

Руководство пользователя (администратора интеграций)
Корпоративная шина данных «Kortex»

Правообладатель:
ООО "Бинари Брейнс"
ИНН: 5029292676

141052, Московская область, г.о. Мытищи, с. Марфино,
ул.Московская, д. 1Ж

Оглавление

1. Об этом документе.....	3
2. О программном обеспечении.....	3
3. Системные требования.....	3
4. Быстрый старт.....	3
4.1. Запуск из JAR.....	3
4.2. Запуск в контейнере.....	3
4.3. Переменные окружения.....	3
5. Конфигурирование интеграций.....	4
5.1. Глобальная конфигурация HTTP-сервера.....	4
5.2. Создание HTTP-эндпоинта (источник).....	4
5.3. Трансформации и фильтры.....	4
5.4. Цели (targets).....	4
5.5. Планирование по расписанию.....	4
6. Эксплуатация и наблюдаемость.....	5
7. Техническая поддержка.....	5

1. Об этом документе

Настоящее руководство содержит инструкции по установке, базовой настройке и эксплуатации «Корпоративной шины данных «Kortex». Документ адресован администраторам интеграций и инженерам сопровождения.

2. О программном обеспечении

«Kortex» — платформа для интеграции и обмена данными между корпоративными системами.

Основные функции:

- прием данных по HTTP(S),
- трансформация и фильтрация данных,
- параллельная обработка данных,
- отправка данных во внешние системы по HTTP,
- планирование запусков по расписанию,
- стандартные ответы пользователю и обработка ошибок.

3. Системные требования

- ОС: Linux/Windows/macOS.
- JRE/JDK: Java 17+.
 - Память/CPU: согласно нагрузке (типично 512 МБ ОЗУ и 1 vCPU на экземпляр).
 - Сеть: открытый TCP-порт HTTP-сервера (по умолчанию 8080).
 - Диск: директория для логов (по умолчанию logs/).

4. Быстрый старт

4.1. Запуск из JAR

- Установите Java 17+.
- Запустите: `java -jar kortex-core.jar`
- По умолчанию HTTP-сервер поднимется на `0.0.0.0:8080`.

4.2. Запуск в контейнере

- Запустите подготовленный Docker-образ из приватного реестра Forgejo ООО «Бинари Брейнс», доступ к которому выдаётся после подписания договора.
- Пробросьте порт 8080, задайте переменные окружения (см. ниже).

4.3. Переменные окружения

- KORTEX_SERVER_HOST — хост HTTP-сервера (по умолчанию 0.0.0.0).
- KORTEX_SERVER_PORT — порт HTTP-сервера (по умолчанию 8080).

5. Конфигурирование интеграций

5.1. Глобальная конфигурация HTTP-сервера

- Хост и порт задаются на уровне приложения Заказчика, разрабатываемого с использованием корпоративной шины данных «Kortex». Все HTTP-источники используют эти значения.

5.2. Создание HTTP-эндпоинта (источник)

- Определите путь (например, /api/events) и метод (GET/POST/PUT/DELETE).
 - Настройте валидацию: ожидаемый Content-Type (application/json), максимальный размер тела, обязательные заголовки.
 - При необходимости включите аутентификацию (API Key/Bearer/Basic) и ограничение частоты.
 - Определите схему ответа: статус-коды, заголовки, JSON-тело ответа.

5.3. Трансформации и фильтры

- Добавляйте шаги преобразования данных (синхронно/асинхронно), фильтруйте нежелательные события.
- Компонуйте преобразования последовательно или ветвите обработку для параллельного выполнения.

5.4. Цели (targets)

- Настройте HTTP-клиент: базовый URL (протокол/хост/порт/путь), заголовки, параметры запроса, таймауты.
- Отправляйте данные в требуемом формате: JSON/XML/текст/форм-данные.

5.5. Планирование по расписанию

- Используйте планировщик для периодических задач (например, опрос внешних API). Настраивается в коде в приложении Заказчика, разрабатываемого с использованием корпоративной шины данных «Kortex».

6. Эксплуатация и наблюдаемость

- Логирование: файл logs/kortex.log и вывод в консоль; уровни логов настраиваются в logback.xml.
- Здоровье сервиса: доступность HTTP-порта и эндпоинтов, ответ на тестовые запросы.
 - Производительность: настройка таймаутов HTTP-клиента/сервера; параллелизм веток обрабатывать с учетом целей.

7. Техническая поддержка

- Email: info@binarybrains.ru
- Выделенная группа в Telegram (создается с представителями Заказчика, Исполнителя после заключения договора)
- Трекер входящих заявок в репозитории проекта

Содержание обращения: версия ПО, среда, журнал (логи), описание, приоритет, шаги воспроизведения.