

**Программный комплекс гарантированной доставки и
трансформации данных «сервисная шина «СинтезПлюс» для
организации взаимодействия различных приложений и сервисов**

Руководство пользователя

Листов 14

Иннополис, 2022 Г.

АННОТАЦИЯ

Настоящий документ является руководством пользователя программного комплекса гарантированной доставки и трансформации данных «сервисная шина «СинтезПлюс» для организации взаимодействия различных приложений и сервисов.

СОДЕРЖАНИЕ

1. Общие сведения.....	4
2. Функциональные характеристики.....	5
3. Информация, необходимая для установки и эксплуатации.....	9
4. Установка, запуск, остановка и удаление	10
4.1. Настройка передачи данных	10
5. Решение типовых проблем.....	12
6. Контактная информация производителя.....	13
6.1. Юридическая информация	13
6.2. Контактная информация службы технической поддержки	13
Перечень сокращений	14

1. ОБЩИЕ СВЕДЕНИЯ

Полное наименование системы: программного комплекса гарантированной доставки и трансформации данных «сервисная шина «СинтезПлюс» для организации взаимодействия различных приложений и сервисов.

Краткое наименование системы: сервисная шина.

Сервисная шина предназначена для гарантированного информационного обмена между системой и сторонними сервисами.

2. ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Сервисная шина должна обеспечивать:

- доступ к разнородным источникам данных (базы данных Postgres, MSSQL, SOAP-сервисы, REST-сервисы, S3-хранилища данных);
- поддержку синхронного и асинхронного способа вызова сервисов;
- использование защищенного транспорта с гарантированной доставкой сообщений, поддерживающего транзакционную модель;
- статическую и алгоритмическую (условную, опирающуюся на значения передаваемых данных) маршрутизацию сообщений;
- доступ к опубликованным сервисам из сторонних информационных систем (далее – ИС) с помощью готовых или специально разработанных модулей;
- отказоустойчивую конфигурацию;
- разнообразные механизмы контроля и управления (аудиты, протоколирование);
- унифицированный событийно-ориентированный обмен данными между различными ИС;
- развитые механизмы масштабирования.

Сервисная шина состоит из основного модуля (esb-process-application), отвечающего за маршруты передачи и предоставления данных, конфигурационный сервер (esb-config-server) и адаптеров взаимодействия с разнородными источниками данных.

Конфигурационная база данных реализована на основе postgres. Взаимодействие всех модулей сервисной шины с конфигурационной базой данных выполняется через esb-config-server.

Кроме этого, в сервисной шине реализован отдельный модуль (esb-experience-api), который является единой точкой взаимодействия с API шины. Так же реализован модуль планировщик, отвечающий за выполнение передач данных по расписанию.

Структура и функциональные возможности сервисной шины представлены в таблице 1.

Таблица 1 – Структура и функциональные возможности сервисной шины

№	Модуль/ Адаптер	Описание	Функционал
1	Esb-experience-api	Приложение, отвечающее за доступ к модулям/адаптерам ESB по REST	– предоставление доступа к ресурсам по протоколу REST; – запуск маршрутов интеграции по протоколу REST; – доступ к esb-onfig-server; – доступ к esb-process-application; – разграничение доступа к ESB

Продолжение таблицы 1

№	Модуль/ Адаптер	Описание	Функционал
2	Esb-rest-adapter	Адаптер для интеграции с системами, работающими по протоколу REST	– предоставление универсального (конфигурируемого) доступа к ресурсам, работающим по протоколу REST; – исполнение rest-запросов; – basic-авторизация; – JWT-авторизация
3	Esb-soap-adapter	Адаптер для интеграции с системами, работающими по протоколу SOAP	– предоставление универсального (конфигурируемого) доступа к ресурсам, работающим по протоколу SOAP; – исполнение SOAP-запросов; – basic-авторизация
4	Esb-postgres-adapter	Адаптер для интеграции с базами данных типа Postgres	– предоставление универсального (конфигурируемого) доступа к базам данных типа POSTGRES; – исполнение SQL-скриптов; – вызов хранимых процедур
5	Esb-MsSql-adapter	Адаптер для интеграции с базами данных типа MSSQL	– предоставление универсального (конфигурируемого) доступа к базам данных типа MSSQL; – исполнение SQL-скриптов; – вызов хранимых процедур
6	Esb-Email-adapter	Адаптер для отправки email сообщений	– отправка email по REST-вызову; – отправка email, полученного через JMS-очередь
7	Esb-Telegram-adapter	Адаптер для отправки сообщений в Telegram	– отправка сообщения в чат-бот по REST-вызову
8	Esb-FileServer-adapter	Адаптер для работы с файловыми хранилищами типа S3	– сохранение файлов в S3-хранилище; – скачивание файлов из S3-хранилища; – скачивание части файла из S3-хранилища; – получение списка файлов, которые хранятся в S3-хранилище; – генерация коротких ссылок для скачивания файлов из S3-хранилища
9	Esb-Hdfs-adapter	Адаптер для работы с файловыми хранилищами типа HDFS	– скачивание файлов из HDFS; – загрузка файлов в HDFS

Продолжение таблицы 1

№	Модуль/ Адаптер	Описание	Функционал
10	Esb-process-application	Основное приложение ESB. Отвечает за работу маршрутов интеграции	– исполнение маршрутов интеграции (передача данных от ресурса-источника до ресурса-приемника с различными вариантами обогащения/преобразования исходных данных); – гарантированная доставка (при критической ошибке «участника» маршрута, данные сохраняются в очереди, и, после восстановления, маршрут продолжит выполнение с этапа прерывания); – трансформация данных с использованием скриптового языка Datasonnet; – перекодировка данных при передаче; – получение данных из нескольких системы; – отправка данных в несколько системы; – повторная отправка успешно переданных данных; – повторная отправка непереданных данных; – автоматический перезапуск ошибочных или частично ошибочных передач данных; – сохранение информации по передаче данных (текущий статус и данные для возможности перезапуска); – определение координат ресурсов на основе конфигурации из базы данных; – логирование в определенном формате для возможности просмотра и поиска данных в Graylog; – отправка информации на почту; – отправка данных в Sentry

Продолжение таблицы 1

№	Модуль/ Адаптер	Описание	Функционал
11	Esb-Config-server	Приложение, отвечающее за хранение/управление конфигурациями приложений и конфигураций маршрутов	– хранение/управление конфигурациями маршрутов интеграции; – хранение/управление конфигурациями приложений; – хранение/управление скриптами трансформаций; – запуск маршрута; – получение описания маршрута; – управление ресурсами и интерфейсами (endpoints); – хранение/управление конфигурациями планировщика задач; – запуск и остановка задачи планировщика; – получение описания задачи планировщика
12	Esb-Scheduler-application	Приложение, отвечающее за запуск маршрутов по расписанию	– отправка сообщений для запуска маршрутов в JMS-очередь по расписанию, указанному в конфигурации маршрута; – логирование в определенном формате для возможности просмотра и поиска данных в Graylog; – отправка информации на почту; – отправка данных в Sentry

3. ИНФОРМАЦИЯ, НЕОБХОДИМАЯ ДЛЯ УСТАНОВКИ И ЭКСПЛУАТАЦИИ

Для установки и настройки необходимо выполнить следующие действия:

- развернуть контейнеры образов в docker или Kubernetes (команды по разворачиванию представлены в инструкции по установке);
- развернуть базу данных postgres и создать учетную запись с возможностью создания базы данных и сущностей (команды по разворачиванию представлены в инструкции по установке);
- развернуть брокер сообщений Active MQ;
- настроить контейнеры приложений через переменные окружения на развернутую базу данных и брокер сообщений;
- обеспечить доступ между компонентами шины до esb-config-server;
- обеспечить доступ к базе данных для esb-config-server.

После запуска esb-config-server в базе данных должна автоматически развернуться база данных esb_repos с необходимым набором сущностей (таблиц и представлений) для работы сервисной шины.

4. УСТАНОВКА, ЗАПУСК, ОСТАНОВКА И УДАЛЕНИЕ

Настройка, запуск и удаление передач данных производится через REST API сервисной шины (Рисунок 1). После запуска сервисной шины должно быть доступно swagger-описание методов сервисной шины. Адрес, по которому должно быть доступно описание:

`http://[IP_адрес_сервера]:31516/webjars/swagger-ui/index.html?url=/api-doc/swagger.json?url=/api-doc/swagger.json#`

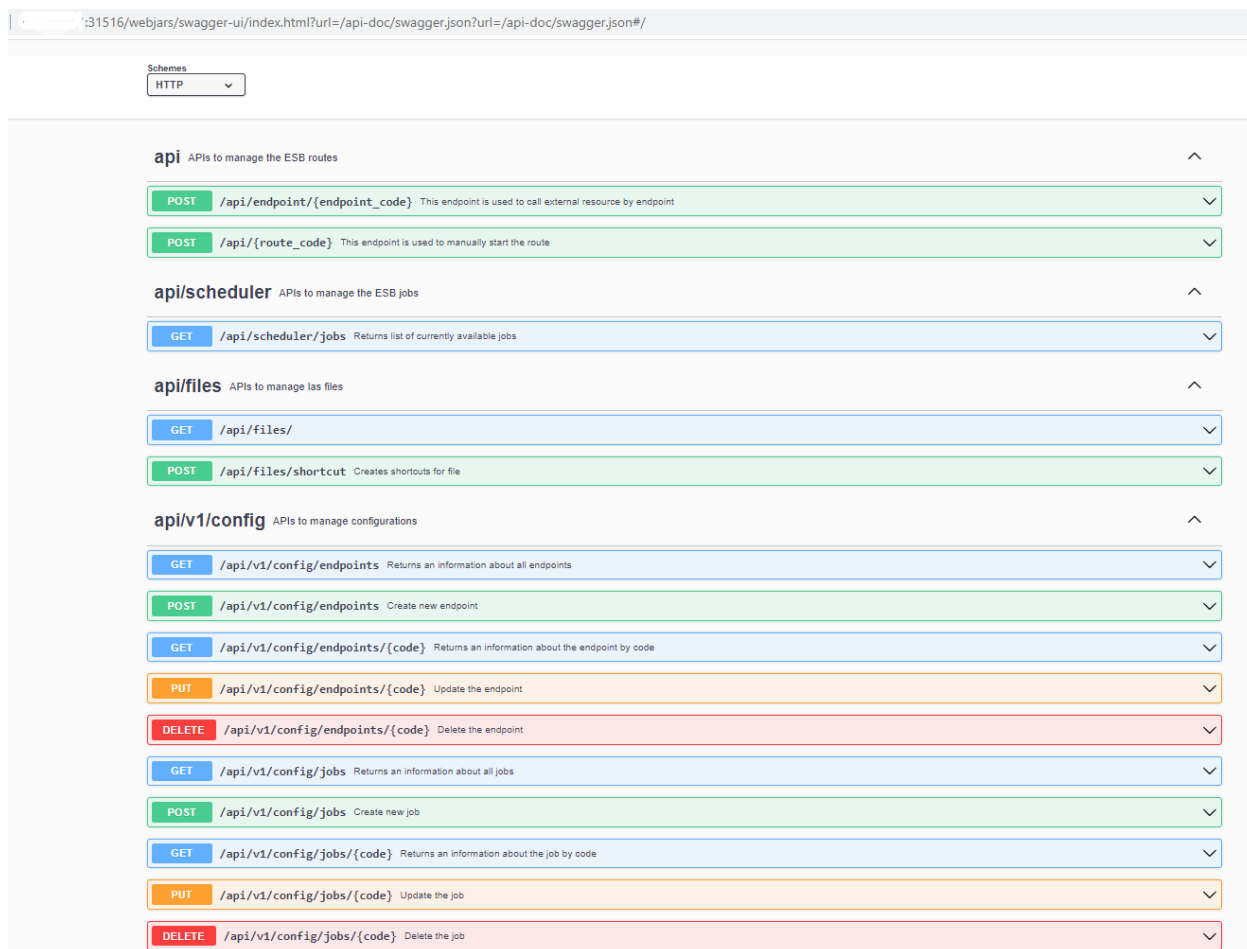


Рисунок 1 – REST API

4.1. Настройка передачи данных

Алгоритм настройки передачи данных:

- 1) создать ресурс (ресурсы), выгружающие данные: `POST/api/v1/config/resources`.
Поддерживаются следующие типы ресурсов: 3-MSSQL, 4-Postgres, 5-REST, 6-SOAP, 7- S3;
- 2) создать ресурс (ресурсы), получающие данные аналогично п. 1;
- 3) создать интерфейсы (endpoint), для получения данных из ресурсов-источников:
`POST/api/v1/config/endpoints`
- 4) создать интерфейсы (endpoint), для отправки данных в ресурсы-приемники аналогично п. 3;

5) создать скрипт по преобразованию данных (datasonnet):

```
POST/api/v1/config/scripts
```

6) создать маршрут передачи данных. При создании маршрута необходимо указать шаблон передачи данных. На текущий момент рекомендуется использовать шаблон передачи данных `routeSimpleAtoBwithEncode`, позволяющий перекодировать данные и передавать в несколько систем:

```
POST/api/v1/config/routes
```

7) перезапустить конфиг-сервер для актуализации конфигурации маршрутов (через `lens` удалить `pod config-server`).

5. РЕШЕНИЕ ТИПОВЫХ ПРОБЛЕМ

Типовые проблемы и способы их решения указаны в таблице 2.

Таблица 2 – Типовые проблемы и способы их решения

№	Описание проблемы	Способы решения
1	В консоли контейнера ошибки на отсутствие доступа к базе данных	Проверить учетные данные, прописанные в переменных окружения контейнера. Проверить доступность базы данных из контейнера по dns-имени или по ip-адресу сервера базы данных
2	В консоли контейнера ошибки при выполнении sql-скриптов.	При первом запуске контейнера в базе данных в схемах (gtm-web для gtm-back и esb-repos для config-server) не должно быть сущностей (объектов). Кроме этого, необходимо при наличии удалить таблицы databasechangelog и databasechangeloglock в схеме public
3	В консоли контейнера ошибки на отсутствие доступа к Active Directory	Проверить учетные данные для AD, прописанные в переменных окружения контейнера. Проверить доступность Active Directory из контейнера (gtm-back) по dns-имени или по ip-адресу сервера базы данных
4	Прочие ошибки в консоли контейнера	Обратиться к вендору

6. КОНТАКТНАЯ ИНФОРМАЦИЯ ПРОИЗВОДИТЕЛЯ

6.1. Юридическая информация

Название компании: АНО ВО «Университет Иннополис»;

Юр. адрес: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д. 1;

ОГРН: 1121600006142

ИНН: 1655258235

6.2. Контактная информация службы технической поддержки

Связаться со специалистами службы технической поддержки можно одним из следующих способов:

- телефон: 8 (843) 203-92-53;
- email: university@innopolis.ru.

Фактический адрес размещения инфраструктуры разработки: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения разработчиков: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения службы поддержки: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения серверов: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

- API – Application Programming Interface – программный интерфейс приложения, описание способов (набор классов, процедур, функций, структур или констант) взаимодействия между компонентами одного программного обеспечения или разными программными обеспечениями;
- ESB – Enterprise Service Bus – сервисная шина предприятия;
- HDFS – Hadoop Distributed File System – распределенная файловая система;
- HTML – Hypertext Markup Language – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере;
- REST – Representational State Transfer – архитектурный стиль взаимодействия компонентов распределенного приложения в сети;
- SOAP – Simple Object Access Protocol – протокол обмена структурированными сообщениями в распределенной вычислительной среде;
- SQL – Structured Query Language – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.