!!!! Важно!!!!

1. Генерация ключей.

Для генерации ключей при создании новых объектов использовать сервис последовательных гуидов. (Если в качестве первичного ключа таблицы, который к тому же является кластерным индексом, выбрано поле типа данных UNIQUEIDENTIFIER, то заполняться оно должно значениями, сгенерированными по последовательному алгоритму. Для этого есть сервис IGuidGenerator. На клиентской стороне также есть реализация данного сервиса (класс SequenceGuidSource). Она, в свою очередь, обращается к серверной реализации).

2. Работа с БД

Если вы хотите выполнять запросы к БД с клиента из фоновых потоков, то необходимо

Переписать (сделать новый) используемый вами серверный метод на использование метода MakeDBTask, иначе у нас будут копиться зависшие соединения.

Пример серверного метода для вызова с клиента:

public FooItem[] GetFooItems(Guid fooUid)

{

var result = new FooItem[0];

//Вариант с аксессором

Task task = MakeDbTask<FooDataAccessor>((acc) =>

{

result = acc.GetFooItems(fooUid);

});

//Вариант без аксессора

Task task = MakeDbTask((db) =>

{

result = (from fi in db.GetTable<FooItem> where fi.FooUid == fooUid select fi).ToArray();

});

task.Start();

task.Wait();

return result.ToArray();

}

Если серверный метод вызван из основного потока, но вы хотите распараллелить запросы на сервере, то поведение примерно следующее

public FooEntity GetFooEntity(Guid fooUid)

{

var result = new FooEntity();

List<Task> tasks = new List<Task>();

tasks.Add(MakeDbTask(db => result.FooItems = GetFooItems(db, fooUid)));

tasks.Add(MakeDbTask<FooDataAccessor>(acc => result.FooNodes = GetFooNodes(acc, fooUid)));

tasks.ForEach(t => t.Start());

Task.WaitAll(tasks.ToArray());

return result;

}

Внутри методов GetFooItems и GetFooNodes необходим использовать только переданные в них экземпляры DbManager и Аксессора, а не из использовать экземпляры из базового класса сервиса.

# Изменения в БД

Предполагаемая схема изменений в БД. Новые таблицы.

# 

Таблица ClbrGraphWS – Рабочие растворы для ГХ

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrGraphWSUid | uniqueidentifier, PK, Not null | Первичный ключ |
| ClbrGraphUid | uniqueidentifier, Not null, FK ClbrGraph | Ссылка на градуировочную характеристику |
| AEUid | uniqueidentifier , Not null, FK AE | Ссылка на ОУ, используемый для градуировки |
| StdDocUid | uniqueidentifier , null, FK STDDoc | Ссылка на НД аналита используемый при приготовлении ОУ (не обязательно) |
| CompositionUid | uniqueidentifier , null, FK Composition | Ссылка на рецептуру. Не обязательна |
| Name | Nvarchar(150), Not null | Наименование раствора (по умолчанию – наименование ОУ) |
| Flags | int, Not null | Битовая маска для хранения настроек (например, используется ли приготовление) |

Таблица ClbrGraphWSCompBatch – партии, используемые для приготовления рабочих растворов

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrGraphWSCompBatchUid | uniqueidentifier PK, Not null | Первичный ключ |
| ClbrGraphWSUid | uniqueidentifier , Not null, FK ClbrGraphWS | Ссылка на рабочий раствор |
| AEBatchUid | uniqueidentifier , Not null, FK AEBatch | Ссылка на партию ОУ |
| DimsId | int, null, FK Dims | габариты для списания ( не обязательны) |

Таблица ClbrGraphSeries – шаблон серии для испытаний

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrGraphSeriesUid | uniqueidentifier, PK, Not null | Первичный ключ |
| ClbrGraphUid | uniqueidentifier, Not null, FK ClbrGraph | Ссылка на градуировочную характеристику |
| Name | nvarchar(150), Not null | Имя серии |
| SeriesType | int, Not null | Тип серии |
| ClbrGraphWSUid | uniqueidentifier, Null, FK ClbrGraphWS | Ссылка на рабочий раствор, не обязательна |

Таблица ClbGraphSeriesTest- точки градуировки шаблона серии испытаний

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrGraphSeriesTestUid | uniqueidentifier PK, Not null | Первичный ключ |
| ClbrGraphSeriesUid | uniqueidentifier , Not null, FK ClbrGraphSeries | Ссылка на шаблон серии испытаний |
| TechTestUid | uniqueidentifier, Not null | ИД показателя методики для испытания. По идее дублирует данные из таблицы ClbrGraph, но оставлю тут во избежание лишних джоинов |
| Value | float ,Not null | Значение |
| AbsErr | float, null | Погрешность |

Таблица ClbrTaskSeries – серия испытаний

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrTaskSeriesUid | uniqueidentifier PK, Not null | Первичный ключ |
| ClbrGraphSeriesUid | uniqueidentifier , Not null, FK ClbrGraphSeries | Ссылка на шаблон серии испытаний |

Таблица ClbrTaskSeriesBatch – расход по партиям для серии испытаний

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrTaskSeriesBatch | uniqueidentifier PK, Not null | Первичный ключ |
| ClbrTaskSeriesUid | uniqueidentifier , Not null, FK ClbrTaskSeries | Ссылка на серию испытаний |
| AEBatchUid | uniqueidentifier , Not null, FK AEBatch | Ссылка на партию ОУ |
| DimsId | int, Not null, FK Dims | габариты для расхода |

Таблица ClbrTaskSeriesTest – точка градуировки для серии испытаний

|  |  |  |
| --- | --- | --- |
| Имя колонки | Тип | Описание |
| ClbrTaskSeriesTestUid | uniqueidentifier PK, Not null | Первичный ключ |
| ClbrTaskSeriesUid | uniqueidentifier , Not null, FK ClbrTaskSeries | Ссылка на серию испытаний |
| ClbrGraphSeriesTestUid | uniqueidentifier , Not null, FK ClbrGraphSeriesTest | Ссылка на точку градуировки шаблона серии сипытаний |
| Value | float ,Not null | Значение |
| AbsErr | float, null | Погрешность |

Изменения в таблице ClbrTask:

Поле ChkSTUid сделать не обязательным.

Добавить поле ClbrTaskSeriesTestUid, uniqueidentifier, null – ссылка на точку градуировки для серии испытаний.

(обсуждаемо) Добавить флаг определяющий типа задания: Образец контроля или ОУ, чтобы не анализировать поля ChkSTUid и ClbrTaskAESeriesTestUid.

# 2. Серверная часть

Библиотеки: Indusoft.LDS.Services.Contracts.Calibration, Indusoft.LDS.Server.Services.Calibration

Сервис: ICalibrationDataService

Обвязать новые таблицы сущностями. Реализовать серверные методы и тесты.

Переработка серверной части

Подготовка серверной части.

Сервер - Indusoft.LDS.Server.Services.Calibration

1. Убрать явно написанные SQL запросы на выборку из кода. При возможности перевести на LINQ , при невозможности – вынести в табличные функции/хранимые процедуры. При переносе в хранимые процедуры/функции стараться избегать большой вариативности параметров (когда на вход функции приходит много параметров, часть из которых может быть пустая, и внутри рулится логика). Лучше сделать несколько процедур/функций.
2. По возможности распараллелить запросы, когда, например, последовательно получаются наборы не связанных между собой сущностей.
3. Постепенно отказаться от выборки в таблицы (DataTable) и перевести клиент на работу с массивами объектов. Они отлично подключаются к гридам на клиенте

# 3. Клиентская часть

Библиотека: Indusoft.LDS.Client.Calibration

Основные классы:

Feature\UI\ CalibrationGraphView.cs – основная форма экземпляра возможности

Dialogs\ ClbrGraphEditDialogForm.cs – Диалог настройки ГХ

Dialogs\ ClbrTaskEditDialog.cs – Диалог настройки задания на калибровку.

Доступ к объектам учета:

Описание модели данных - Indusoft.LDS.Services.Contracts.AE

Сервис - IAEDataService

Переработка клиентского модуля

Переработка модуля данных

Имеющийся класс SA обвязать интерфейсом ICalibrationDataManager (интерфейс надо создать). Необходимо постараться убрать вызовы типа SA.Instance.[DataService].[Метод], заменить их на вызовы методов интерфейса. Экземпляр интерфейса инжектить через конструктор (на будущее).

Переработка представления экземпляра возможности

Feature\UI\ CalibrationGraphView.cs – основная форма экземпляра возможности

1. Отделить представление от реализации. Убрать логику из обработчиков событий, выделить контроллер и представление (возможно разбить на несколько представлений, обсудить).
2. Переработать источники данных для таблиц (GridView, TreeList) в представлении. Отказаться от использования класса DataTable, перейти на использование массива объектов. Переработать методы, связанные с наборами данных.

Прикладные задачи

1. Метод \_bGraphAddClick – Событие по добавлению ГХ. Вынести код создания из обработчика отдельный метод.
2. Переработать диалог ClbrGraphEditDialogForm.
   1. Отвязать реализацию от представления
   2. Добавить закладки, отображаемые в зависимости от типа данных для ГХ (ОУ, ОК)
   3. Переделать валидацию.
   4. Переработать редактирование: убрать публичные свойства у диалога, типа InTech, OutTech. Передавать объект и получать объект.
3. Метод \_bTaskAddClick – Событие по добавлению задания на калибровку. Вынести код из события в отдельный метод, добавить проверки, разделить методы
4. Переработать диалог ClbrTaskEditDialog
   1. Отвязать реализацию от представления
   2. Переработать (сделать новую) закладку для выбора показателей анализа на основе серии
   3. Добавить закладку для определения количества списания. Списание выполнять при закрытии диалога с помощью метода в AEDataService (дописать: найти метод или разработать новый)
   4. Переделать валидацию
5. Регистрация образцы по заданию ГГ. Indusoft.LDS.Client.AnalysisInputControl\Controls\Model\ SampleInfo.cs . Методы:
   1. public SampleInfo(SampleType sampleType, Task taskInfo, DateTime cDate, TaskContent[] taskContents) – констркутор
   2. private bool PrepareTechniqueList(SampleContext sRow) раздел для калибровки, получение ИД показателей по содержимому задания .ClbrManager.GetTechTestByContentUid(taskContent.ContentUid) Проверить работоспособность.
   3. private void AddCalibrationAnalysis(SampleContext sRow, Status newStatusRow, Guid techUid, TaskContent[] taskContents) Собственно само создание анализа. Переработать метод на предмет получения значений для показателей не из образца контроля, а из настроек серии ОУ для ГХ.