GIMS AUTOMATION MANAGER

РАБОЧАЯ ДОКУМЕНТАЦИЯ

ОПИСАНИЕ РАБОТЫ СИСТЕМЫ

Москва 2021

СОДЕРЖАНИЕ

1 ПОДГОТОВКА К РАБОТЕ

1.1 ПОРЯДОК ЗАГРУЗКИ ДАННЫХ И ПРОГРАММ

Вызов программы осуществляется путем ввода в адресной строке интернет-браузера адреса домашней страницы Системы.

1.2 АВТОРИЗАЦИЯ В СИСТЕМЕ

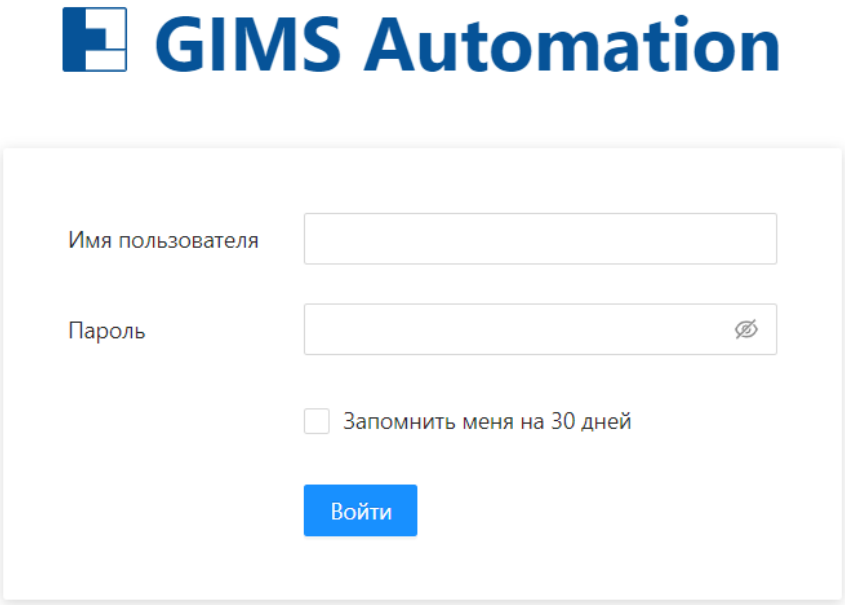


Рисунок 1 – Форма авторизации пользователя

Для доступа к данным, пользователи Системы используют связку логин и пароль, созданные внутри Системы или доменные логин и пароль. Если данные указаны верно, то отображается стартовая страница Системы.

Примечание: при первом запуске Системы вход в программу осуществляется по логину: admin и паролю: 123QWErty.

2 СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ ФУНКЦИОНАЛА

2.1 ПРЕДОСТАВЛЕНИЕ REST API НА БАЗЕ ОБРАБОТКИ ДАННЫХ ИЗ БД

2.1.1 Настройка сценария

Цель операции – Получение обработанной информации от БД через REST API.

Рисунок 2 – Принцип работы сценария

Данный сценарий подходит для взаимодействия со следующими встроенными типами источников данных:

- DB2;
- Hive;
- MSSQL;
- MySQL;
- Oracle;
- PostgreSQL.

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   - Datasources admin;
   - Scenarios admin;
   - Activators admin.
2. В системе должен быть подключены и настроены серверы и собраны в кластеры.

Последовательность действий:

1) Авторизуйтесь в Системе.
2) Перейдите пункт меню **«Источники данных».**
3) Выполните подключение к источнику данных (если соединение уже установлено проверьте что оно активно).

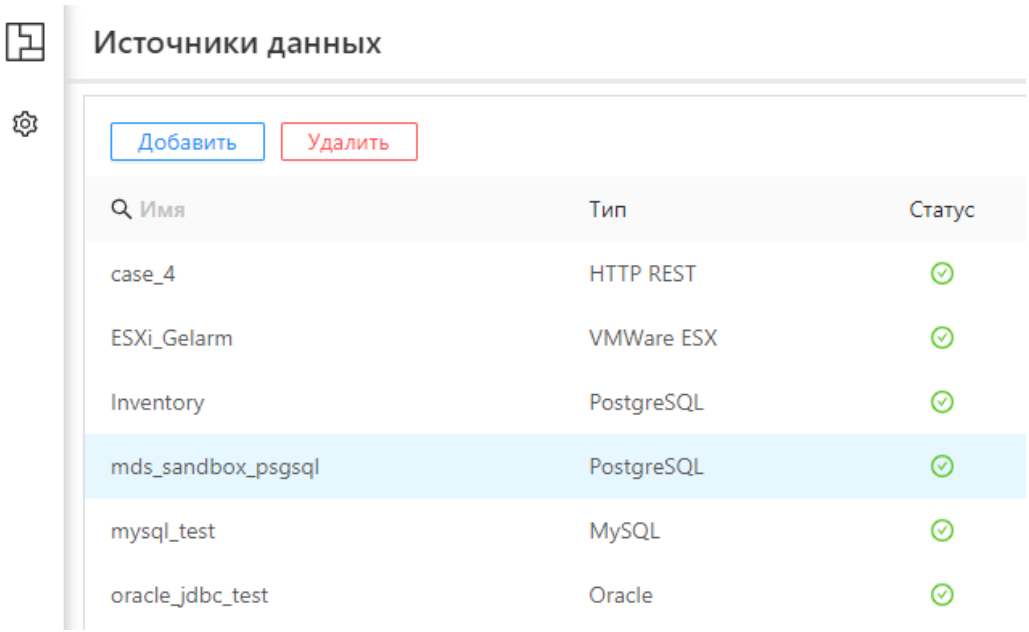


Рисунок 3 – Список источников данных

Пример: *Подключение к БД mds_sandbox_pgssql типа PostgreSQL работает, так как статус соединения с БД активен.*

4) Перейдите пункт меню **«Сценарии автоматизации».**
5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источника. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

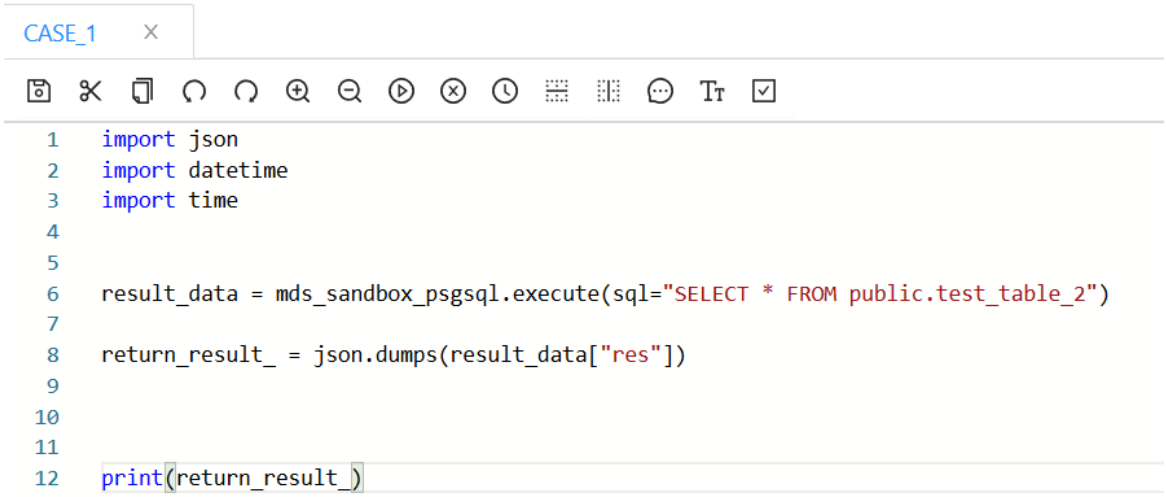


Рисунок 4 – Программный код Сценария автоматизации

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json
import datetime
import time

result_data = mds_sandbox_psgsql.execute(sql="SELECT * FROM public.test_table_2")
return_result_ = json.dumps(result_data["res"])

print(return_result_)
```

В исполняемом коде заложена следующая логика работы Системы:

- *Система обращается к подключенному источнику данных mds_sandbox_pgssql и получает данные из таблицы test_table_2;*
- *Трансформирует полученные данные в формат JSON;*
- *Возвращает данные пользователю.*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте условия, при которых должен будет выполняться сценарий автоматизации.

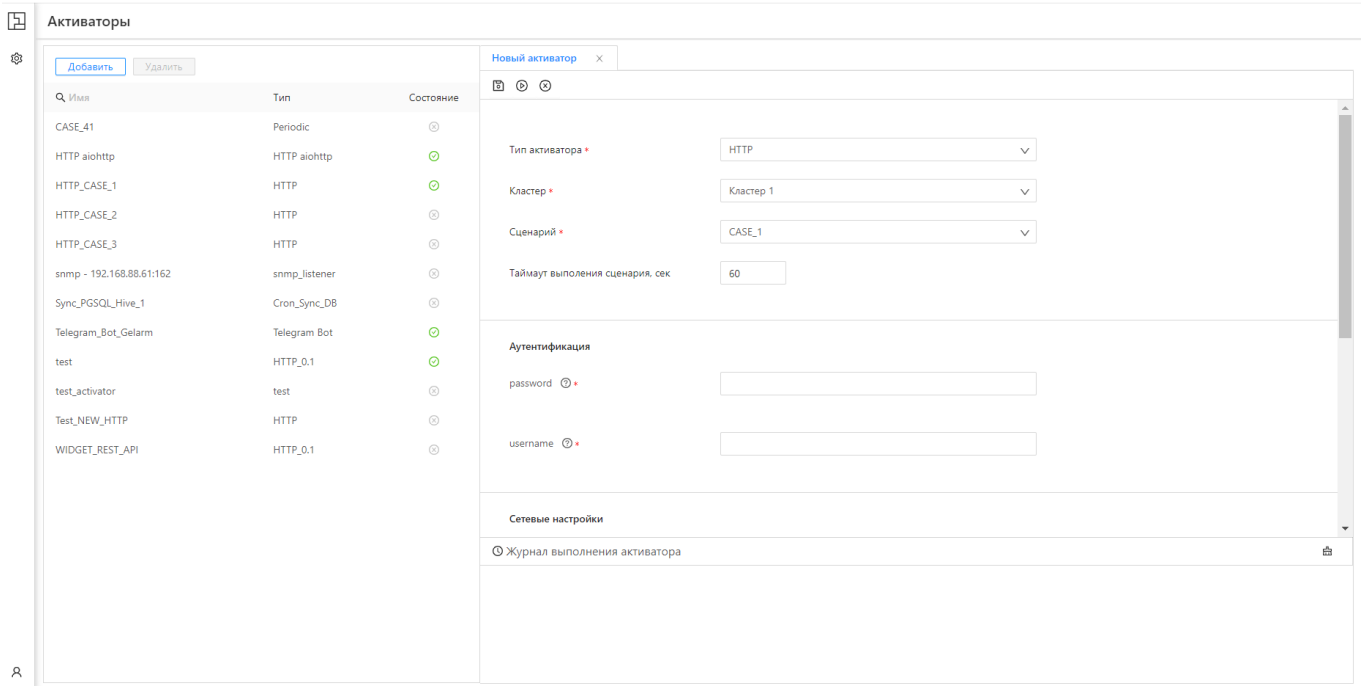


Рисунок 5 – Создание активатора

Пример: *Для создания активатора типа HTTP, работающего на Кластере 1 по сценарию CASE_1 необходимо:*

- *Выбрать из разворачивающегося списка тип активатора – HTTP;*
- *Выбрать из разворачивающегося списка кластер – Кластер 1;*
- *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
- *Назначить таймаут выполнения сценария;*
- *Заполнить раздел аутентификация:*
  - *В поле password задайте пароль для доступа к HTTP;*
  - *В поле username задайте имя пользователя для доступа к HTTP.*
- *Заполнить раздел сетевые настройки:*
  - *В поле port введите номер порта;*
  - *В поле timeout укажите таймаут.*
- *Заполнить раздел параметры обрабатываемых HTTP-запросов:*
  - *Выберете из разворачивающегося списка request_type тип запроса;*
  - *Выберете из разворачивающегося списка content_type_out тип выходных данных;*
  - *Выберете из разворачивающегося списка content_type тип входных данных.*
- *Нажать кнопку*  *для сохранения активатора и ввести имя активатора;*
- *Дождаться пока активатор появится в списке активаторов;*

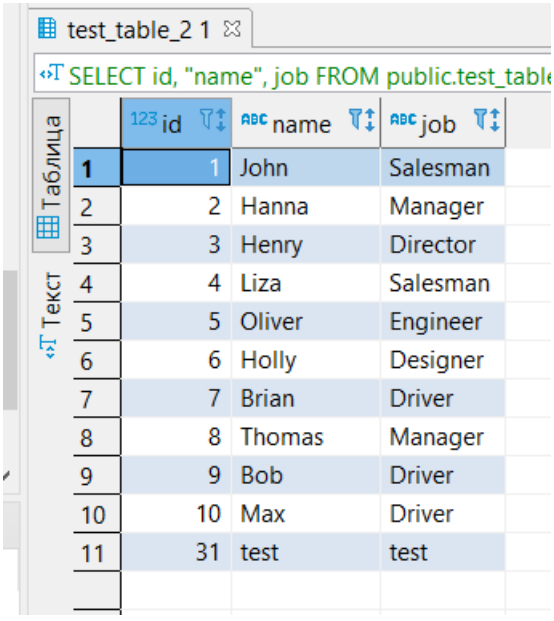
– *Нажать кнопку ⊳ для запуска активатора;*
– *Дождаться пока статус активатора в списке станет активным.*

8) Сохраните внесенные изменения в активатор. В списке активаторов добавится новая строка.
9) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

2.1.2 Демонстрация работы сценария

Описаный выше сценарий выгружает из БД mds_sandbox_pgssql данные таблицы *test_table_2* в формате JSON при активации по HTTP.



	id	name	job
1	1	John	Salesman
2	2	Hanna	Manager
3	3	Henry	Director
4	4	Liza	Salesman
5	5	Oliver	Engineer
6	6	Holly	Designer
7	7	Brian	Driver
8	8	Thomas	Manager
9	9	Bob	Driver
10	10	Max	Driver
11	31	test	test

Рисунок 6 – Содержимое таблицы test_table_2

Для получения данных через REST API необходимо :

1) Определить IP адрес сервера на котором запущен активатор. Для этого необходимо:
   1. В разделе «Конфигуратор инфраструктуры» открыть кластер, на котором работает активатор.

2. Во вкладке Серверы кластера найти значение IP адреса сервера.

Примечание: *Если в разделе «Активаторы» навести указатель на статус запущенного активатора, во всплывающей форме отобразится IP адрес сервера.*

2) Определить порт на котором запущен активатор. Для этого необходимо:
   1. В разделе «Активаторы» открыть запущенный активатор.
   2. В разделе сетевые настройки найти значение порта.

3) Перейти в интернет-браузере по IP адресу сервера на котором запущен активатор с указанием порта: [IP адрес]: [port]

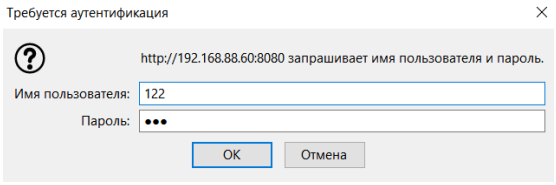


Рисунок 7 - Форма аутентификации

4) В появившейся форме ввести логин и пароль, указанные в разделе аутентификация активатора.
5) В окне интернет-браузера появится информация из БД в формате JSON.

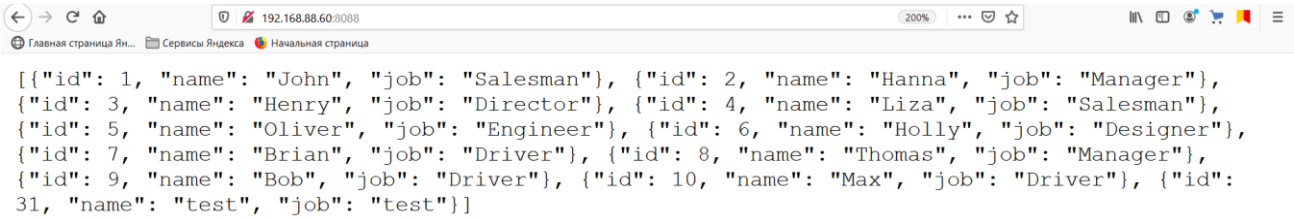


Рисунок 8 – Пример получаемых данных

2.2 Запись данных в БД, поступивших через REST API

2.2.1 Настройка сценария

Цель операции – добавление данных в БД через REST API

Рисунок 9 – Принцип работы сценария

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   - Datasources admin;
   - Scenarios admin;
   - Activators admin.
2. В Системе должен быть подключены и настроены вычислительные мощности.

Последовательность действий:

1) Авторизуйтесь в Системе с правами Администратора системы.
2) Перейдите пункт меню **«Источники данных».**
3) Выполните подключение к БД источника данных (если соединение уже установлено проверьте что оно активно).
4) Перейдите пункт меню **«Сценарии автоматизации».**
5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источника. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

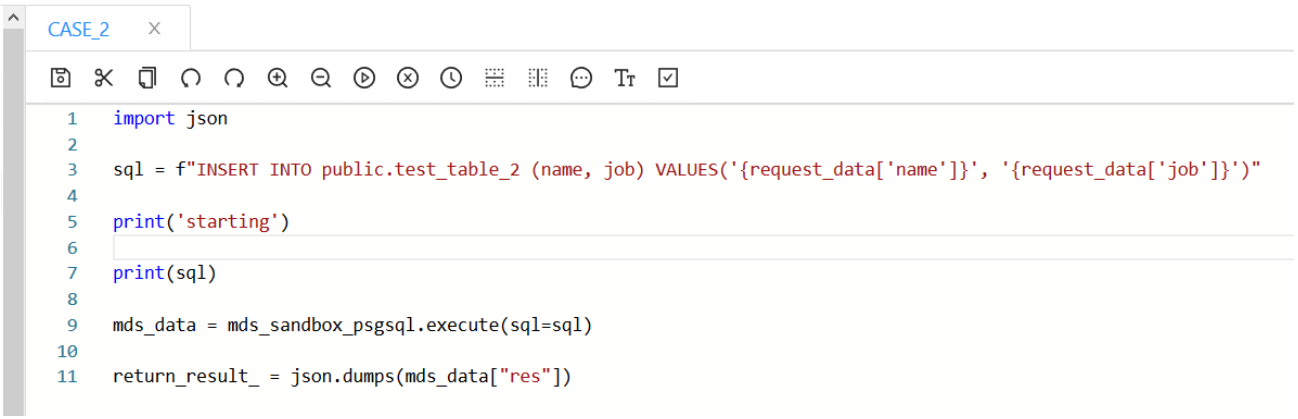


Рисунок 10 - Программный код Сценария автоматизации

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json

sql = f"INSERT INTO public.test_table_2 (name, job) VALUES('{request_data['name']}', '{request_data['job']}')"

print('starting')

print(sql)

mds_data = mds_sandbox_psgsql.execute(sql=sql)

return_result_ = json.dumps(mds_data["res"])
```

В исполняемом коде заложена следующая логика работы Системы:

- *Система считывает данные, введенные через REST API;*
- *Выполняется INSERT в таблицу test_table_2 значение name в колонку «name» и job в колонку «job».*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте условия, при которых должен будет выполняться сценарий автоматизации.

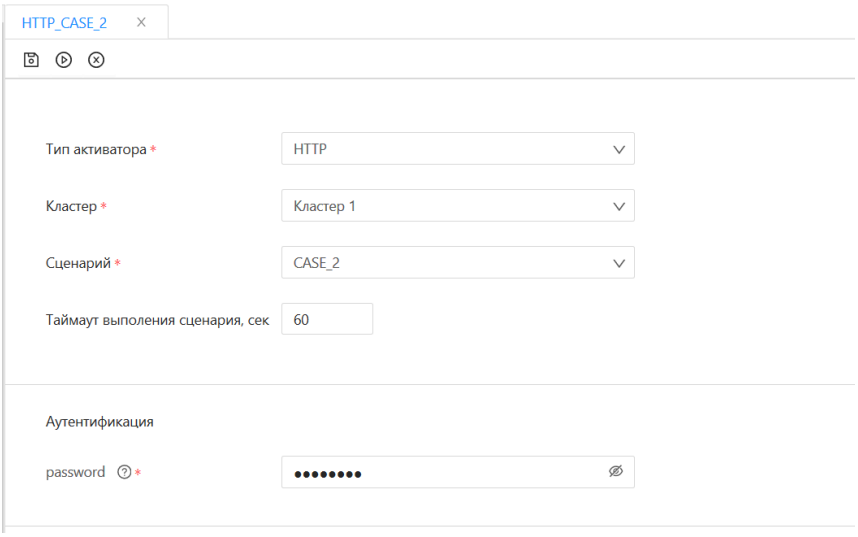


Рисунок 11 – Создание активатора

Пример: *Для создания активатора типа HTTP, работающего на Кластере 1 по сценарию CASE_2 необходимо:*

- *Выбрать из разворачивающегося списка тип активатора – HTTP;*
- *Выбрать из разворачивающегося списка кластер – Кластер 1;*
- *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
- *Назначить таймаут выполнения сценария;*
- *Заполнить раздел аутентификация:*
  - *В поле password задайте пароль для доступа к HTTP;*
  - *В поле username задайте имя пользователя для доступа к HTTP.*
- *Заполнить раздел сетевые настройки:*
  - *В поле port введите номер порта;*
  - *В поле timeout укажите таймаут.*
- *Заполнить раздел параметры обрабатываемых HTTP-запросов:*
  - *Выберете из разворачивающегося списка request_type тип запроса;*
  - *Выберете из разворачивающегося списка content_type_out тип выходных данных;*
  - *Выберете из разворачивающегося списка content_type тип входных данных.*
- *Нажать кнопку*  *для сохранения активатора и ввести имя активатора;*
- *Дождаться пока активатор появится в списке активаторов;*
- *Нажать кнопку*  *для запуска активатора;*
- *Дождаться пока статус активатора в списке станет активным.*

8) Сохраните внесенные изменения в активатор. В списке активаторов добавится новая строка.
9) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

2.2.2 Демонстрация работы сценария

Описаный выше сценарий загружает в БД mds_sandbox_pgssql в таблицу test_table_2 данные, поступившие через REST API.

Для добавления записей в БД необходимо:

1) Определить IP адрес сервера на котором запущен активатор. Для этого необходимо:
    1. В разделе «Конфигуратор инфраструктуры» открыть кластер, на котором работает активатор.
    2. Во вкладке Серверы кластера найти значение IP адреса сервера.

Примечание: *Если в разделе «Активаторы» навести указатель на статус запущенного активатора, во всплывающей форме отобразится IP адрес сервера.*

2) Определить порт на котором запущен активатор. Для этого необходимо:
    1. В разделе «Активаторы» открыть запущенный активатор.
    2. В разделе сетевые настройка найти значение порта.

3) В адресной строке интернет-браузера ввести запрос следующего типа:

    [IP адрес]:[port]/?name=[значение для первой колонки]&job=[значение для второй колонки]

    Например: http://192.168.88.60:8089/?name= Duane&job=Director

    Запрос такого вида добавит в таблицу колонку «name» значение Duane, а в колонку «job» - Director.

4) В появившейся форме ввести логин и пароль, указанные в разделе аутентификация активатора.
5) В окне интернет-браузера появится информация о количестве добавленных строк.

6) Далее, можете подключиться к БД и удостовериться что данные добавлены в таблицу.

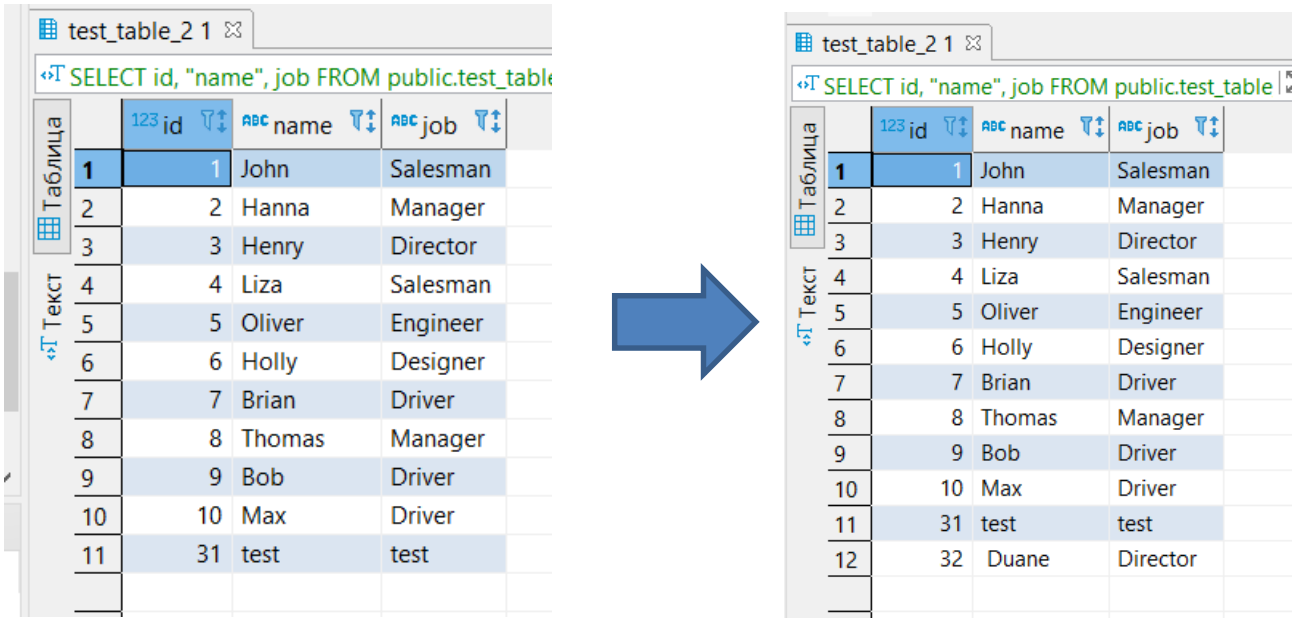


Рисунок 12 – Пример получаемых данных

2.3 Предоставление REST API на базе обработки данных из нескольких источников

2.3.1 Настройка сценария

Цель операции – Получение обработанной информации от нескольких БД через REST API.

Обработка данных происходит на сервере Системы. Поэтому информация может быть получена из разных типов БД и приведена к унифицированному формату.

Данный сценарий подходит для взаимодействия со следующими встроенными типами источников данных:

- Hive;
- MSSQL;
- MySQL;
- Oracle;
- PostgreSQL;

– DB2.

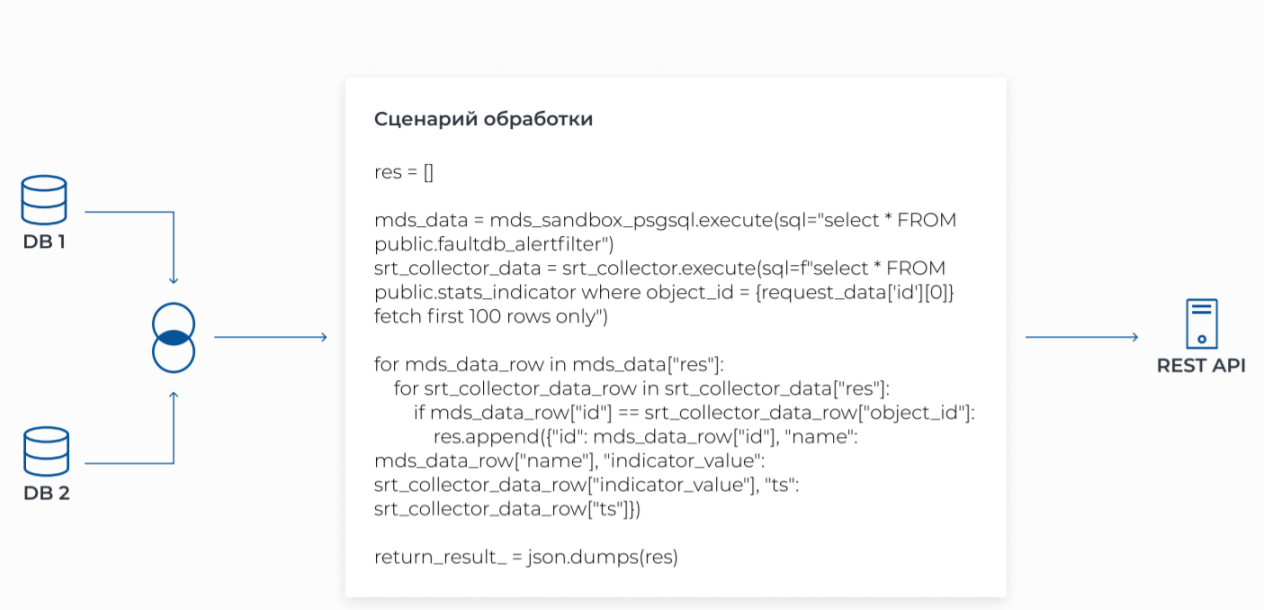


Рисунок 13 – Принцип работы сценария

Сценарий позволяет реализовывать обработку данных из разных типов БД с выводом результата в унифицированном формате.

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   – Datasources admin;
   – Scenarios admin;
   – Activators admin.
2. В системе должен быть подключены и настроены вычислительные мощности.
3. Система должна быть подключена к нужным источникам данных.

Последовательность действий:

1) Авторизуйтесь в Системе с правами Администратора системы.
2) Перейдите пункт меню **«Источники данных».**
3) Выполните настройку подключения ко всем источникам данных (если соединение уже установлено проверьте что оно активно).
4) Перейдите пункт меню **«Сценарии автоматизации».**

5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источников. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

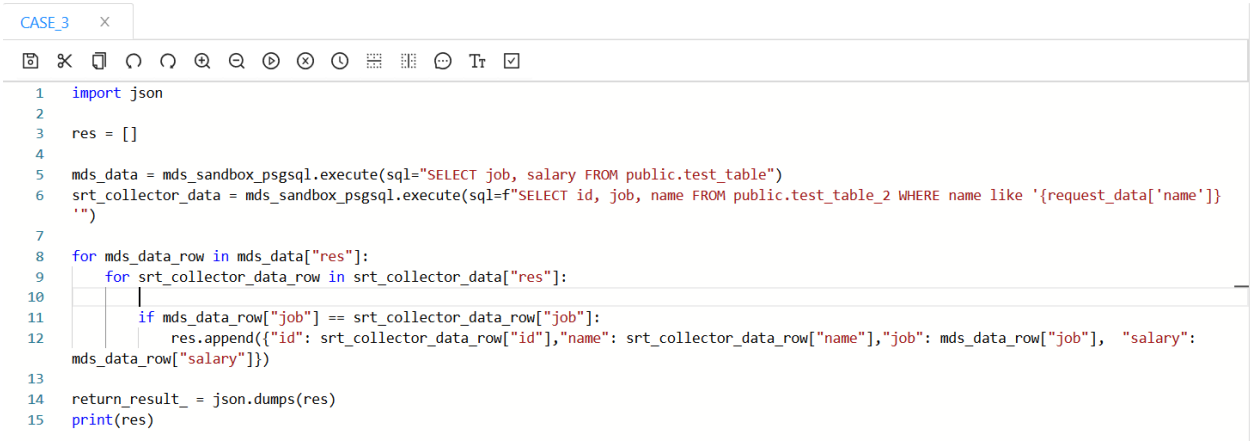


Рисунок 14 - Программный код Сценария автоматизации

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json

res = []

mds_data = mds_sandbox_psgsql.execute(sql="SELECT job, salary FROM public.test_table")
srt_collector_data = mds_sandbox_psgsql.execute(sql=f"SELECT id, job, name FROM
public.test_table_2 WHERE name like '{request_data['name']}'")

for mds_data_row in mds_data["res"]:
   for srt_collector_data_row in srt_collector_data["res"]:
      if mds_data_row["job"] == srt_collector_data_row["job"]:
         res.append({"id": srt_collector_data_row["id"],"name":
srt_collector_data_row["name"],"job": mds_data_row["job"],  "salary": mds_data_row["salary"]})

return_result_ = json.dumps(res)
print(res)
```

В исполняемом коде заложена следующая логика работы Системы:

– *Система обращается к источнику данных mds_sandbox_pgssql и получает данные из таблиц test_table и test_table_2;*

- *Выполняется объединение таблиц по заданному условию (одинаковые значения job);*
- *Осуществляет фильтрацию по значению name;*
- *Трансформирует полученные данные в формат JSON;*
- *Возвращает данные пользователю через REST API.*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте активатор типа HTTP, работающий по сценарию автоматизации созданном на предыдущем шаге.

Пример: *Для создания активатора типа HTTP, работающего на Кластере 1 по сценарию CASE_3 необходимо:*

- *Выбрать из разворачивающегося списка тип активатора – HTTP;*
- *Выбрать из разворачивающегося списка кластер – Кластер 1;*
- *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
- *Назначить таймаут выполнения сценария;*
- *Заполнить раздел аутентификация:*
  - *В поле password задайте пароль для доступа к HTTP;*
  - *В поле username задайте имя пользователя для доступа к HTTP.*
- *Заполнить раздел сетевые настройки:*
  - *В поле port введите номер порта;*
  - *В поле timeout укажите таймаут.*
- *Заполнить раздел параметры обрабатываемых HTTP-запросов:*
  - *Выберете из разворачивающегося списка request_type тип запроса;*
  - *Выберете из разворачивающегося списка content_type_out тип выходных данных;*
  - *Выберете из разворачивающегося списка content_type тип входных данных.*
- *Нажать кнопку*  *для сохранения активатора и ввести имя активатора;*
- *Дождаться пока активатор появится в списке активаторов;*
- *Нажать кнопку*  *для запуска активатора;*
- *Дождаться пока статус активатора в списке станет активным.*

8) Сохраните внесенные изменения в активатор. В списке активаторов добавится новая строка.

9) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

2.3.2 Демонстрация работы сценария

Описаный выше сценарий выгружает из БД mds_sandbox_pgssql данные таблиц *test_table* и test_table_2, выполняет объединение по условию из сценария автоматизации и выводит их в формате JSON при активации по HTTP.

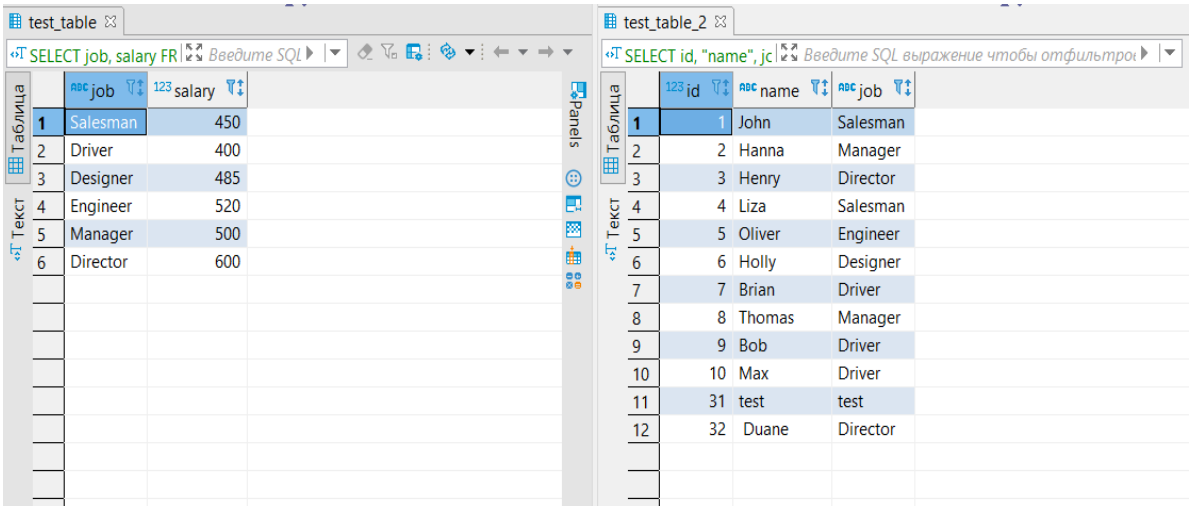


Рисунок 15 – Наполнение таблиц

Для получения данных через REST API необходимо :

1) Определить IP адрес сервера на котором запущен активатор. Для этого необходимо:
   1. В разделе «Конфигуратор инфраструктуры» открыть кластер, на котором работает активатор.
   2. Во вкладке Серверы кластера найти значение IP адреса сервера.

Примечание: *Если в разделе «Активаторы» навести указатель на статус запущенного активатора, во всплывающей форме отобразится IP адрес сервера.*

2) Определить порт на котором запущен активатор. Для этого необходимо:
    1. В разделе «Активаторы» открыть запущенный активатор.
    2. В разделе сетевые настройки найти значение порта.

3) В адресной строке интернет-браузера ввести запрос следующего типа:

    [IP адрес]:[port] /?name=[значение для фильтра колонке name]

    Например: http://192.168.88.60:8080/?name=John

4) В появившейся форме ввести логин и пароль, указанные в разделе аутентификация активатора.
5) В окне интернет-браузера появится информация из БД в формате JSON.

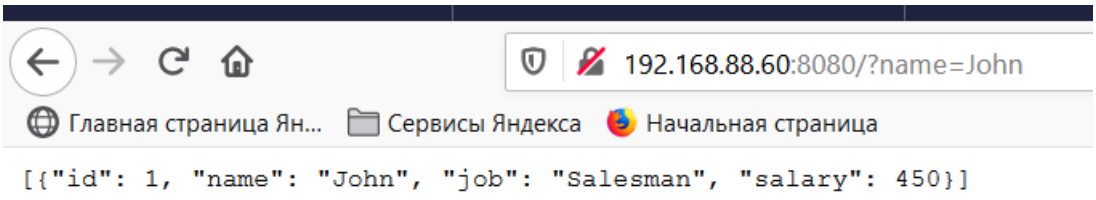


Рисунок 16 – Пример получаемых данных

2.4 РАБОТА АКТИВАТОРА ПО РАСПИСАНИЮ

2.4.1 Настройка сценария

Цель операции – получение периодической информации об изменениях в БД

Рисунок 17 – Принцип работы сценария

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   - Datasources admin;
   - Scenarios admin;
   - Activators admin.
2. В системе должен быть подключены и настроены вычислительные мощности
3. Система должна быть подключена к нужным источникам данных.

Последовательность действий:

1) Авторизуйтесь в Системе с правами Администратора системы.
2) Перейдите пункт меню **«Источники данных».**
3) Выполните подключение к БД источника данных (если соединение уже установлено проверьте что оно активно).
4) Перейдите пункт меню **«Сценарии автоматизации».**
5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источника. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

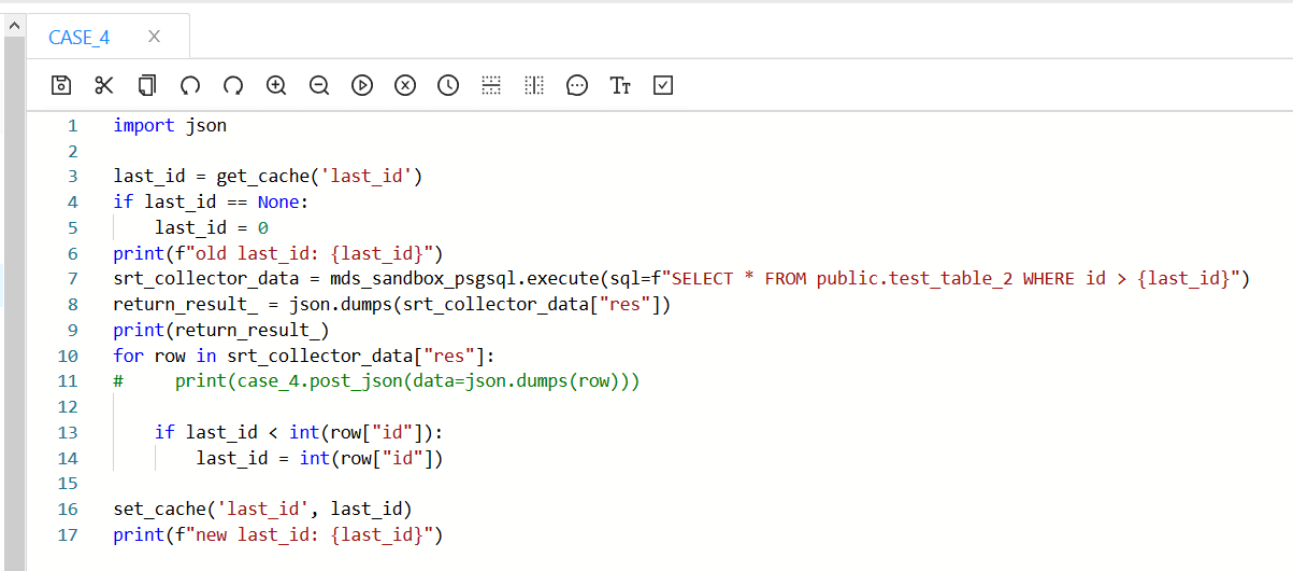


Рисунок 18 – Программный код сценария

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json

last_id = get_cache('last_id')
if last_id == None:
    last_id = 0
print(f"old last_id: {last_id}")
srt_collector_data = mds_sandbox_psgsql.execute(sql=f"SELECT * FROM public.test_table_2
WHERE id > {last_id}")
return_result_ = json.dumps(srt_collector_data["res"])
print(return_result_)
for row in srt_collector_data["res"]:

    if last_id < int(row["id"]):
        last_id = int(row["id"])

set_cache('last_id', last_id)
print(f"new last_id: {last_id}")
```

В исполняемом коде заложена следующая логика работы Системы:

- *Система обращается к источнику данных;*
- *Сохраняет в памяти последнее значение колонки Id;*
- *При изменении значения Система фиксирует факт изменений и перезаписывает последнее значение Id.*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте активатор типа Periodic (по расписанию).

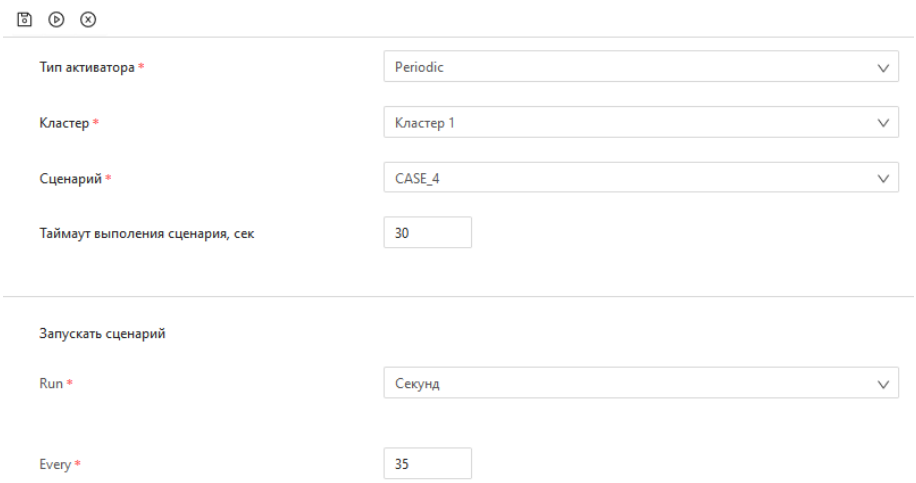


Рисунок 19 – Создание активатора

Пример: *Для создания активатора типа Periodic, работающего на Кластере 1 по ранее созданному сценарию необходимо:*

- *Выбрать из разворачивающегося списка тип активатора –Periodic (по расписанию);*
- *Выбрать из разворачивающегося списка кластер – Кластер 1;*
- *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
- *Назначить таймаут выполнения сценария;*
- *Указать периодичность запуска сценария;*
- *Нажать кнопку для сохранения активатора и ввести имя активатора;*
- *Дождаться пока активатор появится в списке активаторов;*
- *Нажать кнопку для запуска активатора;*
- *Дождаться пока статус активатора в списке станет активным.*

8) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

9) В журнале выполнения активатора отобразится информация о периодичности его выполнения и результатах работы.

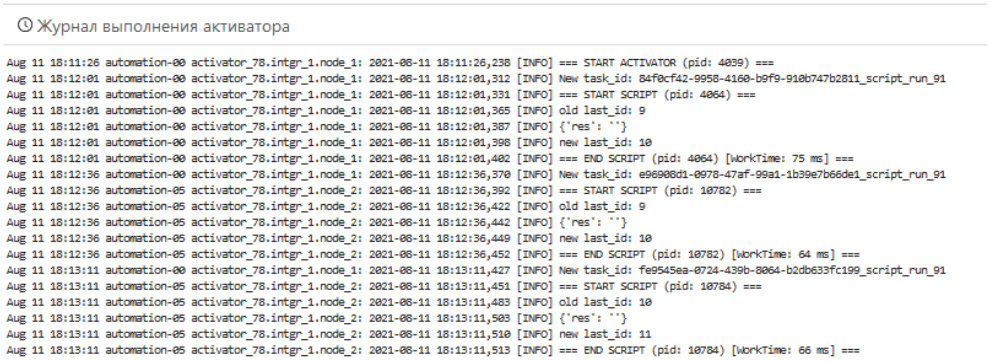


Рисунок 20 – Журнал выполнения активатора.

10) С заданной периодичностью в журнале выпонения активатора будет выводиться инфориация об измнениях в БД.

2.4.2 Демонстрация работы сценария

Описаный выше сценарий каждые 35 секунд регистрирует изменения в БД mds_sandbox_pgssql таблице test_table_2 и автоматически выводит данные, которые были добавлены в источник данных.

Для работы сценария необходимо запустить соответствующий активатор.

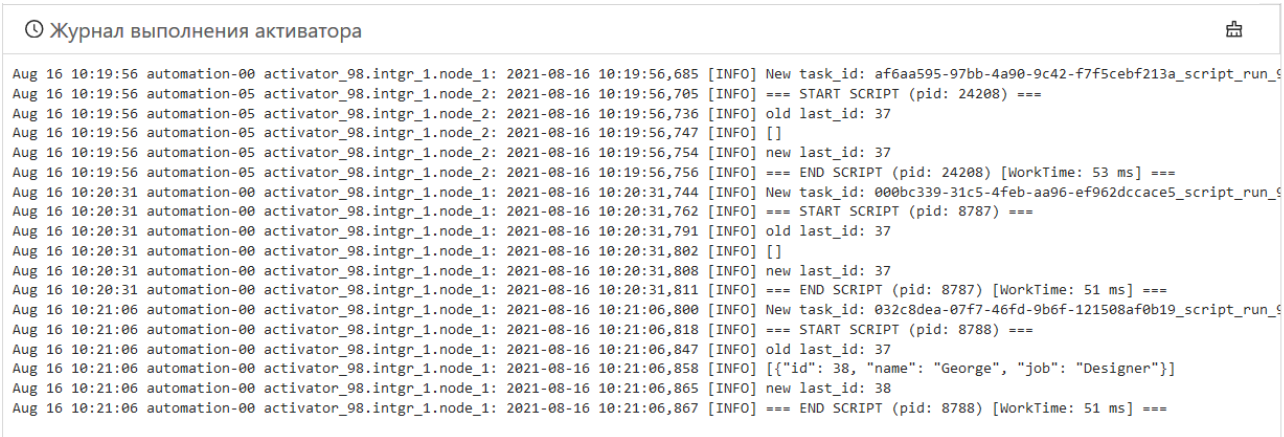


Рисунок 21 – Журнал выполнения активатора

В журнале выполнения активатора отобразится информация о внесенных изменениях в таблице test_table_2 за каждые 35 секунд.

Пример: *2021-08-16 10:21:06 зафиксировано изменение в целевой таблице - была добавлена строка: [{"id": 38, "name": "George", "job": "Designer"}].*

2.5 Вывод данных из БД по запросу из Telegram-бота

2.5.1 Настройка сценария

Цель операции – Получение обработанной информации от БД через Telegram.

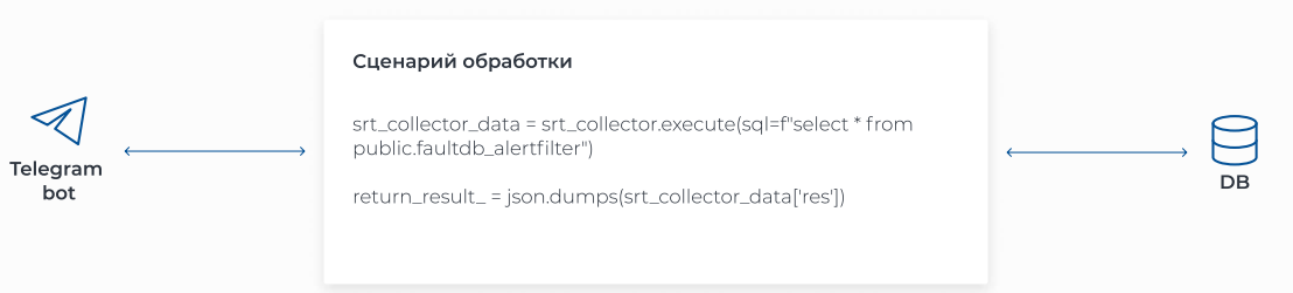


Рисунок 22 – Принцип работы сценария

Сценарий позволяет реализовывать обработку данных из разных типов БД с выводом результата в унифицированном формате.

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   - Datasources admin;
   - Scenarios admin;
   - Activators admin.
2. В системе должен быть подключены и настроены вычислительные мощности.
3. Система должна быть подключена к нужным источникам данных.

Последовательность действий:

1) Авторизуйтесь в Системе с правами Администратора системы.
2) Перейдите пункт меню **«Источники данных».**

3) Выполните подключение к источнику данных типа Telegram Chat (если соединение уже установлено проверьте что оно активно).
4) Перейдите пункт меню **«Сценарии автоматизации».**
5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источника. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

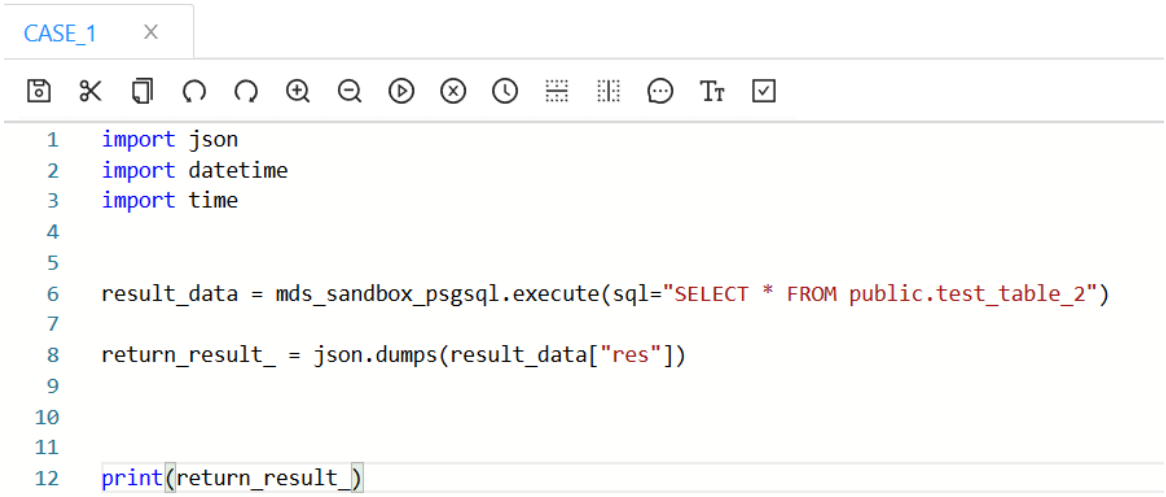


Рисунок 23 – Программный код Сценария автоматизации

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json
import datetime
import time

result_data = mds_sandbox_psgsql.execute(sql="SELECT * FROM public.test_table_2")
return_result_ = json.dumps(result_data["res"])

print(return_result_)
```

В исполняемом коде заложена следующая логика работы Системы:

- *Система обращается к подключенному источнику данных mds_sandbox_pgssql и получает данные из таблицы test_table_2;*
- *Трансформирует полученные данные в формат JSON;*
- *Возвращает данные пользователю.*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте активатор типа Telegram Bot.

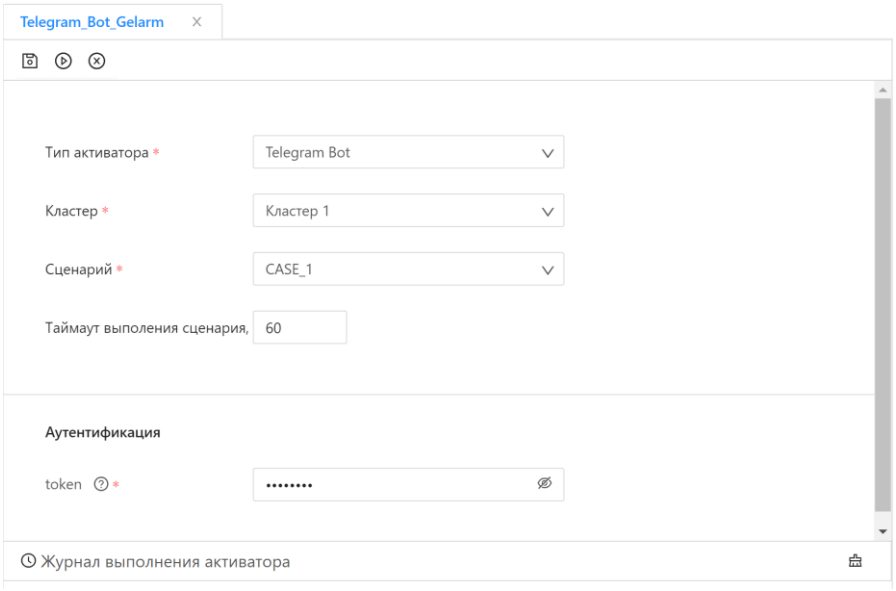


Рисунок 24 – Создание активатора

Пример: *Для создания активатора типа* Telegram Bot*, работающего на Кластере 1 по ранее созданному сценарию необходимо:*

- *Выбрать из разворачивающегося списка тип активатора – Telegram Bot;*
- *Выбрать из разворачивающегося списка кластер – Кластер 1;*
- *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
- *Назначить таймаут выполнения сценария;*
- *Указать токен аутентификации.*
- *Нажать кнопку*  *для сохранения активатора и ввести имя активатора;*
- *Дождаться пока активатор появится в списке активаторов;*
- *Нажать кнопку*  *для запуска активатора;*
- *Дождаться пока статус активатора в списке станет активным.*

8) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

2.5.2 Демонстрация работы сценария

Описаный выше сценарий выгружает из БД mds_sandbox_pgssql данные таблицы *test_table_2* в формате JSON при активации по запросу в Telegram.

Для получения данных через через Telegram-бота необходимо :

1) Направить в адрес Telegram-бота, токен которого указан в настройках активатора команду /data.
2) В ответном сообщении появится информация из БД в формате JSON.

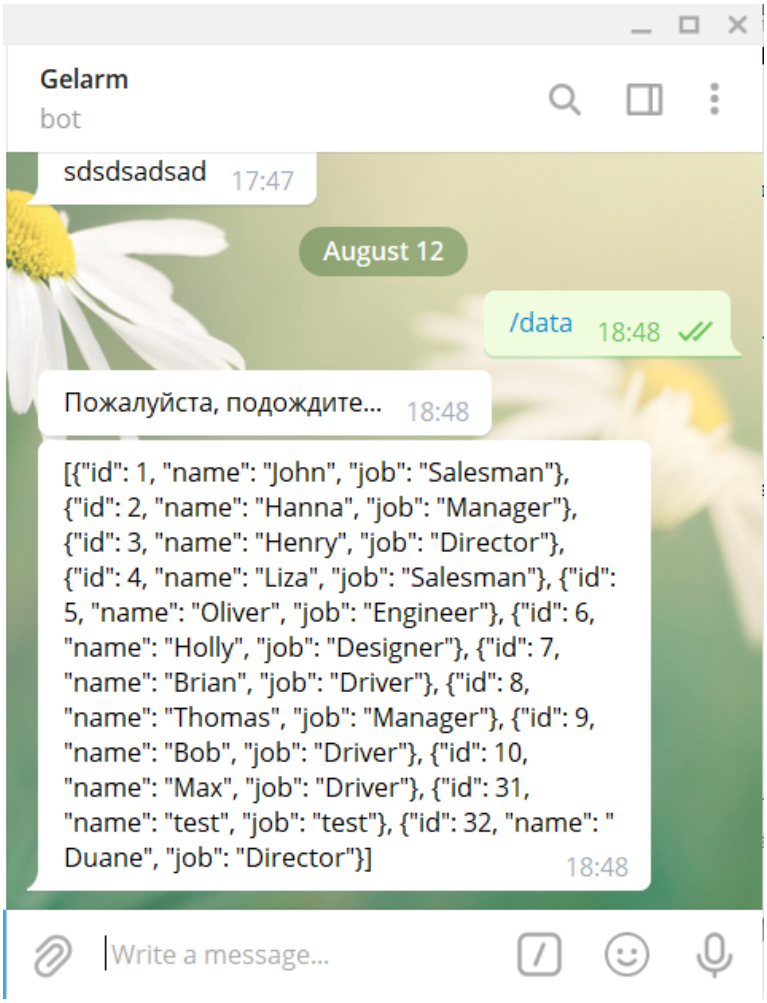


Рисунок 25 – Пример получаемых данных

Система поддерживает гибкую настройку сценариев. При необходимости, возможно задать любую логику обработки данных, поступающих на активатор.

Пример: *можно создать сценарий, поддерживающий распознавание запроса в Telegram с последующей фильтрацией по заданному условию.*

Рисунок 26 – Пример получаемых данных

2.6 АСИНХРОННАЯ ОБРАБОТКА HTTP ЗАПРОСОВ

2.6.1 Настройка сценария

Цель операции – Асинхронное получение обработанной информации от БД через Swagger API.

Настройка сценария осуществляется аналогично п.2.1 Предоставление REST API на базе обработки данных из БД. Единственное отличие – необходимо выбрать тип активатора «HTTP aiohttp»

Тип активатора HTTP aiohttp использует интерфейс Swagger API.

2.6.2 Демонстрация работы сценария

Описаный выше сценарий выгружает из БД mds_sandbox_pgssql данные таблицы *test_table_2* в формате JSON при активации по Swagger API.

test_table_2

SELECT id, "name", job FF | Введите SQL

Таблица | Текст | Запись

	id	name	job
1	1	John	Salesman
2	2	Hanna	Manager
3	3	Henry	Director
4	4	Liza	Salesman
5	5	Oliver	Engineer
6	6	Holly	Designer
7	7	Brian	Driver
8	8	Thomas	Manager
9	9	Bob	Driver
10	10	Max	Driver
11	31	test	test
12	32	Duane	Director
13	33	Bill	Salesman
14	34	Jack	Driver
15	35	Audrey	Driver
16	36	Rebecca	Designer
17	37	Sally	Designer
18	38	George	Designer

Рисунок 27 – Содержимое таблицы test_table_2

Для получения данных через Swagger API необходимо :

1) Определить IP адрес сервера на котором запущен активатор. Для этого необходимо:
    1. В разделе «Конфигуратор инфраструктуры» открыть кластер, на котором работает активатор.
    2. Во вкладке Серверы кластера найти значение IP адреса сервера.

Примечание: *Если в разделе «Активаторы» навести указатель на статус запущенного активатора, во всплывающей форме отобразится IP адрес сервера.*

2) Определить порт на котором запущен активатор. Для этого необходимо:
    1. В разделе «Активаторы» открыть запущенный активатор.

2. В разделе сетевые настройки найти значение порта.

3) В адресной строке интернет-браузера ввести запрос следующего типа:

   [IP адрес]:[port]/api/doc

   Например: http://192.168.88.60:8127/api/doc

4) В появившейся форме ввести логин и пароль, указанные в разделе аутентификация активатора.
5) В окне интернет-браузера появится интерфейс Swagger API.

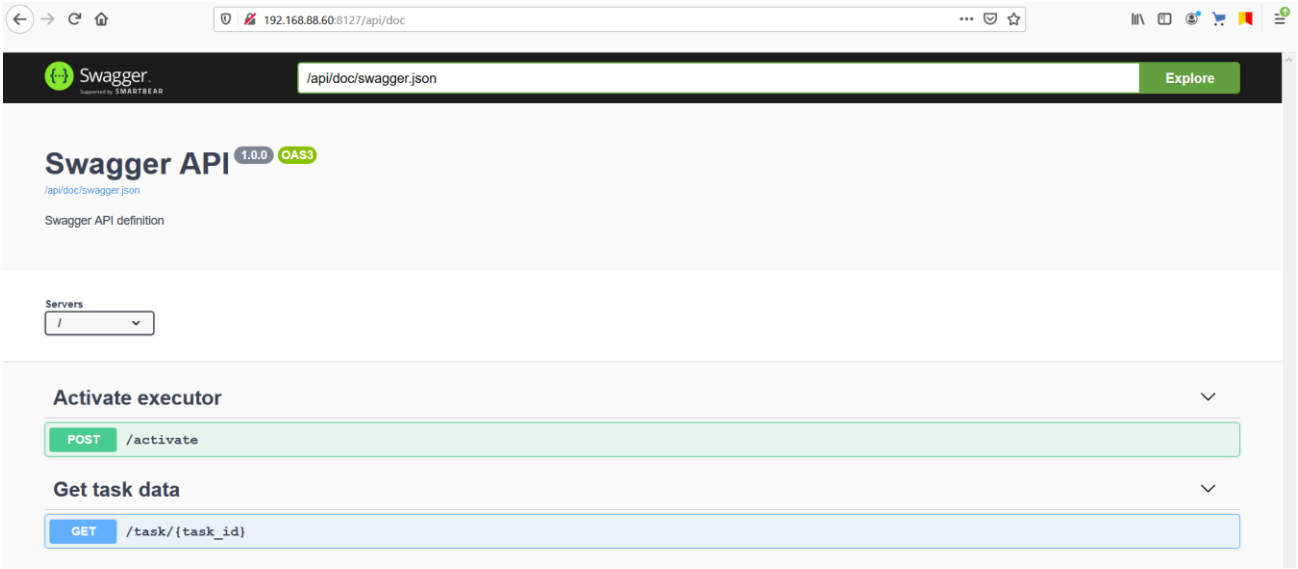


Рисунок 28 – Интерфейс Swagger API

6) Интерфейс поддерживает следующие методы:
   – POST – отправка запроса на получение данных;
   – GET – получение данных.

7) Для отправки запроса на выполнение сценария нажмите на команду «POST».
8) Развернется поле для работы с методом «POST».
9) Нажмите кнопку «Try it out».
10) Нажмите кнопку «Execute».
11) Программа выполнит запрос к источнику данных и сохранит данные таблицы с присвоением уникального номера запроса «task_id».

12) Скопируйте «task_id».

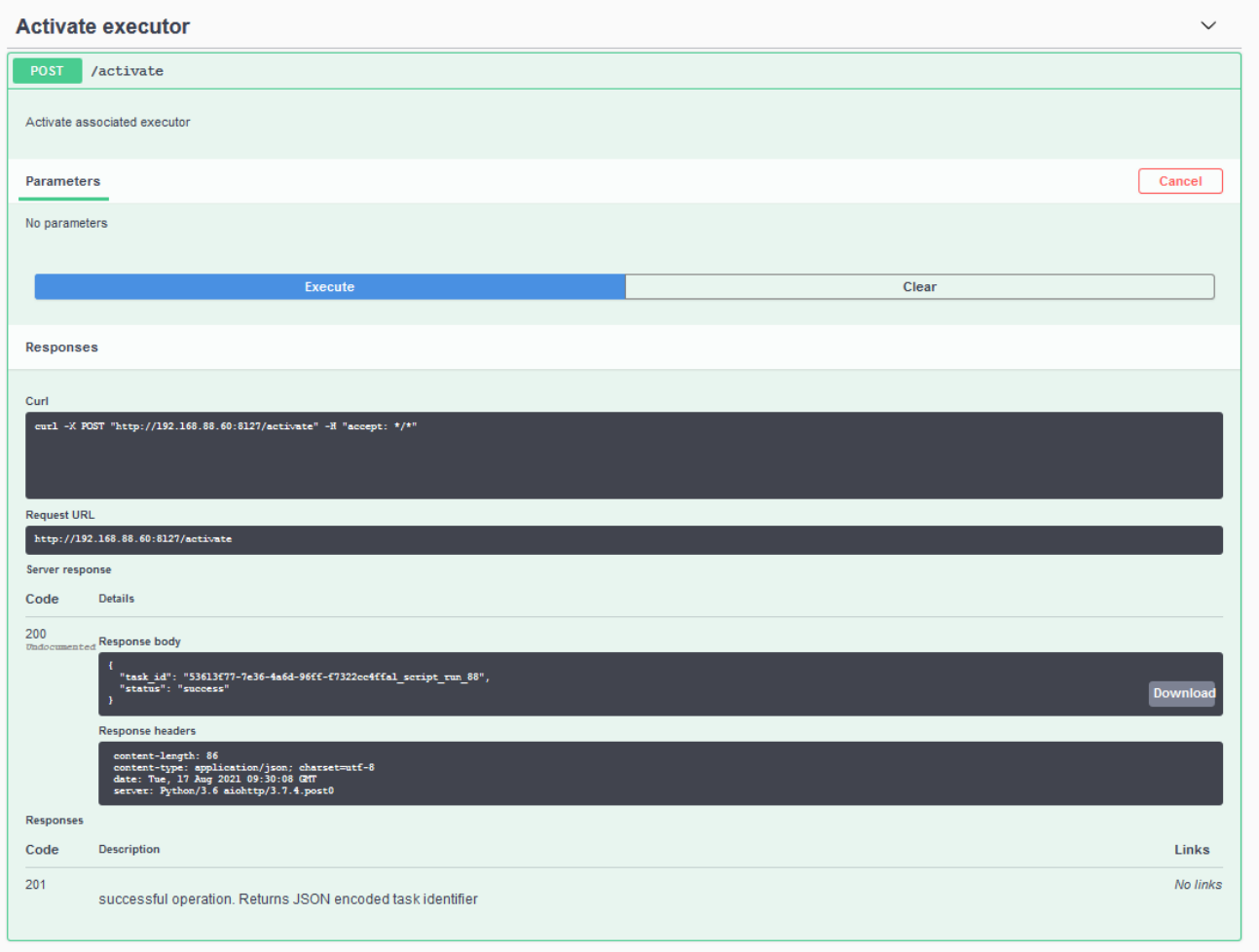


Рисунок 29 – Интерфейс Swagger

13) Для отправки запроса на получение данных нажмите на команду «GET».

14) Развернется поле для работы с методом «GET».

15) Нажмите кнопку «Try it out».

16) В поле «task_id» введите уникальный номер зпроса, полученный в методе «POST»

17) Нажмите кнопку «Execute».

18) В поле Response body появится информация из БД в формате JSON.

Примечание: *Получить данные также можно путем ввода в строке интернет-браузера запроса следующего типа: [IP адрес] :[port]/task/[task_id].*

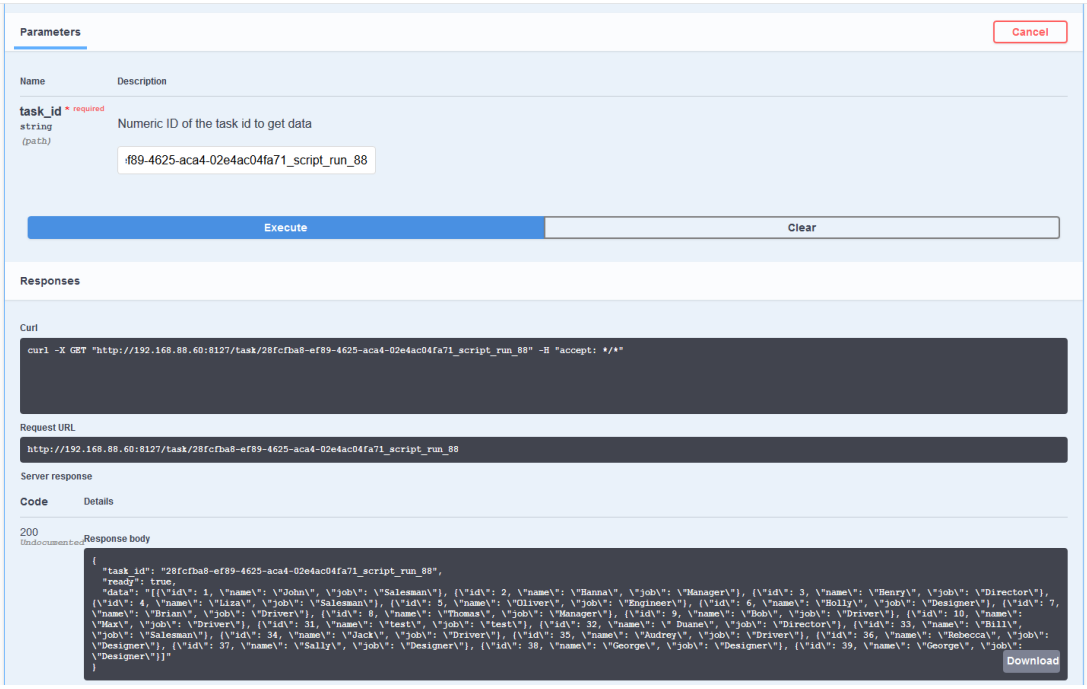


Рисунок 30 – Получение данных через Swagger

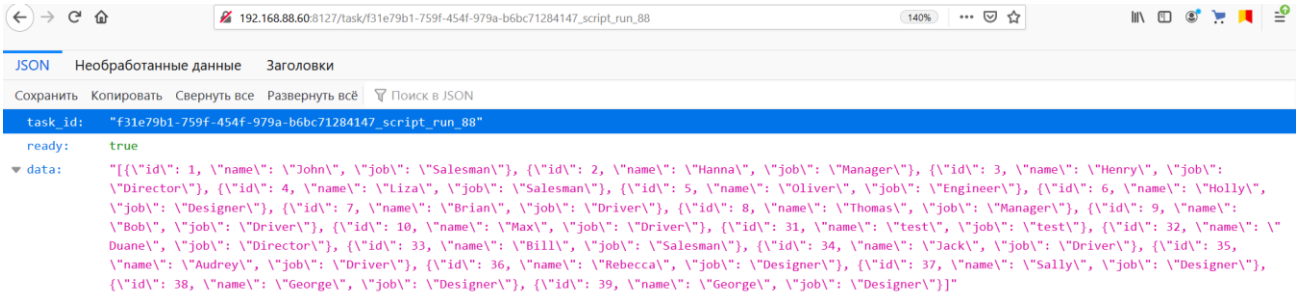


Рисунок 31 – Получение данных через HTTP

2.7 ОБОГАЩЕНИЕ ДАННЫХ

2.7.1 Настройка сценария

Цель операции – Обогащение получаемых данных.

Обработка данных происходит на сервере Системы. Поэтому информация может быть получена из разных типов БД.

Сценарий представляет собой объединение вышеописанных сценариев: Работа активатора по расписанию, Предоставление REST API на базе обработки данных из нескольких источников, Запись данных в БД, поступивших через REST API.

Предусловия выполнения:

1. Сценарий выполняется под учетной записью пользователя, которому назначены следующие роли:
   - Datasources admin;
   - Scenarios admin;
   - Activators admin.
2. В системе должен быть подключены и настроены вычислительные мощности.
3. Система должна быть подключена к нужным источникам данных.

Последовательность действий:

1) Авторизуйтесь в Системе с правами Администратора системы.
2) Перейдите пункт меню **«Источники данных».**
3) Выполните настройку подключения ко всем источникам данных (если соединение уже установлено проверьте что оно активно).
4) Перейдите пункт меню **«Сценарии автоматизации».**
5) Создайте новый сценарий, описывающий действия с данными, получаемыми из источников. Если сценарий автоматизации уже существует (к примеру, ранее разрабатывался для подобной интеграции), то этот шаг можно пропустить.

CASE_5

```
import json

res = []

last_id = get_cache("last_id")
if last_id == None:
    last_id = 0
# last_id = 0
print(f"old last_id: {last_id}")

max_id_get = mds_sandbox_psgsql.execute(sql="SELECT MAX(id) as max_id FROM public.test_table_2")
max_id_res = max_id_get.get("res")[0]
max_id = max_id_res.get("max_id")
print(f"max id: {max_id}")

if max_id > last_id:
    srt_collector_data = mds_sandbox_psgsql.execute(sql=f"SELECT tt2.id, tt2.name, tt2.job, tt.salary FROM public.test_table_2 tt2 join public.test_table tt on tt2.job = tt.job WHERE tt2.id > {last_id}")
    return_result_ = json.dumps(srt_collector_data["res"])
    # print(return_result_)
    sql = f"INSERT INTO public.test_table_3 (name, job, salary) VALUES"
    for row in srt_collector_data["res"]:
        if last_id < int(row["id"]):
            last_id = int(row["id"])
        # print(row["name"])
        sql += "('"+str(row['name'])+"','"+str(row['job'])+"',"+str(row['salary'])+"),"
    sql = sql[:-1]
    print(sql)
    mds_sandbox_psgsql.execute(sql=sql)

    set_cache('last_id', last_id)
```

Рисунок 32 - Программный код Сценария автоматизации

Пример: *Сценарий автоматизации для данной задачи должен быть запрограммирован следующим образом:*

```
import json

res = []

last_id = get_cache("last_id")
if last_id == None:
    last_id = 0
print(f"old last_id: {last_id}")

max_id_get = mds_sandbox_psgsql.execute(sql="SELECT MAX(id) as max_id FROM public.test_tabl
e_2")
max_id_res = max_id_get.get("res")[0]
max_id = max_id_res.get("max_id")
print(f"max id: {max_id}")

if max_id > last_id:
    srt_collector_data = mds_sandbox_psgsql.execute(sql=f"SELECT tt2.id, tt2.name, tt2.job, tt.salary
FROM public.test_table_2 tt2 join public.test_table tt on tt2.job = tt.job WHERE tt2.id > {last_id}")
    return_result_ = json.dumps(srt_collector_data["res"])

    sql = f"INSERT INTO public.test_table_3 (name, job, salary) VALUES"
    for row in srt_collector_data["res"]:
        if last_id < int(row["id"]):
            last_id = int(row["id"])
        sql += "('"+str(row['name'])+"','"+str(row['job'])+"',"+str(row['salary'])+"),"
    sql = sql[:-1]
    print(sql)
    mds_sandbox_psgsql.execute(sql=sql)

    set_cache('last_id', last_id)
    print(f"new last_id: {last_id}")
```

В исполняемом коде заложена следующая логика работы Системы:

- *Система обращается к источнику данных test_table_2;*
- *Сохраняет в памяти последнее значение колонки Id;*
- *При изменении значения Система фиксирует факт изменений и перезаписывает последнее значение Id;*
- *При фиксации изменений система обогащает новые данные данными из таблицы test_table;*

– *Выполняется INSERT обогащенных данных в таблицу test_table_3.*

6) Перейдите пункт меню **«Активаторы»**.
7) Настройте активатор типа HTTP, работающий по сценарию автоматизации созданном на предыдущем шаге.

Пример: *Для создания активатора типа Periodic, работающего на Кластере 1 по сценарию CASE_5 необходимо:*

– *Выбрать из разворачивающегося списка тип активатора – Periodic(по расписанию);*
– *Выбрать из разворачивающегося списка кластер – Кластер 1;*
– *Выбрать из разворачивающегося списка сценарий, созданный на предыдущем шаге;*
– *Назначить таймаут выполнения сценария;*
– *Указать периодичность запуска сценария;*
– *Нажать кнопку*  *для сохранения активатора и ввести имя активатора;*
– *Дождаться пока активатор появится в списке активаторов;*
– *Нажать кнопку*  *для запуска активатора;*
– *Дождаться пока статус активатора в списке станет активным*

8) Сохраните внесенные изменения в активатор. В списке активаторов добавится новая строка.
9) Как только у активатора статус сменится на «Готов к работе» активатор будет готов к работе с источником данных по сценарию автоматизации.

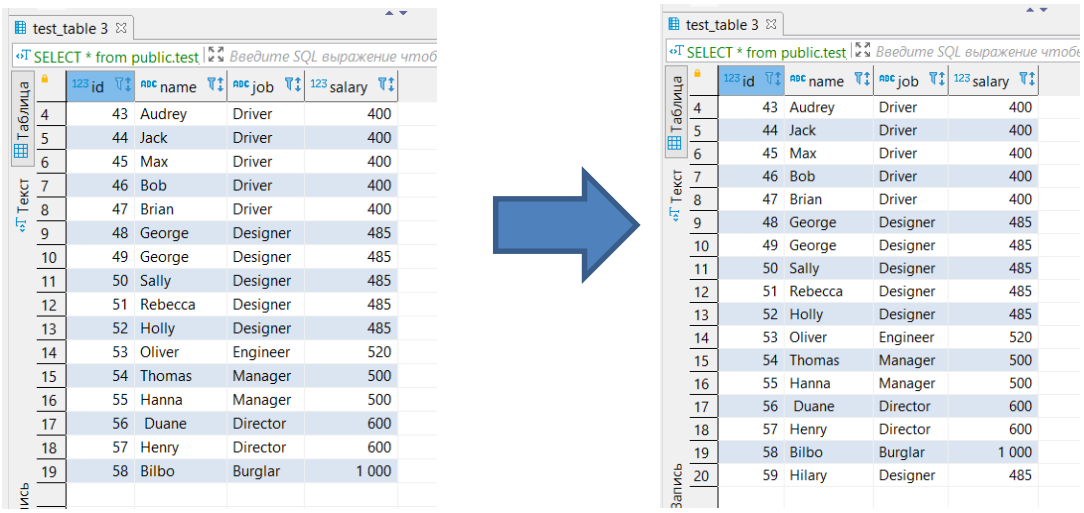
Примечание: *Удостоверьтесь, что статус подключения к источнику данных сменился на «Готов к работе».*

2.7.2 Демонстрация работы сценария

Описаный выше сценарий записывает обогащенные данные из test_table_2 в test_table_3.
Для получения обогащенных данных необходимо :

1) Добавить сырые данные в test_table_2.
2) Для этого можно воспользоваться сценарием «Запись данных в БД, поступивших через REST API».

3) Добавим данные: name - Hilary, job – Designer. Для этого введем адресной строке интернет-браузера 192.168.88.66:8089/?name=Hilary&job=Designer.
4) Система автоматически зафиксирует новые данные, обогатит их информацией о зарплате сотрудника и запишет в целевую таблицу.



test_table 3

SELECT * from public.test

	id	name	job	salary
4	43	Audrey	Driver	400
5	44	Jack	Driver	400
6	45	Max	Driver	400
7	46	Bob	Driver	400
8	47	Brian	Driver	400
9	48	George	Designer	485
10	49	George	Designer	485
11	50	Sally	Designer	485
12	51	Rebecca	Designer	485
13	52	Holly	Designer	485
14	53	Oliver	Engineer	520
15	54	Thomas	Manager	500
16	55	Hanna	Manager	500
17	56	Duane	Director	600
18	57	Henry	Director	600
19	58	Bilbo	Burglar	1 000

test_table 3

SELECT * from public.test

	id	name	job	salary
4	43	Audrey	Driver	400
5	44	Jack	Driver	400
6	45	Max	Driver	400
7	46	Bob	Driver	400
8	47	Brian	Driver	400
9	48	George	Designer	485
10	49	George	Designer	485
11	50	Sally	Designer	485
12	51	Rebecca	Designer	485
13	52	Holly	Designer	485
14	53	Oliver	Engineer	520
15	54	Thomas	Manager	500
16	55	Hanna	Manager	500
17	56	Duane	Director	600
18	57	Henry	Director	600
19	58	Bilbo	Burglar	1 000
20	59	Hilary	Designer	485

Рисунок 33 – Пример получаемых данных

5) Система автоматически зафиксирует новые данные, обогатит их информацией о зарплате сотрудника и запишет в целевую таблицу.
6) При этом в журнале выполнения активатора появится запись о выполненом INSERT.

```
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,002 [INFO] New task_id: 8449dccc-bc19-4f3f-891e-312082d8e8bf_script_run_165
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,023 [INFO] === START SCRIPT (pid: 24220) ===
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,048 [INFO] old last_id: 40
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,061 [INFO] max id: 41
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,065 [INFO] INSERT INTO public.test_table_3 (name, job, salary) VALUES('Hilary','Designer',485)
Aug 17 15:01:26 automation-05 activator_100.intgr_1.node_2: 2021-08-17 15:01:26,077 [INFO] new last_id: 41
```

Рисунок 34 – Журнал выполнения активатора