

Платформа ботов Неовокс

Описание жизненного цикла, поддержки и обслуживания программного обеспечения

Оглавление

| | |
|---|----|
| Оглавление..... | 2 |
| Аннотация..... | 3 |
| Общие сведения..... | 4 |
| Процессы жизненного цикла программного обеспечения..... | 5 |
| Установка..... | 5 |
| 1. Системные требования..... | 5 |
| 2. Подготовка среды установки..... | 5 |
| 3. Получение дистрибутива..... | 5 |
| 4. Настройка конфигурации..... | 6 |
| 5. Процесс установки..... | 7 |
| Эксплуатация..... | 8 |
| Удаление..... | 9 |
| Устранение неисправностей программного обеспечения..... | 10 |
| Совершенствование программного обеспечения..... | 10 |
| Требование к персоналу..... | 11 |

Аннотация

Данный документ содержит:

- описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения;
- устранение неисправностей, выявленных в ходе эксплуатации программного обеспечения;
- совершенствование программного обеспечения;
- информацию о персонале, необходимом для обеспечения такой поддержки.

Данный документ предназначен для администраторов системы «Платформа ботов Неовокс».

Общие сведения

Назначение ПО:

Программное обеспечение «Платформа ботов Неовокс» предназначено для создания голосовых диалоговых систем (IVR-сценариев), их исполнения и обслуживания клиентов в автоматическом режиме.

Включает в себя следующие модули:

- Проектирования IVR-сценариев с использованием графического конструктора на основе блоков и связей.
- Записи, централизованного хранения и версионирования созданных сценариев для контроля изменений.
- Интерпретации сценариев в реальном времени в процессе диалога бота с клиентом.
- Интеграции с системами автоматического распознавания речи (ASR) для извлечения текста из ответов абонента.
- Использования инструментов синтеза речи из текста (TTS) и гибридного синтеза для генерации голоса робота.
- Воспроизведения реплик из встроенной аудиобиблиотеки, позволяющей задействовать предварительно записанные профессиональным диктором фразы.
- Обработки естественного языка с использованием подключаемых NLU-движков и систем классификации текста для определения намерений пользователя.
- Автоматизации обмена данными путем обращения к сторонним сервисам.
- Сбора детальной информации о работе ботов с возможностью формирования отчетности.
- Логирования сессий взаимодействия, необходимого для последующей отладки сценариев и анализа качества обслуживания.

Особенности:

- На текущий момент аутентификация пользователей в платформе ботов не предусматривается.
- ПО работает с СУБД PostgreSQL.
- ПО использует систему телефонии от компании Naumen.
- ПО использует систему распознавания речи ASR от компании ASM.
- ПО использует систему синтеза речи TTS от компании Яндекс.
- ПО использует систему классификации текста Caila от компании Just AI.

Процессы жизненного цикла программного обеспечения

Установка

1. Системные требования

Аппаратное обеспечение

- Процессор: от 2 ядер x86_64/AMD64
- Оперативная память: от 4 ГБ
- Дисковое пространство: от 40 ГБ SSD
- Сеть: доступ к интернету для загрузки Docker-образов

Программное обеспечение

- ОС: Ubuntu 22.04 LTS / Debian 12 / CentOS 8 (или любой Linux с поддержкой Docker)
- Docker: версия 20.10 или выше
- Docker Compose: версия 2.0 или выше (плагин)
- Git: версия 2.25 или выше

2. Подготовка среды установки

Установка необходимого ПО

- Обновление пакетов
`sudo apt-get update && sudo apt-get upgrade -y`
- Установка Docker
`curl -fsSL https://get.docker.com -o get-docker.sh`
`sudo sh get-docker.sh`
- Установка Docker Compose плагина
`sudo apt-get install -y docker-compose-plugin`
- Установка Git, Make и других утилит
`sudo apt-get install -y git make curl wget`
- Добавление пользователя в группу docker
`sudo usermod -aG docker $USER`
`newgrp docker`

Создание сети Docker

- Приложение требует предварительно созданную сеть
`docker network create bothub_network`

3. Получение дистрибутива

- Клонирование репозитория (или копирование файлов дистрибутива)
`git clone <repository-url>`
`cd <project-name>`
- Если дистрибутив получен архивом:
`tar -xzf project-name.tar.gz`
`cd project-name`

4. Настройка конфигурации

Создание файла с переменными окружения

Содержимое .env файла backend API

```
COMPOSE_PROJECT_NAME=bothub_backend

# --- JWT ---
APP_CONFIG__JWT__SECRET_KEY=
APP_CONFIG__JWT__ALGORITHM=HS256
APP_CONFIG__JWT__ACCESS_TOKEN_EXPIRE_MINUTES=720
APP_CONFIG__JWT__REFRESH_TOKEN_EXPIRE_DAYS=7
APP_CONFIG__JWT__COOKIE_NAME=access_token
APP_CONFIG__JWT__REFRESH_COOKIE_NAME=refresh_token
APP_CONFIG__JWT__COOKIE_SECURE=false
APP_CONFIG__JWT__COOKIE_SAMESITE=lax

# --- Rate Limiting ---
APP_CONFIG__RATE_LIMIT__AUTH_SIGNIN=5/minute
APP_CONFIG__RATE_LIMIT__ENABLED=true

# --- Первый администратор ---
APP_CONFIG__FIRST_ADMIN__EMAIL=
APP_CONFIG__FIRST_ADMIN__PASSWORD=
APP_CONFIG__FIRST_ADMIN__USERNAME=

# Database
APP_CONFIG__DB__USER=
APP_CONFIG__DB__PASSWORD=
APP_CONFIG__DB__HOST=
APP_CONFIG__DB__PORT=
APP_CONFIG__DB__DATABASE=
APP_CONFIG__DB__ECHO=
APP_CONFIG__DB__POOL_SIZE=
APP_CONFIG__DB__MAX_OVERFLOW=

# --- Приложение ---
```

```
APP_CONFIG__APP_INFO__APP_NAME=
```

```
APP_CONFIG__APP_INFO__DEBUG=
```

```
APP_CONFIG__APP_INFO__VERSION=
```

5. Процесс установки

Запуск установки через docker:

- `cp $DEV_ENV .env`
- `export COMPOSE_PROJECT_NAME=bothub`
- `docker-compose build bothub`
- `docker-compose up -d --no-deps bothub`

Первоначальная настройка производится специалистами компании во время установки путем внесения параметров в соответствии с указаниями в задании на развертывание ПО.

Важно: для работы платформы требуется наличие Сервиса распознавания от ASM и телефонии от Naumen.

Эксплуатация

Запуск сценария IVR-бота на исполнение осуществляется через систему Naumen аналогично стандартным Python-проектам с использованием Script Framework Protocol:

В веб-интерфейсе Naumen необходимо выбирать целевой проект:

The screenshot shows the Naumen web interface for a project named 'sorokin36003 (FSI)'. The interface is divided into several sections:

- ВХОДЯЩИЙ ПРОЕКТ**: Includes buttons for 'Изменить входящий проект', 'Изменить состояние проекта', 'Копировать', 'Переместить', and 'Экспортировать'. Below are fields for 'Тип проекта' (Очередь), 'Название проекта' (sorokin36003 (FSI)), and 'Состояние' (Активный).
- ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ**: Includes a button 'Изменить дополнительные параметры' and a list of parameters such as 'Длительность хранения системных данных, дн' (0), 'Длительность хранения отчетных данных, дн' (0), 'Длительность хранения записей разговоров, дн.' (0), 'Критерий окончания вызова' (Закрытие экранной формы), 'Целевое время ответа, сек' (0), 'Максимальная длительность короткого вызова, сек' (7), 'Использовать секундомер' (Не использовать), 'Обратный адрес' (servicedesk@yourcompany.com), 'АОН', and 'E-mail'.
- НЕОБХОДИМЫЕ НАВЫКИ**: Includes buttons 'Добавить навык' and 'Изменить'.
- ПАРАМЕТРЫ ОЧЕРЕДИ**: Includes a button 'Изменить параметры очереди' and a list of parameters: 'Адрес' (36003), 'Приоритет' (1,00), 'Тип скрипта' (Простой), 'Версия скрипта' (latest), 'IVR-Скрипт' (/opt/naumen/nauphone/spool/naubuddy/ivr/v3/client/sorokin36003/fsi/general.py torgi), 'Политика очереди', 'Режим обслуживания' (Обычный), and 'Персональная очередь' (нет).
- ПАРАМЕТРЫ ОБРАТНЫХ ЗВОНКОВ**: Includes a button 'Изменить' and a parameter 'Критерий включения' (Не выбран).

Затем указывается путь до интерпретатора — программной реализации, воспроизводящей метаданные.

This is a close-up view of the 'ПАРАМЕТРЫ ОЧЕРЕДИ' section. It shows a list of parameters with their values. The 'IVR-Скрипт' parameter is highlighted with a red box, showing its value: /opt/naumen/nauphone/spool/naubuddy/ivr/v3/client/sorokin36003/fsi/general.py torgi.

| ПАРАМЕТРЫ ОЧЕРЕДИ | |
|----------------------------|---|
| Изменить параметры очереди | |
| Адрес | 36003 |
| Приоритет | 1,00 |
| Тип скрипта | Простой |
| Версия скрипта | latest |
| IVR-Скрипт | /opt/naumen/nauphone/spool/naubuddy/ivr/v3/client/sorokin36003/fsi/general.py torgi |
| Политика очереди | |
| Режим обслуживания | Обычный |
| Персональная очередь | нет |

В качестве параметра передается наименование целевого проекта (из конструктора).

После запуска интерпретатор автоматически загружает активную версию метаданных из базы данных, и дальнейшая работа IVR-бота выполняется в строгом соответствии с визуальной схемой, созданной в конструкторе.

В процессе исполнения все данные (переменные, флаги, результаты распознавания) хранятся в контексте — прослойке между визуальным сценарием и его технической реализацией.

Цикл работы сценария:

- Поступает вызов на телефонию Naumen;
- Naumen запускает целевой скрипт с передачей параметров;
- Целевой скрипт, коим является интерпретатор сценариев, загружает метаданные сценария из сервиса обработки сценариев, если его кеш не был найден;
- Интерпретатор обрабатывает метаданные сценария и генерирует итератор по блокам сценария (нодам);
- Интерпретатор итерируется по нодам и исполняет функционал итерируемой ноды;
- По мере итерирования интерпретатор обращается к сервисам платформы, если того требует соответствующая нода, и отправляет данные для сохранения в БД, логируя системную информацию;
- По окончании итерации следует окончание исполнения скрипта, что телефония Naumen расценивает как завершение вызова.

Удаление

Удаление ПО включает в себя два этапа:

- Удаление БД ПО в СУБД, где хранятся соответствующие данные.
- Удаление контейнера с приложениями.

Устранение неисправностей программного обеспечения

Процесс решения проблем в ПО:

- разрабатывается стратегия менеджмента проблем;
- проблемы регистрируются, идентифицируются и классифицируются;
- проблемы анализируются и оцениваются для определения приемлемого решения (решений);
- выполняется решение проблем;
- проблемы отслеживаются вплоть до их закрытия;
- известно текущее состояние всех зафиксированных проблем.

Штатный порядок работы ПО определяется эксплуатационной документацией, предоставляемой производителем ПО. Поддерживаемый ПО набор функций определяется требованиями технического задания (ТЗ).

В случае обнаружения ошибок в работе ПО, которые являются нарушением требований ТЗ или противоречат порядку работы ПО, описанному в документации, администратор ПО должен направить заявку в службу технической поддержки (СТП) организации путем создания заявки в системе регистрации задач.

СТП проверяет наличие ошибки и рекомендаций по ее устранению в базе знаний технической поддержки.

В случае, если в базе знаний обнаружить описание ошибки не удастся, СТП пытается воспроизвести обнаруженную пользователем ошибку в тестовой среде. После подтверждения найденной ошибки СТП передает разработчикам ПО задание на устранение обнаруженной ошибки.

После устранения неисправности разработчики ПО выпускают обновление к текущей версии ПО или включают исправление в следующую версию ПО. Информация о наличии обновления или новой версии ПО доводится до заявителя.

Совершенствование программного обеспечения

Работа по совершенствованию ПО включает в себя два основных направления:

- повышение качества и надежности ПО;
- актуализация перечня функций, поддерживаемых ПО.

В ходе постоянно проводимой работы по совершенствованию ПО используются хорошо зарекомендовавшие себя методы повышения качества и надежности ПО:

- совершенствование процесса разработки ПО – повышение качества ПО за счет использования современных методик и инструментов разработки;
- совершенствование процесса тестирования ПО – обеспечение необходимой полноты покрытия кода тестами.

Актуализация перечня функций, поддерживаемых ПО, включает в себя:

- добавление новых и изменение существующих функций в соответствии со стратегией развития ПО;

- добавление новых и изменение существующих функций по предложениям;
- исключение устаревших функций.

Требование к персоналу

К эксплуатации ПО «Платформа ботов Неовокс» допускаются лица, ознакомившиеся с эксплуатационной документацией на ПО «Платформа ботов Неовокс», и имеющие практические навыки работы с указанным программным продуктом.

Администратор ПО «Платформа ботов Неовокс» должен иметь навыки:

- Администрирования ОС семейства Linux.
- Администрирование web-сервисов Nginx, Python, JavaScript, TypeScript.
- Администрирование ПО в докер-контейнерах.
- Администрирования СУБД PostgreSQL, Redis.
- Администрирования RabbitMQ.