

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
Научный руководитель, доцент
департамента программной инженерии,
кандидат педагогических наук, доцент.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»

_____ С. А. Виденин
«__» _____ 2024 г.

_____ Н. А. Павлочев
«__» _____ 2024 г.

КофеСкаут

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.06-01 34 02-1-ЛУ

Исполнитель
студент группы БПИ228
_____/ М. Р. Иванов /
«__» _____ 2024 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2024

УТВЕРЖДЕН
RU.17701729.05.06-01 34 02-1-ЛУ

КофеСкаут

Руководство оператора

RU.17701729.05.06-01 34 02-1

Листов 7

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Москва 2024

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	3
1.1. Функциональное назначение	3
1.2. Эксплуатационное назначение.....	3
1.3. Краткая характеристика области применения программы	3
2. УСЛОВИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ.....	4
2.1. Минимальный состав аппаратных средств.....	4
2.2. Минимальный состав программных средств.....	4
3. ВЫПОЛНЕНИЕ ПРОГРАММЫ.....	5
3.1. Загрузка, запуск, выполнение и завершение программы	5
3.2. Управление выполнением программы	5
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	8

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 34 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

1.1. Функциональное назначение

Backend приложения реализованный в виде REST API, предназначен для работы с данными из базы данных и обработки бизнес процессов: заказ напитков, поиск кофеен и напитков.

1.2. Эксплуатационное назначение

API предоставляет возможности для авторизации, размещения заказов пользователями, получения заказов и редактирования меню кофейнями, поиска кофеен и напитков с различными фильтрами.

1.3. Краткая характеристика области применения программы

«КофеСкаут» - приложение для поиска интересующего напитка в кофейнях, расположенных в непосредственной близости от пользователя.

Основная область применения программы - личное использование. Также приложение предоставляет возможность кофейням продвигать свои продукты и рекламировать себя. Клиентам приложение предоставляет возможность выбора желаемого напитка из предложенных с использованием системы фильтров, а после предлагает ближайшие к пользователю кофейни, которые смогут приготовить выбранный напиток, а также предлагает сделать предзаказ. Более того, приложение указывает кратчайший путь до выбранной кофейни, что позволит пользователям сэкономить время как на выборе кофейни, так и на времени, чтобы до неё добраться.

В нынешних реалиях с большим количеством кофеен и ассортимента в них бывает трудно найти желаемый напиток поблизости, поэтому приложение «КофеСкаут» будет пользоваться спросом.

2. УСЛОВИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1. Минимальный состав аппаратных средств

Для запуска backend'а приложения требуется система с двухъядерным процессором и 4мя ГБ оперативной памяти.

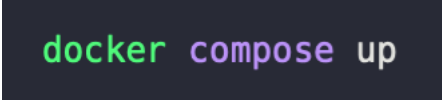
2.2. Минимальный состав программных средств

Для успешного запуска потребуются ОС Linux Debian с установленным на нее Docker engine и docker-compose.yml, .env файлы из репозитория. Также понадобится задать в .env нужные пользователю секретные значения.

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Загрузка, запуск, выполнение и завершение программы

Для запуска приложения требуется перейти в директорию, где расположен `docker-compose.yml` файл и написать в Bash следующую команду:



```
docker compose up
```

Рисунок 1 – команда для развертывания приложения.

После успешного запуска в терминале появится информация об успешном запуске приложения и туда же будут приходить логи.

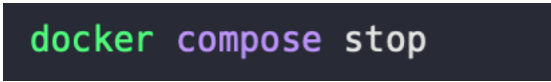
Если требуется обновить приложение, то достаточно написать команду:



```
docker compose pull coffeescoutbackend.api
```

Рисунок 2 – команда для обновления образа приложения.

Для того чтобы завершить программу нужно написать команду:



```
docker compose stop
```

Рисунок 3 – команда для завершения приложения.

3.2. Управление выполнением программы

Управление приложением выполняется через REST Api. Администратор backend'а может авторизоваться, используя данные из секретов приложения. Для администратора доступны следующие функции:

- 1) **[POST] /accounts/cafe-admin/register:** регистрация администратора кофейни. Доступно только администратору серверной части приложения.
- 2) **[POST] /beverage-types:** добавить тип напитка в базу данных. Доступно только администратору серверной части приложения.
- 3) **[PATCH] /beverage-types:** обновить тип напитка в базе данных. Доступно только администратору серверной части приложения.
- 4) **[DELETE] /beverage-types:** удалить тип напитка из базы данных. Доступно только администратору серверной части приложения.

- 5) **[POST] /cafes:** добавить кофейню в базу данных. Доступно только администратору серверной части приложения.
- 6) **[DELETE] /cafes/{id}:** удалить кофейню из базы данных. Доступно только администратору серверной части приложения.

В результате успешного выполнения запросов, возвращается один из Http кодов успеха. В случае неудачи, возвращается Http код проблемы и детали в соответствии с спецификацией Problem details.

4. СООБЩЕНИЯ ОПЕРАТОРУ

Оператору не требуется ориентироваться на какие-либо сообщения от приложения. Все сообщения приложения состоят из логов, которые носят чисто информативный характер. Повторный запуск программы в случае выхода ее из строя или сбоев происходит автоматически.

[illegible]