Рады познакомить Вас с задачником «DATAREON Platform. Специалист»!

Это пособие содержит практические задания, которые помогут подготовиться к сертификационному экзамену «DATAREON Platform. Специалист» и углубить знания о функциональных возможностях платформы.

Структура заданий:

Задания разделены на два типа:

· Интеграционные задачи — для отработки навыков создания и настройки процессов обмена данными между системами.

· Задачи по устранению ошибок — для развития умения анализировать и исправлять неполадки в работе платформы.

Особенности заданий:

Задания в пособии разработаны с упрощениями, чтобы сосредоточиться на ключевых механизмах платформы:

· Структура объектов (поля, типы данных, связи) упрощена для удобства выполнения.

· Необходимо реализовать тот сценарий, что описан в задаче. Выполнение дополнительных настроек приветствуется, но не является обязательным

Выполнить интеграционную задачу

Задача включает в себя интеграционный сценарий, который необходимо выполнить на DATAREON Platform.

Ограничений на использование возможностей платформы – нет. Будет приветствоваться использование различной функциональности для решения задач.

Системы – список внешних систем (типов коннекторов), используемых для выполнения задачи. Источники и получатели данных.

Задача – это тема задачи, включающая общее и поверхностное описание.

Описание задачи – это подробно расписанные условия задачи, включающие проблематику, описание взаимодействия систем друг с другом, название и описание объектов, используемых в рамках конкретной задачи.

Требования – это описание систем и структур. Их выполнение обязательно.

Дополнительные указания – это рекомендации, которые могут помочь решить задачу и убедиться в правильности выполнения и не являются обязательными.

Для успешного выполнения задачи необходимо настроить интеграционный сценарий, соблюдая требования и условия задачи.

Задание №1. Организация передачи данных из системы 1С в систему PostgreSQL с учётом ограничений и особенностей обеих систем

Описание задачи

Системы: 1С, PostgreSQL

Задача: Организовать передачу данных из системы 1С в систему PostgreSQL с учётом ограничений и особенностей обеих систем.

Описание задачи:

Система 1С отправляет записи документа «ИзменениеОстатковТоваров» по каждому изменению отдельным сообщением, а PostgreSQL принимает их только пакетами. Ограничений по объему и количеству записей пакета нет.

У системы PostgreSQL есть ограниченное технологическое окно для загрузки данных. Период доступности технологического окна - каждые 2 часа на 15 минут. Если за данный период все изменения отправить не удалось, то их необходимо оставить на хранение до следующего технологического окна и переотправить частями.

В одном пакете не может быть дублей - иначе процесс загрузки упадет, и пакет отправится в Хранилище сообщений для дальнейшего просмотра содержимого дублирующихся записей. Дубль записи определяется по полю: Номер документа, Дата изменения.

Необходимо накапливать изменения, поступающие из 1С, формировать пакеты данных, исключая дублирующиеся записи, и передавать их в PostgreSQL в период доступности технологического окна.

Требования и условия

1. Источник данных: 1С:

Документ «ИзменениеОстатковТовара» содержит информацию содержит информацию об актуальных остатках товара на складе и передается в формате Json. Объект содержит следующие поля:

Наименование поля	Тип	Описание
Номер документа	String [9, Trim]	Уникальный номер, по которому определяется дубль записи
Проведен	Boolean	Параметр проведения документа
Идентификатор товара	GUID	Идентификатор товара, по которому выполнена операция
Количество	Int	Актуальное количество товара
Склад	GUID	Идентификатор склада хранения товара
Дата изменения	DateTime	Дата выполнения операции. Также используется для выявления дубликата

Тип операции	Перечисление	Тип выполненной операции: • Поступление товара • Перемещение товара • Продажа товара • Возврат товара

2. Получатель PostgreSQL

На стороне получателя информация хранится в таблице **remaining_inventory** со следующими столбцами:

Наименование полей	Тип	Описание
Идентификатор товара	GUID	Идентификатор товара, по которому фиксируется изменение
Наименование	String[255]	Наименование товара
Количество	Int	Количество товаров
Склад	String[100]	Наименование склада
Дата изменения	DateTime	Дата произведения изменения по товару
Тип операции	String[50]	Наименование операции по изменению товаров

Критерии успешного выполнения задачи

Задание считается выполненным, если соблюдены все критерии выполнения.

1. Данные из 1С должны накапливаться в промежуточном хранилище до открытия технологического окна.
2. Перед передачей в PostgreSQL данные должны объединяться в пакеты.
3. В одном пакете не должно быть дублирующихся записей.
4. Если при формировании или загрузке пакета обнаружены дублирующиеся записи, загрузка такого пакета прерывается, а пакет целиком направляется в Хранилище сообщений.

Дополнительные указания

Механизмы Platform и возможные паттерны для решения задачи:

1. Реализация задачи однократной доставки.
2. Слияние нескольких сообщений в одно.
3. Использование Банка данных для обработки передаваемой информации .

Задание №1. Реализация процесса передачи данных о сотрудниках с учетом требований безопасности

Описание задачи

Системы: 1С, Email, MSSQL, REST API

Задача: Реализовать процесс передачи данных о сотрудниках между системами с учётом требований безопасности, избежать повторных проверок одних и тех же данных, снизить нагрузку на службу безопасности и обеспечить корректную маршрутизацию данных.

Описание задачи:

Есть система безопасности (MSSQL), которая контролирует передачу данных в основную систему (Email). Не все данные требуют проверки: часть передаётся автоматически, а часть — только после акцепта (разрешения) со стороны службы безопасности. Служба безопасности не хочет повторно проверять одни и те же данные. Им важны только чувствительные данные — определённые типы и поля, которые могут измениться. Если изменяются нечувствительные данные, объект не должен попадать на акцепт, а решение о его передаче должно приниматься на основе последнего акцепта. Заблокированные данные передаются в хранилище сообщений для анализа. Источниками данных являются системы 1С и Rest API.

Под чувствительными данными подразумеваются данные авторизации - логин и пароль.

Требования и условия

1. Источник данных №1: Корпоративная система управления пользователями (расширенное REST API)

Пакет "УчетнаяЗаписьСотрудника" содержит чувствительные данные **(логин и пароль**) для учетной записи нового сотрудника в корпоративной системе и передается в формате Json. Структура пакета содержит информацию:

Наименование поля	Тип	Описание
Идентификатор сотрудника	GUID	Уникальный идентификатор сотрудника
Email	String [100]	Рабочий электронный адрес сотрудника
Логин	String [50]	Логин сотрудника в системе
Пароль	String [50]	Пароль сотрудника в системе

По полю «Идентификатор сотрудника» происходит сопоставление записей из источников REST API и 1С.

Пример пакета для отправки из REST API:

{

"Идентификатор": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"Email": "ivanov@datareon.ru",

"Логин": "ivanovI",

"Пароль": "Start123"

}

2. Источник данных №2: Система управление персоналом (1С):

Документ «ПереводСотрудника» :

Содержит общую информацию об изменении отдела сотрудника и передается в формате Json. Структура пакета содержит информацию:

Наименование поля	Тип	Описание
Идентификатор сотрудника	GUID	Уникальный идентификатор сотрудника
ФИО сотрудника	String [150]	Фамилия, имя, отчество сотрудника
Старый отдел сотрудника	String [100]	Название отдела, в котором состоит сотрудник
Новый отдел сотрудника	String [100]	Название нового отдела, в который переводят сотрудника
Дата перевода	DateTime	Дата проведения документа о переводе

По полю «Идентификатор сотрудника» происходит сопоставление записей из источников REST API и 1С.

3. Система безопасности: MSSQL.

Структура таблицы в системе содержит следующие поля:

Наименование поля	Тип	Описание
Идентификатор сотрудника	GUID	Уникальный идентификатор сотрудника
Логин сотрудника	String [50]	Логин сотрудника в системе
Пароль сотрудника	String [50]	Пароль сотрудника в системе
Подтверждение	Int	Определяет статус прохождения акцепта: `NULL` (пустое) – запись ожидает проверки службой безопасности; `1` – акцепт пройден, запись готова к выгрузке в Platform; `2` – акцепт не пройден
Флаг	Int	Является триггером для выгрузки записи в Platform. `NULL` или `0` – запись ждет акцепта; при `accept = 1` данные выгружаются, а Флаг становится `1` (выгрузка выполнена)

4. На почту сотрудника (email) должно отправляться письмо, содержащее поля:

Наименование поля	Система-источник
ФИО сотрудника	1С

Логин	Расширенное REST API
Пароль	Расширенное REST API
Отдел	1С

Пример письма, которое отправляется на email сотрудника:

Тема: *Кадровое изменение: Перевод в новый отдел*

Тело письма:

Здравствуйте, Иванов Иван Иванович!

В связи с Вашим переводом направляем актуальные данные для входа в корпоративный портал:

Логин: ivanov.ii

Пароль: ds8Dfhj4

Отдел: Финансовый департамент

Критерии успешного выполнения задания

Задание считается выполненным, если соблюдены все критерии выполнения.

1. Чувствительная информация должна проходить обязательный акцепт (проверку/подтверждение) в системе безопасности (MSSQL).

2. Банк данных должен хранить результаты акцепта и актуальные данные для быстрого доступа. Требований к структуре хранения данных в Банке нет – допускается произвольная реализация, обеспечивающая требуемую логику.

3. Повторная передача чувствительных данных исключается, если они уже проверялись ранее. Используется результат последнего акцепта.

4. Данные, пройденные проверку, формируются в письмо, которое должно включать весь исходный пакет. Письмо отправляется на почту сотрудника.

5. Заблокированные данные* не должны отправляться в систему Email, они сохраняются в Хранилище сообщений.

* Заблокированные данные – это данные, не пройденные акцепт. Настройка правил блокировки данных не требуется – акцепт (значение в поле «Подтверждение») можно проставлять в таблице вручную.

Дополнительные указания

Механизмы Platform и возможные паттерны для решения задачи:

1. Реализация задачи однократной доставки.

2. Использование Банка данных для обработки передаваемой информации.

3. Использование механизма конвейера для обработки процесса акцепта.

4. Механизм определения функции.

5. Требований к реализации механизма акцепта на стороне системы безопасности нет.

Задание №1. Организация согласованной и последовательной передачи объединенных данных с контролем порядка поступления данных

Описание задачи

Системы: REST API, PostgreSQL, 1C

Задача: Обеспечить согласованную и последовательную передачу объединенных данных из систем REST API и PostgreSQL в систему 1C, минимизировав количество обращений к источникам и контролируя порядок поступления данных.

Описание задачи:

Система REST API и система PostgreSQL выдают данные с отметкой даты обновления. Система REST API отправляет всегда полные данные о результатах экзаменов и зачетов, при этом пакет из системы PostgreSQL может содержать незаполненные поля. В таком случае необходимо выполнять дозапрос до момента, когда пользователь PostgreSQL не заполнит все данные. Общий пакет, отправляемый в 1С, должен содержать все поля из обеих систем-источников. Система 1С не контролирует последовательность версий данных — данные записываются в порядке поступления, поэтому важно обеспечить их последовательную и непротиворечивую передачу.

Требования и условия

1. Источник данных №1: Академический портал (расширенное REST API):

   Данные об экзаменах и зачетах поступают из внешней академической системы через расширенное REST API. Пакет «СеместроваяУспеваемость» содержит данные о студенте и его результатах экзаменов и зачетов за семестр и передается в формате JSON. Структура пакета:

Наименование поля	Тип	Описание
Код студента	String [9]	Уникальный идентификатор студента
ФИО студента	String [150]	Фамилия, имя, отчество студента
Группа	String [50]	Наименование учебной группы студента
Семестр	String [50]	Название семестра (например, «Осень 2026»)
Дата отчета	DateTime	Дата и время последнего изменения записи в Семестровой успеваемости
Экзамены	Сложное поле [Массив]	Список экзаменов с оценками
↳ Дисциплина	String [100]	Название учебной дисциплины
↳ Оценка	Int	Балл или оценка
↳ Дата экзамена	DateTime	Дата сдачи экзамена
Зачеты	Сложное поле [Массив]	Список зачетов с результатами
↳ Дисциплина	String [100]	Название учебной дисциплины

↳ Статус	String [10]	Результат зачета (зачет, незачет)
↳ Дата зачета	DateTime	Дата получения зачета

2. Источник данных №2: База практической подготовки студентов (PostgreSQL):

   В PostgreSQL хранится информация о прохождении студентами практики.

   Таблица **practics_result** содержит информацию о прохождении студентом производственной или учебной практики, включая данные о предприятии, руководителе, сроках и итоговой оценке. Пакет передается в формате Json. Структура таблицы **practics_result** содержит поля:

Наименование поля	Тип	Описание
student_id	character varying(9)	Уникальный идентификатор студента
practics_type	character varying(50)	Тип прошедшей студентом практики - производственная или учебная
organization_name	character varying(150)	Полное наименование организации, проводившей практику
organization_inn	character varying(12)	ИНН организации (для производственной практики обязателен)
director_name	character varying(150)	Должность руководителя практики
start_date	date	Дата начала прохождения практики
end_date	date	Дата окончания прохождения практики
grade	integer	Оценка за практику
resume	character varying	Текстовый отзыв руководителя о работе студента
export_flag	integer	Является триггером для выгрузки записи в Platform. `0` – запись не выгружена; при значении = `1` данные выгружены в Platform

3. Система-приемник: Академическая база (1С):

Система содержит документ «**ИтоговаяВедомостьУспеваемостиСтудента**». Структура документа следующая:

Наименование поля	Тип	Описание
Студент	СправочникСсылка.Студенты	Фамилия, имя, отчество студента
Семестр	Строка [50]	Название семестра (например, «Осень 2026»)
Дата	Дата	Дата и время последнего изменения в Академической базе

Практика	**Табличная часть**	Сведения о практике, пройденной студентом в рамках данного семестра
↳ Тип практики	Строка [50]	Тип прошедшей студентом практики - производственная или учебная
↳ Наименование организации	Строка [150]	Полное наименование организации, проводившей практику
↳ ИНН организации	Строка [12]	ИНН организации (для производственной практики обязателен)
↳ ФИО руководителя	Строка [150]	ФИО руководителя практики от предприятия
↳ Должность руководителя	Строка [100]	Должность руководителя практики
↳ Дата начала практики	Дата	Дата начала прохождения практики
↳ Дата окончания практики	Дата	Дата окончания прохождения практики
↳ Оценка практики	Число [2]	Оценка за практику
↳ Отзыв руководителя	Строка [Max]	Текстовый отзыв руководителя о работе студента
Экзамены	**Табличная часть**	Список экзаменов с оценками
↳ Дисциплина	Строка [150]	Название учебной дисциплины
↳ Оценка	Число [2]	Балл или оценка
↳ Дата экзамена	Дата	Дата сдачи экзамена
Зачеты	**Табличная часть**	Список зачетов с результатами
↳ Дисциплина	Строка [150]	Название учебной дисциплины
↳ Статус	Строка [10]	Результат зачета (зачет, незачет)
↳ Дата зачета	Дата	Дата получения зачета

Некоторые реквизиты документа ссылаются на справочники.

Справочник «**Студенты**» имеет следующие реквизиты:

Наименование поля	Тип	Описание
Код	Строка [9]	Уникальный код студента
ФИО студента	Строка [150]	Фамилия, имя, отчество студента
Группа	Строка [50]	Наименование учебной группы студента

По полю «student_id» происходит сопоставление записей из источников REST API и Postgre.

Критерии успешного выполнения задания

Задание считается выполненным, если соблюдены все критерии выполнения.

1. Пакет «**ПолнаяИтоговаяВедомость**» формируется и отправляется в 1С **только после получения данных из обоих источников** (REST API + PostgreSQL). Пакет объединяет информацию об экзаменах, зачетах и практике студента. Пример структуры пакета в формате JSON:


```
{
  "Проведен": false,
  "ПометкаУдаления": false,
  "Дата": "2025-06-20T10:30:00Z",
  "ФИОСтудента": "912c6a70-42f4-468a-8be0-d18333e2b001",
  "Семестр": "Осенний",
  "Экзамены": [
    {
      "Дисциплина": "Математика",
      "Оценка": 5,
      "ДатаЭкзамена": "2025-06-10T00:00:00Z"
    }
  ],
  "Зачеты": [
    {
      "Дисциплина": "История",
      "Статус": "Зачет",
      "ДатаЗачета": "2025-06-01T00:00:00Z"
    }
  ],
  "Практика": [
    {
      "ТипПрактики": "Учебная",
      "НаименованиеОрганизации": "АО Информационные Системы",
      "ИННОрганизации": "123456789012",
      "ФИОРуководителя": "Иванов И.И.",
      "ДолжностьРуководителя": "Руководитель отдела разработки",
      "ДатаНачалаПрактики": "2025-07-01T00:00:00.000Z",
      "ДатаОкончанияПрактики": "2025-07-14T00:00:00.000Z",
      "ОценкаПрактики": 5,
      "ОтзывРуководителя": "Студент Успешно справился с поставленными
задачами, освоил стандарты разработки."
    }
  ]
}
```


2. 1С не контролирует порядок поступления данных — все данные должны поступать последовательно, без дубликатов.
3. Из системы Postgres необходимо забирать данные по расписанию - каждые 30 секунд.
4. **Контроль последовательности:**
    - Данные из систем REST API и PostgreSQL должны объединяться в платформе с учётом актуальности (по дате обновления).
5. Если данных недостаточно для формирования полного пакета, платформа должна выполнить дозапрос в систему PostgreSQL
6. Система REST API отправляет пакет со всеми заполненными полями.
7. **Хранение данных:**
    - Все объединенные данные сохраняются в платформе для минимизации повторных обращений к системам источникам.
    - Правил к структуре, хранимой в банке - нет.
8. В системы Postgres и REST API данные поступают независимо друг от друга.

Дополнительные указания

1. Реализация задачи последовательной доставки (для контроля порядка данных).
2. Слияние сообщений (для объединения данных из REST API и PostgreSQL).
3. Использование Банка данных для обработки и хранения передаваемой информации.

Задание №1. Реализация поиска информации об автомобиле по производителю и модели с возвратом данных в формате JSON

Описание задачи

Системы Расширенное REST API, MSSQL, PostgreSQL

Задача Реализация поиска информации об автомобиле по производителю и модели с возвратом данных в формате JSON.

Описание задачи:

Необходимо реализовать функционал поиска информации об автомобиле на основе его производителя и модели через два HTTP-запроса:

- Первый запрос (GET) — параметры передаются в строке запроса.
- Второй запрос (POST) — параметры передаются в теле сообщения (JSON).

Система должна возвращать данные в формате JSON, объединяя информацию из двух источников: MSSQL (основные данные об автомобиле) и PostgreSQL (отзывы).

Требования и условия

Формат запроса: GET запрос по адресу api/vehicle?manufacturer=lada&model=vesta.

Формат запроса: POST запрос /api/vehicle

Формат ответа: Ответ должен быть в формате JSON, как показано ниже:

```
{
  "id": 123,
  "manufacturer": "lada",
  "model": "vesta",
  "avgPrice": 1500000,
  "reviews": [
    {"user": "Alice", "rating": 5},
    {"user": "Bob", "rating": 4}
  ]
}
```

Источники данных:

1. Источник MSSQL хранить информацию об автомобилях в таблице со следующими полями:

Наименование	Тип поля	Описание
Id	uniqueidentifier	Идентификатор транспортного средства
Manufacturer	varchar(max)	Производитель
Model	varchar(max)	Модель транспортного средства
AvgPrice	nchar(10)	Стоимость

2. Источник PostgreSQL хранит информацию об отзывах в таблице со следующими полями:

Наименование	Тип поля	Описание
id	integer	Идентификатор записи отзыва
username	character varying(100)	Имя пользователя
rating	integer	Оценка
vehicle_id	character varying(50)	Идентификатор транспортного средства. Это ключ связи таблицы с отзывами PostgreSQL и таблицы с транспортными средствами MSSQL.

3. Расширенный REST API - используется для выполнения запросов на получение информации

Связь объектов происходит по ключу Id и VehicleId.

Критерии успешного выполнения задания

Задание считается выполненным, если соблюдены все критерии выполнения.

1. **Валидация параметров:**

- Если отсутствует любой из параметров: manufacturer или model, необходимо вернуть сообщение об ошибке, указывающее, какое именно поле не заполнено.
- Если не заполнены оба поля, вернуть сообщение, что оба поля не заполнены.

2. **Обработка отсутствующих данных:**

- Если автомобиль с указанными параметрами не найден, вернуть соответствующее сообщение пользователю.
- Если для найденного автомобиля отсутствуют отзывы, вернуть пустой список отзывов.

Дополнительные указания

1. Убедитесь, что система корректно обрабатывает запросы и возвращает данные в требуемом формате.
2. Реализуйте механизм валидации параметров запроса.
3. Обеспечьте корректное объединение данных из различных источников.

Задание №1. Автоматизация отправки уведомлений менеджерам о создании и изменении партнеров в 1С

Описание задачи

Системы 1С, Email, Банк данных (необязательно)

Задача Автоматизация отправки уведомлений менеджерам о создании и изменении партнеров в 1С

Описание задачи: Компания А ставит задачу по автоматизации отправки уведомлений менеджерам о создании партнера и внесении изменений в описание партнера в системе управления продаж. Сообщение должно быть выслано в текстовом виде на почту привязанного к партнеру менеджера.

Изменение партнера отслеживается по полям: Наименование партнера, Описание партнера, Юридический адрес, Тип, Партнерский менеджер.

Требования

Источник данных :

Система управления продаж (1С) хранит информацию о партнерах и связанных сущностях:

1. **Справочник "Партнеры":**

Содержит информацию о партнерах:

Наименование	Тип	Описание
Код	Строка	Код записи справочника
Наименование партнера	Строка	Наименование организации. Изменения по полю необходимо передавать в письме.
Описание партнера	Строка (500)	Описание организации. Изменения по полю необходимо передавать в письме.
ИНН	Строка(12)	ИНН организации. Изменения по полю не передавать в письме.
Юридический адрес	Строка (500)	Юридический адрес организации. Изменения по полю необходимо передавать в письме.
Тип	ПеречислениеСсылка.ТипКонтрагента	Тип организации. • Юр. лицо

		• Физ. лицо • ИП • Публично- правовая организация Изменения по полю необходимо передавать в письме.
Веб-сайт	Строка (50)	Веб-сайт организации. Изменения по полю не передавать в письме.
Партнерский менеджер	СправочникСсылка.Сотрудники	Ссылка на справочник "Сотрудники". Партнерский менеджер организации, изменения по полю необходимо передавать в письме.
Контактные лица	Табличная часть	Ссылка на справочник (Контактные лица). Изменения в табличной части (добавление, удаление, изменение) не передавать в письме.

Уникальность записи справочника «Партнеры» при обработке определяется по полю - **ИНН**

2. **Справочник "Сотрудники"**

Содержит информацию о менеджерах:

Наименование	Тип	Описание
Фамилия	Строка(500)	Фамилия сотрудника
Имя	Строка(500)	Имя сотрудника
Отчество	Строка(500)	Отчество сотрудника
Электронная почта	Строка(100)	Электронная почта сотрудника
Дата рождения	Дата	Дата рождения сотрудника
Пол	ПеречислениеСсылка.Пол	Пол сотрудника: мужской, женский
Номер телефона	Строка(20)	Номер телефона сотрудника
Город	Строка(100)	Город нахождения сотрудника
Подразделение	Строка(500)	Подразделение сотрудника
Должность	Строка(500)	Должность сотрудника

Уникальность записи справочника «Сотрудники» при обработке определяется по полю **Ссылка** на стороне 1С.

3. **Справочник "Контактные лица"**

Содержит информацию о контактных лицах:

Наименование	Тип	Описание
Фамилия	Строка(500)	Фамилия контактного лица
Имя	Строка(500)	Имя контактного лица
Отчество	Строка(500)	Отчество контактного лица
Электронная почта	Строка(100)	Электронная почта контактного лица
Дата рождения	Дата	Дата рождения контактного лица
Пол	ПеречислениеСсылка.Пол	Пол контактного лица: мужской, женский
Номер телефона	Строка(20)	Номер телефона контактного лица
Город	Строка(500)	Город нахождения контактного лица
Подразделение	Строка(500)	Подразделение контактного лица
Должность	Строка(500)	Должность контактного лица

Контактные лица - сотрудники организации партнера.

Уникальность записи справочника «Контактные лица» при обработке определяется по полю **Ссылка** на стороне 1С.

Критерии успешного выполнения задания

Задание считается выполненным, если соблюдены все критерии выполнения.

Условия получения письма коннектором Email:

1. **Настройка уведомлений о создании партнера:**
   - Настроить систему для автоматической отправки уведомлений менеджерам при создании нового партнера в справочнике "Партнеры".
   - Уведомление должно быть отправлено по факту внесения новой записи.
   - Уведомление должно содержать информацию о новом партнере.
2. **Настройка уведомлений об изменении описания партнера:**
   - Настроить систему для автоматической отправки уведомлений менеджерам при внесении изменений в описание партнера.
   - Уведомление должно быть отправлено по факту внесения изменений в основную информацию о партнере (все поля, кроме табличной части - Контактные лица).
   - Уведомление должно содержать актуальную информацию.
   - В письме быть выделены те поля, что были изменены.
3. **Формирование и отправка писем:**
   - Настроить шаблоны писем для уведомлений о создании и изменении партнеров.
   - Убедиться, что письма содержат всю необходимую информацию и отправляются на корректные адреса электронной почты менеджеров.

Дополнительные указания

1. Убедиться, что данные корректно передаются между системами.
2. Проверить корректность работы системы уведомлений.

Примеры писем, которые отправляются на email сотрудника:

При создании:

Тема: Добавлен новый партнер ИноФерм
Тело письма:
Здравствуйте, Иванов Иван Иванович!

Добавлен новый партнер - «ИноФерм».

При изменении:

Тема: Изменены данные о партнере “ИноФерм”
Тело письма:
Здравствуйте, Иванов Иван Иванович!

Изменены данные о прикрепленном Вам партнере:

Наименование партнера: ИноФерм

Юридический адрес: ~~Россия, Москва, ул. Боборов 16стр3~~ -> Россия, Москва, ул. Боборов 16стр4

Содержимое рассылки может отличаться и включать в себя дополнительную информацию, но должно обязательно содержать те пункты, что описаны выше в условиях.

Задание на исправление некорректной настройки

Задача на исправление ошибки включает в себя общее описание ошибки, которую необходимо исправить.

Сам сценарий для воспроизведения ошибки создан и развернут на предоставленном для прохождения экзамена стенде.

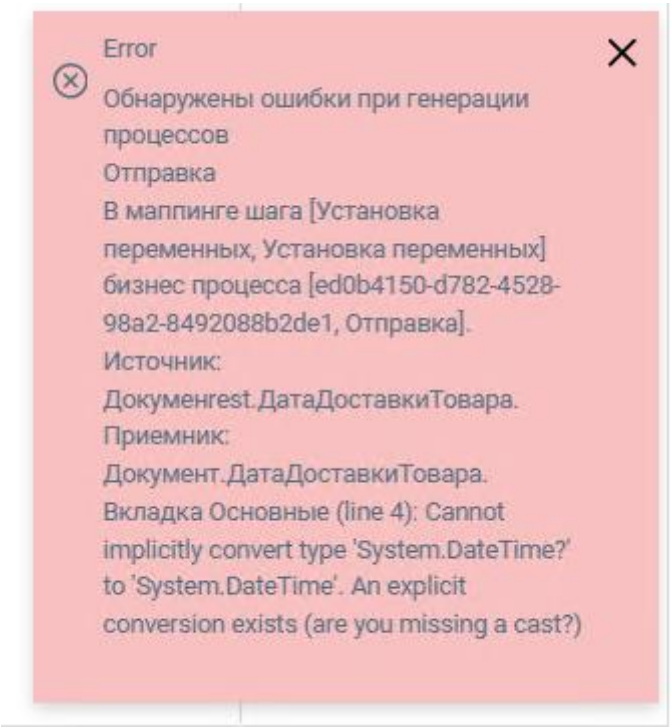
Описание проблемы – это описание ошибки и используемых для воспроизведения ошибки объектов.

Для выполнения задачи необходимо устранить ошибку и указать на причину ее возникновения.

Задание №2. Не применяется конфигурация.

Описание

При применении конфигурации появляется ошибка:



Версия 1С:Предприятия - 8.3.24.1691

Режим совместимости - 8.3.22

Версия подсистемы - 3.0.3.5

Задача:

Необходимо определить, в чем именно заключается ошибка и каким образом ее можно устранить, чтобы применение конфигурации выполнилось и сообщение, отправленное из swagger системы REST API, дошло до системы-получателя 1С.

Задание №2. Проверка выполнения сценария при работе с Email

Описание

При отправке сообщений на почту происходит ошибка:

```
Код 500: [Ошибка обработки сообщения на стороне сервера] System.Exception: Не указан получатель для отправки email

   at DT.Adapter.Adapters.Email.PlatformSmtpClient.SendEmail(MailMessage mailMessage)

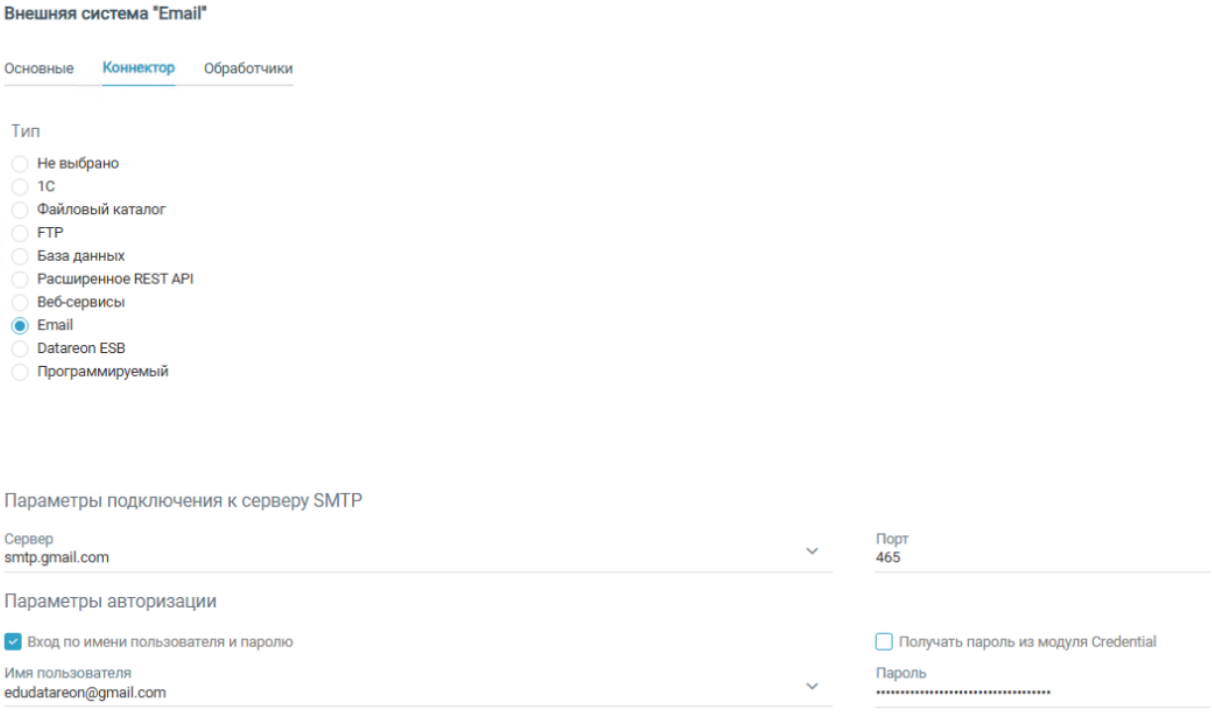
   at DT.Adapter.Adapters.Email.EmailAdapter.SendEmail(MailMessage mailMessage)

   at DT.Adapter.Adapters.Email.EmailAdapter.SendMessageAsync(InternalMessage internalMessage)

   at DT.Adapter.Senders.UpstreamOutgoingMessagesSender.ProcessSendMessageAsync(InternalMessage message, Boolean sendToArchive)
```

В ЦМ состояние внешней системы – Рабочее.

Настройки внешней системы – Система Email:



Код обработчика:

```
GetMailMessage();
```

Задание №2. Организовать единый процесс передачи сообщений в хранилище без ошибок

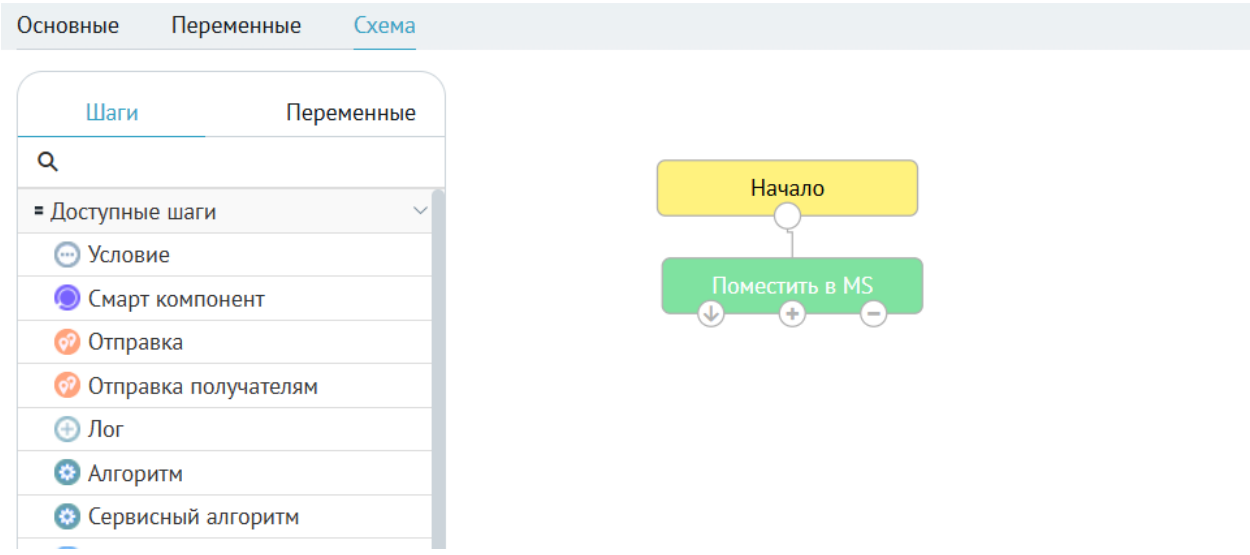
Описание

Изначально необходимо реализовать маршрут, по которому все полученные сообщения записываются в Хранилище сообщений.

При тестировании схемы на отправку сообщения в Хранилище сообщений возникает ошибка:

```
Не удалось запустить обработку сообщения Id=d5d0e5e1-5d6f-4290-b848-1754105c9c84,StreamId=1685dadd-d218-45ce-8706-adbbafe47126,PlatformCorrelationId=,MessageType=DataMessage,ProcessId=ccab7f01-30e4-4c98-a0fa-050b6886a593,Source=c238977c-dcd7-40bf-a71c-6bbc670bd9b5,BodySize=55,Properties=[Dt_HandlerId = 71757cd3-1c2f-415e-b80c-5a5bc23a24df; ProcessingStage = 10; RouteFunctions = ccab7f01-30e4-4c98-a0fa-050b6886a593; DelayCount = 0; ], так как Не определены получатели для маршрутов для сообщения Id=d5d0e5e1-5d6f-4290-b848-1754105c9c84,StreamId=1685dadd-d218-45ce-8706-adbbafe47126,PlatformCorrelationId=,MessageType=DataMessage,ProcessId=ccab7f01-30e4-4c98-a0fa-050b6886a593,Source=c238977c-dcd7-40bf-a71c-6bbc670bd9b5,BodySize=55,Properties=[Dt_HandlerId = 71757cd3-1c2f-415e-b80c-5a5bc23a24df; ProcessingStage = 10; RouteFunctions = ccab7f01-30e4-4c98-a0fa-050b6886a593; DelayCount = 0; ]
```

Маршрут – Отправка в Хранилище



При этом сообщение физически записывается в Хранилище, и по цепочке событий также видно, что отправка в хранилище выполнена успешно.

Необходимо определить, в чем именно заключается ошибка и каким образом ее можно устранить, выполнив исходную задачу.

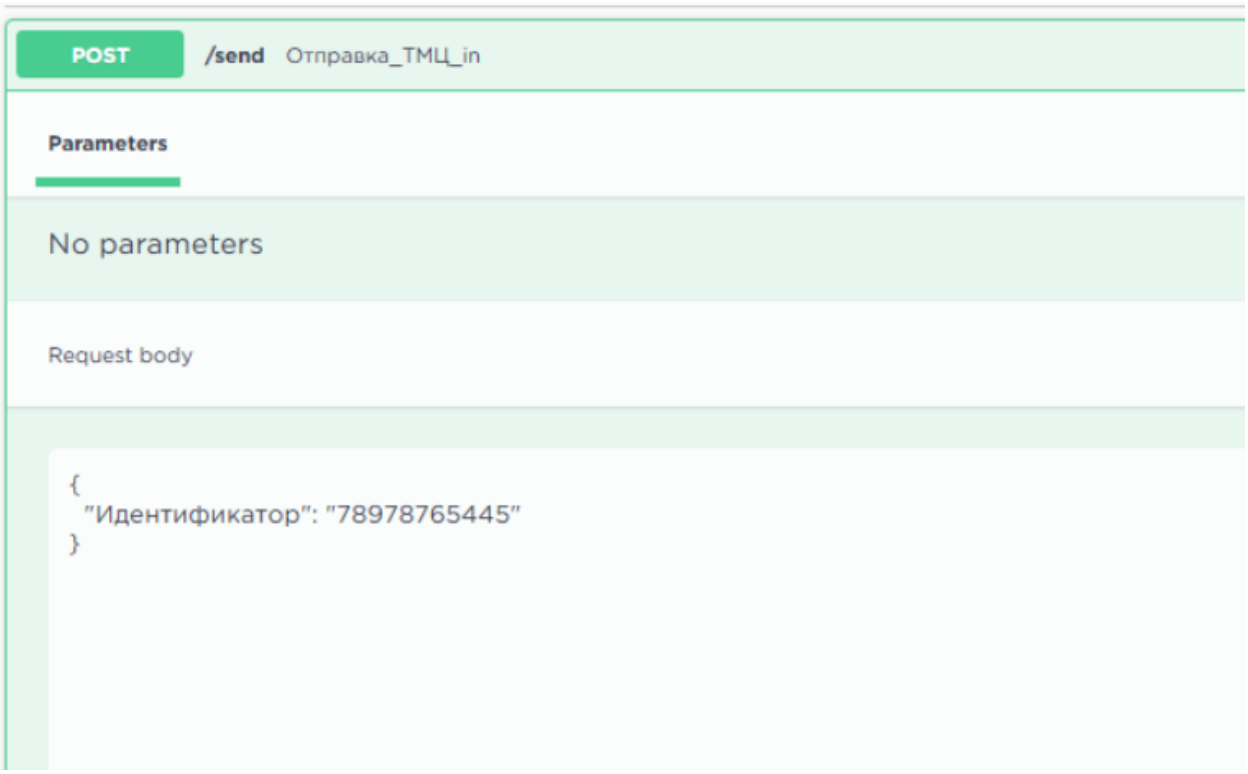
Задание №2. Устранить ошибки настройки интеграционного сценария при обогащении

Описание

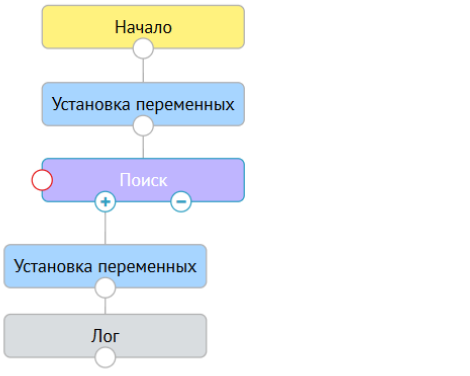
Реализован процесс сверки данных с записью в Банке данных.

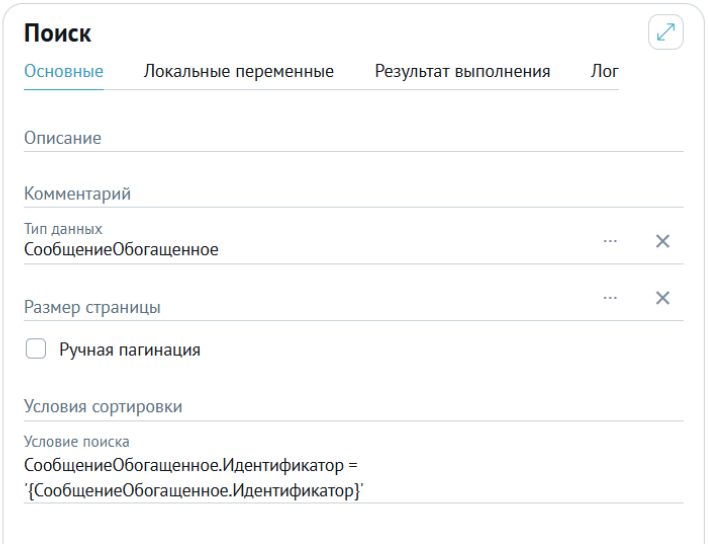
Данный процесс выполняет такую задачу: поиск по входящему идентификатору, получение записи и обогащение сообщения.

1. Источник - система Веб отправляет сообщение:

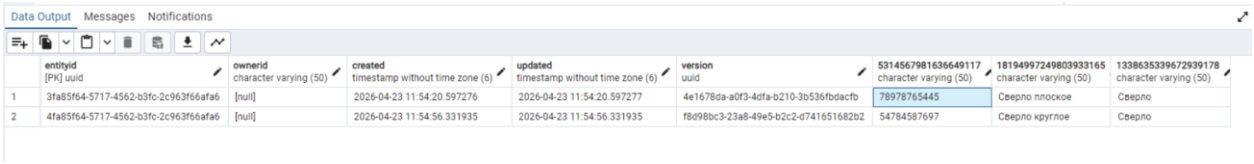


2. Бизнес процесс запускается по типу данных, в шаге «Поиск» осуществляется поиск записей в банке по полю «Идентификатор»:





3. В Банке данных существует только одна запись с соответствующим идентификатором:



Data Output | Messages | Notifications

	entityid [PK] uuid	ownerid character varying (50)	created timestamp without time zone (6)	updated timestamp without time zone (6)	version uuid	5314567981636649117 character varying (50)	1819499724980393316 character varying (50)	1338635339672939178 character varying (50)
1	3fa85f64-5717-4562-b3fc-2c963f66afa6	[null]	2026-04-23 11:54:20.597276	2026-04-23 11:54:20.597277	4e1678da-a0f3-4dfa-b210-3b536fbdacfb	78978765445	Сверло плоское	Сверло
2	4fa85f64-5717-4562-b3fc-2c963f66afa6	[null]	2026-04-23 11:54:56.331935	2026-04-23 11:54:56.331935	f8d98bc3-23a8-49e5-b2c2-d741651682b2	54784587697	Сверло круглое	Сверло

Ошибка в процессе:

Ошибка при выполнении процесса [05f5ada7-87fe-4d3c-b4f6-be72771262be, ОбработкаКлассификаторов], шаг [Установка переменных, Установка переменных], Id = 07d8bc90-1366-48fa-a88a-e67a2c6e1ade: System.ArgumentOutOfRangeException: Index was out of range. Must be non-negative and less than the size of the collection. (Parameter 'index') at System.Collections.Generic.List`1.get_Item(Int32 index) at SystemModels.OwnList1.get_Item(Int32 index) at SystemModels.SystemModel.ОбработкаКлассификаторов.GetValue_1516024358(OwnList1 РезультатШагаПоиск) at SystemModels.SystemModel.ОбработкаКлассификаторов.SetVariables_3a6d1b4c(LocalScope local) at SystemModels.SystemModel.ОбработкаКлассификаторов.SetVariables() at DT_Processes.Engine.Context.InnerContext.SetVariables() at DT_Processes.Engine.Context.ProcessContext.SetVariables() at DT_Processes.Engine.Steps.SetVariablesStep.ProcessMapping(IProcessContext context) at DT_Processes.Engine.Steps.SetVariablesStep.Run(CancellationToken token, IProcessEnvironment env, IProcessContext context, ProcessData data, ProcessLoggingType logType) at DT_Processes.Engine.Execution.BPExecutor.DoRunStep(IProcess bp, ProcessData data, IProcessContext context, IRunnableStep step, CancellationToken token) at DT_Processes.Engine.Execution.BPExecutor.ExecuteStep(IProcess bp, ProcessData data, IProcessContext context, IRunnableStep step, CancellationToken token) System.ArgumentOutOfRangeException: Index was out of range. Must be non-negative and less than the size of the collection. (Parameter 'index') at System.Collections.Generic.List`1.get_Item(Int32 index) at SystemModels.OwnList1.get_Item(Int32 index) at SystemModels.SystemModel.ОбработкаКлассификаторов.GetValue_1516024358(OwnList1 РезультатШагаПоиск) at SystemModels.SystemModel.ОбработкаКлассификаторов.SetVariables_3a6d1b4c(LocalScope local) at SystemModels.SystemModel.ОбработкаКлассификаторов.SetVariables() at DT_Processes.Engine.Context.InnerContext.SetVariables() at DT_Processes.Engine.Context.ProcessContext.SetVariables() at DT_Processes.Engine.Steps.SetVariablesStep.ProcessMapping(IProcessContext context) at DT_Processes.Engine.Steps.SetVariablesStep.Run(CancellationToken token, IProcessEnvironment env, IProcessContext context, ProcessData data, ProcessLoggingType logType) at DT_Processes.Engine.Execution.BPExecutor.DoRunStep(IProcess bp, ProcessData data, IProcessContext context, IRunnableStep step, CancellationToken token) at DT_Processes.Engine.Execution.BPExecutor.ExecuteStep(IProcess bp, ProcessData data, IProcessContext context, IRunnableStep step, CancellationToken token)

Необходимо устранить ошибку, выполнив изначальную задачу сверки и обогащения данных.

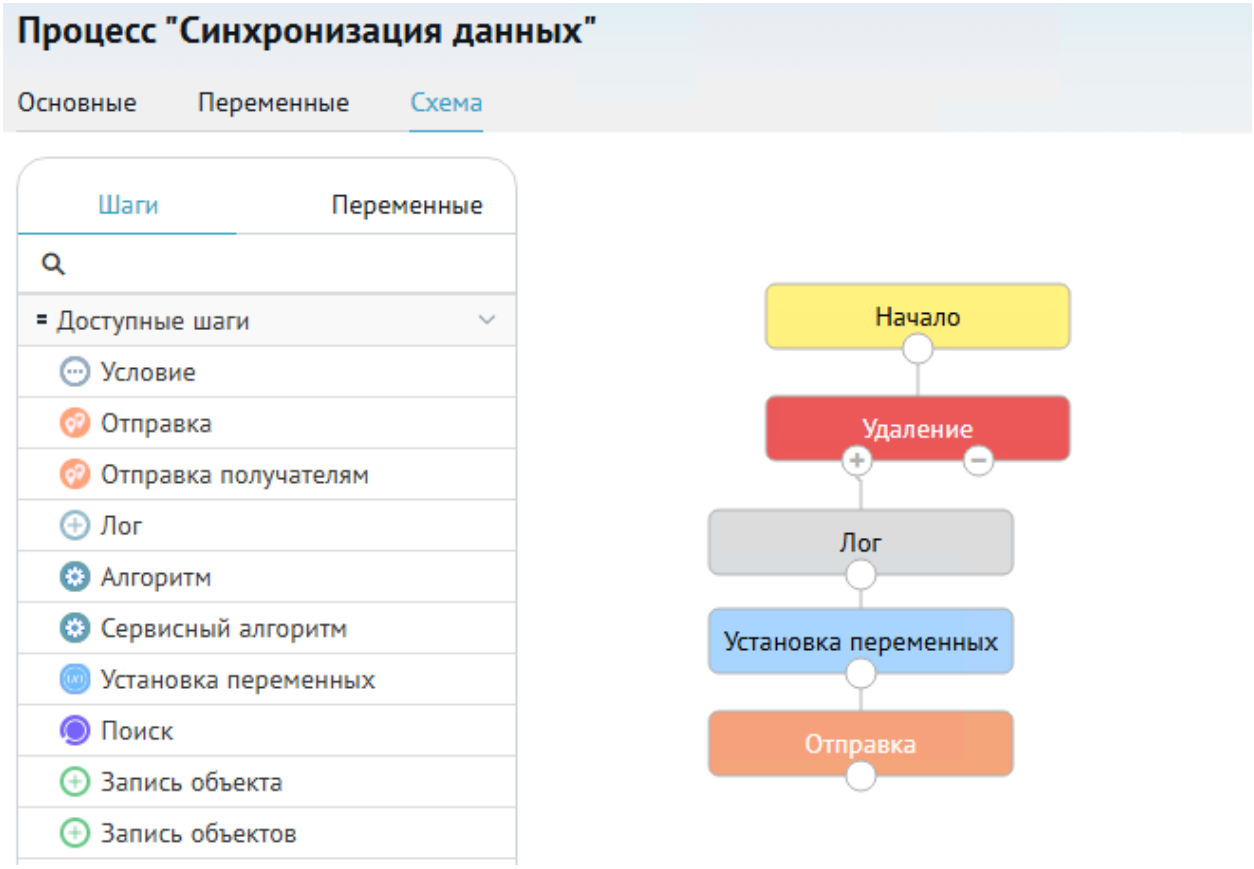
Задание №2. Организовать удаление записей из банка с устранением исключений

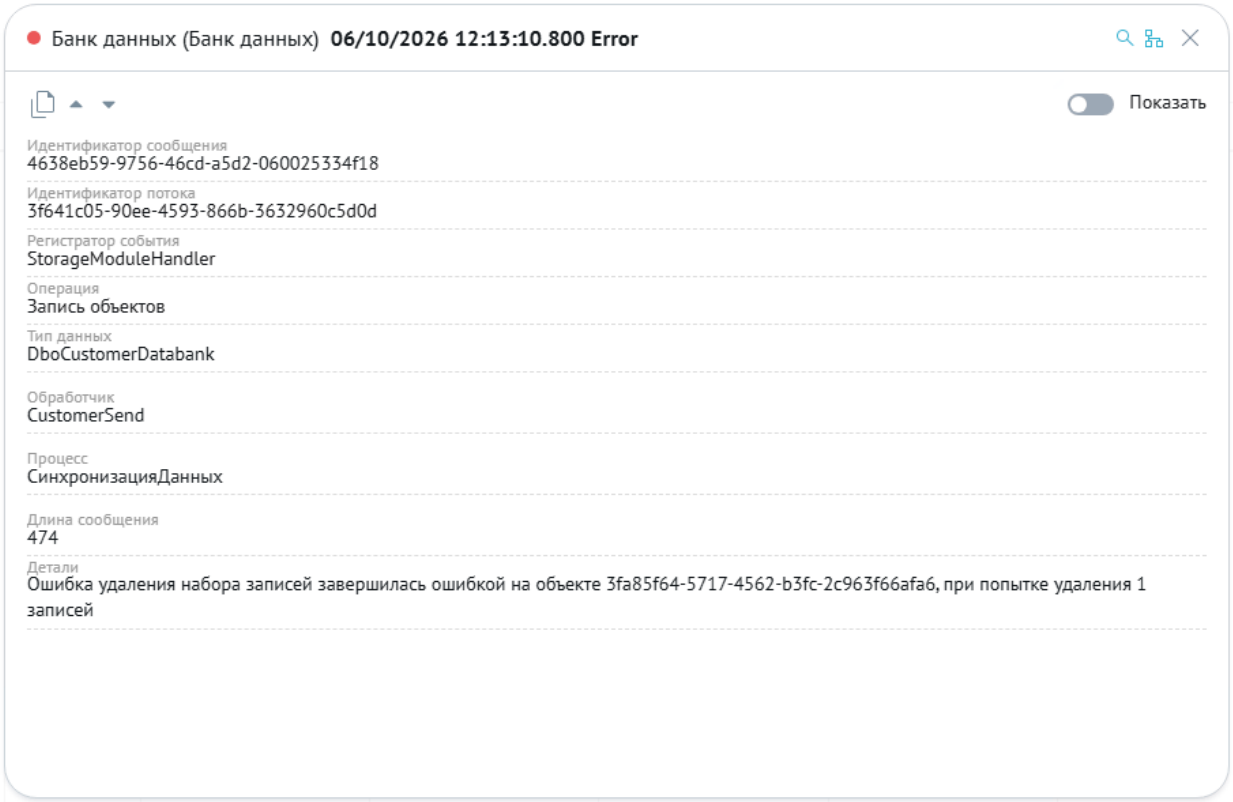
Поставили задачу – производить удаление записей из Банка данных. Идентификаторы записей синхронизированы между внешней системой «dbo_Customer» и Банком данных, то есть они одинаковы. На тестовой среде все работает корректно. В продуктивной среде же часть записей уже удалена ранее — и внезапно процесс валится. В итоге ночная очистка падает и блокирует интеграцию.

Необходимо изучить запуск процесса – «Синхронизация данных».

В платформу поступает сообщение, запускается процесс. В этом процессе есть шаг удаление, куда передаются записи(массив) банка данных.

В шаге Удаление настроен только левый выход - когда запись найдена, и удаление произошло.





При отправке такого пакета:

{

"EntityId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"name": "Петров Максим Сергеевич",

"email": "petrov@mail.ru",

"phone": "89304879032",

"address": "г. Москва ул. Ленина 1 д. 1"

}