Документация IT Support Bot для ВКонтакте

IT Support Bot — это бот для мессенджера ВКонтакте, предназначенный для автоматизации процессов IT-поддержки в РАНХиГС. Основные функции:

1. Создание заявок в IT-поддержку
2. Просмотр статуса существующих заявок
3. Дополнение заявок новой информацией
4. Авторизация по корпоративной почте @ranepa.ru

Технологический стек:

1. Python 3.9+
2. VKBottle 3.x (фреймворк для работы с VK API)
3. PostgreSQL 13+ (хранение данных пользователей и кодов верификации)
4. SimpleOne API (система управления заявками)
5. aiosmtplib (отправка email-уведомлений)

Установка и настройка

Предварительные требования

1. Python 3.9 или новее
2. PostgreSQL 13+
3. Аккаунт ВКонтакте с настроенным сообществом
4. Доступ к SimpleOne API
5. SMTP-сервер для отправки email

Процесс установки

Клонирование репозитория:

git clone https://github.com/ karandash1111k/it-support-bot.git

cd it-support-bot

Установка зависимостей:

pip install -r requirements.txt

Настройка переменных окружения (создайте файл .env):

ini

# Настройки VK

VK\_TOKEN=your\_vk\_group\_token

# Настройки БД

DB\_HOST=host

DB\_PORT=5432

DB\_NAME=it\_support\_bot

DB\_USER=postgres

DB\_PASSWORD=secure\_password

# Настройки SMTP

SMTP\_HOST=smtp.example.com

SMTP\_PORT=587

SMTP\_USER=user@example.com

SMTP\_PASSWORD=email\_password

SMTP\_FROM=noreply@example.com

# Настройки SimpleOne

SIMPLEONE\_URL=https://api.simpleone.ru

SIMPLEONE\_TOKEN=your\_api\_token

DEFAULT\_TICKET\_TYPE=incident

DEFAULT\_PRIORITY=medium

Конфигурация

Основные настройки хранятся в классе Config (src/core/config.py):

Обязательные переменные

Переменная Описание

VK\_TOKEN Токен группы ВКонтакте

DB\_HOST Хост PostgreSQL

DB\_NAME Имя базы данных

DB\_USER Пользователь БД

DB\_PASSWORD Пароль БД

SMTP\_HOST SMTP-сервер

SMTP\_PORT Порт SMTP

SMTP\_USER Пользователь SMTP

SMTP\_PASSWORD Пароль SMTP

SMTP\_FROM Email отправителя

SIMPLEONE\_URL URL API SimpleOne

SIMPLEONE\_TOKEN Токен для доступа к SimpleOne API

Структура БД

Таблица users:

email (VARCHAR) - корпоративная почта (первичный ключ)

vk\_id (INTEGER) - ID пользователя ВКонтакте

is\_verified (BOOLEAN) - статус верификации

created\_at, updated\_at (TIMESTAMP) - метки времени

Таблица verification\_codes:

email (VARCHAR) - ссылка на пользователя

code (VARCHAR) - код подтверждения

expires\_at (TIMESTAMP) - срок действия кода

created\_at (TIMESTAMP) - время создания

Интеграция с SimpleOne

Методы API

create\_ticket() - создание новой заявки:

await simpleone\_api.create\_ticket(

email="user@ranepa.ru",

category="Категория",

subcategory="Подкатегория",

description="Описание проблемы",

attachments=["url1", "url2"]

)

get\_user\_tickets() - получение заявок пользователя:

tickets = await simpleone\_api.get\_user\_tickets("user@ranepa.ru")

add\_comment\_to\_ticket() - добавление комментария к заявке:

await simpleone\_api.add\_comment\_to\_ticket(

ticket\_id="123",

comment="Дополнительная информация",

attachments=["url1"]

)

Формат заявки

При создании заявки передается следующий JSON:

{

"type": "incident",

"priority": "medium",

"title": "Категория - Подкатегория",

"description": "Описание проблемы",

"custom\_fields": {

"category": "Категория",

"subcategory": "Подкатегория",

"user\_email": "user@ranepa.ru"

},

"attachments": ["url1", "url2"]

}

Механизмы аутентификации

Процесс аутентификации

1. Пользователь вводит email @ranepa.ru
2. Генерируется 6-значный код и сохраняется в БД
3. Код отправляется на email пользователя
4. Пользователь вводит код в боте
5. Код проверяется, при успехе - аутентификация завершена

Обработка заявок

Создание заявки

1. Пользователь выбирает категорию → подкатегорию
2. Вводит описание проблемы
3. Прикрепляет файлы (опционально)
4. Заявка создается в SimpleOne

Логирование и мониторинг

Уровни логирования:

1. INFO: основные события (запуск бота, создание заявки)
2. WARNING: нештатные ситуации (ошибка SMTP)
3. ERROR: критические ошибки (ошибка БД, API)

Рекомендуемые метрики для мониторинга

1. Количество активных пользователей
2. Среднее время обработки заявки
3. Количество созданных заявок в день
4. Процент успешных аутентификаций
5. Время отклика SimpleOne API

Техническое обслуживание

1. Обновление зависимостей
2. Мониторинг нагрузки
3. Резервное копирование БД

Тестирование

Рекомендуется:

1. Модульные тесты для критических компонентов (аутентификация, работа с БД)
2. Интеграционные тесты для проверки взаимодействия с VK API и SimpleOne
3. End-to-end тесты ключевых сценариев (создание заявки)

Оптимизация производительности

1. Используйте connection pool для БД
2. Кэшируйте часто запрашиваемые данные (например, категории заявок)
3. Оптимизируйте запросы к SimpleOne API
4. Рассмотрите возможность использования asyncpg для работы с PostgreSQL

Для успешной эксплуатации бота рекомендуется следовать описанным практикам и регулярно обновлять систему в соответствии с изменениями в API ВКонтакте и SimpleOne.