Changing 2.0

* Postup vytvoreni projektu je popsan v MojePoznamky => Castle Windsor
* Otestovat spojeni s databazi se da pomoci Unit testu -> MojePoznamky -> Unit Testy -> ConnectionTest
* Projekt po vytvoreni jsem ulozil na plochu do slozky ChangingProjekty
* Problem se SplashScreenem (MojePoznamky -> SplashScreen)

Media data model je tam jak davame scripty

C:\Pool\Admosphere\data\DatabaseModel\MediaData.Database\Model

Da se otevrit v programu EnterpriseArchitekt trial na 30 dni. Mitroz rikal o nejake comunity edici . Další moznost je ADOIT comunity edition

* Zkusit dojit od message az po ownera pomoci innerjoinu

Z tabulky media.mediamessage do tabulky owner creativu pres motiv, pres motivVersion pres motivlet pres ownera

Databaze testovaci je MediaData3Auto

Nainstalovat Xaml styler online extensions ctrl s neklikat aby se dalo lepe mergovat

V Xamlu staci dat prave tlacitko mysi a vybrat FormatXaml Styler vsechno uklidi.

[‎12/‎6/‎2017 9:34 AM] Peter Hlavenka:

Nevim jestli jsem dobre pochopil co je vlastne Message.  Rekl bych ze je to proste vyskyt reklamy . Jakmile se nekde objevi reklama mame message neboli jakysi soubor informaci o reklame. Message ma nejaky motiv , jeho verze v case a popis produktu je v motivletu. To je to co holky vidi ve zmenovadle. Oprav me pokud je to spatne .

[‎12/‎6/‎2017 9:48 AM] Filip Čálek:

ano jednoduse receno message je nejaka zakladni informace o vyskytu reklamy v nejakem mediu. v novinach, v radiu, v TV...

ma creativu, ktera se dal sklada z dalsich podtabulek, creativeItem, frame... atd.

[‎12/‎6/‎2017 9:48 AM] Filip Čálek:

tam jsou v pripade tisku jednotlive vystrihy dane reklamy. at uz je to vystrizeny obr nebo video, mp3...

[‎12/‎6/‎2017 9:49 AM] Filip Čálek:

kazda creativa ma Motiv a jeho verze platne v case. "Nivea akcni nabidka za srpen", "nabidka Nivea pro rok 2017"

kazda verze se sklada z jednoho ci vice Motivletu. Popis produktu. Je to sampon, kondicioner a krem na prdel zaroven. coz jsou 3 motivlety:)

Message

* zakladni informace o vyskytu reklamy v nejakem mediu. v novinach, v radiu, v TV...

**Norma**

* Při prvnim vyskytu reklamy se tato nova reklama vezme jako norma . Tato první reklama ma svoji creativu a zaroven svoji normu. To znamena, ze když Matching ( picture, video, audio) nalezne stejnou reklamu ) tato nova reklama ma vlastní creativu, ale normu uz ma te první nalezene reklamy

Creativa

* samotna reklama muze to byt bilboard, reklama v televizi , novinach atd .. Je slozena z podtabulek creativeItem, frame... atd., tam je reklama zaarchivovana. V pripade tisku je tam napr vystrih dane reklamy, ale muze tam byt třeba video, mp3…

Motiv

* Motiv ma v podstate jen ID-cko a sloupce Created/By a Modified/By

**MotivVersion**

* **Je to platna verze Motivu v case. Nekdy je potreba zmenit motiv v case příklad: Niveu koupi Shauma ale dozvi se to pozdeji protože az ted jim to schvalila nejaka nadnarodni společnost a proto chcou zmenit MotivVersnu třeba od 1.6.2017. Kdysi se změna provedla proste na Nivee a cela historie se prepsala. Proto vznikly MotivVersny. Jde o stejny motiv ale plati od – do**

**MotivLet**

* Kazda verze se sklada z jednoho ci vice Motivletu. Popis produktu. Je to sampon, kondicioner a krem na prdel zaroven. coz jsou 3 motivlety:) , Nebo v jedne reklame bylo vice vyrobku soucasne, mame taky vice motivletu.

Tabulka Meda.Motivlet

Když si otevru tabulku Media.Motivlet vidim sloupce ve kterych jsou jen ID odkazujici do jinych tabulek (ForeignKeys)



V puvodnim zmenovadle byly zobrazeny motivlety jenz měli sloupce:



Media.Owner Media.CompanyBrand Media.Market Media.Gender Media.Platform

Media.ProductBrand Media.Catebory Media.Telco Media.Motivlet.Id

Media.ProductDetail Media.Role

Tento motivlet je soucasti jedne nebo vice MotivVersny a ty jsou soucasti nejakeho motivu.

aplikace ARES Ministerstva financí je souhrnně zpřístupnit údaje z informačních systémů pro vedení registrů a evidencí veřejné správy o ekonomických subjektech. ARES přehledně zpřístupňuje údaje, přebírané ze zdrojových registrů do databáze ARES a současně umožňuje přímé přepnutí do www aplikací orgánů veřejné správy, které příslušné informační systémy provozují, pokud tyto aplikace již existují.

Tabulka Media.MediaMessage

Ma nasl. Sloupce:



A sloupce pokracuji:



Sloupec: Z jake tabulky Co tabulka obsahuje

[Id] Id

[ImportId] Media.Import

[MediaTypeId] Media.MediaType

[AdvertisementTypeId] Media.AdvertisementType

[AdvertisementSourceId] Media.AdvertisementSource

[AdvertisedFrom] DateTime

[AdvertisedTo] DateTime

[CodingHintContextId] Media. CodingHintContext

[CodingPlausibilityId] Media.CodingPlausability

[Ready] bit

[MediumId] Media.Medium

[PriceValue] Money

[PriceCurrencyId]

[PriceScopeId]

[IsPriceUnknown] bit

[PlacementId]

[IsSelfPromotion] bit

[PremiereMessageId]

[Note] varchar

[NormCreativeId]

[CreativeId]

[Created] datetime

[CreatedBy]

[Modified]

[ModifiedBy]

[DeletedByBusinessRules] datetime

[OriginalId] varchar

[Version] timestamp

V podstate tam kde konci nazev na Id je cizi klic který ukazuje na Id do tabulku která ma stejny nazev : NecoId ukazuje do tabulky Něco na sloupec Id.

V databazi MediaData3Auto jsou tabulky které zacinaji Něco. Něco. Něco

První nazev oznacuje oblast. Jsou to oblasti jako: Export, Import, History, Common, logging, Membership, ATMediaMatching, Media a Creativa

Druhy nazev uz blize specifikuje o co se jedna. Priklad : Všechny tabulky které zacinaji na Creative mají něco spolecneho s Creativou. Najdeme tu podtabulky Creativy jako Import, CreativeToCreativeItem, CreativeItem atd . Je to jedna velka oblast vztahujici se ke Creative. Tabulku Motiv sice na vytistenem planku najdeme protože creativa ma primou vazbu na motiv, ale je prazdna protože jeji obsah nenalezi do oblasti Creative. Nenajdeme tu ani tabulky motivlet a motivVersion. Ty patri do oblasti Media.

Creative.Creative => existuje

Creative.Motive => neexistuje protože tabulka Motive patri do oblasti Media a najdeme ji jako Media.Motive.

Další oblast která nas zajima je Media. Tady se najde Media.Motive, Media.MotivVersion, Media.Motivlet Media.MediaMessage atd.

Cesta z Media.Message na Ownera :

* Media.MediaMessage ma cizi klic do tabulky Creative (CreativeId) .
* Vezmu jednu message napr Id = 1 a budu chtit najit ownera
* Najdu teda Creativu která patri k MediaMessage s Id = 1

select mm.Id messageId, cr.Id creativeId from Media.MediaMessage mm

join Creative.Creative cr on mm.CreativeId = cr.Id

where mm.Id = 1

* Když uz mam creativu mam i MotiveId. Pridame teda join do tabulky Media.Motive. Nezapomenout nahoru pokazde pridat novy sloupec který chceme vypsat.

join Media.Motive mot on cr.MotiveId = mot.Id



* Jakmile mam motiveId uz muzu pripojit tabulku MotiveVersion

join Media.MotiveVersion motver on mot.Id = motver.MotiveId



* A k ni tabulku Motivlet prez vazebni tabulku MotiveToMotiveVersion

join Media.MotiveVersionToMotivlet mvtml on motver.Id = mvtml.MotiveVersionId

join Media.Motivlet let on mvtml.MotivletId = let.Id



* Jakmile mam motivet mam i Ownera protože OwnerId je jeden sloupec z tabulky Motivlet

join Media.Owner ow on let.OwnerId = ow.Id



Cely script:

select mm.Id MessageId, cr.Id CreativeId, mot.Id MotiveId, motver.Id MotiveVersionId, let.Id MotivLetId, ow.Name OwnerName

from Media.MediaMessage mm

join Creative.Creative cr on mm.CreativeId = cr.Id

join Media.Motive mot on cr.MotiveId = mot.Id

join Media.MotiveVersion motver on mot.Id = motver.MotiveId

join Media.MotiveVersionToMotivlet mvtml on motver.Id = mvtml.MotiveVersionId

join Media.Motivlet let on mvtml.MotivletId = let.Id

join Media.Owner ow on let.OwnerId = ow.Id

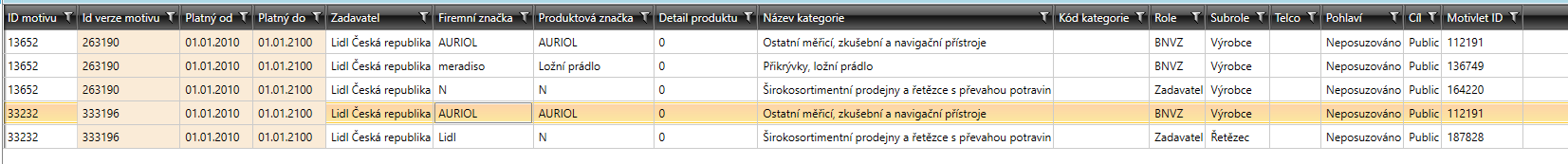
where mm.Id = 1

27. listopadu 2018

Metoda MotiveVersionDao.LoadMotivesByParameters mi vrati podle zadanych parametru všechny motiveVersny ve kterych jsou vyhledavane slova:

Zadavatel: Lidl Česká republika

Produktova znacka: AURIOL



V tomto pripade naslo dve motiveVersny : 263190 a 333196. Tyto motiveVersny ale maji vice motivletu a procedura vrati vsechny. Musime vracene hodnoty prefiltrovat tak, aby byly vyplsany jen motivlety ve kterych jsou hledane hodnoty.

Storovka **Media.Codding.FilterMotive** (Databaze MediaData3Auto)

* V ObjectiveType tride je navic properta ActiveTo . Ta neni potrebna mohla by zmizet.
* Ve filtrech a v ObjectiveType chybi Telco ktere je zobrazene v motivletech . Neni potreba filtrovat podle Telco ?

Metoda LoadByParameters mi vrati stringy pro owner, companyBrand, productBrand a productDetail. Nemel bych mit I jejich Id ? Na wrapperu jsem si vytvoril property jako CategoryName ale tady naopak pouzivam jen CategoryId . Tyto property by mely zmizet

* Ve filtrech musí byt objekt který bude mit napr CategoryId a zaroven CategoryName .
* Metoda LoadMotivesByParameters je omezena na vraceni 100 zaznamu. Mela by ale vracet všechny nalezene. Napr. Lidl ma vice nez 10000 motivletu. Když zvysim metode hranici kolik zaznamu ma nalezt, spadne to na

**BLToolkit.Data.DataException:** 'Časový limit provedení vypršel. Časový limit uplynul před dokončením operace nebo server neodpovídá.' **BLToolkit.Data.DataException:** 'Časový limit provedení vypršel. Časový limit uplynul před dokončením operace nebo server neodpovídá.'

Když si necham vyhledat ProductBrand kde je hodnota Lasagne, aplikace mi nenajde motivlet kde je Id = 28079

select ProductBrand.Name, \* from media.Motivlet let

join media.ProductBrand on let.ProductBrandId = ProductBrand.Id

where let.Id = 28079



Metoda LoadMotives mi tento motivlet vůbec nenajde, problem není ve filtrech. Uz storovka Media.Codding.FilterMotive tento zaznam nevrati.

// dalsi parametry nejsou stringy, request musi predat mapperu datove typy podle typu Id coz uz dela, ted jen mit v combu objekty ktere maji spravne Id.

Premyslel jsem nad optimalizaci nacitani. Myslim ze ciselniky musim v query joinout. EnumTableValues muzu pouzit v pripade, ze jiz znam Id -> (GetValueById). Když napr. uzivatel vyhledava podle ownera nezname GenderId a  hledanemu objektu musime zobrazit Gender tj musim GenderId dostat z databaze.

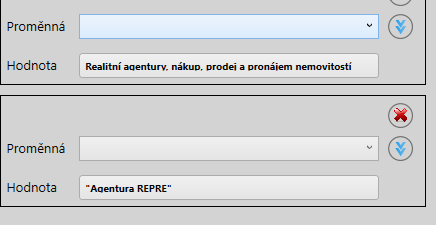
Premyslel jsem nad optimalizaci nacitani. Myslim ze ciselniky musim v query joinout. EnumTableValues muzu pouzit v pripade, ze jiz znam Id -> (GetValueById). Když napr. uzivatel vyhledava podle ownera nezname GenderId a  hledanemu objektu musime zobrazit Gender tj musim GenderId dostat z databaze.

ComboBoxViewModel.LoadObjectives vola provider.GetObjectives . Rozhrani IObjectiveProvider implementuji ruzne tridy. Jednou z nich je CompanyBrandObjectiveProvider

Kde je metoda GetMotivletsSubSet : return m\_motivletDao.SelectByPropertyType(((CompanyBrandObjective)objective).Id, "CompanyBrand", 100, out allMotivletsCount);

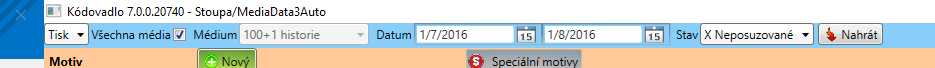
Vyhodil jsem ChangingActionBase a naimportoval jsem projekt MIR.Common2Common ve kterem je nova bazovka pro servisni akce které potrebuji AuditableProvidera a teda meni něco v databazi. Nacitaci akce dedi jen od ServiceAction. Nova Bazovka je

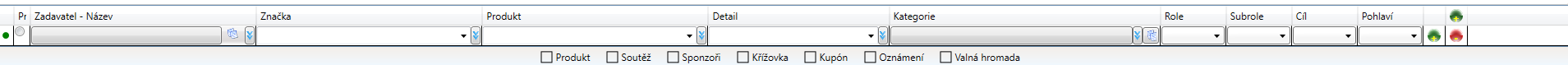
AuditableServiceActionBase



Ve skenovadle chcou akce které potrebuji AuditableProvidera referenci na MIR.Common2.Common. Muzu pridat?

Kodovadlo:



Tlacitko Novy:

Opravy bugu ve skenovadle:

#### Dám si do vyhledávacího panelu vpravo název média (jakýkoliv), automaticky se mi ukáže Zpracovatel, ale dovolí mi to si navolit Zpracovatele z výběru o 2 řádky níže.

Pridal jsem na processorCombo do xamlu propertu IsEnabled bindovanou na processorInfoVisibility s trueToFalseConverterem.

#### Když vložím do vyhledávacího panelu vpravo "Datum do", které je menší než "Datum od" a neukáže mi to chybu.

Pridal jsem validaci

#### Do okna Upravit publikaci je možné vložit historický datum vydání (3.1.2000) i datum připomenutí.

Nastaveno na 2010 jak ve validatoru aby zobrazilo error, tak v metode CheckInputs která brani ulozeni nespravnych vstupu.

#### Do čísla vydání, roku vydání a plánovaného čísla vydání je možné zadat i záporné hodnoty a uloží se to!

Workaround, nova properta IssueCountAsShort. IssueCount je byte a nemuze nabyvat zapornych hodnot. Zaporne hodnoty uz nejdou ulozit.

#### Dám Vytvořit publikaci, vymažu číslo vydání, uloží se to s číslem vydání 1.

Nejde ulozit publikaci bez cisla vydani. Issue number je predvyplneno podle max hodnoty vracene z databaze.

#### Vytvořím publikaci, když ji edituji, změním datum vydání a uložím, tak se mi nezaktualizuje seřazení.

Serazeno po editaci.

#### Když se do okna Upravit publikaci vloží do roku/čísla vydání písmeno, tak se ukáže, že byl záznam aktualizován, ale v DB se nechá ta hodnota, která tam byla předtím.

Pokud uzivatel vlozi pismeno do TextBoxu bindovaneho na ciselny datovy typ, do setteru se vůbec nedostanu abych to mohl resit. Uzivateli se ted zobrazi cervena validace a message, ze neplatne hodnoty byly ignorovane.

Merge : Pull -> commitnu normalne, když uz někdo commitnul prede mnou updatnu se na nej, ted jsem na jeho vetvi a mergnu si do ni to svoje.

Bez pristupu do databaze není mozne prefiltrovat ValueComboItems. To bychom museli mit v pameti uplne všechny motivlety z databaze, protože query na vyber moznych hodnot do dalsiho comba říká napr: vyber z motivletů jen ty productBrandy, kde je Owner napr Lidl. Takze se prehrabavame ve všech existujicich motivletech . Otazka je, jestli je chceme všechny motivlety nacashovat. Je jich 567 000

ToDo:

* Export gridu do Excelu
* Razeni v gridu dle vice sloupcu zjistit zda Telerik podporuje
* Info na klavesove zkratky a barvy
* Zmena MotiveName se musi udelat pro vsechny Motivlety za datumem.
* Kontrola jestli motivlet po zmene nejake hodnoty uz existuje. Pokud ano , musim pouzit tento motivlet.

Zmenovani musí probehnout v DataViewModelu protože tady je kolekce Motivlets a zaroven na wraperech zname property které urcuji co se bude menit.

Musim vytvorit wrapper ChangeVersionsWrapper a ConfirmChangesWrapper a pomoci agregatoru predat do dataVM kolekci NewValues kde budou nove hodnoty které se budou menit. (selectedValues). V dataVM projdeme motivlety a u tech které mají propertu spravne, zmenime v transakci hodnoty.

NewValues comba radsi nacitam rovnou na zacatku. Asynchronni nacitani zpusobovalo zaseknuti nebo pad aplikace pokud uzivatel zacal psat do comba novou hodnotu drive nez byly nacteny data. Stejne musel cekat az byly data nacteny . V combu MotiveNames trva dlouho i prefiltrovani jestli tam takova hodnota je.

// bude se menit u wrapperu kde je AllConditionsMet = true a isBeforeSelectedDate = false

// a budou se menit jen property podle vlastnosti na wrapperu (IsOwnerSearched atd..)

// vymenovat se bude za nove hodnoty z kolekce wrappers napr wrapers kde je argument owner , selectedValues.First()

// nejdrive musim zmenit platnost puvodniho motivletu podle SelectedDate.

// vytvorim novy motivlet jehoz zacatek bude stejny jako konec puvodniho

Factory trida v coru bude mit nejaky v konstruktoru. castle resolver jako genericky argument bude prijimat value datacontract a vracet bude tridu která mi bude insertovat misto 8 serv akci budu mit 8 inserteru.

Doplnit zmeny na dao tride a servisni akci pro gender a platform tak aby brali z daa jen id a potom pomoci ~~mapperu predelat uz od GetCategory (pridavat Name) viz composedDataContractMapper~~

Text="{Binding RelativeSource={RelativeSource FindAncestor, AncestorType=telerik:GridViewDataColumn}, Path=DataMemberBinding}"

Tlacitkem potvrdit zmeny, zmenim na motivletum, které splnuji kriteria vyhledavani, property zvolene v novych hodnotach. Pokud není zadany cas od ktereho menim, zmenim jen property na všech nalezenych motivletech , nemenim jejich verze ani Id.

Když mam zadany cas od ktereho motivlety menim, musim ukoncit stavajici motivVersnu k tomuto datumu, vlozit novou MotiveVersnu která zacina na tento datum a bud probiha anebo ma konec tak aby na ni navazovala nasledujici versna.

Tato vlozena motivVersna bude mit uz motivlety se zmenenymi propertami , stejne jako další navazujici . Musime tedy vytvorit nove motivlety .

Když updatnu motivlet tak bude zmeneny pro všechny motivy které na nej odkazuji takze muzu updatnout jen pokud není další motiv který na ne odkazuje , jinak musim vytvorit nove motivlety.

Pokud nemam zadany cas, updatnu propertu na nalezenem motivletu ve všech MotiveVersnach ve kterych se nachazi.

Pokud mam zadany cas od, musim ukoncit motivversnu k tomuto datumu , vytvorit novou motivVersnu, vytvorit nove motivlety podle puvodnich ale se zmenenymi novymi hodnotami a updatnout motivlety i v navazujicich motiveVersnach.

Když vytvarim novy motivlet(zmeneny) musim zkontrolovat zda uz takovy motivlet neexistuje a pokud ano musim ho pouzit.

Stare motivletId se nebude nikdy mazat zustane na motivletech kterym se omezuje platnost.

Id motivu a verze rovnou zgrupovat

Když je zadany datum nebo Motive musim insertovat

Hlidat duplicity , pouzit motiv pokud uz existuje

Pokud menim primarni motivlet musim updatnout i primaryId

Odstranit tlactika zmenit verzi a sloucit filtrovat

Zvyraznit nejak motiveversny které patri k sobe

Tlacitko filtrovat se presune dolu , na jeho stlaceni se musí spustit command prirazeny jak puvodnimu tlacitku , tak tlacitku nahradit.

27. listopadu 2018

1. To co vyhovuje filtrum se nemeni takze se to muze odstranit ze zmenovane kolekce pomoci setnuti AllMotivletsMet = false . Zobrazenou kolekci to neovlivni ta zustane podbarvena spravne.
2. Mohly by se zobrazit pocty jako kolik itemu by mělo byt spravne nacteno a kolik se jich bude menit.
3. Zmenovani by mělo probehnout asynchronne. Nez se ale dodela první operace, musí byt tlacitko potvrdit disablovane. V oznamovaci oblasti ukazat, ze probiha operace (jako v skenovadle) a zamezit zavreni aplikace.

4.3.2018

Co se teda deje při zmenovani všech motivletu když není selectnuty datum ani MotiveName?

private List<MotiveVersionToMotivletDataContract> ChangeAll(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

1. Ziskame motivlety ktere odpovidaji Filters (nejoinovane)
2. Zjistime jestli zmenou nevytvori duplicitu
   1. Zmenim jim property podle NewValues
   2. Vlozim zmenene do TempMotivlet
   3. Prohledam duplicity **GetDuplicities**
3. Ziskam Id verzi kterych se to tyka
4. Ziskam vsechny stare vazby pro allConMet vcetne jejich sourozencu.
5. Vlozim do TempMtm
6. Vytvorim si TempMtm kde bude Id verze , predchazejici MotivletId a nove MotivletId.
7. Zgroupnu je podle verze a letId
8. Kde je nove letId vratim verzi a nove Id

Kde je null vratim verzi a puvodni letId jen pro skupiny ktere maji vice nez jednu polozku => duplicity

* 1. Prohledam duplicity v rozsahu jedne verze (kdyz se meni vice motivletu ve verzi)

1. Ziskam neexistujici vazby (jen u nich muze byt duplicita v ramci verze.)
2. Vlozim je do tempMtm
3. TempMotivlet zgrupuju podle vlastnosti a vyberu jen skupiny > 1
4. TempMtm joinu se zgrupovanym dotazem a beru vazby kde neni null => mam vazby pro motivlety se stejnymi vlastnostmi (Nemusi byt na stejne verzi)
5. Zgroupuju je podle VersionId
6. Skupiny vazeb kde je count > 1 jsou duplicity.
7. Pokud je duplicita zobrazim ji a konec.
8. Motivlety ktere neexistuji zgroupnu podle vlastnosti a updatnu jen prvni z kazde skupiny
9. Ostatnim updatnu primaryId a zmenim vazby.
10. Zmenim MotiveName (pokud je vybrany ke zmenovani)

Co se deje při zmenovani všech motivletu když není selectnuty datum ale je selectnuty MotiveName?

private List<MotiveVersionToMotivletDataContract> ChangeWithMotiveName(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

1. Ziskam Composed datacontracty ktere splnuji podminky z filtru. Mam na nich i cislo verze
2. Z nich si vyselectuju vazby
3. Z vazeb ziskam cisla verzi
4. Duplicity potrebuju posuzovat pro vsechny verze ne jen pro ty, ktere uzivatel vybere. Ziskam vsechny motivlety podle filters, joinute s verzi

Nebude jich milion protoze je to omezene vyberem verze.

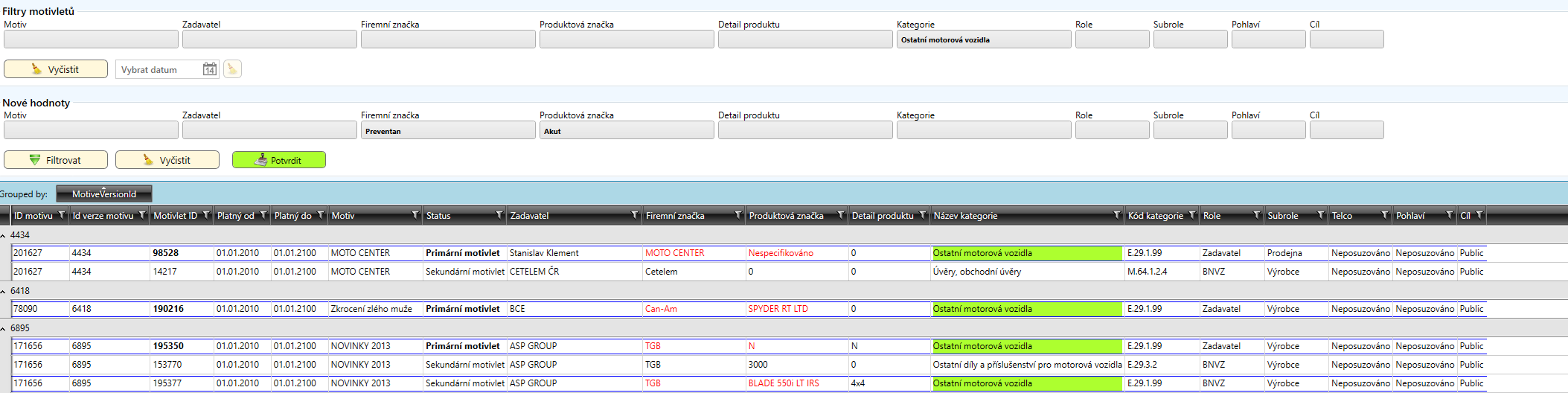
1. Vsem motivletum zmenim property a insertnu je do TempMotivlet.
2. Posoudim duplicity pro vsechny verze (I ty nevybrane uzivatelem)
3. Mam duplicity ? zobrazim a konec
4. Do TempMotivlet si vlozim jen lety ktere splnuji vsechny parametry (vcetne filtru MotiveName)
5. Posoudim zmenovane motivlety podle toho, jestli existuji I na jinych verzich. Pokud ano, nemuzu je updatnout, musim insertnout novy s vlastnostmi na jaky chci zmenit.
6. Motivlety k insertu si zgrupuju podle vlastnosti a insertnu jen jeden s takovymi vlastnostmi.
7. Motivlety ktere nejsou na jinych verzich muzu updatnout
8. Do TempMtm si vlozim bindingy jen pro verze ktere jsou vybrane
9. Updatnu primaryId jen pro verze ktere jsou vybrane
10. Zmenim vazby jen pro verze ktere jsou vybrane.
11. Zmenim pripadne motiveName

Co se bude dit pri selectnutem datumu a nevybranem MotiveName?

Chtelo by to omezeni na ciselniky tady kdyz se bude menit Role Zadavatel kde je 300 000 motivletu tak je potrebuju joinout na verze. Tohle samotne upadne..

1. V kazdem pripade potrebuju motivlety ktere vyhovuji filtrum
2. Mohl bych je sebrat nejoinovane jako v changeAll
3. Zmenit jim property a s puvodnim letId je vrazit do TempMotivlet.
4. Tabulku Mot si joinu s Mtm a Ver a muzu ukoncit platnost probihajicich verzi.
5. Hodila by se mozna dalsi tabulka nebo sloupec kam by se ukladal puvodni datum platnosti . Neco jako verMtm sloupce: IdVerze, puvodni konec platnosti a novy konec platnosti. (novy konec je selectedDate) spise budeme potrebovat sloupec pro nove Id verze (insertovanej)
6. Rekneme : dej mi verze kde je datum od < selectedDate < datum do. Zmen jim konec platnosti a vrat jejich bindingy.
7. Ziskame motivlety podle bindingu, zmenime jim property
8. Motivlety na rozdelene verzi se nesmi updatovat . Musi se bud najit takovy jaky chci vytvorit a zmenit primaryId a vazba, anebo ho vytvorit
9. Vytvorime si prazdne verze . Musime si pro kazdou pamatovat cislo te puvodni
10. Motivlety se zmenenymi prop se daji do TempMotivlet a do nove verze hledame uz existujici , nebo zgroupneme property a prvni insertneme a vlozime do verzny jeho cislo.
11. Duplicity se tady hledat nemusi protoze zadne nejsou. Bud takovy motivlet najdeme nebo ne.

Duplicity navic:



//ToDo :

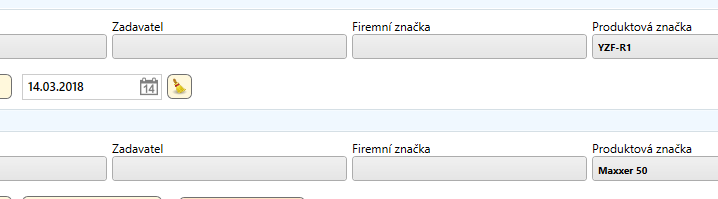
1. Pri stlaceni tlacitka filtrovat se nezobrazuje tocic , jen tesne pred zobraznim dat => udelat to asynchronne.
2. Dodelat cislovani stranek pri posouvani ( 2000 - 3000)
3. Posouvani stranek pri velkem mnozstvi stranek posila pro kazde posunuti dotaz takze muzu naklikat 20 dotazu ktere jeste nedobehly a mam problem
4. Nacitani poctu motivletu probiha jeste po zobrazeni dat . U velkeho mnozstvi nacteni poctu sourozencu docela trva. (Motivlety jsou nejoinovane, takze po zmene se vyprazdni tabulka a kdyz ukazuje nejaky pocet motivletu jsou to sirotci)
5. jinak nam vznikaj dalsi duplicity a sice MotiveVersion. dejme tomu, ze mame dve verze jednoho Motivu a kazda ma ruznej motivlet. pokud ale dojde ke sjednoceni motivletu, tak mame dve verze s jednim motivletem. databazove to je ok, ale neni to spravne:)

ty verze by se mely sjednotit v jednu, ale to asi nechejme az na pozdeji

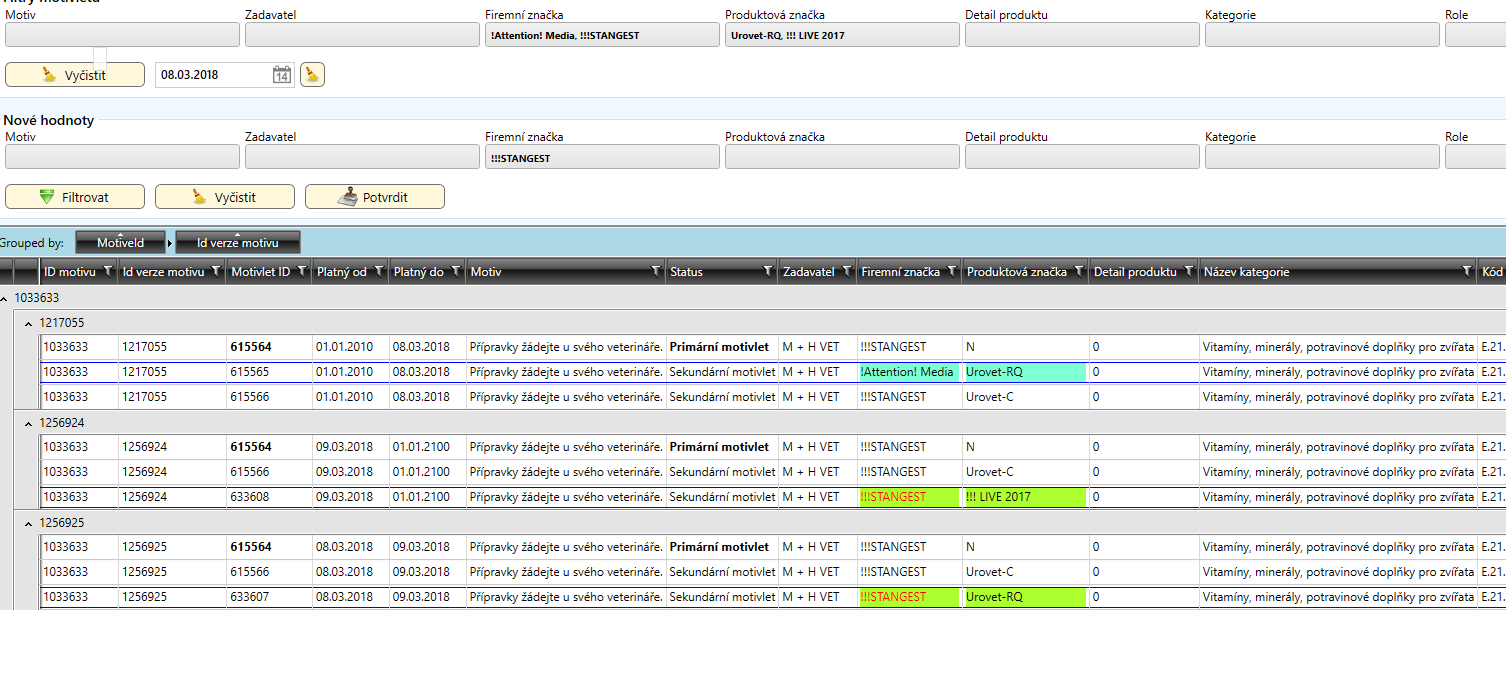
1. Pri zmenovani s datumem by se melo zjistovat, jestli je nutne tvorit novy motivlet, nebo jen updatnout stavajici.

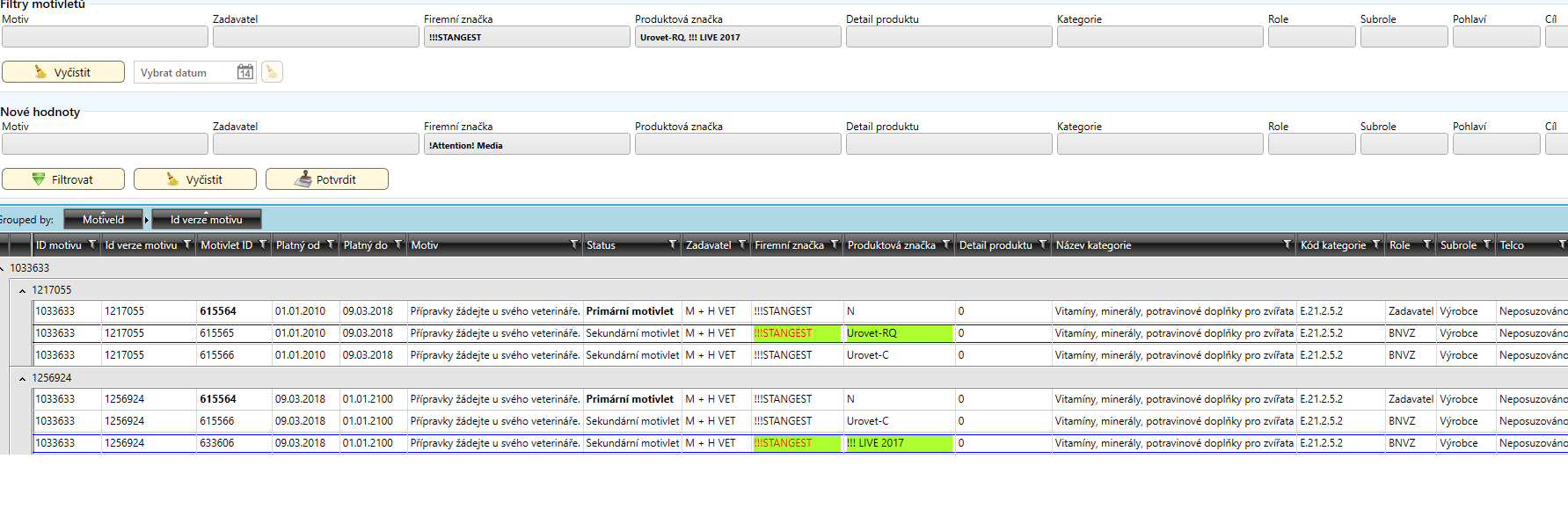
//////////////////

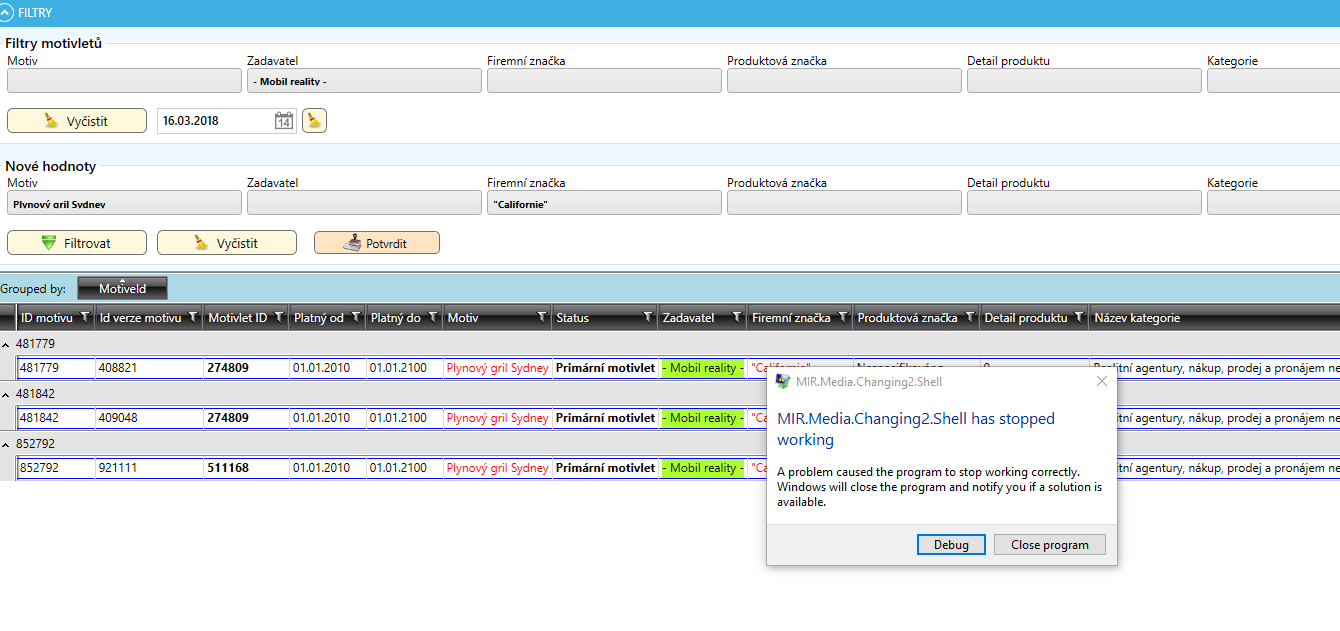
Bohuzel nemuzu pouzit ChangeAll po volani metody ChangeWithDate. ChangeAll posoudi motivlety na zmeny , ty ktere existuji pouzije a zmeni vazby a ty ktere neexistuji updatne to ale nemuzu udelat kdyz zmenuju od data. Tam by se updatli I vsechny pred datumem.

Ty ktere existuji pouziju a ty ktere neexistuji musim vytvorit (ledaze by tento motivlet neexistoval na jine verzne pred datumem. To bych ho pak updatnout mohl)

Nejprve otestujeme duplicity od nejakeho datumu . Pokud na puvodnich verzich nevznikaji , teprve ted si insertneme nove verze. Duplicity jsou otestovane, muzeme zmenit vazby . Ty ktere by se ale pri zmenovani vseho jen updatli, tady musim insertnout . ( jen prvni ze stejnych a vazby.)





B.R.D => Stangest

Chce to z TempMotivlet nejak odstranit ty ktere se jen updatujou

1. Dej mi ty co se updatujou
2. Vloz je do TempMotivlet
3. Zgrupni je a updatni prvni jako v ChangeAll
4. Do TempMotivlet zase changeMotivlets
5. Pokracujeme od metody MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

Po vikendu prestane pracovat ClickOnce => presype se ostra databaze a nebude tam tabulka TempMotiveVersion na mediaData3Auto

Rekapitulace => zmenovadlo funguje (jde k otestovani)

27. listopadu 2018

Tlacitko Potvrdit posle do DataViewModelu Parametry z Filters. Nejdulezitejsi servisni akce je ChangeMotivesAction.

Zni se podle toho jestli je selectnuty datum a MotiveName volaji tridy, které dedi od bazovky ChangingServiceBase na které jsou všechny spolecne metody. Jsou to:

## ChangeAllService

Vola se když není selectnuty motivName ani datum.

Protože není vybrane zadne omezeni proste menime vse bez ohledu na datum a name. Musime ale vždy seskupit updatovane podle vlastnosti a updatnout jen jeden motivlet. Ostatní které se chcou zmenit na takovyto motivlet dostanou Id toho updatnuteho

Overi jestli se nasli duplicity : Ano => zobrazi je a konec. Ne => pokracuje ve zmenovani. // Při hledani duplicit se do TempMotivlet dostanou changedMotivlets : to jsou motivlety odpovidajici Filters ale se zmenenymi propertami podle NewValues

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet(); => // Najde motivlety ktere neexistuji, **zgrupne** je podle vlastnosti a updatne jen prvni z motivletu ktere by mely stejne vlastnosti. V tabulce TempMotivlet jsou ted jen motivlety bez duplicity.

MotivletDao.UpdateMotivletsWithoutDuplicity(); // Po teto operaci uz vsechny motivlety na ktere se chcou puvodni zmenit existuji.

InsertIntoTempMotivlet(changedMotivlets); => // Do TempMotivlet se musi vratit vsechny motivlety nejen ty zgroupnute.

MotiveVersionDao.UpdatePrimaryIds(null); // Postara se o zmenu PrimaryId na versne

MotiveVersionToMotivletDao.UpdateMotivletToMotiveVersions(null); // Metody jsou pouzite I pri selectnutem datumu . Tady je null. Postara se o zmenu vazeb. Najde takovy motivlet na jaky se chce puvodni zmenit a updatne vazbu

ChangeMotiveName(request, null); // Metody jsou pouzity I pri vybranem MotiveName . Tady neni vybrany tedy null. Pokud je v NewValues vybrany novy motiveName tak se zmeni. ( Zmena se projevi pro vsechny motivlety na versne)

### kod

namespace MIR.Media.Changing2.Core.ChangingServices

{

public class ChangeAllService : ChangingServiceBase

{

public ChangeAllService(IMotiveVersionDao motiveVersionDao,

IMotiveVersionToMotivletDao motiveVersionToMotivletDao, IMotivletDao motivletDao,

IMapper<ParametersDataContract, MotivletFilterParamLists> filterMapper, IMapper<Motivlet, TempMotivlet> motivletMapper

)

: base(motiveVersionDao, motiveVersionToMotivletDao, motivletDao, filterMapper, motivletMapper)

{

}

public List<MotiveVersionToMotivletDataContract> ChangeAll(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

{

var motivlets = MotivletDao.GetUnjoinedMotivlets(filtersParamLists);

var changedMotivlets = GetDuplicities(request, out var duplicities, motivlets);

if (duplicities.Any())

{

return duplicities.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).Distinct().ToList();

}

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.UpdateMotivletsWithoutDuplicity();

InsertIntoTempMotivlet(changedMotivlets);

MotiveVersionDao.UpdatePrimaryIds(null);

MotiveVersionToMotivletDao.UpdateMotivletToMotiveVersions(null);

ChangeMotiveName(request, null);

return null;

}

}

}

## ChangeWithMotiveNameService

Vola se při selectnutem MotiveName bez datumu

var allConditionMets = MotivletDao.GetComposedDataContracts(filtersParamLists, null); // tady vybira i podle MotiveNamu ale ne podle datumu. Misto datumu dame null. Najde lety podle parametru spoji je s verzemi a vrati slozeny contract.

var allConditionMetBindings = GetAllConditionsMetBindings(allConditionMets); // sebereme vazby

public List<MotiveVersionToMotivletDataContract> ChangeWithMotiveName(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

{

var allConditionMets = MotivletDao.GetComposedDataContracts(filtersParamLists, null);

var allConditionMetBindings = GetAllConditionsMetBindings(allConditionMets);

var versions = allConditionMetBindings.Select(d => d.MotiveVersionId).ToList(); // sebereme Id verzi

var motivlets = MotivletDao.GetJoinedMotivlets(filtersParamLists); // z databaze si podle Filters ziskame motivlety (joinovane)

var changedMotivlets = ResolveContracts(motivlets, request.Values); // zmenime jim property

InsertIntoTempMotivlet(changedMotivlets); // vlozime je do TempMotivlet

var duplicityList = GetDuplicityList(versions, filtersParamLists, request); // tady se prohledavaji duplicity trochu jinak vola se metoda GetDuplicityList

if (duplicityList.Any()) // kdyz se najdou duplicity zobrazime je a nepokracujeme dal.

{

return duplicityList.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).ToList();

}

InsertIntoTempMotivlet(changedMotivlets);

var updatableMotivlets = MotivletDao.GetUpdatableMotivlets(filtersParamLists.MotiveNames); //vrati jen existujici motivlety, ktere nejsou na jinych verzich

var grouped = (from temp in updatableMotivlets

group temp by new

{

Owner = temp.OwnerId,

Company = temp.CompanyBrandId,

ProductBrand = temp.ProductBrandId,

ProductDetail = temp.ProductDetailId,

Category = temp.CategoryId,

Role = temp.RoleId,

Market = temp.MarketId,

Gender = temp.GenderId,

Platform = temp.PlatformId

}

into gr

select gr.First()).ToList();

MotivletDao.TempMotivletInsertBatch(grouped); //zgrupneme podle vlastnosti a vlozime do tempMotivlet

MotivletDao.UpdateMotivletsWithoutDuplicity();

InsertIntoTempMotivlet(changedMotivlets); // vrati do temp tabulky changedMotivlets

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet(); // Najde neexistujici, zgrupne je a vlozi do tempMotivlet

MotivletDao.InsertMotivletsFromTempMotivlets(); // Insertne neexistujici z TempMotivlet

InsertIntoTempMotivlet(changedMotivlets); // vrati do temp tabulky changedMotivlets

InsertIntoTempMotiveVersionToMotivlet(allConditionMetBindings);

MotiveVersionDao.UpdateRestricteredPrimaryIds();

MotiveVersionToMotivletDao.UpdateRestricteredMotivletToMotiveVersions();

ChangeMotiveName(request, versions);

return null;

}

}

## ChangeWithDateService

Vola se při selectnutem datumu

Při vybranem datumu je potreba rozdelit probihajici verzi. Probihajici se ukonci v selectedDate a zadne zmeny na ni se nedeji. Vlozime novou verzi která bude v selectedDate zacinat a koncit tam kde koncila puvodni verze. Tato vlozena verze dostane stejne motivlety jako mela probihajici, na nich se provedou pozadovane zmeny a ty budou i na všech verzich které navazuji na tuto vlozenou verzi. Pokud uzivatel vybere datum který odpovida zacatku verze, nic se nerozdeluje. Motivlety které jsou za timto datumem se zmeni. Stare motivlety ale musí zustat na puvodni verzi, coz znamena, ze nove se musí insertnout. Pokud ale tento motivlet není na zadne verzi která byla před zadanym datumem, tak se jenom updatuje.

Overi jestli se nasli duplicity : Ano => zobrazi je a konec. Ne => pokracuje ve zmenovani. // Při hledani duplicit se do TempMotivlet dostanou changedMotivlets : to jsou motivlety odpovidajici Filters ale se zmenenymi propertami podle NewValues

var runningVersions = MotiveVersionDao.GetRunningVersions(filtersParamLists, selectedDate); // Najde bezici verze ( je nutne je zgrupovat podle Ver.Id . Na jedne verzi totiz muze byt vice motivletu ktere se chcou stat jinym motivletem)

InsertTempVersions(runningVersions); // Vlozi verze do tabulky TempMotiveVersion

MotiveVersionDao.UpdateTempVersionsMotiveName(newMotiveName); // Pokud je vybrany Name v NewValues, zmeni na novych a nasledujicich verzich Name

MotiveVersionDao.InsertMotiveVersionsFromTempTable(); // Vytvori nove prazdne verze v tabulce MotiveVersion

var versionToChangeMotiveName = MotiveVersionDao.GetVersionIds(filtersParamLists, selectedDate); // dej mi verze ktere odpovidaji filtrum ve Filters a selectnutemu datumu. Verze musim dostat po pridani novych verzi, ale jeste pred tim, nez se zmeni motivlety na pridanych verzich

MotiveVersionToMotivletDao.InsertBindingsFromPreviousVersions(); // Naplnim prazdne verze puvodnimi motivlety

// kdyz verzi rozdeluju, vsechny motivlety existuji na starsi verzi => ta rozdelovana je ted ta starsi => nic nemuzu updatnout

// kdyby byl selectedDate jeste pred zacatkem bezici verze tak by se mohly motivlety jen updatnout coz se taky stane.

// V TempMotivlet jsou vsechny motivlety ktere odpovidaji Filters ( i ty z verzi pred datumem) uz se zmenenymi propertami. Staci je zgroupnout a insertnout prvni ze skupiny

// Tohle je zgrupne podle propert, resetne TempMotivlet a vlozi tam z kazde skupiny jen jeden motivlet (updatuje se vzdy jen jeden ze skupiny)

// tady ale nic neupdatuju, takovy motivlet jaky je ted v TempMotivlet insertnu

var lets = MotivletDao.**GetBeforeDateNotExistingMotivlets**(selectedDate); // Jeli selectnuty Name nesmi vracene lety byt na jinych verzich nez na vyselectovanych coz tady neni protoze pri selectnutem se pouziva jina servica

MotivletDao.TempMotivletInsertBatch(lets); // Vlozi tyto prebrane motivlety do temp tabulky

// Poreseni updatu:

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet(); // bere to co je v TempMotivlet zgrupne a vlozi zase do TempMotivlet bez duplicit

MotivletDao.UpdateMotivletsWithoutDuplicity(); // Updatne jen prvni ze zgrupovanych

InsertIntoTempMotivlet(changedMotivlets); // vrati do temp tabule zase vsechny zmenene motivlety

// Poreseni insertu:

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet(); // bere to co je v TempMotivlet zgrupne a vlozi zase do TempMotivlet bez duplicit

MotivletDao.InsertMotivletsFromTempMotivlets(); // Insertne jen prvni ze zgrupovanych

InsertIntoTempMotivlet(changedMotivlets); // // Vratim tempTabulku do puvodniho stavu.

MotiveVersionDao.UpdatePrimaryIds(selectedDate); // Pokud se meni motivletId na versne musi se updatnout I PrimaryId podle toho zda byl puvodni motivlet primarni.

MotiveVersionToMotivletDao.UpdateMotivletToMotiveVersions(selectedDate); // Meni se vazby. Ted pro motivlety jake jsme chteli vytvorit vzdy takovy najde at uz se puvodni updatly nebo insertly

ChangeMotiveName(request, versionToChangeMotiveName); // zmena namu

MotiveVersionDao.DeleteTempTable(); // tabulka TempMotiveVersion existuje v databazi . ( # tabulka z nejakeho duvodu neakceptovala insertovane hodnoty, proto jsme vytvorili normalni tabulku ) Musi se udrzovat prazdna.

### Kod

using System;

using System.Collections.Generic;

using System.Linq;

using Mediaresearch.Framework.Mapping;

using MIR.Common2.MediaData.DataContracts;

using MIR.Entities.MediaData.Dao.Media;

using MIR.Entities.MediaData.Media;

using MIR.Media.Changing2.Common.Requests;

namespace MIR.Media.Changing2.Core.ChangingServices

{

public class ChangeWithDateService : ChangingServiceBase

{

public ChangeWithDateService(IMotiveVersionDao motiveVersionDao,

IMotiveVersionToMotivletDao motiveVersionToMotivletDao, IMotivletDao motivletDao,

IMapper<ParametersDataContract, MotivletFilterParamLists> filterMapper, IMapper<Motivlet, TempMotivlet> motivletMapper

)

: base(motiveVersionDao, motiveVersionToMotivletDao, motivletDao, filterMapper, motivletMapper)

{

}

public List<MotiveVersionToMotivletDataContract> ChangeWithDate(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

{

var selectedDate = (DateTime) request.SelectedDate;

var motivlets = MotivletDao.GetUnjoinedMotivlets(filtersParamLists);

var changedMotivlets = GetDuplicities(request, out var duplicities, motivlets);

if (duplicities.Any())

{

return duplicities.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).Distinct().ToList();

}

var runningVersions = MotiveVersionDao.GetRunningVersions(filtersParamLists, selectedDate);

InsertTempVersions(runningVersions);

MotiveVersionDao.UpdateRunningVersionsActiveToDate(selectedDate);

MotiveVersionDao.UpdateTempVerisionsActiveFromDate(selectedDate);

var newMotiveName = ResolveMotiveName<MotiveNameDataContract>(request.Values);

if (!string.IsNullOrEmpty(newMotiveName))

{

MotiveVersionDao.UpdateTempVersionsMotiveName(newMotiveName);

}

MotiveVersionDao.InsertMotiveVersionsFromTempTable();

var versionToChangeMotiveName = MotiveVersionDao.GetVersionIds(filtersParamLists, selectedDate);

MotiveVersionToMotivletDao.InsertBindingsFromPreviousVersions();

var lets = MotivletDao.GetBeforeDateNotExistingMotivlets(selectedDate);

MotivletDao.TempMotivletInsertBatch(lets);

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.UpdateMotivletsWithoutDuplicity();

InsertIntoTempMotivlet(changedMotivlets);

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.InsertMotivletsFromTempMotivlets();

InsertIntoTempMotivlet(changedMotivlets);

MotiveVersionDao.UpdatePrimaryIds(selectedDate);

MotiveVersionToMotivletDao.UpdateMotivletToMotiveVersions(selectedDate);

ChangeMotiveName(request, versionToChangeMotiveName);

MotiveVersionDao.DeleteTempTable();

return null;

}

}

}

## ChangeWithMotiveNameAndDate

Vola se při selectnutem datumu a vybranem MotiveName

Pokud je vybrany datum a Name je potreba menit jen vybrane verze. Navíc pokud motivlet není před datumem tak se muze updatnout jenom pokud není ani na zadne verzi která není uzivatelem vybrana ve Filters.

Update motivletu => motivlet není před vybranym datumem ani na jiné nevybrane verzi.

Je skoro totozna s tridou ChangeWithDate az na několik odlisnosti:

var lets = MotivletDao.GetBeforeDateNotExistingMotivlets(selectedDate, filtersParamLists.MotiveNames); // Ma navic filtersParamLists.MotiveNames protoze vybira motivlety I podle toho, jestli je verze na ktere je tento motivlet mezi vybranymi ve filters

toto nahrazuje dva radky v ChangeWithDate

var allConditionMets = MotivletDao.GetComposedDataContracts(filtersParamLists, selectedDate); // vybere vsechny motivlety podle Filters vcetne Namu a datumu

var allConditionMetBindings = GetAllConditionsMetBindings(allConditionMets); // z nich vybereme bindingy

InsertIntoTempMotiveVersionToMotivlet(allConditionMetBindings); // vlozime do tempMtm tabulky

MotiveVersionDao.UpdateRestricteredPrimaryIds(); // updatneme primId jen pro omezene verze datumem a namem

MotiveVersionToMotivletDao.UpdateRestricteredMotivletToMotiveVersions(); // to same pro vazby.

### Kod

using System;

using System.Collections.Generic;

using System.Linq;

using Mediaresearch.Framework.Mapping;

using MIR.Common2.MediaData.DataContracts;

using MIR.Entities.MediaData.Dao.Media;

using MIR.Entities.MediaData.Media;

using MIR.Media.Changing2.Common.Requests;

namespace MIR.Media.Changing2.Core.ChangingServices

{

public class ChangeWithMotiveNameAndDateService : ChangingServiceBase

{

public ChangeWithMotiveNameAndDateService(IMotiveVersionDao motiveVersionDao, IMotiveVersionToMotivletDao motiveVersionToMotivletDao, IMotivletDao motivletDao,

IMapper<ParametersDataContract, MotivletFilterParamLists> filterMapper, IMapper<Motivlet, TempMotivlet> motivletMapper)

: base(motiveVersionDao, motiveVersionToMotivletDao, motivletDao, filterMapper, motivletMapper)

{

}

public List<MotiveVersionToMotivletDataContract> ChangeWithMotiveNameAndDate(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

{

var selectedDate = (DateTime) request.SelectedDate;

var motivlets = MotivletDao.GetUnjoinedMotivlets(filtersParamLists);

var changedMotivlets = GetDuplicities(request, out var duplicities, motivlets);

if (duplicities.Any())

{

return duplicities.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).Distinct().ToList();

}

var runningVersions = MotiveVersionDao.GetRunningVersions(filtersParamLists, selectedDate);

InsertTempVersions(runningVersions);

MotiveVersionDao.UpdateRunningVersionsActiveToDate(selectedDate);

MotiveVersionDao.UpdateTempVerisionsActiveFromDate(selectedDate);

var newMotiveName = ResolveMotiveName<MotiveNameDataContract>(request.Values);

if (!string.IsNullOrEmpty(newMotiveName))

{

MotiveVersionDao.UpdateTempVersionsMotiveName(newMotiveName);

}

MotiveVersionDao.InsertMotiveVersionsFromTempTable();

var versionToChangeMotiveName = MotiveVersionDao.GetVersionIds(filtersParamLists, selectedDate);

MotiveVersionToMotivletDao.InsertBindingsFromPreviousVersions();

var lets = MotivletDao.GetBeforeDateNotExistingMotivlets(selectedDate, filtersParamLists.MotiveNames);

MotivletDao.TempMotivletInsertBatch(lets);

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.UpdateMotivletsWithoutDuplicity();

InsertIntoTempMotivlet(changedMotivlets);

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.InsertMotivletsFromTempMotivlets();

InsertIntoTempMotivlet(changedMotivlets);

var allConditionMets = MotivletDao.GetComposedDataContracts(filtersParamLists, selectedDate);

var allConditionMetBindings = GetAllConditionsMetBindings(allConditionMets);

InsertIntoTempMotiveVersionToMotivlet(allConditionMetBindings);

MotiveVersionDao.UpdateRestricteredPrimaryIds();

MotiveVersionToMotivletDao.UpdateRestricteredMotivletToMotiveVersions();

ChangeMotiveName(request, versionToChangeMotiveName);

MotiveVersionDao.DeleteTempTable();

return null;

}

}

}

## CustomChangingService

Je doplneni zmenovadla, tak, aby bylo mozne zmenovat jen polozky které uzivatel naklika.

namespace MIR.Media.Changing2.Core.ChangingServices

{

public class CustomChangingService : ChangingServiceBase, ICustomChangingService

{

private readonly IAuditableIdentityProvider m\_auditableIdentityProvider;

public CustomChangingService(IAuditableIdentityProvider auditableIdentityProvider, IMotiveVersionDao motiveVersionDao,

IMotiveVersionToMotivletDao motiveVersionToMotivletDao, IMotivletDao motivletDao, IMapper<Motivlet, TempMotivlet> motivletMapper

)

: base(auditableIdentityProvider, motiveVersionDao, motiveVersionToMotivletDao, motivletDao, motivletMapper)

{

m\_auditableIdentityProvider = auditableIdentityProvider;

}

// zmenuji se vybrane motivlety jen na motivech ktere uzivatel oznacil

public List<MotiveVersionToMotivletDataContract> CustomChange(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists)

{

// pokud motivlet existuje na jinych motivech, nebo na tomto a neni selectnuty a nevznikne duplicita v ramci tohoto motivu, tvorim novy

// pokud na jinych motivech neexistuje, neni naco ho updatnout , tvorim novy

// pokud neni vybrany datum, menim vsechno co uzivatel naklikal.

// pokud je vybrany datum, menim vsechno co uzivatel naklikal, s tim, ze probihajici verzi rozdeluju.

// kontrola duplicity. Pokud by zmenou vznikl motivlet ktery uz na teto versne je, nebo se meni dva motivlety z jedne versny na stejny , oznacim duplicitu a konec.

var duplicities = CheckDuplicities(request, out var changedMotivlets, out var originalBindings);

if (duplicities.Any())

{

return duplicities; // List

}

if (request.SelectedDate == null)

{

MotivletDao.InsertMotivletsWithoutDuplicityToTempMotivlet();

MotivletDao.InsertMotivletsFromTempMotivlets(m\_auditableIdentityProvider);

InsertIntoTempMotivlet(changedMotivlets);

MotiveVersionToMotivletDao.InsertTempBindings(originalBindings);

MotiveVersionDao.CustomUpdatePrimaryIds(null);

MotiveVersionToMotivletDao.CustomUpdateMotivletToMotiveVersions(null);

var versions = request.SelectedMotivlets.Select(d => d.MotiveVersionId).ToList();

ChangeMotiveName(request, versions);

}

else

{

Console.WriteLine(@"request.SelectedDate != null");

}

return new List<MotiveVersionToMotivletDataContract>();

}

// kontrola duplicity. Pokud by zmenou vznikl motivlet ktery uz na teto versne je, nebo se meni dva motivlety z jedne versny na stejny , oznacim duplicitu a konec.

private List<MotiveVersionToMotivletDataContract> **CheckDuplicities**(ChangeMotivesRequest request, out List<Motivlet> changedMotivlets, out List<TempMotiveVersionToMotivlet> originalBindings)

{

var duplicities = new List<TempMotiveVersionToMotivlet>();

// motivlety podle vybranych motivletId

var motivlets = MotivletDao.GetMotivletsById(request.SelectedMotivlets.Select(d => d.MotivletId).ToList());

// zmenime jim property podel NewValues

changedMotivlets = ResolveContracts(motivlets, request.Values);

// vlozim do TempMotivlet

InsertIntoTempMotivlet(changedMotivlets);

// do TempMtm si dam soucasne vazby uzivatelem vybranych motivletu (znam je)

originalBindings = request.SelectedMotivlets.Select(d => new TempMotiveVersionToMotivlet

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId,

PreviousId = d.MotivletId

}).ToList();

// vlozim do TempMtm

MotiveVersionToMotivletDao.InsertTempBindings(originalBindings);

// updatnu v tempMtm sloupec motivletId podle porovnanych propert (temp a mot)

MotiveVersionToMotivletDao.UpdateNewBindings();

// v tabulce tempMtm mam tri naplnene sloupce. Joinu k tomu mtm a beru ty, kde v mtm neni null => duplicita TIMTO ZJISTIM, JESTLI VZNIKNE DUPLICITA TIM ZE NA VERZI UZ TAKOVY LET JE

duplicities.AddRange(MotiveVersionToMotivletDao.GetExistingBindings());

duplicities.AddRange(GetDuplicitiesAtVersionRange(request)); // pridam duplicity ktere muzou vzniknout tak, ze se dva motivlety z jedne verzny meni na jeden stejny

var list = new List<MotiveVersionToMotivletDataContract>();

list.AddRange(duplicities.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).Distinct().ToList());

list.AddRange(duplicities.Select(d => new MotiveVersionToMotivletDataContract

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.PreviousId

}).Distinct().ToList());

return list;

}

//POKUD MAJI DVA CONTRACTY SE ZMENENYMI PROPERTAMI STEJNE VER.iD => DUPL.

[C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\SQL + LINQ](file:///C:\Users\phlavenka\OneDrive\Nielsen%20%20prace\Moje%20poznamky%20Nielsen\SQL%20+%20LINQ) Group by into LINQ.docx

private List<TempMotiveVersionToMotivlet**> GetDuplicitiesAtVersionRange**(ChangeMotivesRequest request)

{

var changedComposedContracts = ResolveContracts(request.SelectedMotivlets, request.Values);

var query = from con in changedComposedContracts

group con by con.MotiveVersionId

into g

select new

{

Group = g,

SubGroups = from c in g

group c by new

{

c.OwnerId,

c.CompanyBrandId,

c.ProductBrandId,

c.ProductDetailId,

c.CategoryId,

c.RoleId,

c.MarketId,

c.GenderId,

c.PlatformId

}

into g2

where g2.Count() > 1

select g2

};

var groups = query.Select(d => d.SubGroups.SelectMany(x => x)).ToList();

var dal = groups.Select(d => d.Select(x => x)).SelectMany(d => d).ToList();

return dal.Select(d => new TempMotiveVersionToMotivlet

{

MotiveVersionId = d.MotiveVersionId,

MotivletId = d.MotivletId

}).ToList();

}

}

public interface ICustomChangingService

{

List<MotiveVersionToMotivletDataContract> CustomChange(ChangeMotivesRequest request, MotivletFilterParamLists filtersParamLists);

}

}

## Rekapitulace

Zaciname klasicky v MainViewModelu

MainViewModel

FilterViewModel

DataViewModel

FilterPartViewModel

NewValues

FilterPartViewModel

Filters

FilterPartViewModel je jen jeden, ale je pouzity jednou jako Filters a podruhe jako NewValues. Každý FilterPartViewModel ma kolekci public BindableCollection<IFilterElement> Filters { get; set; }

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IFilterElement

IfilterElement je rozhrani pro tridu FilterElement. Ve View je pouzity **MultiSelectComboBox** z Frameworku. Trida FilterElement ho ovlada. Mame tedy ovladac pro každý comboBox jak z Filters, tak z NewValues. Filtrovani zacina kliknutim na comboBox cimz se dostaneme do property :

FilterElement.IsMultiComboBoxOpen a odtamtud do metody GetValues. Zde se rozhodujeme podle toho o jake filtry se jedna a nacitame data do comboBoxu. Data se nacitaji pomoci dotazu do databaze, kde se vyselectuji zadane property na Motivletech. Ve Filters nacitame data pro první combobox všechny data, pro každý další uz jen itemy které odpovidaji i datum z predchozich comboboxu.

V NewValues nacitame vždy všechny data.

**Tlacitko Filtrovat** je povolene když jsou vybrane data v hornich i dolnich filtrech. Z FilterPartViewModelu se posle event, jehož posluchacem je FilterViewModel. Ten vi z kterych filtru event prichazi protože ma instance FilterPartViewModelu

Ma privatni propertu IeventAggregator pomoci které eventy dal m\_eventAggregator.Publish do DataViewModelu. To same plati i pro event zmeny datumu.

**DataViewModel** prijima eventy od aggregatoru , které rozlisuje pomoci parametru metody Handle. DataViewModel ovlada Grid, StatusBar a zobrazeni ErrorInfo v pripade duplicit. Po stlaceni tlacitka Filtrovat sem pride event, a zavola se servisni akce GetDataAction

**GetDataAction**  ziska data z databaze na zaklade Filters, vrati je do kolekce Motivlets. Po vraceni se vola další metoda DataViewModelu, která pomoci triggeru ve DataView podbarvi radky v gridu.

**ComposedWrapper** je wrapper který slouzi pro podbarvovani motivletu (slozenych spolu s verzi a motivem ) v gridu. Funguje to na zaklade bool propert jako napr IsBeforeSelectedDate nebo AllConditionsMet atd.

**GetDataCountAction**  nacita data do statusBaru

Abych se vratil ke comboboxum :

**FilterElement** je genericka trida : public class FilterElement<T> : PropertyChangedBase, IFilterElement where T : ValueDataContract, new()

**ValueDataContract** je abstraktni bazovka pro napr OwnerDataContract, CategoryDataContract atd

Pri vytvareni instance filterElementu se musi urcit genericky argument kterym rikame ze v tomto comboboxu muzou byt itemy jen nejakeho typu (datacontractu).

private void CreateFilters()

{

Filters.Add(new FilterElement<MotiveNameDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.MotiveVersionName, DisableConfirmButton));

Filters.Add(new FilterElement<OwnerDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Owner, DisableConfirmButton));

Filters.Add(new FilterElement<CompanyBrandDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.CompanyBrand, DisableConfirmButton));

Filters.Add(new FilterElement<ProductBrandDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.ProductBrand, DisableConfirmButton));

Filters.Add(new FilterElement<ProductDetailDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.ProductDetail, DisableConfirmButton));

Filters.Add(new FilterElement<CategoryDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Category, DisableConfirmButton));

Filters.Add(new FilterElement<RoleDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Role, DisableConfirmButton));

Filters.Add(new FilterElement<MarketDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Market, DisableConfirmButton));

Filters.Add(new FilterElement<GenderDataContract>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Gender, DisableConfirmButton));

Filters.Add(new FilterElement<**PlatformDataContract**>(m\_publisher, IsNewValuesFilterPart, CreateParameters, ObjectiveTypeValues.Platform, DisableConfirmButton));

}

Na zaklade tohoto generickeho parametru se zaroven v DataViewModelu vola genericky request **GetComboBoxValuesRequest**

public class GetComboBoxValuesRequest<T> : ChangingRequestBase<ListResponse<T>> where T : ValueDataContract

Tento request se vola primo ve FilterElementu, proto vime jaky argument prave pouzivame a potrebujeme . Je to psano obecne => parametr <T> se sem preda prave z argumentu který se sem dostane při samotnem vytvoreni filtru.

ValueComboItems = m\_publisher.Publish(new GetComboBoxValuesRequest<T>(Parameters)).Items.ToList();

Pro ruzne parametry se volaji k jednomu requestu ruzne servisni akce. Toto chovani je zajisteno ve tride CustomServiceActionSubscriber (musi byt zaregistrovany v installeru v castlu).

public void Subscribe()

{

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<MotiveNameDataContract>, ListResponse<MotiveNameDataContract>>(typeof(GetMotiveNamesByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<OwnerDataContract>, ListResponse<OwnerDataContract>>(typeof(GetOwnersByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<CompanyBrandDataContract>, ListResponse<CompanyBrandDataContract>>(typeof(GetCompanyBrandsByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<ProductBrandDataContract>, ListResponse<ProductBrandDataContract>>(typeof(GetProductBrandsByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<ProductDetailDataContract>, ListResponse<ProductDetailDataContract>>(typeof(GetProductDetailsByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<CategoryDataContract>, ListResponse<CategoryDataContract>>(typeof(GetCategoriesByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<RoleDataContract>, ListResponse<RoleDataContract>>(typeof(GetRolesByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<MarketDataContract>, ListResponse<MarketDataContract>>(typeof(GetMarketsByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<GenderDataContract>, ListResponse<GenderDataContract>>(typeof(GetGendersByParametersAction));

ServiceActionSubscriber.Subscribe<GetComboBoxValuesRequest<PlatformDataContract>, ListResponse<PlatformDataContract>>(typeof(GetPlatformsByParametersAction));

}

Viz také poznamky Genericita.docx

**Tlacitko potvrdit**  vyvola dialogove okno a nasledne posle dotaz který zmenuje data. Deje se to asynchronne a po dodani dat se ceka ještě na dotaz který do statusBaru dodava pocty zmenovanych dat.

**Zmenovani** vzhledem k velkemu mnozstvi zmenovanych dat probiha vetsinou na serveru pomoci #tempTabulek. Jen u TempMotiveVersion se z nejakeho duvodu musela udelat tabulka normalni (temp nebrala null)

**ChangeMotivesAction**  Jelikoz byla tato serv. Akce prilis velka rozdelil jsem ji do nekolika mensit trid jejihz metody se tady jen zavolaji na zaklade toho, zda byl selectnuty datum a MotiveName.

Jednotlive tridy a jejich metody pak osetruji jen kod který nastane v těchto pripadech .

Na vytvoreni temp tabulek se pouziva sql kod volany v daoTridach.

Knihovny Teleriku na export do xlsx.

~~Zobrazit v gridu status jako string~~

// ValueComboItems.Add(new T {Name = m\_filter});

**27. listopadu 2018**

~~Pennny Market , PENNY MARKET => sjednotit~~

Klik do gridu a zmenovani jen vybranych

Prazdne mista na statusu (vykreslovani)

Code rewiew

CanUserSelect="False"

IsFilteringAllowed="True"

**21. května 2018**

# Zalozka prejmenovani

Prejmenovani musí probihat pomoci request – response. Vytvorim gen. Request kteremu zaregistruju několik servisnich akci podle parametru. Argumentem bude ValueDatacontract, request vraci tak doneresponse.

**Otazky:**

1. Udelal jsem export do xlsx muzu nechat ? negroupuje
2. Typed factory facility funguje, ale neresi mi to rozhodovaci proces, který request se ma resolvnout
3. DomainUserSource
4. Ideployment poanglictit vterinovy odpocet

# Bug poskozena data

Ahoj , u toho zmenovadla to bylo nahrazovani bez datumu ? Muzes mi prosim presne popsat postup zmenovani ?  Krok za krokem .

[‎5/‎28/‎2018 7:23 AM] Drahomira Tržilová:

Ahoj, verzovala jsem od 1.1.2018. Nahrazovala jsem hodnoty ve 3 krocích.

1. Produkt - blíže neurčeno, Cíl - Blíže neurčeno na Nespecifikováno a Public.
2. Potom produkt N a cíl Blíže neurčeno na cíl Public,
3. Produkt zůstal stejný a poslední jen cíl Blíže neurčeno na Public.

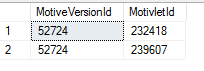
**Na motivu 228604 je jen jedna verze :**

select \* from media.MotiveVersion where MotiveId = 228604

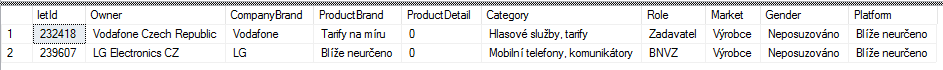
****

**Na jeho versne jsou dva motivlety : 232418 a 239607:**

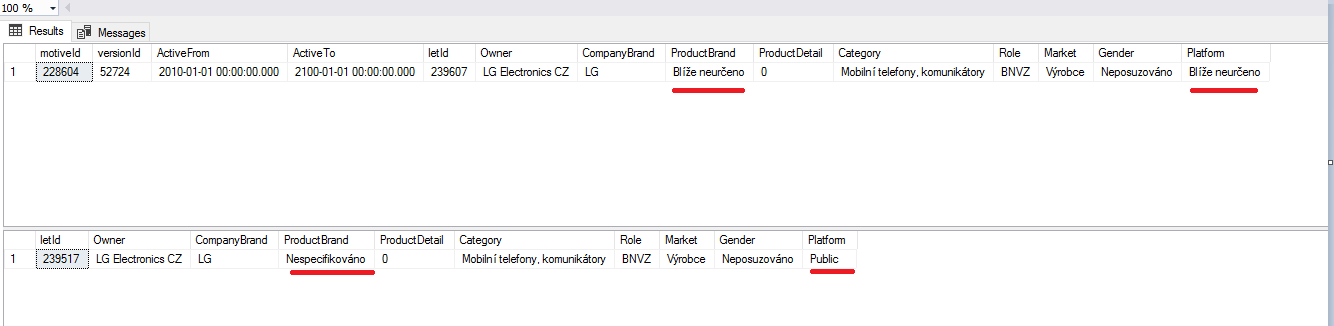
select \* from Media.MotiveVersionToMotivlet where MotiveVersionId = 52724

****

**V prvni zmene se meni ProductBrand Blize neurceno. Meni se tedy jen motivlet 239607:**

****

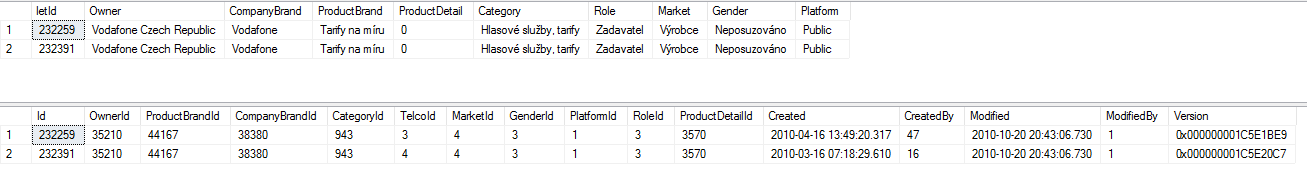
**Motivletu 239607 se dotkla jen první změna a spravne se zmenil na 239517 :**

****

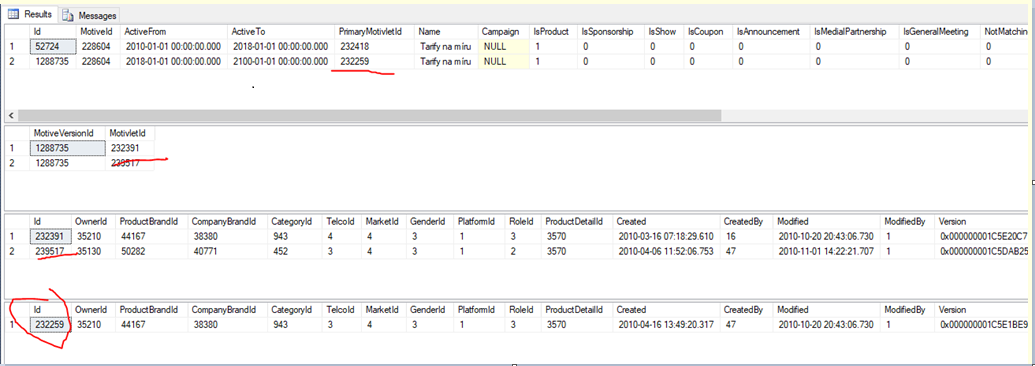
**Druhou zmenou tyto motivlety dotceny nebyly protože se menil ProductBrand N .**

**Ve treti zmene se menil jen cil . Blize neurceno na Public . Tato změna se dotkla motivletu 232418 který se zmenil spravne na 232391**

**Motivlety 232259 a 232391 se lisi jen v TelcoId podle ktereho pravdepodobne neporovnavam, coz je chyba a opravim ji. :**

****

**Z těchto dvou skoro stejnych motivletu se teda nahodne vybral 232391 jako motivlet který se pomoci vazby vlozil na versnu ale jako primarni motivlet se urcil 232259. :**

****

**Oprava :**

1. **Do hledani spravneho motivletu zahrnout i TelcoId,**
2. **Nove primaryId se musí nastavit pro cely motiv, nejen pro část za selectnutym datumem.**

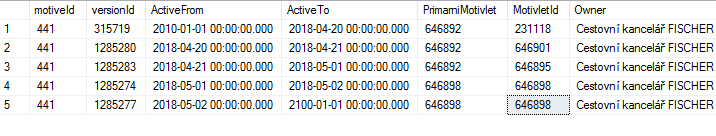
**Změna MotiveName + date**

Metoda UpdateRestricteredPrimaryIds() meni primaryMotivletId jen v radcich kde je primarni motivlet . :

query.Where(d => d.Ver.PrimaryMotivletId == d.Mtm.MotivletId).Set(d => d.Ver.PrimaryMotivletId, x => x.Mot.Id).Update();

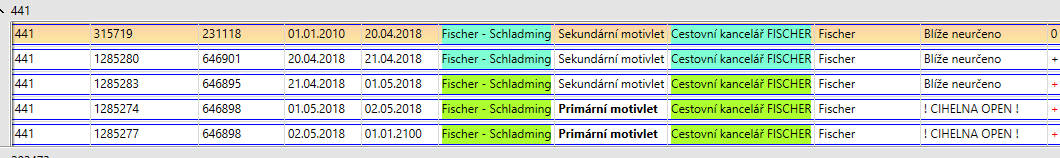
Tim ale nastava situace kde verze které neobsahuji primar mají spatne primaryMotivletId:

Spatne

****

Spravne

V ramci jednoho motivu musí byt jen jedno primaryId at je verzi kolik chce .



**4. června 2018**

**Zalozka prejmenovani**

1. Filtrovani na zalozce Changing se projevi i v gridu na zalozce Prejmenovani
2. Na zalozce prejmenovani se v comboBoxu na stlaceni sipky dolu , rovnou vybira selectedItem a v setteru rovnou filtruju. Chtelo by to poresit anebo pridat tlacitko Filtrovat.
3. Zobrazovat stare Ico
4. Nejde mi publishnout BGProduction
5. Upravil jsem FrameworkHg deploymentService. Jak vydat balicek ? V Shellu uz je MediaResearch.Gui nareferencovane proc se u mne neprojevi zmeny ? Framework jsem zbuildil

**Vydani balicku:**

1. Ve FrameworkHg.src je batka packAllPackages a PushAllPackages
2. Spustime první a vybereme jen to z ceho chceme udelat balicek
3. Spustime druhou batku

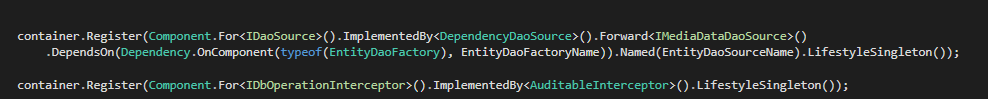
**AuditableProvider**

1. Jak podle tohoto kodu BLToolkit vi, ze se ma podivat do tabulky Security.UserType ? Je to podle ciziho klice v databazi ? Musi byt tabulka UserType namapovana ?

[Association(ThisKey = "UserTypeId", OtherKey = "Id", CanBeNull = false)]

public UserType UserType { get; set; }

1. Proc mame Security.User a Membership.User ?
2. BLToolkit DataAccess => jake interceptory se nevolaji , jak najdu interceptor na serveru ?
3. Jakto, ze když resolvnu IMediaDataDaoSource, Castle zna a resolvne mi všechny daoTridy ?

****

1. Musí se updatnout Modified na motivu když updatuju Owner.Name ?

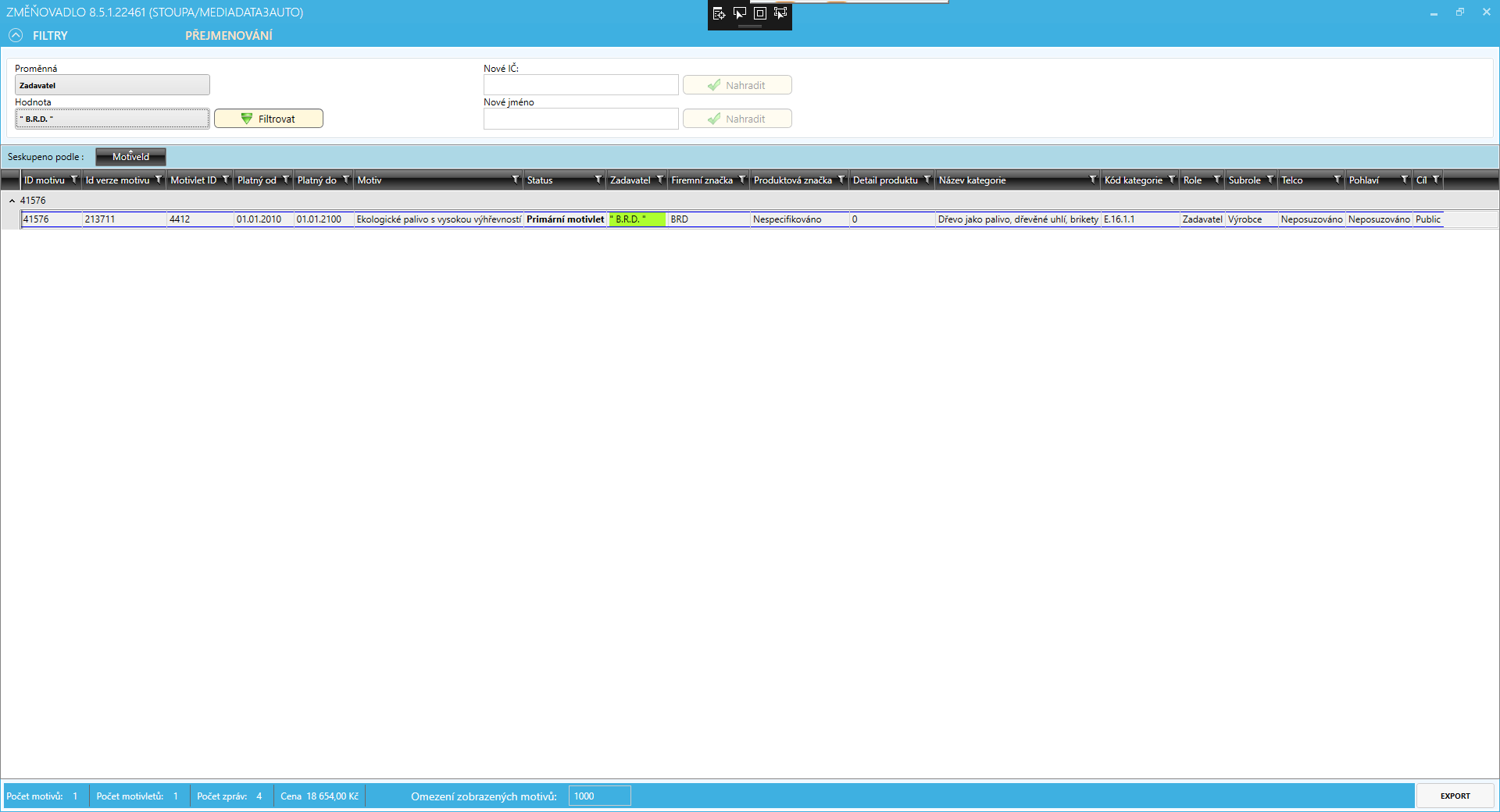
Při kazde uprave entity se zvedne sloupec version(timestamp).

timestamp is a synonym for rowversion

**Aby fungovaly interceptory na auditable, musim mit entitu k dispozici v programu. Pokud menim entitu na serveru, musim auditni stopu zadat explicitne .**

**System.Resources.MissingManifestResourceException:** 'Nelze najít žádné prostředky, které by byly vhodné pro uvedenou (nebo neutrální) jazykovou verzi. Zkontrolujte, zda během kompilace byl objekt Mediaresearch.Framework.Gui.Deploy.DeployResources.resources do sestavení MIR.Media.Changing2.Shell.CZ.Test správně vložen či propojen nebo zda všechna vyžadovaná satelitní sestavení je možné načíst a jsou plně podepsána.'

Zalozka Prejmenovani:



# Změňovadlo - Nezaznamenává auditní stopu 50190 TFS

Včera upravovala Dráža ve Změňovadle nějaké motivy a v tabulce [MediaData3].[History].[MotiveVersion] je záznam, že to změnil userId = 9 místo 2.  Upoutalo mě to v kodovadle protože Laďka ( userID 9) není v práci.

Vypadá to, že pokud je pole ModifiedBy prázdné, tak se při zápisu aktualizuje, pokud už je vyplněné tak se ponechá jeho hodnota.

Příklad: Motiveversion ID:

1273376 – nezměněno, zkopírováno userid 9

273182 – změněno, protože bylo NULL

Dělá to zřejmě pouze změňovadlo, kodovadlo se chová korektně.

/\*\*\*\*\*\* Script for SelectTopNRows command from SSMS \*\*\*\*\*\*/

SELECT TOP (1000)[HistoryCode]

,[HistoryDate]

,[HistoryVersion]

,[HistoryBy]

,[Id]

,[MotiveId]

,[Name]

,[Campaign]

,[Created]

,[CreatedBy]

,[Modified]

,[ModifiedBy]

FROM [MediaData3].[History].[MotiveVersion] where Id = 273182

SELECT TOP (1000)[HistoryCode]

,[HistoryDate]

,[HistoryVersion]

,[HistoryBy]

,[Id]

,[MotiveId]

,[Name]

,[Campaign]

,[Created]

,[CreatedBy]

,[Modified]

,[ModifiedBy]

FROM [MediaData3].[History].[MotiveVersion] where Id = 1273376

