

**Руководство администратора ПО  
«Сервис передачи одноразовых паролей через Telegram»**

# **1. Введение**

## **1.1. Область применения**

Настоящий документ предназначен для администраторов, эксплуатирующих программное обеспечение «Сервис передачи одноразовых паролей через Telegram» (далее — «Система»). Руководство содержит порядок установки, настройки, администрирования и взаимодействия с API Системы.

## **1.2. Перечень выполняемых функций администратора/оператора**

В перечень выполняемых функций администратора Системы входят:

- Установка и настройка Системы в контейнерной среде.
- Настройка переменных окружения и подключение к внешним сервисам (Telegram Bot API, Valkey).
- Мониторинг работы Системы и анализ логов.
- Восстановление работоспособности при сбоях.
- Конфигурация параметров безопасности (TTL, лимиты попыток).

## **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

### 2.1. Системные требования

#### 2.1.1. Аппаратные требования

- Количество логических ядер процессора: 4
- Семейство процессоров: x86
- Частота процессора: 3.0 ГГц
- Объем оперативной памяти: 16 ГБ

#### 2.1.2. Программные требования

- ОС: Debian 11 или аналогичная, поддерживающая Docker
- Docker: 24.0.2 или выше
- Valkey (Redis-совместимая СУБД): 8.0 или выше
- Дополнительно (опционально):
  - Grafana Loki 2.6.1 (для логирования)
  - Grafana 9.2.2 (для мониторинга)

#### 2.1.3. Язык программирования

Система разработана на Golang 1.22.

## 2.2. Порядок установки

1. Смонтируйте диск с дистрибутивом в папку /mnt
2. Скопируйте из дистрибутива исходники из папки /mnt в папку /root
3. Отредактируйте файл docker-compose.yml, в соответствии с пунктами 3.2 данного документа
4. Смените текущую папку на /root и выполните в ней 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. Конфигурируемые параметры

Для корректной работы модуля обработки запросов и доставки сообщений, необходимо настроить для него следующие переменные окружения:

- HOST — адрес и порт, который слушает сервис.
- LOG\_LEVEL — уровень логирования (error, warn, info, debug).
- TELEGRAM\_HASH — токен бота Telegram, полученный от @BotFather.
- TELEGRAM\_BOT\_NAME — имя бота (например, MyOtpBot).
- DB\_HOST, DB\_PASSWORD, DB\_ID — параметры подключения к Valkey.
- OTP\_STORE\_INTERVAL — время жизни OTP в секундах (по умолчанию 300 сек).
- ACCESS\_KEY — ключ доступа к API

### Пример настройки модуля:

```
services:
  otp-bot:
    build: .
    restart: always
    ports:
      - "80:80"
    environment:
      HOST: ":80"
      LOG_LEVEL: "info"
      TELEGRAM_HASH: "ваш_токен_бота"
      TELEGRAM_BOT_NAME: "имя_бота_без_@"
      DB_HOST: "valkey:6379"
      DB_PASSWORD: ""
      DB_ID: "0"
      OTP_STORE_INTERVAL: "300s"
      ACCESS_KEY: "key"
    depends_on:
      - valkey
    networks:
      - otp-net

valkey:
  image: valkey/valkey:8.0
  restart: always
  networks:
    - otp-net
```

## 4. Описание API

### 4.1. Общие сведения

Система предоставляет JSON-RPC API по HTTP.

Все методы, кроме ping, доступны по адресу /. Доступ к API защищен с помощью Bearer token, передаваемого в заголовке Authorization.

Метод ping доступен по адресу /public

Заголовок Content-Type должен быть application/json.

### 4.2. Методы API

#### 4.2.1. ping

Проверка доступности сервиса.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "ping",
  "id": 1
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": "pong",
  "id": 1
}
```

#### 4.2.2. startcode.create

Генерация уникальной ссылки для авторизации пользователя в боте.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "startcode.create",
  "id": 2,
  "params": "user123"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": {
    "code": "550e8400-e29b-41d4-a716-446655440000",
    "url": "https://telegram.me/MyOtpBot?start=550e8400-e29b-41d4-a716-446655440000"
  },
  "id": 2
}
```

#### 4.2.3. otp.create

Генерация OTP для пользователя и отправка его через Telegram.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "otp.create",
  "id": 3,
  "params": "user123"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 3
}
```

#### 4.2.4. otp.check

Проверка введённого OTP.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "otp.check",
  "id": 4,
  "params": {
    "user_id": "user123",
    "code": "123456"
  }
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 4
}
```

В случае неверного кода или отсутствия OTP возвращается false.