

Руководство администратора ПО
**«Система обмена табличными данными посредством CSV-
файлов»**

1. Введение

1.1. Область применения

Настоящий документ предназначен для сотрудников эксплуатирующей организации и отражает основные функциональные возможности и порядок действий при выполнении операций, связанных с администрированием программного обеспечения «Система обмена табличными данными посредством CSV-файлов» (далее - «Система»)

1.2. Перечень выполняемых функций администратора/оператора

В перечень выполняемых функций администратора Системы входят:

- Установка и настройка Системы
- Реализация планов устранения сбоев и нетиповых нештатных ситуаций
- Выполнение сбора и предоставление в вышестоящую линию технической поддержки информации для воспроизведения технических проблем и выработки решений по их разрешению
- Реализация рекомендаций по устранению нештатных ситуаций, полученных с вышестоящей линии поддержки
- Восстановление работоспособности Системы при сбоях в работе функциональных модулей
- Разработка решения по устранению технических проблем в работе функциональных модулей

1.3. Уровень подготовки администратора/оператора

Администратор/оператор (далее по тексту Администратор) Системы должен уметь пользоваться и настраивать среду функционирования контейнеров или систему оркестрации, используемую на предприятии.

Рекомендуемая численность персонала для эксплуатации Системы — 1 штатная единица.

Администраторы Системы должны пройти обязательную общую и специальную подготовку для работы с Системой.

Общая подготовка должна включать в себя получение знаний и навыков работы с Системой в качестве администратора.

Специальная подготовка должна включать в себя получение знаний и навыков в объеме, необходимом для выполнения своих должностных обязанностей

1.4. Перечень документации

В состав документации, с которой необходимо ознакомиться администратору Системы входят:

- описание функциональных характеристик Системы
- описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения.

2. Установка Системы

В данном разделе будет описана установка Системы на Debian Linux. Предполагается, что были предварительно установлены также Docker, Docker Compose, а также, используемая СУБД: PostgreSQL.

2.1. Системные требования к ПО

Минимальные аппаратные требования:

- Операционная система, способная запускать контейнеры. Предпочтительно Linux.
- Система управления контейнерной виртуализацией. Предпочтительно Docker Swarm или Kubernetes.
- Количество логических ядер процессора: 4
- Семейство процессоров: x86
- Частота процессора: 3.0. ГГц
- Объем установленной памяти: 16 Гб

2.1.2. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО

- Debian 11 (Открытая лицензия GNU)
- Docker 24.0.2 (open-source community edition)
- Postgresql 14 (Открытая лицензия PostgreSQL license)

При необходимости внешнего логгирования и мониторинга допускается использовать дополнительное сервисное ПО:

- Grafana Loki 2.6.1 (Открытая лицензия GNU)
- Grafana 9.2.2 (Открытая лицензия GNU)

2.1.3. Языки программирования

При разработке Системы был использован язык программирования GoLang 1.20 (открытая лицензия BSD)

2.2. Порядок установки

1. Смонтируйте диск с дистрибутивом в папку /mnt
2. Скопируйте из дистрибутива исходники из папки /mnt в папку /root
3. Смените текущую папку на /root
4. Отредактируйте файл docker-compose.yml, в соответствии с пунктом 3.2 данного документа
5. Создайте и отредактируйте файл настроек для модуля, в соответствии с пунктом 3.3 данного документа
6. Выполните команду
`docker compose -up -d --build`

3. Настройка Системы

3.1. Общие сведения

В данном документе приводятся примеры настройки Системы с использованием среды Docker Compose. Настройка операционной системы, СУБД, а также возможная настройка использования систем оркестрации, находятся вне компетенции этого документа и не будут тут описаны.

3.2. Модуль импорта данных

Модуль требует настройки следующих переменных окружения:

- HOST - адрес сервиса, который будет слушать модуль. На этот адрес следует пробросить внешний порт или настроить проксирующий сервер для поддержки протокола https.
- SETTINGS_FILE - путь к файлу с настройками запросов. Файл подключается к контейнеру с помощью volume. В данной переменной окружения указывается локальный путь внутри контейнера, к которому был примонтирован volume.
- DB_HOST — имя хоста или IP PostgreSQL сервера
- DB_PORT – номер порта PostgreSQL сервера. Значение по умолчанию 5432.
- DB_USER — имя пользователя PostgreSQL сервера
- DB_PASSWORD — пароль пользователя PostgreSQL сервера
- DB_NAME — имя базы данных
- DB_CONN_MAX_TIME — устанавливает максимальное время перед переиспользованием соединения
- DB_MAX_OPEN_CONNECTIONS — максимальное количество соединений с базой данных
- DB_MAX_IDLE_CONNECTIONS — максимальное количество неиспользуемых соединений с базой данных
- LOG_LEVEL - уровень логгирования. Поддерживаемые значения:
 - error
 - warn
 - info
 - debug

Пример настройки модуля:

```
http_in:
  build:
    context: ./http-in/
  restart: always
  ports:
    - "8080:8080"
  environment:
    DB_HOST: host.docker.internal
    DB_PORT: 5432
    DB_USER: postgres
    DB_PASSWORD: LDP8brN7B8ZLf5Q879QXuJYXCwm9nP
    DB_NAME: test
    HOST: :8080
    SETTINGS_FILE: /config.json
    LOG_LEVEL: debug
  volumes:
    - ./http-in/config.json:/config.json:ro
  extra_hosts:
    - "host.docker.internal:host-gateway"
```

3.3. Настройка импорта данных

Для того, чтобы Система могла импортировать CSV-файл конкретной структуры, необходимо провести определенные настройки. Такие настройки хранятся в файле настроек, путь к которому указан в переменной `SETTINGS_FILE`, описанной в разделе 3.2. Файл настроек должен содержать json-объект следующей структуры:

`requests` – объект, содержащий описание http-маршрутов и настройки импорта данных

Ключами объекта `requests` являются маршруты, содержащие в себе относительные `url`.

Элементами объекта `requests` также являются объекты `json` со следующей структурой:

- `lazy_quotes` – булевый флаг. Если имеет значение `true`, кавычки могут отображаться в поле без кавычек, а одинарные кавычки могут отображаться в кавычках.
- `comma` – символ-разделитель полей
- `sql` – запрос к СУБД с параметрами, в который будет передан `map` с ключами в виде имен столбцов в CSV-файле и значениями соответствующих полей.

Пример файла настройки обработчика запросов:

```
{
  "requests": {
    "/upload":{"lazy_quotes": false, "comma": ";", "sql": "insert into test (id, caption) values
(cast(:id as int), cast(:name as varchar)) on conflict (id) do update set
caption=cast(EXCLUDED.caption as varchar)"}
  }
}
```