

Bot.one

Информационная система **Bot.one**

Описание архитектуры и стек технологий

Москва, 2024

Содержание

1. Общие сведения и функциональная архитектура.....	3
2. Описание используемых технологий.....	5
3. Физическая архитектура.....	5
4. Технологические и программные требования для работы системы.....	7
4.1. Технологические требования.....	7
4.2. Программные требования.....	7

1. Общие сведения и функциональная архитектура

Система формирования автоматических диалогов Bot.one (далее – система, Bot.one) предназначена для автоматизации общения с клиентами через создание и развертывание автоматических ботов ведения диалогов.

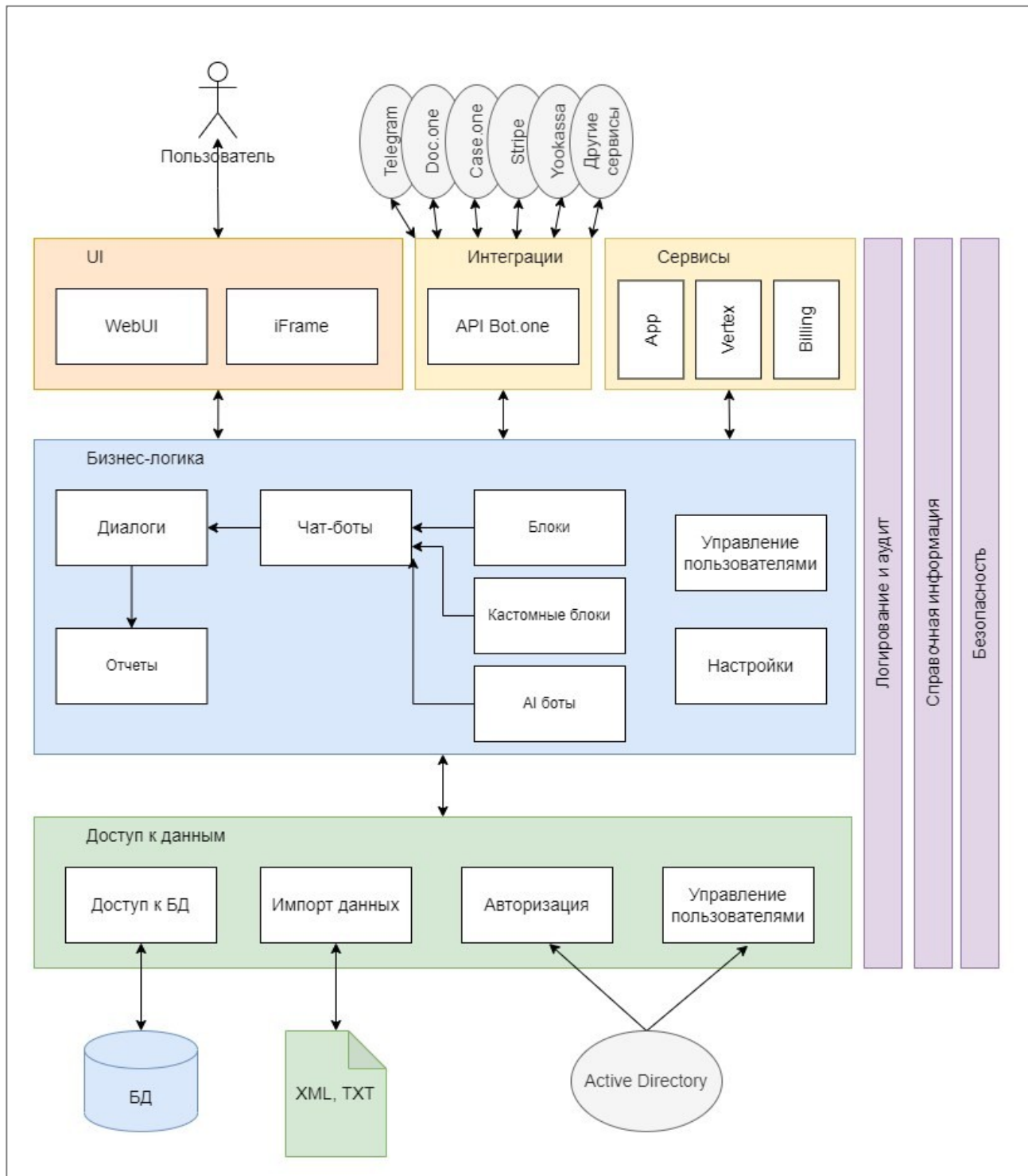


Рисунок 1. Функциональная архитектура Bot.one

Bot.one содержит модули:

1. **Чат-боты** – конструктор ботов.
2. **Диалоги** – хранение диалогов.

3. **AI боты** – настройка интеллектуального поиска по ботам.
4. **Кастомные блоки** – создание, редактирование и установка кастомных (пользовательских) блоков.
5. **Отчеты** – просмотр и формирование отчетов.
6. **Настройки** – настройка параметров системы и управление доступом пользователей.
7. Модуль редактирования профиля и пароля пользователя.
8. Интеграция с внешними системами.

Работа пользователей с системой осуществляется через веб-интерфейс.

2. Описание используемых технологий

Система разработана на PHP и JavaScript.

Взаимодействие с другими системами осуществляется через API сервиса.

Для хранения данных используется СУБД MySQL и MongoDB.

Для запуска системы используются контейнеры Docker.

3. Физическая архитектура

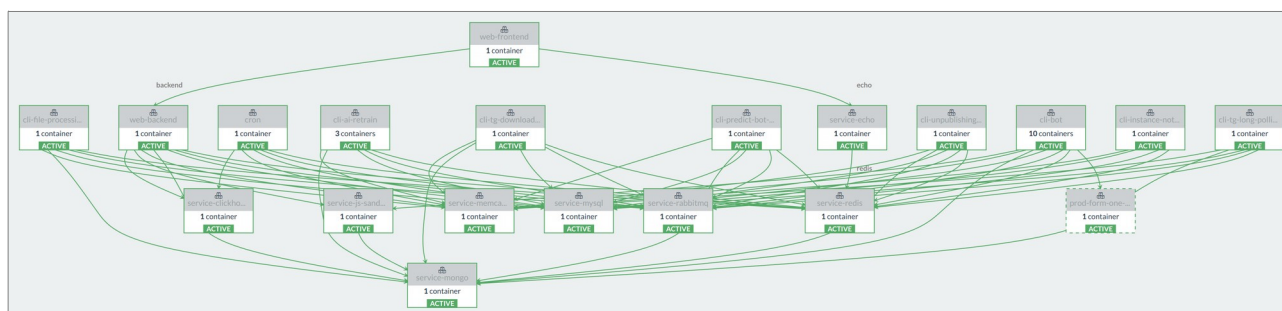


Рисунок 2. Физическая архитектура Doc.one

Описание контейнеров:

- web-frontend – веб-сервер (NGINX), отдает статический контент и проксирует запросы на бэкенд;

- web-backend – приложение (PHP-FPM), обрабатывает данные, работает с кэширующим сервисом и базами данных;
- service-mysql – база данных (MySQL), хранит данные приложения (шаблоны, документы, файлы и т.д.);
- service-redis – быстрое хранилище данных типа «ключ-значение» в памяти (Redis), используется в качестве кэша перед передачей данных в MySQL, позволяет снизить нагрузку на приложение;
- service-memcached – быстрое хранилище данных типа «ключ-значение» в памяти (Memcached), хранит сессии и результаты запросов от базы данных;
- service-clickhouse – аналитическая СУБД, используется для формирования отчетов;
- service-mongo – документоориентированная система управления базами данных (MongoDB), используется для хранения файлов (GridFS);
- service-rabbitmq – менеджер очередей, используется для передачи сообщений в cli-воркеры.
- service-js-sandbox – сервис-песочница для выполнения javascript кода, используется для блока Script;
- service-echo – websocket сервер для обмена данными в реальном времени;
- cli-bot – consumer, выполняет обработку сообщений из RabbitMQ;
- cli-file-processing – consumer, выполняет обработку загруженных файлов со списком пользователей или ключевых фраз для AI блока;
- cli-predict-bot-by-phrase – consumer, выполняет обработку добавленных вручную фраз для обучения AI бота (из файла или через интерфейс);
- cli-ai-retrain – consumer, переобучает AI блок. Запускается по планировщику (Cron) или вручную в настройках редактора блока;
- cli-unpublishing-bot – consumer, выполняет снятие бота с публикации;
- cli-tg-long-polling – consumer, получает и обрабатывает обновления от Telegram, если в конфигурационном файле установлена настройка получения обновления через long-polling;
- cli-tg-download-file – consumer, получает файлы из Telegram, переданные в диалоге;
- cli-instance-notify – consumer, отправляет нотификации в приложение, в частности используется для отправки нотификаций по истечении лимитов лицензии;
- cron – планировщик (Cron), выполняет периодические задачи.

4. Технологические и программные требования для работы системы

Требования к конфигурации не учитывают использование какого-либо стороннего ПО на серверах. Для обеспечения оптимальной производительности и стабильности работы системы не рекомендуется устанавливать на серверы прочее программное обеспечение, не относящееся к работе системы.

4.1. Технологические требования

Минимальные аппаратные требования для развертывания системы на серверах заказчика:

- Без использования технологии интеллектуального поиска (AI боты):
 - CPU 4 ядра по 2 GHz;
 - RAM 8 Gb;
 - SSD 50+ Gb;
 - SSD для хранения файлов документов – объем диска зависит от количества и объема документов.
- С использованием технологии интеллектуального поиска (AI боты):
 - CPU 8 ядра по 2 GHz;
 - RAM 16 Gb;
 - SSD 50+ Gb;
 - SSD для хранения файлов документов – объем диска зависит от количества и объема документов.

4.2. Программные требования

Система разворачивается на сервере в Docker-контейнерах.

Минимальные требования к операционной системе сервера приложений, СУБД, локального каталога расширений:

- Ubuntu 18.04 и выше;
- Debian 10 и выше

- Astra Linux 1.7 SE;
- Другие дистрибутивы Linux, поддерживающие работу Docker и Docker-compose последних версий (после предварительного тестирования).

На сервере приложения и СУБД должно быть установлено следующее ПО (указаны минимальные версии):

- NGINX;
- JavaScript (ReactJS 17);
- PHP 8.0;
- Redis;
- Memcached;
- RabbitMQ 3.7.8;
- MySQL 5.7;
- ClickHouse 21.3.4.25;
- MongoDB 4.0.5.

На сервере локального каталога расширений должно быть установлено следующее ПО (указаны минимальные версии):

- nginx;
- php 8.0;
- MySQL 5.7.

Для работы пользователей с системой необходимо наличие установленной актуальной версии браузеров на устройстве пользователя:

- Google Chrome;
- Safari;
- MS Edge;
- Opera;
- Mozilla Firefox.