Тестовое задание на позицию Golang-разработчика

Оглавление

[1. Общее описание задания 1](#_Toc118113842)

[2. Технические подробности реализации 1](#_Toc118113843)

[2.1 Модуль управления операторами 1](#_Toc118113844)

[2.2 Модуль управления проектами 2](#_Toc118113845)

[2.3 База данных 2](#_Toc118113846)

[3. Вспомогательные библиотеки, которые необходимо использовать при выполнении задания 2](#_Toc118113847)

[4. Что необходимо предоставить по готовности 2](#_Toc118113848)

[5. Дедлайны и дальнейшие действия 3](#_Toc118113849)

# Общее описание задания

|  |  |
| --- | --- |
| Предметная область | Система контроля и учёта операторов контактного центра |
| Описание предметной области | Система представляет собой http-сервис с реляционной базой данной и состоит из двух модулей:   * Модуль управления операторами * Модуль управления проектами |
| Модуль управления операторами | Должен обладать следующими функциональными возможностями:   * Создавать новых операторов; * Просматривать операторов; * Редактировать операторов; * Удалять операторов; |
| Модуль управления проектами | Должен обладать следующими функциональными возможностями:   * Создавать проекты; * Просматривать проекты; * Редактировать проекты; * Удалять проекты; * Назначать операторов на проект; * Удалять операторов с проекта; |

# Технические подробности реализации

## 2.1 Модуль управления операторами

Описание набора полей для сущности «Оператор»:

|  |  |
| --- | --- |
| Оператор | * Имя * Фамилия * Отчество * Город * Номер телефона * E-mail * Пароль |

Необходимо реализовать http-методы, которые будут выполнять действия, описанные в пункте 1 в строке «Модуль управления операторами».

Каждый метод должен быть с валидацией отправляемых данных и в случае ошибки – возвращать её описание (валидация: e-mail, номера телефона в формате 8\*\*\*\*\*\*\*\*\*\* и тд и тп).

Дополнительно, каждому оператору должен присваиваться свой уникальный циферно-буквенный идентификатор и генерироваться случайный пароль.

## 2.2 Модуль управления проектами

Описание набора полей для сущности «Проект»:

|  |  |
| --- | --- |
| Проект | * Название проекта * Тип проекта (входящий/исходящий/автоинформатор) |

Необходимо реализовать http-методы, которые будут выполнять действия, описанные в пункте 1 в строке «Модуль управления проектами».

Каждый метод должен быть с валидацией отправляемых данных и в случае ошибки – возвращать её описание.

Дополнительно, каждому проекту должен присваивается свой уникальный циферно-буквенный идентификатор.

## 2.3 База данных

В качестве реляционной базы данных необходимо использовать PostgreSQL.

Необходимо разработать архитектуру и написать миграции.

Обязательно должны присутствовать первичные и внешние ключи, правильно подобранные типы данных и индексы.

# Вспомогательные библиотеки, которые необходимо использовать при выполнении задания

|  |  |  |
| --- | --- | --- |
| Библиотека | Для чего | Ссылка на github |
| gin | Основной http-фреймворк | https://github.com/gin-gonic/gin |
| Bun | ORM для PostgreSQL | https://github.com/uptrace/bun |
| Migrate | Утилита для создания, применения и откатки миграций для баз данных | https://github.com/golang-migrate/migrate |

# Что необходимо предоставить по готовности

|  |  |
| --- | --- |
| Исходный код | Полный исходный код сервиса + миграции для базы данных |
| Dockerfile | Файл с инструкциями сборки вашего сервиса в Docker |
| docker-compose.yaml | Набор всех необходимых сервисов для запуска вашего приложения в Docker |
| Список всех http-методов | Каждый метод с описанием для чего он служит, какой тип запроса ожидает, какой пример ответа возвращает и пример отправляемых в него данных |
| Инструкция | Подробная инструкция с шагами, что нужно сделать, чтобы запустить ваш сервис и начать с ним работу |

# Дедлайны и дальнейшие действия

На выполнение задания вам отводится 7 дней.

Далее мы его проверяем и даем обратную связь по нему.

Возможно, потребуется что-то доработать или переделать.