

Импорт справочника тех. позиций в ПО

- [Основные сведения](#)
 - [Входные данные](#)
 - [Выходные данные](#)
 - [Ограничения и предложения](#)
 - [Соглашение об именовании](#)
- [2. Контракты API Сервиса Импорта](#)
 - [Входной контракт](#)
 - [Выходные контракты](#)
 - [Выходной контракт валидации файла Справочника](#)
 - [Выходной контракт импорта Справочника](#)
 - [Структура сообщения для Apache Kafka](#)
- [3. Физическая модель данных](#)
- [4. Алгоритм импорта Справочника](#)
 - [4.1. Инициация алгоритма импорта](#)
 - [4.2. Графический пользовательский интерфейс для запуска импорта файла Справочника](#)
 - [Макет дизайна](#)
 - [Требования к графическому пользовательскому интерфейсу для запуска импорта файла Справочника](#)
 - [4.3. Алгоритм валидации файла Справочника](#)
 - [4.4. Алгоритм импорта Справочника](#)
- [5. Логирование импорта](#)
 - [5.1. Уровни логирования](#)
 - [5.2. Формат сообщений](#)
 - [5.3. Место хранения логов](#)
 - [5.4. Объём логирования](#)
 - [5.5. Безопасность](#)
- [Приложения](#)
 - [Приложение А. Редактируемые файлы PlantUML](#)
 - [Приложение Б. Список связанных задач](#)

Примечание о правках

Ниже в тексте утратившие силу части постановки обозначены ~~зачёркнутым шрифтом~~.

Новое в постановке обозначено шрифтом [синего цвета](#).

Основные сведения

В настоящей статье описаны требования к сервису импорта (далее - Сервиса) Справочника тех. позиций (далее - Справочника). Сервис должен позволять:

- сотруднику группы поддержки ПО импортировать посредством графического пользовательского интерфейса файл Справочника в формате *.XLSX;
- загружать содержимое этого файла в базу данных;
- обновлять витрины отчётов.

Входные данные

На вход Сервиса поступают:

- Справочник в виде файла электронной таблицы Excel в формате .XLSX (актуальный шаблон см. [Ведение справочника тех. позиций в ПО](#), приложение Ж). [Файл справочника может содержать как данные только одного Общества Группы, так и данные сразу множества разных Обществ Группы.](#);
- содержимое текстового поля "Причина импорта*" из GUI страницы импорта Справочника;
- ФИО пользователя, который производит импорт Справочника.

Выходные данные

Результатом работы Сервиса являются:

- добавление содержимого файла Excel в базу данных,
- обновление витрин данных для отчётов,
- загрузка файла в BlobStore,
- добавление записи в таблицу Журнала импорта в БД.

Ограничения и предложения

Файл Справочника имеет стандартные для Excel-файла ограничения. Ниже приведены некоторые из них, имеющие непосредственное отношение к работе Сервиса:





- Количество листов в файле - ограничено объёмом оперативной памяти, по умолчанию - 1 лист.
- Количество символов в ячейке — 32 767 знаков.

Помимо этого существуют следующие ограничения:

- Запрещается наличие дублей по столбцу "UUID тех.позиции" в файле Справочника;
- Допустимая дата, записанная в столбец "Дата действия" в Справочнике - не ранее, чем текущая дата минус 365 дней.

Соглашение об именовании

Далее мы исходим из того, что будут использованы следующие наименования сущностей:

Наименование сущности	Тип сущности	Описание
techposes-dictionary-import- контур*	 kafka Топик Apache Kafka	Топик Apache Kafka, в который Сервис импорта Справочника осуществляет запись сообщения
контур*_reports_data	 База данных	База данных для хранения витрин данных отчётов
dm_techposes_dictionary_historical	 таблица БД	Таблица витрины данных для отчёта "Изменения Справочника тех. позиций"
import_log	 таблица БД	Таблица для хранения данных Журнала импорта Справочника

Примечание

контур* в наименовании сущностей может принимать одно из следующих значений:

- test
- dev
- demo
- master
- prod (**TBD**)

2. Контракты API Сервиса Импорта

В настоящем разделе приведены входной и выходные контракты Сервиса.

Входной контракт

```
// Входной контракт сервиса импорта файла Справочника
{
  "file" : {
    "type" : "file",
```

```

        "contentType" : "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
        "description" : "импортируемый файл Справочника"
    },
    "reasonForImport" : {
        "type" : "string",
        "minLength" : 10,
        "maxLength" : 200,
        "description" : "Причина импорта"
    },
    "userName" : {
        "type" : "string",
        "description" : "ФИО пользователя, производящего импорт Справочника"
    },
    "required" : ["file", "reasonForImport"]
}

```

Выходные контракты

Выходной контракт валидации файла Справочника

```

// Выходной контракт валидации файла Справочника
{
    "validationStatus" : {
        "type" : "string",
        "value" : "fail"
    },
    "errorCode" : {
        "type" : "integer",
        "example" : 1
    },
    "errorMessage" : {
        "type" : "string",
        "minLength" : 10,
        "example" : "Обнаружены дубли по столбцу UUID тех. позиции"
    },
    "required" : ["validationStatus", "errorCode", "errorMessage"]
}

```

Выходной контракт импорта Справочника

```

// Выходной контракт импорта файла Справочника
{
    "importResult" : {
        "type" : "string",
        "description" : "Результат импорта файла Справочника"
        "value" : "success"
    },
    "importMessage" : {
        "type" : "string",
        "minLength" : 10,
        "description" : "Дополнительное сообщение о результате импорта.",
        "example" : "Обнаружены uuid тех. позиций, отсутствующие в Модели
АКТИВОВ"
    },
    "required" : ["importResult", "importMessage"]
}

```

Структура сообщения для Apache Kafka

Сервис импорта Справочника отправляет сообщение в формате JSON в топик **techposes-dictionary-import-контур*** Apache Kafka согласно следующего контракта:

```

//Выходной контракт импорта Справочника - сообщение в Apache kafka
{
    "importStatus" : {"type" : "string", "value" : "success"},
    "object_id" : {"type" : "string", "description" : "идентификатор (ключ) файла в хранилище BlobStore (C3)"},
}

```

```

    "import_filename" : {"type" : "string", "description" : "Наименование файла
Справочника"},
    "file_content" : {
        "type" : "object",
        "properties" : {
            "created" : {
                "type" : "array",
                "description" : "Перечень добавленных в Справочник тех.
позиций - содержимое списка lstCreated",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "uuid" : {"type" : "string",
                                "description" : "uuid тех. позиции"},
                        "og_title" : {"type" : "string",
                                    "description" : "Наименование ОГ"},
                        "og_code" : {"type" : "string",
                                    "description" : "Код ОГ"},
                        "object_title" : {"type" : "string",
                                        "description" : "Наименование объекта"},
                        "object_code" : {"type" : "string",
                                        "description" : "Код объекта"},
                        "object_part" : {"type" : "string",
                                        "description" : "часть объекта"},
                        "location" : {"type" : "string",
                                    "description" : "Код положения для тех.позиции (LOCATION)"},
                        "rnmposition" : {"type" : "string",
                                        "description" : "Обозначение тех. позиции на схеме (RNMPOSITION)"},
                        "description" : {"type" : "string",
                                        "description" : "Описание тех.позиции"},
                        "date_of_action" : {"type" : "string",
                                            "description" : "Дата изменения", "format" : "date-time"}
                    }
                },
                "updated" : {
                    "type" : "array",
                    "description" : "Перечень тех. позиций Справочника, в
которых внесены изменения - содержимое списка lstUpdated",
                    // далее - аналогично блоку "added_techposes"
                },
                "deleted" : {
                    "type" : "array",
                    "description" : "Перечень удаляемых из Справочника тех.
позиций - содержимое списка lstDeleted",
                    // далее - аналогично блоку "added_techposes"
                }
            }
        },
        "import_result_added" : {"type" : "integer", "description" : "Количество добавленных
тех. позиций"},
        "import_result_changed" : {"type" : "integer", "description" : "Количество изменённых
тех. позиций"},
        "import_result_deleted" : {"type" : "integer", "description" : "Количество удалённых
тех. позиций"},
        "import_result_total" : {"type" : "integer", "description" : "Общее количество тех.
позиций в Справочнике"},
        "import_log_file" : {"type" : "string", "description" : "Наименование файла лога
импорта Справочника"},
        "log_object_id" : {"type" : "string", "description" : "object_id файла лога импорта Справочника"},
        "required" : ["importStatus", "batchID", "import_filename", "file_content",
"import_result_added", "import_result_changed", "import_result_deleted", "import_result_total",
"import_log_file"]
    }
}

```

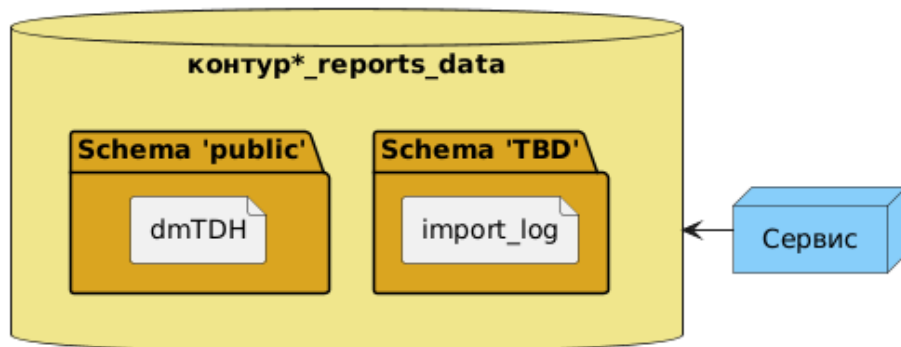
Примечание

Далее ниже по тексту выходные атрибуты выходного контракта будут отображены в тексте **зелёным цветом шрифта жирного начертания**.

3. Физическая модель данных

Взаимосвязь между Сервисом и базами данных представлена на схеме ниже:


Взаимосвязь Сервиса с базами данных



Условные обозначения таблиц БД:
dmTHD - dm_techposes_dectionary_historical

Ниже представлены структуры таблицы dmTDH:

Таблица dm_techposes_dictionary_historical

Наименование столбца в БД	Заголовок в файле Справочника	Тип данных	Комментарий
og_title	Наименование ОГ	текст	
og_id	Код ОГ	текст	Источником данных является бизнес-справочник <u>Маппинг ОГ Справочника тех. позиций</u> , мэппинг производится по столбцу og_title
 UUID	UID тех.позиции	текст	
object_title	Наименование объекта	текст	
object_code	Код объекта	текст	
object_part	Часть объекта	текст	
location	Код положения для тех.позиции (LOCATION)	текст	

rnmposition	Обозначение тех. позиции на схеме (RNMPOSITION)	текст	
description	Описание тех.позиции	текст	
action	Правка	текст	
date_of_action	Дата правки	дата (ДД.ММ.ГГГГ)	
import_reason	Основание изменений	текст	
upload_timestamp	Дата импорта	дата+время	

Ниже представлена структура таблицы Журнала импорта Справочника:

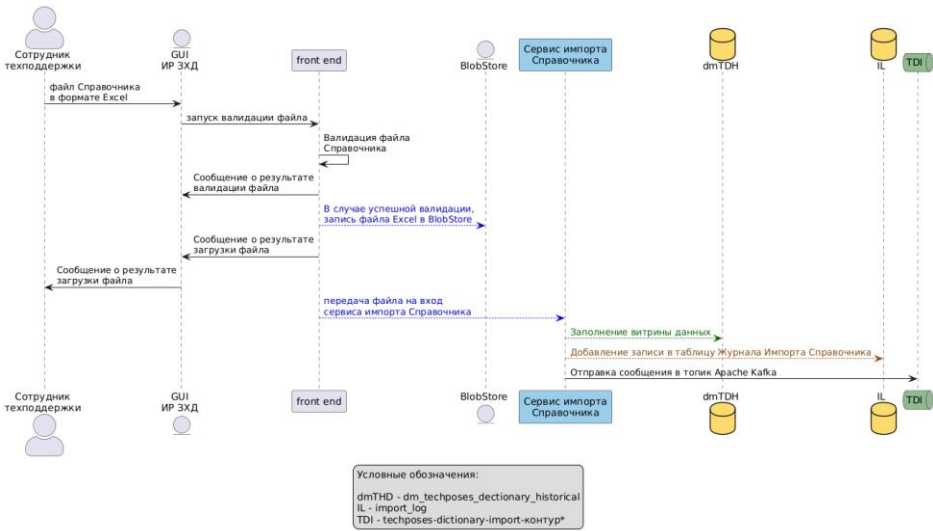
Таблица import_log

Наименование столбца в БД	Заголовок в файле Справочника	Тип данных	
imported_filename	Наименование исходного файла	текст	
object_id	id импортированного в BlobStore файла Справочника	текст	
og_title	Наименование ОГ	текст	
og_id	Код ОГ	текст	Источником данных является бизнес-справочник <u>Маппинг ОГ Справочника тех. позиций</u> , мэппинг производится по столбцу og_title
import_status	Статус	текст	
import_began_at	Начало	дата (ДД.ММ.ГГГГ чч:мм:сс)	
import_end_at	Завершение	дата (ДД.ММ.ГГГГ чч:мм:сс)	

username	ФИО пользователя	текст	
user_id	Идентификатор пользователя	текст	
import_reason	Причина импорта	текст	
import_result_added	Добавлено	целое число	
import_result_deleted	Удалено	целое число	
import_result_changed	Изменено	целое число	
import_result_total	Всего тех. позиций	целое число	
import_log_filename	Наименование файла лога импорта	текст	
import_log_object_id	id импортированного в BlobStore файла логирования	текст	

4. Алгоритм импорта Справочника

Общая схема импорта Справочника представлена на следующем рисунке:



Укрупнённо алгоритм импорта состоит из следующих шагов:

1. валидация Excel файла Справочника. После успешной валидации, файл сохраняется в BlobStore;

2. непосредственно импорт содержимого файла в БД после успешной валидации -
заполнение витрин данных, добавление записи в таблицу Журнала импорта
Справочника;
3. отправка сообщения в топик Apache Kafka с изменениями Справочника - то есть с
Добавленными, Изменёнными и Удалёнными тех. позициями;
4. вывод информации в GUI об успешном импорте Справочника.

4.1. Инициация алгоритма импорта

В ПО должна быть возможность выбора одного файла Справочника в формате Excel (*.XLSX) для последующего импорта.

4.2. Графический пользовательский интерфейс для запуска импорта файла Справочника

Макет дизайна

Здесь картинка и ссылка на макет.

Требования к графическому пользовательскому интерфейсу для запуска импорта файла Справочника

Инициирование запуска импорта Excel файла Справочника должно быть доступно из блока **"Импорт справочника тех. позиций"**.

Должна быть возможность **развернуть/свернуть** блок **"Импорт справочника тех. позиций"**.

Алгоритм обработки клика кнопки **"Выбрать файл"**:

- Должно открываться стандартное диалоговое окно выбора файла для загрузки. При этом должна быть возможность выбора **только одного файла в формате *.XLSX**.
 - При подтверждении выбора файла, его название должно отобразиться в блоке **"Импорт справочника тех. позиций"**.
Справа от названия файла должна быть размещена маленькая круглая кнопка удаления файла из перечня загружаемых (*).

Алгоритм обработки клика кнопки **"Загрузить"**:

- В случае **отсутствия значения** в текстовом поле **"Причина импорта*"** или если **длина содержимого текстового поля "Причина импорта*" < 10** кнопка **"Загрузить"** должна быть не кликабельна, т.к. указанное поле является обязательным для заполнения.
- Запуск **алгоритма валидации** выбранного Excel файла справочника.
 - **В случае неуспешной валидации Excel файла Справочника:**
 - Возникает ошибка **"Файл Справочника не валиден"** и выводится сообщение об ошибке в формате: **"[Номер ошибки] [Иконка ошибки] [Текстовое сообщение об ошибке]"**
 - Выйти из алгоритма.

- В случае успешной валидации Excel файла Справочника производится выполнение алгоритма импорта файла Справочника.
 - По результату импорта файла **вывести сообщение** в блоке "Импорт справочника тех. позиций":
 - импорт прошёл успешно: "[Наименование файла] загружен [иконка успешной загрузки]";
 - импорт прошёл не успешно: "[Наименование файла] не загружен [иконка ошибочной загрузки]";
 - добавить запись о результате импорта в [Журнал импорта Справочника](#).

4.3. Алгоритм валидации файла Справочника

Алгоритм валидации включает в себя следующие шаги:

Проверить структуру первого листа файла: удостовериться, что первый лист импортируемого файла содержит все необходимые заголовки столбцов в первой строке, согласно их перечня (см. раздел 1 настоящей статьи). Порядок следования столбцов при этом может отличаться, но не их количество и текстовое содержимое заголовков (с учётом регистра). В случае обнаружения несоответствия структуры заголовков столбцов выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла Справочника
{
  "validationStatus" : "fail",
  "errorCode" : "1",
  "errorMessage" : "формат содержимого файла не соответствует утверждённому шаблону
Справочника"
}
```

Проверить по всем строкам данных **наличие пустых ячеек по столбцу UUID** и в случае, если обнаружены пустые ячейки по столбцу uuid выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла Справочника
{
  "validationStatus" : "fail",
  "errorCode" : "2",
  "errorMessage" : "пропуски данных в обязательном для заполнения столбце 'UUID тех.
позиции'"
}
```

Проверить по всем строкам данных **наличие пустых ячеек по столбцу Код ОГ** и в случае, если обнаружены пустые ячейки по столбцу og_title выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла Справочника
{
  "validationStatus" : "fail",
  "errorCode" : "3",
  "errorMessage" : "пропуски данных в обязательном для заполнения столбце 'код ОГ'"
}
```

Проверить по всем строкам данных **наличие дубликатов по столбцу UUID** и в случае, если обнаружены дубликаты по столбцу UUID выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла справочника
{
    "validationStatus" : "fail",
    "errorCode" : "4",
    "errorMessage" : "обнаружены дубли в столбце 'UUID тех. позиции'"
}
```

Проверить по всем строкам данных **наличие пустых ячеек по столбцу location** и в случае, если обнаружены пустые ячейки по столбцу location выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла справочника
{
    "validationStatus" : "fail",
    "errorCode" : "5",
    "errorMessage" : "пропуски данных в обязательном для заполнения столбце 'Код положения для тех.позиции (LOCATION)'"
}
```

Проверить по всем строкам данных **наличие пустых ячеек по столбцу rnmposition** и в случае, если обнаружены пустые ячейки по столбцу rnmposition выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла справочника
{
    "validationStatus" : "fail",
    "errorCode" : "6",
    "errorMessage" : "пропуски данных в обязательном для заполнения столбце 'Обозначение тех. позиции на схеме (RNMPOSITION)'"
}
```

Проверить по всем строкам данных **все ли наименования Обществ Группы (ОГ) находятся в бизнес-справочнике ogTitlesForCT** и в случае, если обнаружены наименования, отсутствующие в справочнике, выйти из алгоритма с контрактом:

```
// Выходной контракт валидации файла справочника
{
    "validationStatus" : "fail",
    "errorCode" : "7",
    "errorMessage" : "обнаружены не существующие в ПО наименования ОГ"
}
```

В случае, если все проверки пройдены успешно - осуществить запись данных в атрибуты:

import_filename = наименование файла Справочника;

import_begin_date = дата и время в формате дд.мм.гггг чч.мм.сс;

importer_username = имя пользователя, осуществляющего импорт Справочника;

import_reason = значение поля "Причина импорта*".

Перейти к п. 4.4.

4.4. Алгоритм импорта Справочника

После успешной валидации (см. п. 3.3) алгоритм импорта справочника в БД состоит из следующих шагов:

1. **Создать файл логирования**, формат *.txt. Наименование файла должно быть составлено из наименования импортируемого файла и суффикса "_log".
2. Осуществить запись в лог-файл:

TIMESTAMP [INFO] запуск импорта справочника тех. позиций...

3. Зафиксировать текущий TIMESTAMP в переменную **import_began_at**.
4. Присвоить значение 0(ноль) выходным атрибутам **import_result_added, import_result_changed, import_result_deleted, import_result_total**.
5. Создать три пустых списка для хранения тех. позиций: **lstCreated, lstUpdated, lstDeleted** (используются для составления сообщения для Apache Kafka, см. выходной контракт).
6. Присвоить значение false переменной **hasSomeWarnings**.
7. Загрузить Excel файл Справочника в **BlobStore**, получив ссылку на него и метку времени записи файла в хранилище. Ссылку записать в параметр **object_id**;
8. Осуществить запись в лог-файл:

TIMESTAMP [INFO] файл Справочника загружен в BlobStore. object_id
= [object_id]

9. Далее мы принимаем соглашение, что в случае возникновения любых фатальных ошибок в ходе обработки таблиц базы данных, будут выполнены следующие шаги алгоритма:

1. Осуществить запись в лог-файл:

TIMESTAMP [FATAL] Критическая ошибка произошла при обработке записи Справочника для [UUID]. Импорт прерван.

2. Добавить запись в таблицу Журнала импорта Справочника. Код приведён ниже:

код работы с таблицей БД

```
-- производим добавление данных в таблицу Журнала импорта Справочника
INSERT INTO import_log (imported_filename, og_id, og_title, object_id,
import_status, import_began_at, import_end_at, username, import_reason,
import_result_added, import_result_deleted, import_result_changed,
import_result_total, import_log_filename, import_log_object_id)
VALUES (import_filename, og_id, og_title, object_id, 'fatal', import_began_at,
import_end_date, importer_username, import_reason, import_result_added,
import_result_changed, import_result_deleted, import_result_total,
import_log_filename, log_object_id);
```

3. Завершить работу алгоритма с выходным контрактом:

```
// Выходной контракт импорта файла Справочника
{
    "importResult" : "fail",
    "importMessage" : "Ошибка обработки содержимого файла Справочника при записи в
базу данных"
}
```

10. Для каждой строки данных в файле Справочника:

1. Если содержимое ячейки в столбце **"Дата изменения"** <

дата(import_began_at) минус 365 дней:

1. осуществить запись в файл лога:

TIMESTAMP [WARNING] "дата изменения тех. позиции " + [UUID]
+ " меньше минимально допустимой для Справочника тех. позиций. "

2. Присвоить переменной **hasSomeWarnings** значение true.

3. перейти к следующей строке данных.

2. Если содержимое ячейки в столбце **"Дата изменения"** = **пустая строка**
или null:

1. осуществить запись в файл лога:

TIMESTAMP [WARNING] для тех. позиции [UUID] не указана дата
изменения.

2. Присвоить переменной **hasSomeWarnings** значение true.

3. перейти к следующей строке данных.

3. Если **UUID найден** в таблице **dm_techposes_dictionary_historical** :

1. Если записанная ранее в

таблицу **dm_techposes_dictionary_historical** дата в

столбце date_of_action для данного UUID больше той, что находится
в ячейке в столбце **"Действие"** импортируемого файла:

1. Осуществить запись в лог-файл:

TIMESTAMP [WARNING] для тех. позиции [UUID] в файле
указана дата меньше той, что записана в базу данных.

2. Присвоить переменной **hasSomeWarnings** значение true.

3. Перейти к следующей строке данных.

2. Если содержимое ячейки в столбце **"Действие"** = **Добавить:**

1. осуществить запись в лог:

TIMESTAMP [WARNING] попытка повторного добавления тех. позиции [UUID], имеющейся в базе данных.

2. Присвоить переменной **hasSomeWarnings** значение true.
3. Перейти к следующей строке данных в файле.
3. Если содержимое ячейки в столбце "Действие" = Удалить:
 1. осуществить инкрементацию на 1 выходного атрибута **import_result_deleted**;
 2. осуществить запись значений строки данных из файла в качестве нового элемента списка **lstCreated**;
 3. добавить запись для обрабатываемой строки данных по тех. позиции в таблицу **dm_techposes_dictionary_historical**, при этом `upload_timestamp = import_began_at`, `object_id = object_id`;
 4. осуществить запись в файл лога:

TIMESTAMP [DEBUG] удалена тех. позиция [UUID]

4. Если содержимое ячейки в столбце "Действие" = Изменить:
 1. инкрементировать на 1 значение атрибута **import_result_changed**
 2. осуществить обновление записи данных в таблице **dm_techposes_dictionary_historical** для данного UUID;
 3. осуществить запись значений строки данных из файла в качестве нового элемента списка **lstChanged**;
 4. осуществить запись в файл лога:

TIMESTAMP [DEBUG] изменена тех. позиция [UUID]

5. инкрементировать на 1 значение атрибута **import_result_total**.
6. Перейти к следующей строке данных.
4. Если **UUID не найден** в таблице **dm_techposes_dictionary_historical**:
 1. Если содержимое ячейки в столбце "Действие" = Добавить:
 1. инкрементировать на 1 значение атрибута **import_result_added**, перейти к следующей строке данных.
 2. **добавить запись в таблицу dm_techposes_dictionary_historical** для данного UUID, при этом `upload_timestamp = import_began_at`, `object_id = object_id`;

3. осуществить запись значений строки данных из файла в качестве нового элемента списка **lstCreated**.
4. инкрементировать на 1 значение атрибута **import_result_total**.
5. осуществить запись в файл лога:

TIMESTAMP [DEBUG] добавлена тех. позиция [UUID]

6. перейти к следующей строке данных.
2. Если содержимое ячейки в столбце "**Действие**" = **Изменить** или "**Действие**" = **Удалить**:
 1. Осуществить запись в лог:

TIMESTAMP [WARNING] Невозможно внести запись о действии над тех. позицией [UUID] - она не найдена в базе данных.

2. Присвоить переменной **hasSomeWarnings** значение true.
3. Перейти к следующей строке данных в файле Справочника.
3. Если содержимое ячейки в столбце "**Действие**" = **пустой строке** или **null**:
 1. Осуществить запись в лог:

TIMESTAMP [WARNING] отсутствует значение в столбце 'Действие' для тех. позиции [UUID]

2. Перейти к следующей строке данных в файле Справочника.
11. Зафиксировать дату и время окончания импорта: **import_end_date** = TIMESTAMP.
12. Осуществить запись в лог-файл:

TIMESTAMP [INFO] процедура импорта Справочника завершалась **import_end_date**. Импортировано: **import_result_total** тех. позиций, из них добавлено: **import_result_added**, изменено: **import_result_changed**, удалено: **import_result_deleted**.

13. Загрузить файл лога в BlobStore, получив его id. Наименование файла лога записать в параметр **import_log_filename**, а **import_log_object_id** файла лога импорта записать в параметр **log_object_id**.
14. Осуществить запись в таблицу Журнала импорта Справочника. Код представлен ниже:
Если значение переменной **hasSomeWarnings** = false, то использовать код:

код работы с таблицей БД

-- производим добавление данных в таблицу Журнала импорта Справочника


```
INSERT INTO import_log (imported_filename, og_id, og_title, object_id,
import_status, import_began_at, import_end_at, username, import_reason,
import_result_added, import_result_deleted, import_result_changed,
import_result_total, import_log_filename, import_log_object_id)
VALUES (import_filename, og_id, og_title, object_id, 'success',
import_began_at, import_end_date, importer_username, import_reason,
import_result_added, import_result_changed, import_result_deleted,
import_result_total, import_log_filename, log_object_id);
```

В противном случае использовать код:

код работы с таблицей БД

```
-- производим добавление данных в таблицу Журнала импорта Справочника
INSERT INTO import_log (imported_filename, og_id, og_title, object_id,
import_status, import_began_at, import_end_at, username, import_reason,
import_result_added, import_result_deleted, import_result_changed,
import_result_total, import_log_filename, import_log_object_id)
VALUES (import_filename, og_id, og_title, object_id, 'warnings',
import_began_at, import_end_date, importer_username, import_reason,
import_result_added, import_result_changed, import_result_deleted,
import_result_total, import_log_filename, log_object_id);
```

15. Отправить сообщение в топик **techposes-dictionary-import-контур*** согласно структуры выходного контракта (см. раздел Выходные контракты), используя подготовленные поля данных и заполненные списки **lstCreated**, **lstUpdated**, **lstDeleted**.
16. Если значение переменной **hasSomeWarnings** = false. то завершить работу алгоритма с выходным контрактом:

```
// Выходной контракт импорта файла Справочника
{
    "importResult" : "success",
    "importMessage" : ""
}
```

17. Если значение переменной **hasSomeWarnings** = true. то завершить работу алгоритма с выходным контрактом:

```
// Выходной контракт импорта файла Справочника
{
    "importResult" : "warnings",
    "importMessage" : ""
}
```

5. Логирование импорта

Основной задачей логирования в работе Сервиса является отслеживание хода выполнения алгоритма и отладка ошибок.

5.1. Уровни логирования

Мы принимаем, что уровни логирования (LEVEL) в работе Сервиса включают в себя следующий перечень:

- **DEBUG:** Используется для детальной отладки.
- **INFO:** Информация о нормальной работе системы.
- **WARNING:** Предупреждение о потенциальных проблемах.
- **ERROR:** Ошибки, требующие внимания.
- **FATAL:** Критические ошибки, ведущие к остановке системы.

5.2. Формат сообщений

Мы принимаем следующий формат сообщений в лог-файлах импорта Справочника:

TIMESTAMP [LEVEL] MESSAGE

5.3. Место хранения логов

Файл логирования импорта Справочника должен храниться в BlobStore.

5.4. Объём логирования

Каждый файл логирования включает в себя информацию о логировании одного прохода работы алгоритма импорта Справочника и потому ограничен количеством строк в файле Excel Справочника.

5.5. Безопасность

Просмотр файлов логирования импорта Справочника доступен для сотрудников с ролью Специалист технической поддержки.

Приложения

Приложение А. Редактируемые файлы PlantUML

- [алгоритм импорта - укрупненно версия 3.txt](#)
- [схема взаимосвязи сервиса и баз данных 3.txt](#)

Приложение Б. Список связанных задач