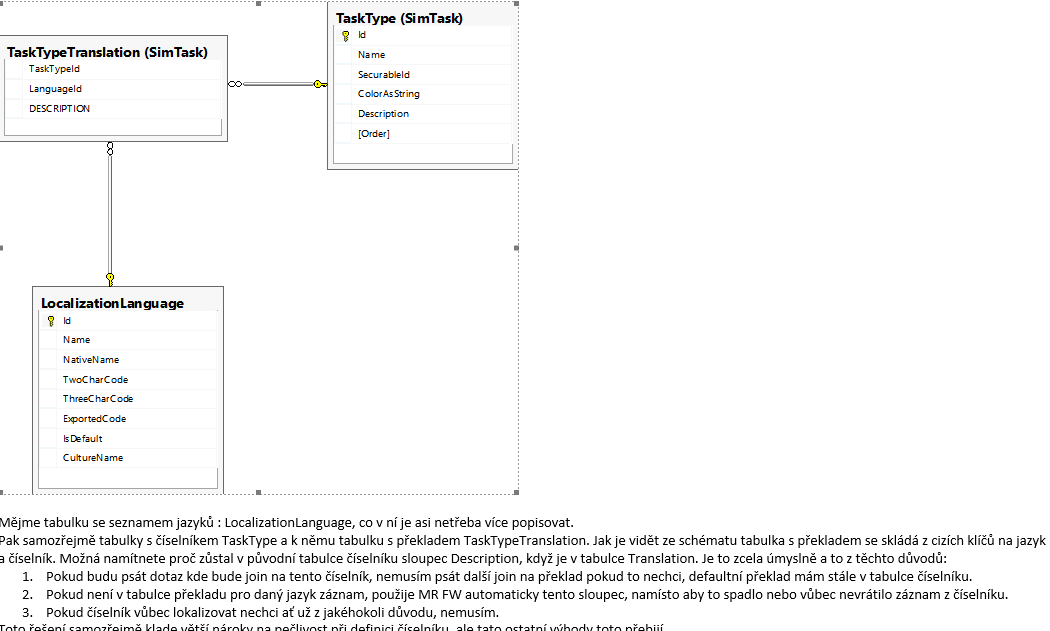
Ciselnikem se rozumi jakysi vycet hodnot v tabulce v databazi. Ciselnik musí byt osazen atributem EnumTable

* Rekneme , ze potrebujeme prelozit sloupec v nektere tabulce do vice jazyku. To provedeme pomoci tzv. ciselniku. Mitroz na mail v outlooku posilal: Konečné řešení lokalizační otázky číselníků - dobre precist.
* Vytvorime **ciselnik** (uplne normalni vazebni tabulka ktera slouzi jen k prekladani )
* 

* V databazi musi pribyt nova tabulka PublicationStatusTranslation . Vznikne spustenim scriptu napr. pro lokalizaci PartName tabulky Ve scanovadle byl potreba nasledujici script :

**Na vytvoreni tabulky LocalizationLanguage:** C:\Pool\Admosphere\data\DatabaseModel\PrintStorage.Database\PrintStorage.SchemaVersion\_021

**Na vytvoreni tabulky PartNameTranslation:**  C:\Pool\Admosphere\data\DatabaseModel\PrintStorage.Database\PrintStorage.SchemaVersion\_021

* Musi se dodrzet stabni kultura nazvu teda pro tabulku PartName vznikne PartNameTranslation.
* V tabulce PartName vytvorime sloupec Description : ten se pouziva jako default kdyz neni vic jazyku (viz mail Mitroz)

GO

ALTER TABLE dbo.PartName ADD [Description] NVARCHAR(4000) NULL

GO

BEGIN TRANSACTION

UPDATE dbo.PartName SET Description = Name

COMMIT TRANSACTION

**Mapovani entit**

* Tabulku PartName je potreba **namapovat** abychom mohli v daotridach volat prez tabulky dotazy. Mapovane entity jsou samozrejme ve slozce Entities, stejne jako jejich dao tridy. Jejich dataContracty jsou ve slozce Entites/MIR.Common2.PrintStorage.DataContracts.
* Kazda namapovana tabulka se musi pridat do dataModelu . V tomto pripade je to trida**: PrintStorageDataModel**

public Table<PartName> PartName => GetTable<PartName>();

* **Dao** tridy jsou ve sl. Entities pod PrintStorage\Dao\Dbo
* A **mapovaci tridy** entit : pod PrintStorage\Dbo
* Nad mapovanou tabulku je potreba pridat **BLToolkit znacky** . :

namespace MIR.Entities.PrintStorage.Dbo

{

**=> [TableName]**

**=> [LocalizedEnumTable(typeof(LocalizedPartName))] => definuje , ze se jedna o tabulku s lokalizaci**

public class PartName : **LightDatabaseEntityByteKey<PartName>, IEnumerableTableWithDescriptionAndOrder<byte>**

**trida dedi podle toho jestli ma identitu a inkrementaci , byte je v tomto pripade proto, ze primarni klic tabulky je typu tinyint coz je byte (8 bitu)**

{

public string Name { get; set; }

public bool IsSheet { get; set; }

public bool IsCover { get; set; }

public bool PictureMatching { get; set; }

**tady neni [mapIgnore]**

public string Description { get; set; }

[MapIgnore]

public byte IdValue => Id;

[MapIgnore] **=> definuje , ze sloupec neni v tabulce PartName prezto je namapovany**

public string NameValue => Name;

[MapIgnore]

public byte OrderValue { get; set; }

[MapIgnore]

public string DescriptionValue

{

get => Description;

set => Description = value;

}

public class LocalizedPartName : DefaultNamingConventionEnumTableMappingByteKey<PartName>

**definuje , ze se ma hledat preklad v tabulce ktera se jmenuje jako tato a na konci ma Translation napr PartName a PartNameTranslation.**

{

}

}

* Po namapovani a pridani do tridy modelu by mela byt mapovana trida viditelna na modelu v dao tride .
* Preklad dostaneme tak ze si normalni cestou vytahneme datacontract ktery potrebujeme a potom zmenime sloupec ktery se ma prelozit pomoci volani metody **GetValueId( nejakeId ). DescriptionValue.**

Predpoklada se uz vytvoreny sloupec description v databazi. :D

protected override ListResponse<PublicationDataContract> ExecuteInternal(GetPublicationsRequest request)

{

var items = m\_publicationDao.GetPublications(request.AdvertisedFrom, request.AdvertisedTo, request.MediumId, request.PublicationStatusId , request.ProcessorId,request.Comment);

foreach (var item in items)

{

item.Status = m\_publicationStatuts.**GetValueById(item.PublicationStatusId).DescriptionValue;**

**item.Status je ten prekladany sloupec.Timto dostameme prelozeny sloupec . V metode vlastne predame id radku na ktere lezi slovo ktere chceme prekladat , a vrati se jeho descriptionValue. To je v pripade ,ze je vice jazyku hodnota sloupce description v tabulce necoTranslation.**

}

return new ListResponse<PublicationDataContract>(items);

}

* Mitroz mi opravil mapovani takto: Puvodne jsem pomoci dotazu v MediumDao chtel MediumDataContract ktery mel data ze tri tabulek a jeste je tridil pomoci logiky v metode v MediumDao. Ta byla prilis dlouha a neprehledna . Mitroz namapoval entitu Medium ktere pridal navic Vlastnost MediumPublicationPeriodicity a MediumScanningPresset coz jsou ve skutecnosti dve tabulky v databazi . Takto mame v jedine namapovane tride to , co by jinak bylo ve trech .

[TableName]

[DaoFactory(DaoType = typeof(MediumDao))]

public class Medium : LightDatabaseEntityShortKey<Medium>, IAuditable

{

[Nullable]

public string OriginalId { get; set; }

public string Name { get; set; }

[Nullable]

public string Comment { get; set; }

[Nullable]

public byte? ProcessorId { get; set; }

[Nullable]

public short? PageWidth { get; set; }

[Nullable]

public short? PageHeight { get; set; }

public DateTime Created { get; set; }

public byte CreatedBy { get; set; }

public DateTime? Modified { get; set; }

public byte? ModifiedBy { get; set; }

[Association(CanBeNull = true, ThisKey = nameof(Id), OtherKey = "MediumId")]

public MediumPublicationPeriodicity Periodicity { get; set; }

[Association(CanBeNull = true, ThisKey = nameof(Id), OtherKey = "MediumId")]

public MediumScanningPreset ScanningPreset { get; set; }

}

**Postup pri prekladu nove tabulky:**

* **Pomoci scripctu vytvorime v databazi tabulku LocalizationLanguage**

CREATE TABLE [dbo].[LocalizationLanguage](

[Id] [tinyint] NOT NULL,

[Name] [nvarchar](4000) NOT NULL,

[NativeName] [nvarchar](4000) NOT NULL,

[TwoCharCode] [nchar](2) NOT NULL,

[ThreeCharCode] [nchar](3) NOT NULL,

[ExportedCode] [nvarchar](100) NOT NULL,

[IsDefault] [bit] NOT NULL CONSTRAINT [DF\_LocalizationLanguage\_IsDefault] DEFAULT ((0)),

[CultureName] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_LocalizationLanguage] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

* **Vytvorime celou, pripadne upravime o sloupec Description tabulku ktera se bude prekladat**

CREATE TABLE dbo.PublicationStatusTranslation

(

PublicationStatusId TINYINT NOT NULL CONSTRAINT FK\_PublicationStatusTranslation\_PublicationStatusId FOREIGN KEY REFERENCES dbo.PublicationStatus(Id),

LanguageId TINYINT NOT NULL CONSTRAINT FK\_PublicationStatusTranslation\_LanguageId FOREIGN KEY REFERENCES dbo.LocalizationLanguage(Id),

[Description] NVARCHAR(4000) NOT NULL

)

GO

ALTER TABLE [dbo].[PublicationStatusTranslation] ADD CONSTRAINT [PK\_PublicationStatusTranslation] PRIMARY KEY CLUSTERED

(

[PublicationStatusId] ASC,

[LanguageId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

GO

BEGIN TRAN

INSERT INTO dbo.LocalizationLanguage(Id, Name, NativeName,TwoCharCode,ThreeCharCode, ExportedCode, IsDefault, CultureName)

VALUES (1, N'English' , N'English' , N'en' , N'eng', 'ENG', 1, 'en')

INSERT INTO dbo.LocalizationLanguage(Id, Name, NativeName,TwoCharCode,ThreeCharCode, ExportedCode, IsDefault, CultureName)

VALUES (2, N'Czech' , N'Čeština' , N'cs' , N'cze', 'CZ', 0, 'cs-CZ')

INSERT INTO dbo.PublicationStatusTranslation( PublicationStatusId ,LanguageId ,Description)

VALUES (1 , 1 , N'Released' )

INSERT INTO dbo.PublicationStatusTranslation( PublicationStatusId ,LanguageId ,Description)

VALUES (2 , 1 , N'Delivered' )

COMMIT TRAN

**Samostatne pridani sloupce Decription vypada takhle:**

GO

ALTER TABLE dbo.PartName ADD [Description] NVARCHAR(4000) NULL

BEGIN TRANSACTION

UPDATE dbo.PartName SET Description = Name // toto rika ze hodnoty ve sloupci description se vezmou ze sloupce Name.

COMMIT TRANSACTION

* Ted se musi namapovat tabulka , nastavit ji hlavicku , podedit EntityDaoFactory , implementovat rozhrani a pridat vnitrni trida. :

[TableName]

[LocalizedEnumTable(typeof(LocalizedPartName))]

public class PartName : LightDatabaseEntityByteKey<PartName>, IEnumerableTableWithDescriptionAndOrder<byte>

sloupce ktere nejsou skutecne v tabulce jsou mapIgnore

[MapIgnore]

public string DescriptionValue

{

get => Description;

set => Description = value;

}

public class LocalizedPartName : DefaultNamingConventionEnumTableMappingByteKey<PartName>

{

}

* **Ciselniky se musí v servisni akci dostat z EnumTable :**

public GetPublicationPartsAction(IPartDao partDao , **IEnumTableValues<PartName, byte > partNames**) // byte proto , ze primarni klic tabulky je tinyint (8 bits)

{

m\_partDao = partDao;

m\_partNames = partNames;

}

protected override ListResponse<PartDataContract> ExecuteInternal(GetPublicationPartsRequest request)

{

var items = m\_partDao.GetParsByPublicationId(request.PublicationId);

// Preklad PartName

foreach (var item in items)

{

item.PartName = **m\_partNames**.GetValueById( item.NameId).DescriptionValue;

}

return new ListResponse<PartDataContract>(items);

}

Navod :

====================================================================================

Aby se prekladal sloupec PeriodicityType v tabulce **MediumPublicationPeriodicity** musime vytvorit pro tento sloupec ciselnik. Tabulka MediumPublicationPeriodicity je jen halda bez primarniho klice.

* Vytvorime teda novou tabulku  **PeriodicityType** ktera bude mit : Id (PK) , Name (Nvarchar) a Description (Nvarchar)

|  |  |  |
| --- | --- | --- |
| Id | Name | Description |
| 1 | Daily | Daily |
| 2 | Weekly | Weekly |
| 3 | Monthly | Monthly |

Slouperc Description je default kdyz nechceme lokalizovat vezme se hodnota z tohoto sloupce.

* Pomoci scriptu (vzdy pomoci scriptu aby byly scripty ulozene pro nove pouziti ) vytvorime tabulku se stejnym **nazvem + Translation** na konci.

Tady teda **PeriodicityTypeTranslation** ktera bude mit :

PeriodicityTypeId (PK, FK, tinyint , not null )

LocalizationLanguageId (PK, FK, tinyint , not null )

Description (nvarchar(4000), not null )

|  |  |  |
| --- | --- | --- |
| PeriodicityTypeId | LocalizationLanguageId | Description |
| 1 | 1 | Daily |
| 1 | 2 | Denne |
| 2 | 1 | Weekly |
| 2 | 2 | Tydne |
| 3 | 1 | Monthly |
| 3 | 2 | Mesicne |

* Novou tabulku PeriodicityType **namapujeme .** Dulezite jsou zvyraznene věci:

namespace MIR.Entities.PrintStorage.Dbo

{

[TableName]

**[LocalizedEnumTable(typeof(LocalizedPeriodicityType))]**

public class PeriodicityType : LightDatabaseEntityByteKey<PeriodicityType>, **IEnumerableTableWithDescription<byte>**

{

public string Name { get; set; }

public string Description { get; set; }

//from interface:

[MapIgnore]

public string DescriptionValue

{

get => Description;

set => Description = value;

}

**[MapIgnore]**

public byte IdValue => Id;

[MapIgnore]

public string NameValue => Name;

**public class LocalizedPeriodicityType : DefaultNamingConventionEnumTableMappingByteKey<PeriodicityType>**

**{**

}

}

}

* V servisni akci si zavolame nejakou metodu na tride MediumDao , ta nam vrati nejaky List<MediumDataContract> a predtim nez v servisni akci vratime ListResponse< List<MediumDataContract>> , prelozime nekterou vlastnost datacontractu pomoci metody

GetValueById ( IdPrekladanehoRadku).Description :

public class GetMediumTabItemsAction : ScanningActionBase<GetMediumTabItemsRequest, ListResponse<MediumDataContract>>

{

private readonly IMediumDao m\_mediumDao;

private readonly IEnumTableValues<PeriodicityType, byte> m\_periodicityTypes;

private readonly IMapper<Medium, MediumDataContract> m\_mediumMapper;

public GetMediumTabItemsAction(IMediumDao mediumDao, IEnumTableValues<PeriodicityType, byte> **periodicityTypes**, //NA TETO TABULCE SE VOLA GetValueById().De

IMapper<Medium, MediumDataContract> mediumMapper)

{

m\_mediumDao = mediumDao;

m\_periodicityTypes = periodicityTypes;

m\_mediumMapper = mediumMapper;

}

protected override ListResponse<MediumDataContract> ExecuteInternal(GetMediumTabItemsRequest request)

{

var items = m\_mediumDao.GetMediumTabItems();

List<MediumDataContract> result = new List<MediumDataContract>();

foreach (var medium in items)

{

var dataContract = m\_mediumMapper.Map(medium); // TADY PROBEHNE MAPOVANI POMOCI MAPPERU (do media se pridaji jeste dve mapovane tabulky

PublicationPeriodicity (na ktere je PeriodicityType vlastnost ) a ScanningPresset

if (medium.Periodicity != null)

{

dataContract.Periodicity.PeriodicityType = **m\_periodicityTypes.**GetValueById(medium.Periodicity.PeriodicityType).DescriptionValue;

//TOHLE JE TAK DIVNE PROTO, ZE PERIODICITY JE PRIDANE NA MEDIUM POMOCI JOINU VYGENEROVANEHO BLTOOLKITEM NA ZAKLADE ATRIBUTU ASSOCIATION

//

// [Association(CanBeNull = true, ThisKey = nameof(Id), OtherKey = "MediumId")]

// public MediumPublicationPeriodicity Periodicity { get; set; }

//

// JINAK BY TAM BYLO NECO JAKO : item.PartName = m\_partNames.GetValueById( item.NameId).DescriptionValue;

}

result.Add(dataContract);

}

return new ListResponse<MediumDataContract>(result);

}

}

Konec navodu

================================================================================================================