УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия Дисциплина «Облачные и туманные вычисления»

**Проект**

Разработка Telegram бота "InterviewBot"

Студент

*Кузнецов М. А.*

*P34131*

Преподаватель

*Перл О. В.*

Санкт-Петербург, 2023 г.

Содержание

[Сведения о приложении 2](#_Toc154594662)

[Реализация решения 3](#_Toc154594663)

# Сведения о приложении

Телеграмм-бот InterviewBot, который в себе будет содержать вопросы по разным топикам и темам, относящихся к IT в целом. Например, градация вопросов по языкам, алгоритмическим задачам и т. п. Возможны также и более общие темы -- устройство Linux, System Design и т. д. Еще, как возможность, пользователи сами смогут расширять существующую базу вопросов, то есть, через интерфейс бота добавлять свои вопросы. Другими словами – все это является некой совместной расширяемой базой знаний.

# Реализация решения

На предыдущем этапе был определен следующий стек разработки:

* [Go](https://go.dev/) – в качестве главного языка программирования бота.
* [Telegram Bot API](https://github.com/go-telegram-bot-api/telegram-bot-api/) – библиотека для удобного взаимодействия с API Telegram.
* [PostgreSQL](https://www.postgresql.org/) – open-source база данных, которая предоставит весь необходимый функционал для хранения данных.
* [pgx](https://github.com/jackc/pgx) – библиотека для Go для работы с базой данных PostgreSQL.

А также следующие роли:

1. Гость – это любой человек, который может найти бота через поиск, перейти по ссылке, взаимодействовать с ним. Данная роль разрешает использование основных и базовых команд бота. Назначается по-умолчанию каждому новому пользователю.
2. Администратор – для получения данной роли требуется admin-key. Данный ключ генерируется автоматически при каждом запуске экземпляра бота на сервере. С помощью команды /adminkey можно авторизоваться в боте и получить дополнительный доступ к другим командам, которые позволяют управлять вопросами и темами, просматривать предложения.

К приложению была также добавлена роль super-admin, которую перенимает на себя разработчик бота. Данная роль конфигурируется только через .yml файл конфигурации путем передачи ID пользователя Telegram, которому будет назначена данная роль. Пользователь с данной ролью в дополнении ко всем прочим командам может:

1. /block – запретить глобальный доступ к боту
2. /allow – разрешить глобальный доступ к боту
3. /blocka – запретить авторизацию для новых администраторов
4. /allowa – разрешить авторизацию для новых администраторов

Также не возникла необходимость в создании хранилища для фотографий для вопросов. Оказалось, что Telegram позволяет хранить изображения на своих серверах, доступ к которым можно получить по их уникальному ID. Таким образом, в базе данных храниться ID изображения, которое при показывании вопроса просто извлекается напрямую с серверов Telegram, а не через хранилище на сервере.

В остальном функционал бота остался прежним и удовлетворяет заданным на 1-м этапе требованиям.

# Детали реализации

Структура проекта:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

.postgresql – хранит сертификат на подключение к БД Yandex Cloud.

.ssh – необходимые ssh-ключи для подключения к Compute Yandex Cloud.

build/Dockerfile – *(не используется)* Dockerfile для сборки проекта. Используется при локальной разработке, не используется при разворачивании на сервере.

cmd – содержит запускающий бота файл. Точка входа в программу.

Internal – содержит внутренний код работы и бизнес-логику бота.  
 app – основные настройки и запуск бота на основе [Telegram Bot API](https://github.com/go-telegram-bot-api/telegram-bot-api/).

keygen – генератор случайного админ-ключа. Может быть перезаписан на свой через файл конфигурации.

metrics – работа с метриками для Prometheus.

pkg – код, относящийся к работе с хранилищем и БД:

db – содержит конфигурацию по настройке и подключению к БД Postgres.

model – модели для работы с БД.

repository – реализация паттерна Репозиторий для работы с БД.

app-config.yml – файл конфигурации приложения. Он имеет следующий вид:

Изображение выглядит как текст, снимок экрана, Шрифт

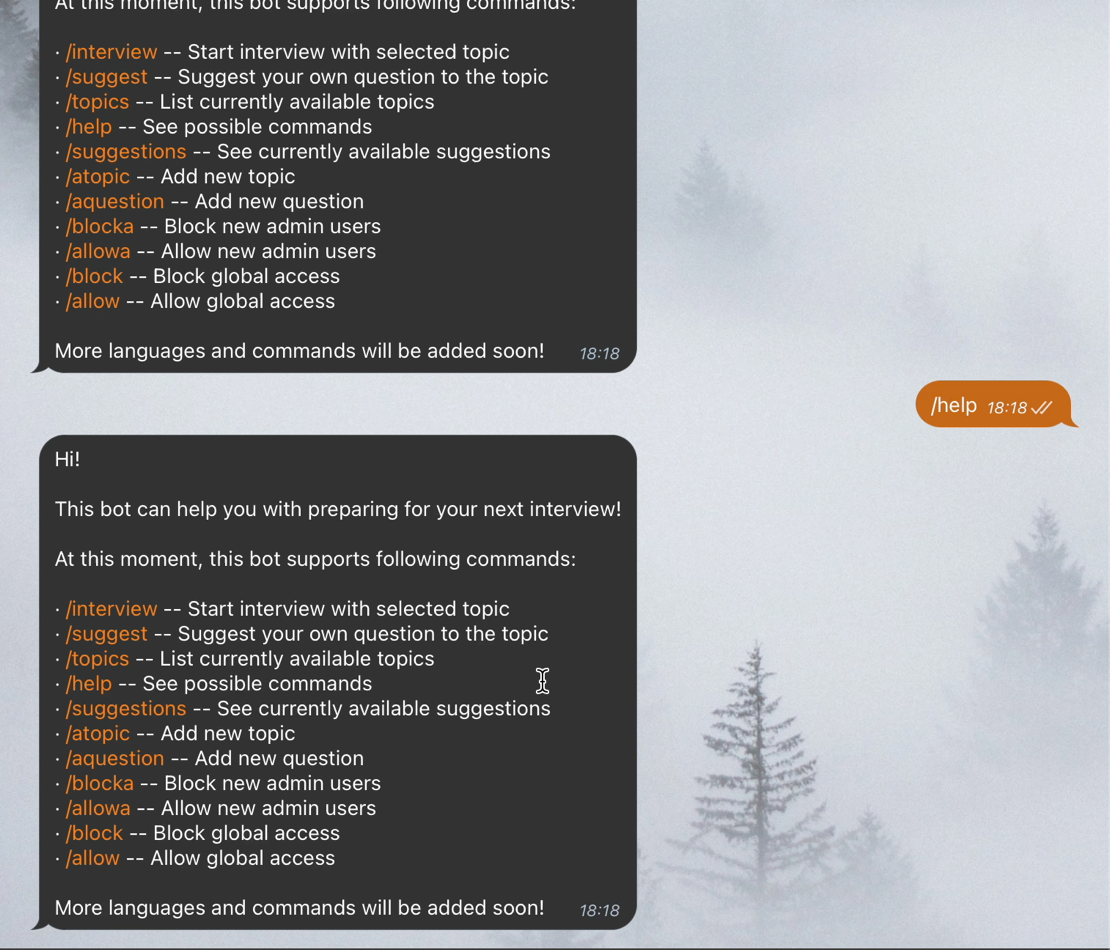
Автоматически созданное описание

force-admin-key – если true, то перезапишет случайный ключ на указанный в поле admin-key.

docker-compose.yml – *(не используется)* компоуз-файл для разворачивания dev-среды. Не используется при разворачивании на сервере.

Makefile – содержит набор команд для сборки и выгрузки приложения на сервер.

# Пример работы



(пкм – воспроизведение)

Ссылка: @ITIQTelegram\_Bot

# Сервисы Yandex Cloud

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Apollo – виртуальная машина, минимальная конфигурация, прерываемая. На ней запускается собранная программа:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

.postgresql – содержит .crt для подключения к БД

app-config.yml – конфигурация приложения

itiq – запускающий файл

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

DataLens – для сбора метрик. В качестве примера созданы два чарта\дашборда:

1. prom-Botmetrics (QL-чарт) для Prometheus как пример вывода по времени использования команд:  
   Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение, текст

   Автоматически созданное описание
2. Topics-Questions для PostgreSQL как пример вывода содержимого БД:

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение, снимок экрана

Автоматически созданное описание

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, снимок экрана, текст

Автоматически созданное описание

Hephaestus – DbaaS от Yandex Cloud, в котором мы храним наши вопросы, топики, предложения.