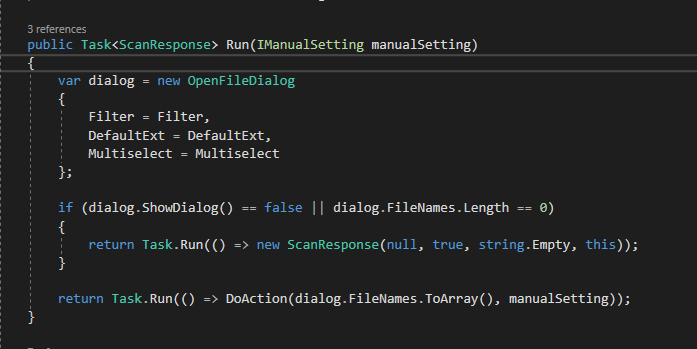
Jelikoz je ItemsSource kolekce rozhrani uz v setteru property selectedItem je mozne vyuzit jednu z vlastnosti na tride ktera implementuje toto rozhrani : if (m\_selectedImageSource.DissableAutoSettingForPdf){...}

Zmacknutim tlacitka **Load** neboli  **Scan** se presuneme do metody DoLoad()

se zavola asynchronne :

var result = await SelectedImageSource.Run(manualScanSettings);

Kazda trida ImageSource ma na (spolecne) bazovce metodu Run ktera vraci bezici Task (ulohu)

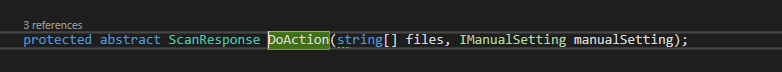


Otevre se dialog pro vyber souboru

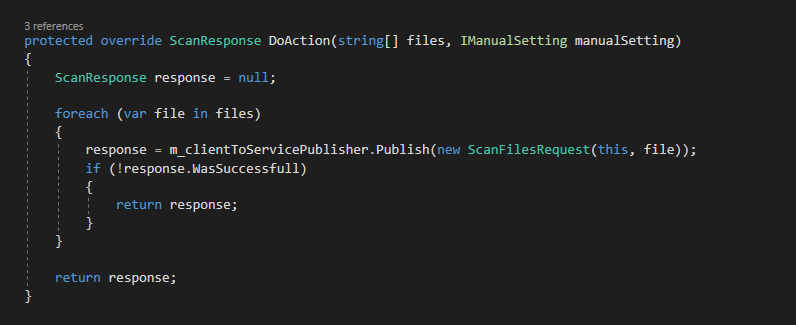
Pokud se dialog zavre, nebo nejsou vybrane soubory vrati bezici ulohu s novou (“ prazdnou ”) instanci ScanResponse

Jinak vrati bezici ulohu ktera provadi metodu DoAction

Metoda DoAction je na bazovce definovana jako abstract takze ji potomci musi prekryvat



A prekryti napr v FilesImageSource:



Je to vlastne uplne obycejna metoda jako byva v servisnich akcich ktera vraci ScanResponse, akorat se nevola pomoci clientToServicePublishera a nedotazuje se do databaze, ale teprve tady se

Publisherem vola servisni akce response = m\_clientToServicePublisher.Publish(new ScanFilesRequest(this, file));

Protoze jsme ve tride FilesImageSource, vime, ze zdrojem jsou soubory a podle toho volame servisni akci ScanFilesRequest

Kdybychom byli ve tride PdfImageSource volal by se ScanPdfRequest

A ve tride ScannerImageSource se zase vola return Task.Run(() => m\_publisher.Publish(new ScanRequest(this){ManualSetting = manualSetting}));

(Requesty maji samozrejme taky spolecnou bazovku )

**Tim ze jsme dali do comba rozhrani ktere zastupuje nejakou tridu muzeme nastavovat ruzne parametry v programu podle propert na jednotlivych implementacich rozhrani. Kazda implementace ma svoji asychronni metodu Run ktera vola ruzne servisni akce.**

**Castle.MicroKernel.Resolvers.DependencyResolverException:** 'Could not resolve non-optional dependency for 'TelevisionMessageBrowserViewModel

Could not resolve non-optional dependency for 'TelevisionMessageBrowserViewModel' (MIR.Pricing.MessagePricing.MessageBrowsing.Television.TelevisionMessageBrowserViewModel). Parameter 'currentUser' type 'MIR.Entities.MediaData.Security.IUser''