**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТОМСКИЙ ЭКОНОМИКО-ПРОМЫШЛЕННЫЙ КОЛЛЕДЖ»**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ API СЕРВИСА**

***ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ***

*СПЕЦИАЛЬНОСТЬ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ*

Выполнил:

Студент группы 1912с

\_\_\_\_\_\_\_\_\_\_\_\_ / Заикин В.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Проверил:

Руководитель УП

\_\_\_\_\_\_\_\_\_\_\_\_ / Уляхин В.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Томск 2024

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 5](#_Toc166089750)

[**Информация о проекте и команде разработчиков** 5](#_Toc166089751)

[**Назначение и структура документа** 5](#_Toc166089752)

[**Глоссарий** 5](#_Toc166089753)

[**ОБЩЕЕ ОПИСАНИЕ** 6](#_Toc166089754)

[**Общий взгляд на предлагаемое решение** 6](#_Toc166089755)

[**Ролевая модель** 7](#_Toc166089756)

[**ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ** 11](#_Toc166089757)

[**Концептуальная модель данных** 11](#_Toc166089758)

[**Логическая модель данных** 12](#_Toc166089759)

[**Сущности** 12](#_Toc166089760)

[**Описание API** 16](#_Toc166089761)

[***Формат ответа общих требований*** 16](#_Toc166089762)

[Регистрация 17](#_Toc166089763)

[Авторизация 18](#_Toc166089764)

[Просмотр категорий товаров 18](#_Toc166089765)

[Просмотр товаров определенной категории 19](#_Toc166089766)

[Просмотр конкретного товара 19](#_Toc166089767)

[Просмотр всех товаров 19](#_Toc166089768)

[Просмотр отзывов у товара 20](#_Toc166089769)

[Просмотр всех отзывов 20](#_Toc166089770)

[Просмотр всех новостей 20](#_Toc166089771)

[Просмотр конкретной новости 21](#_Toc166089772)

[Просмотр способов оплаты 21](#_Toc166089773)

[Просмотр статусов заказов 21](#_Toc166089774)

[Выход пользователя 22](#_Toc166089775)

[Просмотр профиля пользователя 22](#_Toc166089776)

[Изменение пароля в профиле 22](#_Toc166089777)

[Добавление товара в корзину 23](#_Toc166089778)

[Просмотр своей корзины 23](#_Toc166089779)

[Оформление заказа 24](#_Toc166089780)

[Редактирование своей корзины 24](#_Toc166089781)

[Удаление товара из корзины 25](#_Toc166089782)

[Просмотр всех заказов текущего пользователя 25](#_Toc166089783)

[Создание отзыва для купленного товара 25](#_Toc166089784)

[Выход 26](#_Toc166089785)

[Просмотр всех пользователей 26](#_Toc166089786)

[Просмотр конкретного пользователя 27](#_Toc166089787)

[Создание смены 27](#_Toc166089788)

[Редактирование смены 28](#_Toc166089789)

[Удаление сотрудника со смены 28](#_Toc166089790)

[Добавление сотрудника на смену 29](#_Toc166089791)

[Просмотр смен 29](#_Toc166089792)

[Просмотр конкретной смены 29](#_Toc166089793)

[Прикрепление к заказу 30](#_Toc166089794)

[Открепление от заказа 30](#_Toc166089795)

[Создание новости 31](#_Toc166089796)

[Редактирование новости 31](#_Toc166089797)

[Удаление новости 32](#_Toc166089798)

[Создание товара 32](#_Toc166089799)

[Редактирование товара 33](#_Toc166089800)

[Удаление товара 34](#_Toc166089801)

[Просмотр ролей 34](#_Toc166089802)

[Создание роли 35](#_Toc166089803)

[Редактирование роли 35](#_Toc166089804)

[Удаление роли 36](#_Toc166089805)

[Удаление смены 36](#_Toc166089806)

[Создание нового пользователя с присвоением роли 37](#_Toc166089807)

[Редактирование пользователя 38](#_Toc166089808)

[Удаление пользователя 39](#_Toc166089809)

[Редактирование отзыва 39](#_Toc166089810)

[Удаление отзыва 40](#_Toc166089811)

[Создание категории 40](#_Toc166089812)

[Обновление категории 41](#_Toc166089813)

[Удаление категории 41](#_Toc166089814)

[Создание статуса заказа 42](#_Toc166089815)

[Обновление статуса заказа 42](#_Toc166089816)

[Удаление статуса заказа 43](#_Toc166089817)

[Редактирование способа оплаты 43](#_Toc166089818)

[Создание способа оплаты 44](#_Toc166089819)

[Удаление способа оплаты 44](#_Toc166089820)

[Просмотр заказа 45](#_Toc166089821)

[Редактирование заказа 45](#_Toc166089822)

[Удаление заказа 46](#_Toc166089823)

[Просмотр состава заказа по ID заказа 46](#_Toc166089824)

[Просмотр состава заказа 46](#_Toc166089825)

[Создание состава заказа 47](#_Toc166089826)

[Обновление состава заказа 47](#_Toc166089827)

[Удаление товара из состава заказа 48](#_Toc166089828)

[Выручка товара за всё время 49](#_Toc166089829)

[Выручка заказов за период 49](#_Toc166089830)

[Выручка товара за период 50](#_Toc166089831)

[Выручка категории за период 50](#_Toc166089832)

[**ТЕСТИРОВАНИЕ** 51](#_Toc166089833)

[**Авторизация пользователя** 51](#_Toc166089834)

[**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 52](#_Toc166089835)

# **ВВЕДЕНИЕ**

## **Информация о проекте и команде разработчиков**

|  |  |
| --- | --- |
| Название проекта |  |
| Цель проекта |  |
| Задачи проекта | 1. Разработать технического задания API сервиса. 2. Разработать технического задания веб-сайта. 3. Разработать технического задания десктоп приложения. 4. Разработать API сервиса на фреймворке Laravel 10. 5. Разработать веб-сайта на фреймворке Vue 3. 6. Разработать десктоп приложения на C# WPF. 7. Разработать автоматизированные тесты и провести тестирование API сервиса. 8. Разработать автоматизированные тесты и провести тестирование веб-сайта. 9. Разработать автоматизированные тесты и провести тестирование десктоп приложения. |
| Название команды | Команда «Батискаф» |
| Состав команды и роли | Заикин Вадим Анатольевич – Аnalyst, Backend developer, QA.  Шейкина Елизавета Викторовна – Аnalyst, Frontend developer, QA.  Гусейнов Мурад Шахин Оглы – Аnalyst, C# developer, QA. |

## **Назначение и структура документа**

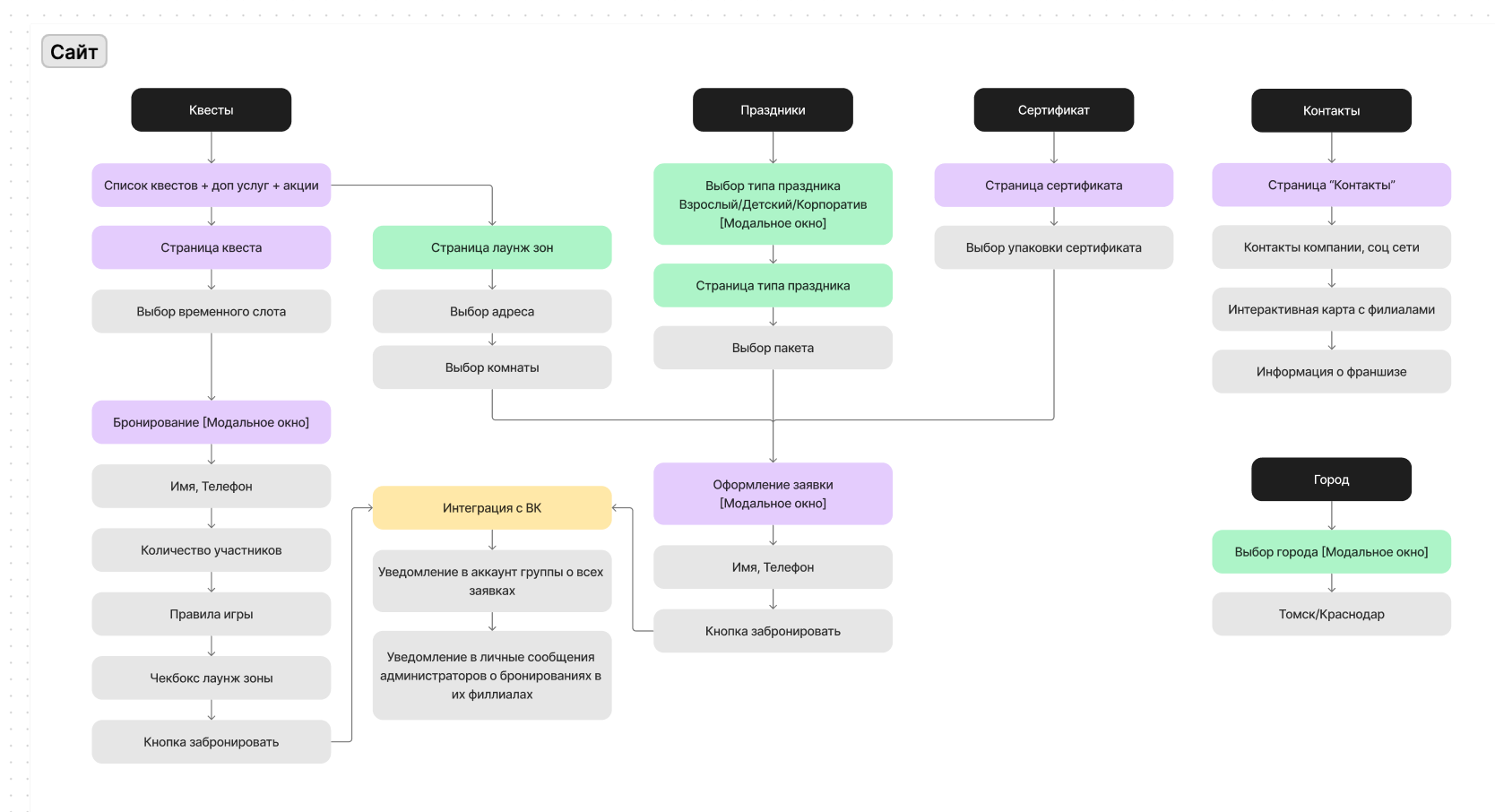
Настоящий документ рассчитан на использование в проектной команде в качестве ключевого документа, определяющего требование по реализации задачи по разработке API сервиса для проекта «…».

## **Глоссарий**

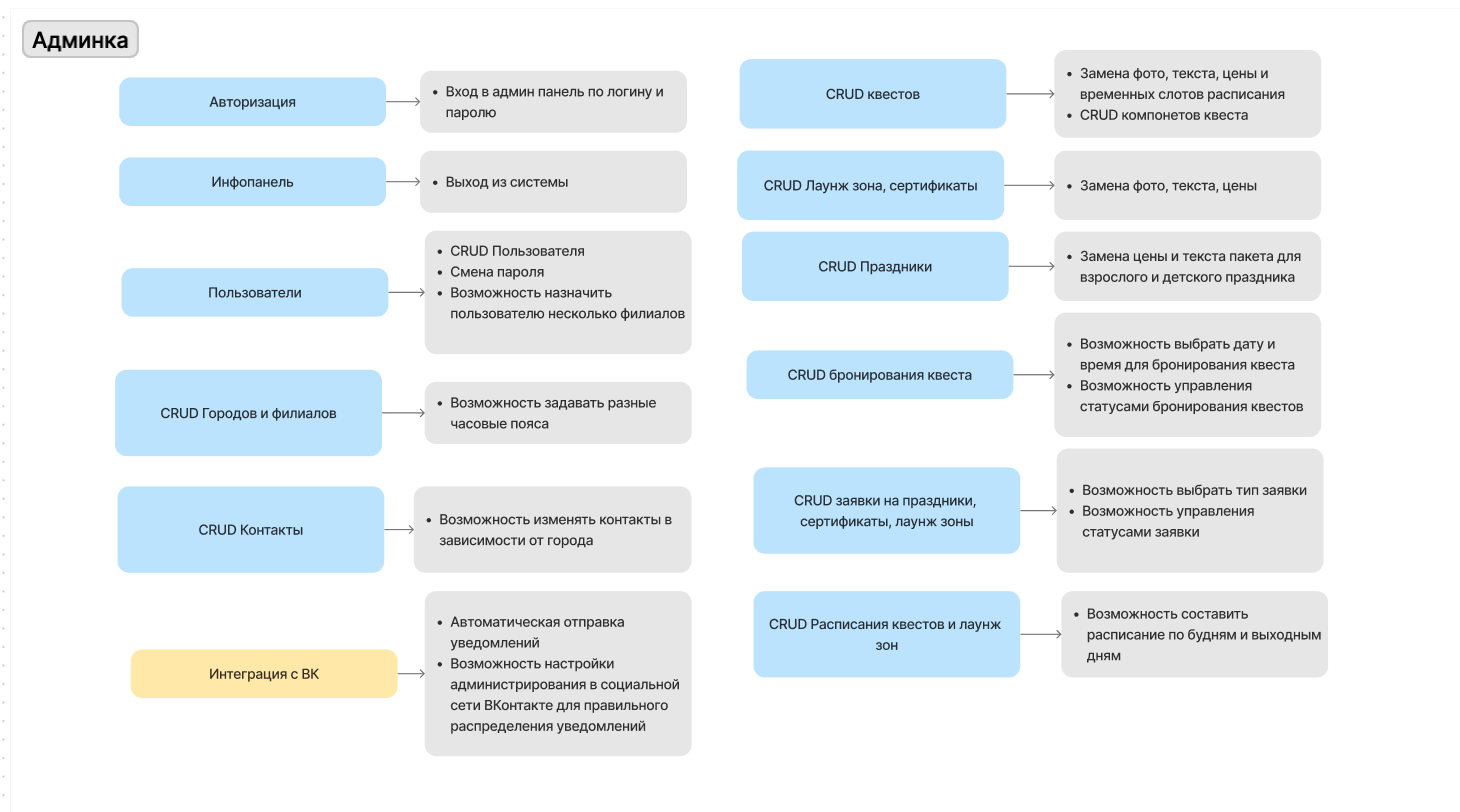
|  |  |
| --- | --- |
| **Понятие** | **Определение понятия** |
| API | API (интерфейс программирования приложений) – это набор определенных правил и соглашений, который используется для взаимодействия между различными программными компонентами. API определяет, как различные части программного обеспечения могут взаимодействовать друг с другом, обычно путем обмена данными и выполнения определенных операций. |
| Postman | Postman – это инструмент для тестирования и разработки API. Он предоставляет удобный интерфейс для отправки HTTP запросов к API, а также для анализа ответов. |
|  |  |
|  |  |
|  |  |

# **ОБЩЕЕ ОПИСАНИЕ**

## **Общий взгляд на предлагаемое решение**



(вставить свою концепцию)



(вставить свою концепцию)

## **Ролевая модель**

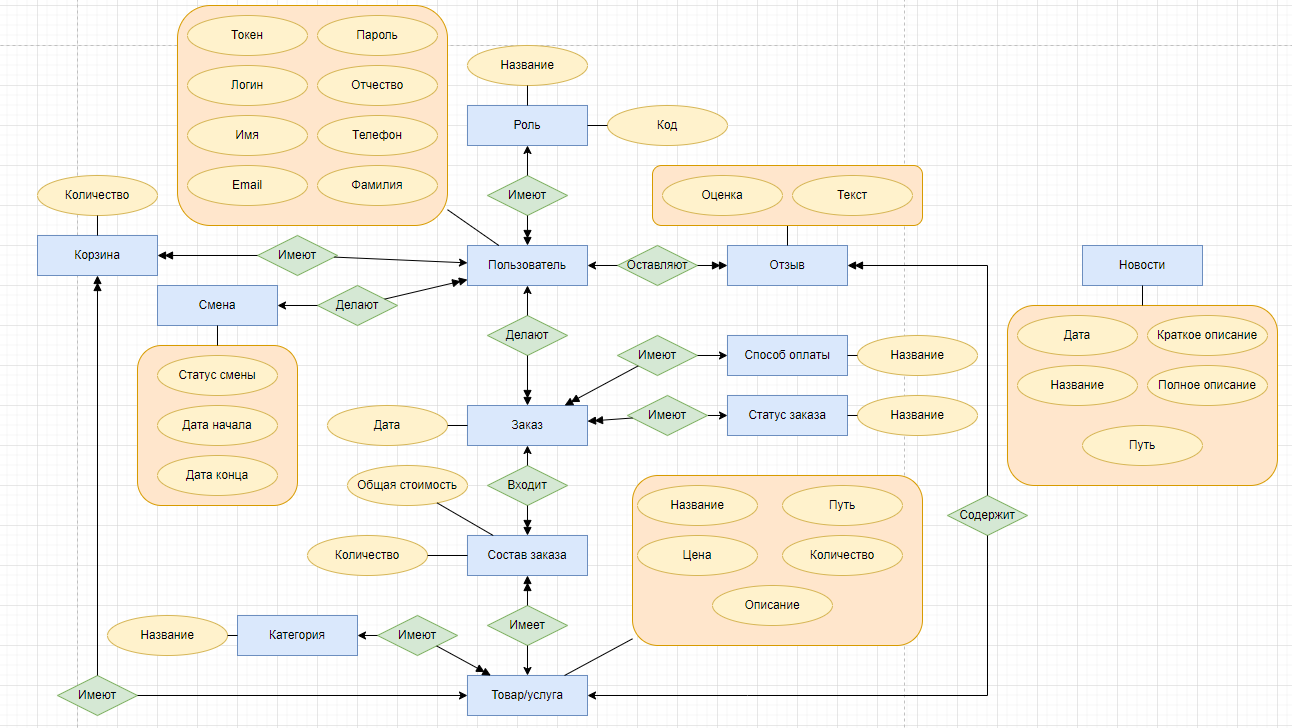
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Раздел** | **Фичи** | **Админ** | **Менеджер** | **Сотрудник** | **Клиент** |
| Инфопанель | Выход из системы | да | да | да | да |
| Пользователи | Просмотр списка пользователей | да | да | нет | нет |
| Просмотр своего профиля | да | да | да | да |
| Регистрация | да | да | да | да |
| Создание пользователя с нужной ролью | да | нет | нет | нет |
| Редактирование пользователя | да | нет | нет | нет |
| Изменение пароля пользователя | да | да (только себя) | да (только себя) | да (только себя) |
| Удаление пользователя | да | нет | нет | нет |
| Просмотреть конкретного пользователя | да | да | нет | нет |
| Роли | Создание роли | да | нет | нет | нет |
| Удаление роли | да | нет | нет | нет |
| Редактирование роли | да | нет | нет | нет |
| Смена | Добавление на смену | да (только сотрудников) | да (только сотрудников) | нет | нет |
| Удаление со смены | да (только сотрудников) | да (только сотрудников) | нет | нет |
| Просмотр смен | да | да | да | нет |
| Просмотр конкретной смены | да | да | да | нет |
| Создание смены | да | да | нет | нет |
| Удаление смены | да | да | нет | нет |
| Редактирование смены | да | да | нет | нет |
| Корзина | Добавление товара в корзину | да | да | да (только в свою) | да (только в свою) |
| Редактирование корзины | да | да | да (только свою) | да (только свою) |
| Удаление товара из корзины | да | да | да (только из своей) | да (только из своей) |
| Просмотр корзины другого пользователя | да | да | нт | нет |
| Просмотр своей корзины | да | да | да | да |
| Новости | Просмотр новостей | да | да | да | да |
| Просмотр конкретной новости | да | да | да | да |
| Редактирование новости | да | нет | нет | нет |
| Удаление новостей | да | нет | нет | нет |
| Создание новостей | да | нет | нет | нет |
| Фото новостей | Просмотр всех фото конкретной новости | да | да | да | да |
| Добавление фото | да | нет | нет | нет |
| Удаление фото | да | нет | нет | нет |
| Редактирование фото | да | нет | нет | нет |
| Фото товаров | Просмотр всех фото конкретного товара | да | да | да | да |
| Добавление фото | да | нет | нет | нет |
| Удаление фото | да | нет | нет | нет |
| Изменение фото | да | нет | нет | нет |
| Отзывы | Просмотр отзывов | да | да | да | да |
| Просмотр отзывов у конкретного товара | да | да | да | да |
| Создание отзыва | да (только на купленный товар) | да (только на купленный товар) | да (только на купленный товар) | да (только на купленный товар) |
| Редактирование отзыва | да | нет | нет | нет |
| Удаление отзыва | да | нет | нет | нет |
| Способы оплаты | Просмотр способов оплаты | да | да | да | да |
| Редактирование способов оплаты | да | нет | нет | нет |
| Создание способа оплаты | да | нет | нет | нет |
| Удаление способа оплаты | да | нет | нет | нет |
| Статус заказов | Просмотр статуса заказов | да | да | да | да |
| Создание статуса заказа | да | нет | нет | нет |
| Редактирование статуса заказа | да | нет | нет | нет |
| Удаление статуса заказа | да | нет | нет | нет |
| Категории | Просмотр категорий | да | да | да | да |
| Создание категорий | да | нет | нет | нет |
| Редактирование категорий | да | нет | нет | нет |
| Удаление категорий | да | нет | нет | нет |
| Состав заказа | Просмотр конкретного состава заказа | да | да | да | да |
| Просмотр составов заказа по идентификатору заказа | да | да | да | да |
| Редактирование состава заказа | да | нет | нет | нет |
| Удаление состава заказа | да | нет | нет | нет |
| Заказы | Оформление заказа | да | да | да | да |
| Редактирование заказа | да | нет | нет | нет |
| Удаление заказа | да | нет | нет | нет |
| Просмотр заказов | да | да | да (только своих) | да (только своих) |
| Прикрепление сотрудника к заказу | да | да | да (только себя) | нет |
| Открепление сотрудника от заказа | да | да | да (только себя) | нет |
| Отчёты | Просмотр выручки товара за всё время | да | нет | нет | нет |
| Просмотр выручки заказов за период | да | нет | нет | нет |
| Просмотр выручки товара за период | да | нет | нет | нет |
| Просмотр выручки категории за период | да | нет | нет | нет |

# **ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ**

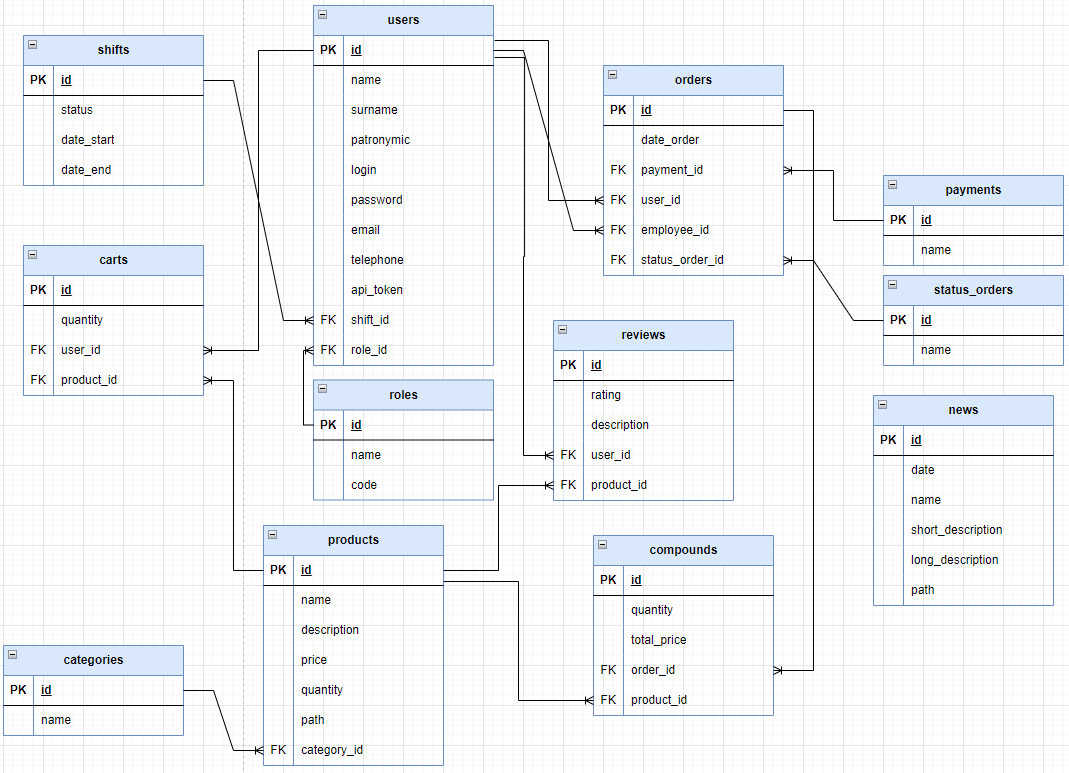
## **Концептуальная модель данных**

Описание концептуальной модели:

1. Пользователь: содержит информацию о пользователе, которые имеют доступ к системе внутри компании. Может содержать поля, такие как имя, фамилия, логин, пароль, телефон, почта, токен.
2. Роль: содержит информацию о роли пользователя. Может содержать поле название.
3. Смена: может содержать поля, такие как дата начала, дата конца, статус смены.
4. Отзыв: содержит информацию о тексте отзывы, а также его оценки. Может содержать поля, такие как текст, оценка.
5. Новости: содержит информацию о название новости, дате публикации, а также описание. Может содержать поля, такие как дата, название, краткое описание, полное описание, путь.
6. Категория: содержит информацию о название категории. Может содержать поле название.
7. Товар: содержит информацию о названии, описании, пути, цены и количества.
8. Способ оплаты: содержит информацию о наименовании способа оплаты.
9. Заказ: содержит информацию о дате заказа.
10. Состав заказа: содержит информацию об общей стоимости, количества.
11. Статус: содержит информацию о названии статуса.
12. Корзина: содержит информацию о количестве товара.



## **Логическая модель данных**



## **Сущности**

**roles (роли)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор роли |
| name | varchar (64) | NOT NULL | Имя роли |
| code | varchar (64) | NOT NULL  UNIQUE | Код роли |

**users (пользователи)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор пользователя |
| name | varchar (128) | NOT NULL | Имя пользователя |
| surname | varchar (128) | NOT NULL | Фамилия пользователя |
| patronymic | varchar (128) | NULL | Отчество пользователя |
| login | varchar (128) | NOT NULL  UNIQUE | Логин пользователя |
| password | varchar (128) | NOT NULL | Пароль пользователя |
| email | varchar (128) | NOT NULL  UNIQUE | Электронная почта пользователя |
| telephone | biginteger | NOT NULL  UNIQUE | Номер телефона пользователя |
| api\_token | varchar (255) | NULL  UNIQUE | Токен пользователя |
| shift\_id | biginteger | FK  NULL | Идентификатор смены |
| role\_id | biginteger | FK  NOT NULL | Идентификатор роли |

**products (товары)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор товара |
| name | varchar (64) | NOT NULL | Название товара |
| description | text | NULL | Описание товара |
| price | decimal (15, 2) | NOT NULL | Цена товара |
| quantity | int | NOT NULL | Количество товара |
| path | varchar (255) | NULL | Путь до фото товара |
| category\_id | biginteger | FK  NOT NULL | Идентификатор категории |

**carts (корзина)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор корзины |
| quantity | int | NOT NULL | Количество товара в корзине |
| user\_id | biginteger | NOT NULL | Идентификатор пользователя |
| product\_id | biginteger | NOT NULL | Идентификатор товара |

**shifts (смены)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор смены |
| status | varchar (64) | NOT NULL | Статус смены |
| date\_start | datetime | NOT NULL | Начало смены |
| date\_end | datetime | NOT NULL | Конец смены |

**categories (категории)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор категорий |
| name | varchar (64) | NOT NULL | Название категории |

**orders (заказы)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор заказа |
| date\_order | datetime | NOT NULL | Дата оформления заказа |
| payment\_id | biginteger | FK  NOT NULL | Идентификатор способа оплаты |
| user\_id | biginteger | FK  NOT NULL | Идентификатор пользователя |
| employee\_id | biginteger | FK  NULL | Идентификатор сотрудника |
| status\_order\_id | biginteger | FK  NOT NULL | Идентификатор статуса заказа |

**reviews (отзывы)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор отзыва |
| rating | int | NOT NULL | Оценка отзыва |
| description | varchar (255) | NOT NULL | Описание отзыва |
| user\_id | biginteger | FK  NOT NULL | Идентификатор пользователя |
| product\_id | biginteger | FK  NOT NULL | Идентификатор товара |

**compounds (состав заказов)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор состава заказа |
| quantity | int | NOT NULL | Количество товаров |
| total\_price | decimal (15, 2) | NOT NULL | Итоговая цена |
| order\_id | biginteger | FK  NOT NULL | Идентификатор заказа |
| product\_id | biginteger | FK  NOT NULL | Идентификатор товара |

**payments (способы оплаты)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK | Идентификатор |
| name | varchar (64) | NOT NULL  AUTO\_INCREMENT | Название способа оплаты |

**status\_orders (статусы заказов)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор статуса заказа |
| name | varchar (64) | NOT NULL | Название статуса заказа |

**news (новости)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | biginteger | PK  AUTO\_INCREMENT | Идентификатор новости |
| date | datetime | NOT NULL | Дата публикации новости |
| name | varchar (64) | NOT NULL | Название новости |
| path | varchar (255) | NULL | Путь до фото новости |
| short\_description | varchar (255) | NOT NULL | Короткое описание новости |
| long\_description | text | NOT NULL | Длинное описание новости |

## **Описание API**

### ***Формат ответа общих требований***

1. Попытка отправить некорретный запрос

Status: 400

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 400,      "message": "Некорректный запрос"  } |

1. Попытка доступа гостя к защищенным авторизацией функциям системы.

Status: 401

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 401,      "message": "Ошибка авторизации"  } |

1. Попытка доступа авторизованного пользователя к функциям недоступным для него.

Status: 403

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 403,      "message": "Доступ запрещён"  } |

1. Попытка получить несуществующий ресурс.

Status: 404

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 404,      "message": "Не найдено"  } |

1. Попытка отправить неприемлимый запрос

Status: 406

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 406,      "message": "Неприемлимо"  } |

1. В случае ошибок связанных с валидацией данных.

Status: 422

Content-Type: application/json

Body:

|  |
| --- |
| {      "success": **false**,      "code": 422,      "message": {          "key1": [              "Error message 1."          ],          "keyN": [              "Error message N."          ]      }  } |

## Регистрация

Описание ресурса: Метод позволяет зарегистрировать нового пользователя.

Метод запроса: POST.

URL-адрес: {{host}}/register

Заголовки запроса:

Content-Type: application/json

Параметры запроса: отсутствуют

Тело запроса:

json

{

"login": "example\_user",

"password": "example\_password",

"name": "Иван",

"surname": "Иванов",

"email": "ivan@example.com",

"telephone": "1234567890"

}

Статус успешного ответа: 200.

Тело успешного ответа:

json

{

"message": "Пользователь успешно зарегистрирован"

}

Возможные статусы: 200, 422

## Авторизация

Описание ресурса: Метод позволяет авторизировать пользователя и получить токен доступа.

Метод запроса: POST.

URL-адрес: {{host}}/login

Заголовки запроса:

Content-Type: application/json

Параметры запроса: отсутствуют

Тело запроса:

json

{

"login": "example\_user",

"password": "example\_password"

}

Статус успешного ответа: 200.

Тело успешного ответа:

json

{

"token": "random\_token"

}

Возможные статусы: 200, 422

## Просмотр категорий товаров

Описание ресурса: Метод позволяет получить список всех категорий товаров.

Метод запроса: GET.

URL-адрес: {{host}}/categories

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив категорий товаров

Возможные статусы: 200

## Просмотр товаров определенной категории

Описание ресурса: Метод позволяет получить список всех товаров определенной категории.

Метод запроса: GET.

URL-адрес: {{host}}/category/{id}

Заголовки запроса: отсутствуют

Параметры запроса:

id (integer): идентификатор категории товаров

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив товаров в указанной категории

Возможные статусы: 200

## Просмотр конкретного товара

Описание ресурса: Метод позволяет получить информацию о конкретном товаре.

Метод запроса: GET.

URL-адрес: {{host}}/product/{id}

Заголовки запроса: отсутствуют

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: информация о товаре

Возможные статусы: 200, 404

## Просмотр всех товаров

Описание ресурса: Метод позволяет получить список всех товаров.

Метод запроса: GET.

URL-адрес: {{host}}/products

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив всех товаров

Возможные статусы: 200

## Просмотр отзывов у товара

Описание ресурса: Метод позволяет получить список отзывов о конкретном товаре.

Метод запроса: GET.

URL-адрес: {{host}}/product/{id}/reviews

Заголовки запроса: отсутствуют

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив отзывов о товаре

Возможные статусы: 200, 404

## Просмотр всех отзывов

Описание ресурса: Метод позволяет получить список всех отзывов.

Метод запроса: GET.

URL-адрес: {{host}}/reviews

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив всех отзывов

Возможные статусы: 200

## Просмотр всех новостей

Описание ресурса: Метод позволяет получить список всех новостей.

Метод запроса: GET.

URL-адрес: {{host}}/news

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив всех новостей

Возможные статусы: 200

## Просмотр конкретной новости

Описание ресурса: Метод позволяет получить информацию о конкретной новости.

Метод запроса: GET.

URL-адрес: {{host}}/news/{id}

Заголовки запроса: отсутствуют

Параметры запроса:

id (integer): идентификатор новости

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: информация о новости

Возможные статусы: 200, 404

## Просмотр способов оплаты

Описание ресурса: Метод позволяет получить список всех доступных способов оплаты.

Метод запроса: GET.

URL-адрес: {{host}}/payments

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив всех способов оплаты

Возможные статусы: 200

## Просмотр статусов заказов

Описание ресурса: Метод позволяет получить список всех доступных статусов заказов.

Метод запроса: GET.

URL-адрес: {{host}}/status\_orders

Заголовки запроса: отсутствуют

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: массив всех статусов заказов

Возможные статусы: 200

## Выход пользователя

Описание ресурса: Метод позволяет пользователю выйти из системы.

Метод запроса: GET.

URL-адрес: {{host}}/logout

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном выходе

Возможные статусы: 200

## Просмотр профиля пользователя

Описание ресурса: Метод позволяет пользователю просмотреть свой профиль.

Метод запроса: GET.

URL-адрес: {{host}}/profile

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: данные профиля пользователя

Возможные статусы: 200

## Изменение пароля в профиле

Описание ресурса: Метод позволяет пользователю изменить свой пароль.

Метод запроса: POST.

URL-адрес: {{host}}/profile/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса:

json

{

"new\_password": "новый\_пароль"

}

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном изменении пароля

Возможные статусы: 200, 422

## Добавление товара в корзину

Описание ресурса: Метод позволяет пользователю добавить товар в корзину.

Метод запроса: POST.

URL-адрес: {{host}}/cart/product/{id}

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном добавлении товара в корзину

Возможные статусы: 200, 422

## Просмотр своей корзины

Описание ресурса: Метод позволяет пользователю просмотреть свою корзину.

Метод запроса: GET.

URL-адрес: {{host}}/cart

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: данные корзины пользователя

Возможные статусы: 200

## Оформление заказа

Описание ресурса: Метод позволяет пользователю оформить заказ на основе товаров в корзине.

Метод запроса: POST.

URL-адрес: {{host}}/checkout

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: данные нового заказа

Возможные статусы: 200, 422

## Редактирование своей корзины

Описание ресурса: Метод позволяет пользователю редактировать свою корзину (изменять количество товаров).

Метод запроса: POST.

URL-адрес: {{host}}/cart/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса:

json

{

"product\_id": "идентификатор\_товара",

"quantity": "новое\_количество"

}

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном обновлении корзины

Возможные статусы: 200, 422

## Удаление товара из корзины

Описание ресурса: Метод позволяет пользователю удалить товар из своей корзины.

Метод запроса: DELETE.

URL-адрес: {{host}}/cart/product/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении товара из корзины

Возможные статусы: 200, 422

## Просмотр всех заказов текущего пользователя

Описание ресурса: Метод позволяет пользователю просмотреть список всех своих заказов.

Метод запроса: GET.

URL-адрес: {{host}}/orders

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: список заказов пользователя

Возможные статусы: 200

## Создание отзыва для купленного товара

Описание ресурса: Метод позволяет пользователю создать отзыв для определенного купленного товара.

Метод запроса: POST.

URL-адрес: {{host}}/product/{id}/review/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса:

json

{

"rating": "рейтинг\_отзыва",

"description": "текст\_отзыва"

}

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном создании отзыва

Возможные статусы: 200, 422

## Выход

Описание ресурса: Метод позволяет пользователю выйти из системы (завершить сеанс аутентификации).

Метод запроса: GET.

URL-адрес: {{host}}/logout

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном выходе

Возможные статусы: 200

## Просмотр всех пользователей

Описание ресурса: Метод позволяет менеджеру просмотреть список всех пользователей системы.

Метод запроса: GET.

URL-адрес: {{host}}/users

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: список пользователей

Возможные статусы: 200

## Просмотр конкретного пользователя

Описание ресурса: Метод позволяет менеджеру просмотреть информацию о конкретном пользователе.

Метод запроса: GET.

URL-адрес: {{host}}/user/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор пользователя

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: информация о пользователе

Возможные статусы: 200, 404

## Создание смены

Описание ресурса: Метод позволяет менеджеру создать новую смену.

Метод запроса: POST.

URL-адрес: {{host}}/shift/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса:

json

{

"date\_start": "дата\_начала\_смены",

"date\_end": "дата\_окончания\_смены",

"status": "статус\_смены"

}

Статус успешного ответа: 200.

Тело успешного ответа: информация о созданной смене

Возможные статусы: 200, 422

## Редактирование смены

Описание ресурса: Метод позволяет менеджеру изменить информацию о существующей смене.

Метод запроса: POST.

URL-адрес: {{host}}/shift/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор смены

Тело запроса:

json

{

"date\_start": "новая\_дата\_начала\_смены",

"date\_end": "новая\_дата\_окончания\_смены",

"status": "новый\_статус\_смены"

}

Статус успешного ответа: 200.

Тело успешного ответа: информация о обновленной смене

Возможные статусы: 200, 404, 422

## Удаление сотрудника со смены

Описание ресурса: Метод позволяет менеджеру удалить сотрудника из указанной смены.

Метод запроса: DELETE.

URL-адрес: {{host}}/shift/{id}/user/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор смены

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении сотрудника из смены

Возможные статусы: 200, 404

## Добавление сотрудника на смену

Описание ресурса: Метод позволяет менеджеру добавить сотрудника на указанную смену.

Метод запроса: POST.

URL-адрес: {{host}}/shift/{id}/user/add

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор смены

Тело запроса:

json

{

"user\_id": "идентификатор\_сотрудника"

}

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном добавлении сотрудника на смену

Возможные статусы: 200, 404, 422

## Просмотр смен

Описание ресурса: Метод позволяет сотруднику просмотреть список всех доступных смен.

Метод запроса: GET.

URL-адрес: {{host}}/shifts

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: список доступных смен

Возможные статусы: 200

## Просмотр конкретной смены

Описание ресурса: Метод позволяет сотруднику просмотреть информацию о конкретной смене.

Метод запроса: GET.

URL-адрес: {{host}}/shift/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор смены

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: информация о смене

Возможные статусы: 200, 404

## Прикрепление к заказу

Описание ресурса: Метод позволяет сотруднику прикрепить себя к указанному заказу.

Метод запроса: POST.

URL-адрес: {{host}}/order/{id}/employee/attach

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном прикреплении к заказу

Возможные статусы: 200, 404, 422

## Открепление от заказа

Описание ресурса: Метод позволяет сотруднику открепить себя от указанного заказа.

Метод запроса: POST.

URL-адрес: {{host}}/order/{id}/employee/detach

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном откреплении от заказа

Возможные статусы: 200, 404, 422

## Создание новости

Описание ресурса: Метод позволяет администратору создать новую новость.

Метод запроса: POST.

URL-адрес: {{host}}/news/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

title (string): заголовок новости

content (string): содержание новости

Тело запроса:

json

{

"title": "Новость",

"content": "Содержание новости"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданная новость

Возможные статусы: 200, 422

## Редактирование новости

Описание ресурса: Метод позволяет администратору отредактировать существующую новость.

Метод запроса: POST.

URL-адрес: {{host}}/news/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор новости

title (string): новый заголовок новости

content (string): новое содержание новости

Тело запроса:

json

{

"title": "Новый заголовок",

"content": "Новое содержание"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированная новость

Возможные статусы: 200, 404, 422

## Удаление новости

Описание ресурса: Метод позволяет администратору удалить существующую новость.

Метод запроса: DELETE.

URL-адрес: {{host}}/news/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор новости

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении новости

Возможные статусы: 200, 404

## Создание товара

Описание ресурса: Метод позволяет администратору создать новый товар.

Метод запроса: POST.

URL-адрес: {{host}}/product/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): название товара

description (string): описание товара

price (decimal): цена товара

category\_id (integer): идентификатор категории товара

Тело запроса:

json

{

"name": "Название товара",

"description": "Описание товара",

"price": 1000.00,

"category\_id": 1

}

Статус успешного ответа: 200.

Тело успешного ответа: созданный товар

Возможные статусы: 200, 422

## Редактирование товара

Описание ресурса: Метод позволяет администратору отредактировать существующий товар.

Метод запроса: POST.

URL-адрес: {{host}}/product/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

name (string): новое название товара

description (string): новое описание товара

price (decimal): новая цена товара

category\_id (integer): новый идентификатор категории товара

Тело запроса:

json

{

"name": "Новое название",

"description": "Новое описание",

"price": 1200.00,

"category\_id": 2

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный товар

Возможные статусы: 200, 404, 422

## Удаление товара

Описание ресурса: Метод позволяет администратору удалить существующий товар.

Метод запроса: DELETE.

URL-адрес: {{host}}/product/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении товара

Возможные статусы: 200, 404

## Просмотр ролей

Описание ресурса: Метод позволяет администратору просмотреть список всех ролей в системе.

Метод запроса: GET.

URL-адрес: {{host}}/roles

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса: отсутствуют

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: список ролей

Возможные статусы: 200

## Создание роли

Описание ресурса: Метод позволяет администратору создать новую роль.

Метод запроса: POST.

URL-адрес: {{host}}/role/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): название новой роли

code (string): уникальный код новой роли

Тело запроса:

json

{

"name": "Название роли",

"code": "Код роли"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданная роль

Возможные статусы: 200, 422

## Редактирование роли

Описание ресурса: Метод позволяет администратору отредактировать существующую роль.

Метод запроса: POST.

URL-адрес: {{host}}/role/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор роли

name (string): новое название роли

code (string): новый уникальный код роли

Тело запроса:

json

{

"name": "Новое название роли",

"code": "Новый код роли"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированная роль

Возможные статусы: 200, 404, 422

## Удаление роли

Описание ресурса: Метод позволяет администратору удалить существующую роль.

Метод запроса: DELETE.

URL-адрес: {{host}}/role/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор роли

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении роли

Возможные статусы: 200, 404

## Удаление смены

Описание ресурса: Метод позволяет администратору удалить существующую смену.

Метод запроса: DELETE.

URL-адрес: {{host}}/shift/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор смены

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении смены

Возможные статусы: 200, 404

## Создание нового пользователя с присвоением роли

Описание ресурса: Метод позволяет администратору создать нового пользователя и назначить ему роль.

Метод запроса: POST.

URL-адрес: {{host}}/user/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): имя пользователя

surname (string): фамилия пользователя

patronymic (string): отчество пользователя

login (string): логин пользователя

email (string): адрес электронной почты пользователя

password (string): пароль пользователя

telephone (string): телефон пользователя

role\_id (integer): идентификатор роли, назначаемой пользователю

Тело запроса:

json

{

"name": "Имя пользователя",

"surname": "Фамилия пользователя",

"patronymic": "Отчество пользователя",

"login": "Логин пользователя",

"email": "Адрес электронной почты пользователя",

"password": "Пароль пользователя",

"telephone": "Телефон пользователя",

"role\_id": "Идентификатор роли"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданный пользователь с присвоенной ролью

Возможные статусы: 200, 422

## Редактирование пользователя

Описание ресурса: Метод позволяет администратору отредактировать существующего пользователя.

Метод запроса: POST.

URL-адрес: {{host}}/user/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор пользователя

name (string): новое имя пользователя

surname (string): новая фамилия пользователя

patronymic (string): новое отчество пользователя

login (string): новый логин пользователя

email (string): новый адрес электронной почты пользователя

password (string): новый пароль пользователя

telephone (string): новый телефон пользователя

Тело запроса:

json

{

"name": "Новое имя пользователя",

"surname": "Новая фамилия пользователя",

"patronymic": "Новое отчество пользователя",

"login": "Новый логин пользователя",

"email": "Новый адрес электронной почты пользователя",

"password": "Новый пароль пользователя",

"telephone": "Новый телефон пользователя"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный пользователь

Возможные статусы: 200, 404, 422

## Удаление пользователя

Описание ресурса: Метод позволяет администратору удалить существующего пользователя.

Метод запроса: DELETE.

URL-адрес: {{host}}/user/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор пользователя

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении пользователя

Возможные статусы: 200, 404

## Редактирование отзыва

Описание ресурса: Метод позволяет администратору изменить существующий отзыв.

Метод запроса: POST.

URL-адрес: {{host}}/review/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор отзыва

rating (integer): новый рейтинг отзыва

description (string): новое описание отзыва

Тело запроса:

json

{

"rating": "Новый рейтинг отзыва",

"description": "Новое описание отзыва"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный отзыв

Возможные статусы: 200, 404, 422

## Удаление отзыва

Описание ресурса: Метод позволяет администратору удалить существующий отзыв.

Метод запроса: DELETE.

URL-адрес: {{host}}/review/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор отзыва

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении отзыва

Возможные статусы: 200, 404

## Создание категории

Описание ресурса: Метод позволяет администратору создать новую категорию товаров.

Метод запроса: POST.

URL-адрес: {{host}}/category/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): название категории

Тело запроса:

json

{

"name": "Название категории"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданная категория

Возможные статусы: 200, 422

## Обновление категории

Описание ресурса: Метод позволяет администратору изменить существующую категорию товаров.

Метод запроса: POST.

URL-адрес: {{host}}/category/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор категории

name (string): новое название категории

Тело запроса:

json

{

"name": "Новое название категории"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированная категория

Возможные статусы: 200, 404, 422

## Удаление категории

Описание ресурса: Метод позволяет администратору удалить существующую категорию товаров.

Метод запроса: DELETE.

URL-адрес: {{host}}/category/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор категории

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении категории

Возможные статусы: 200, 404

## Создание статуса заказа

Описание ресурса: Метод позволяет администратору создать новый статус заказа.

Метод запроса: POST.

URL-адрес: {{host}}/status\_order/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): название статуса заказа

Тело запроса:

json

{

"name": "Название статуса заказа"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданный статус заказа

Возможные статусы: 200, 422

## Обновление статуса заказа

Описание ресурса: Метод позволяет администратору изменить существующий статус заказа.

Метод запроса: POST.

URL-адрес: {{host}}/status\_order/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор статуса заказа

name (string): новое название статуса заказа

Тело запроса:

json

{

"name": "Новое название статуса заказа"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный статус заказа

Возможные статусы: 200, 404, 422

## Удаление статуса заказа

Описание ресурса: Метод позволяет администратору удалить существующий статус заказа.

Метод запроса: DELETE.

URL-адрес: {{host}}/status\_order/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор статуса заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении статуса заказа

Возможные статусы: 200, 404

## Редактирование способа оплаты

Описание ресурса: Метод позволяет администратору изменить существующий способ оплаты.

Метод запроса: POST.

URL-адрес: {{host}}/payment/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор способа оплаты

name (string): новое название способа оплаты

Тело запроса:

json

{

"name": "Новое название способа оплаты"

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный способ оплаты

Возможные статусы: 200, 404, 422

## Создание способа оплаты

Описание ресурса: Метод позволяет администратору создать новый способ оплаты.

Метод запроса: POST.

URL-адрес: {{host}}/payment/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

name (string): название способа оплаты

Тело запроса:

json

{

"name": "Название способа оплаты"

}

Статус успешного ответа: 200.

Тело успешного ответа: созданный способ оплаты

Возможные статусы: 200, 422

## Удаление способа оплаты

Описание ресурса: Метод позволяет администратору удалить существующий способ оплаты.

Метод запроса: DELETE.

URL-адрес: {{host}}/payment/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор способа оплаты

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении способа оплаты

Возможные статусы: 200, 404

## Просмотр заказа

Описание ресурса: Метод позволяет просмотреть информацию о конкретном заказе.

Метод запроса: GET.

URL-адрес: {{host}}/order/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: информация о заказе

Возможные статусы: 200, 404

## Редактирование заказа

Описание ресурса: Метод позволяет администратору изменить информацию о конкретном заказе.

Метод запроса: POST.

URL-адрес: {{host}}/order/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

order\_data (object): объект с новыми данными заказа (например, статус, дата)

Тело запроса:

json

{

"order\_data": {

"status": "новый статус",

"date": "новая дата"

}

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный заказ

Возможные статусы: 200, 404, 422

## Удаление заказа

Описание ресурса: Метод позволяет администратору удалить конкретный заказ.

Метод запроса: DELETE.

URL-адрес: {{host}}/order/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении заказа

Возможные статусы: 200, 404

## Просмотр состава заказа по ID заказа

Описание ресурса: Метод позволяет просмотреть состав заказа по его идентификатору.

Метод запроса: GET.

URL-адрес: {{host}}/compounds/order/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: состав заказа

Возможные статусы: 200, 404

## Просмотр состава заказа

Описание ресурса: Метод позволяет просмотреть состав заказа по его идентификатору.

Метод запроса: GET.

URL-адрес: {{host}}/order/compound/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор состава заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: состав заказа

Возможные статусы: 200, 404

## Создание состава заказа

Описание ресурса: Метод позволяет создать новый состав заказа.

Метод запроса: POST.

URL-адрес: {{host}}/order/{id}/compound/create

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор заказа

compound\_data (object): объект с данными о составе заказа (например, товар, количество)

Тело запроса:

json

{

"compound\_data": {

"product\_id": "идентификатор\_товара",

"quantity": "количество\_товара"

}

}

Статус успешного ответа: 200.

Тело успешного ответа: созданный состав заказа

Возможные статусы: 200, 404, 422

## Обновление состава заказа

Описание ресурса: Метод позволяет администратору обновить данные о составе заказа.

Метод запроса: POST.

URL-адрес: {{host}}/order/compound/{id}/update

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор состава заказа

compound\_data (object): объект с обновленными данными о составе заказа (например, товар, количество)

Тело запроса:

json

{

"compound\_data": {

"product\_id": "новый\_идентификатор\_товара",

"quantity": "новое\_количество\_товара"

}

}

Статус успешного ответа: 200.

Тело успешного ответа: отредактированный состав заказа

Возможные статусы: 200, 404, 422

## Удаление товара из состава заказа

Описание ресурса: Метод позволяет администратору удалить товар из состава заказа.

Метод запроса: DELETE.

URL-адрес: {{host}}/order/compound/{id}/delete

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор состава заказа

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: сообщение об успешном удалении товара из состава заказа

Возможные статусы: 200, 404

## Выручка товара за всё время

Описание ресурса: Метод позволяет получить суммарную выручку по конкретному товару за всё время.

Метод запроса: GET.

URL-адрес: {{host}}/orders/product/{id}

Заголовки запроса:

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

Тело запроса: отсутствует

Статус успешного ответа: 200.

Тело успешного ответа: суммарная выручка по товару за всё время

Возможные статусы: 200, 404

## Выручка заказов за период

Описание ресурса: Метод позволяет получить суммарную выручку по всем заказам за определенный период.

Метод запроса: POST.

URL-адрес: {{host}}/orders/between

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

start\_date (string): начальная дата периода (в формате "YYYY-MM-DD")

end\_date (string): конечная дата периода (в формате "YYYY-MM-DD")

Тело запроса:

json

{

"start\_date": "начальная\_дата",

"end\_date": "конечная\_дата"

}

Статус успешного ответа: 200.

Тело успешного ответа: суммарная выручка за период

Возможные статусы: 200, 422

## Выручка товара за период

Описание ресурса: Метод позволяет получить суммарную выручку по конкретному товару за определенный период.

Метод запроса: POST.

URL-адрес: {{host}}/orders/product/{id}/between

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор товара

start\_date (string): начальная дата периода (в формате "YYYY-MM-DD")

end\_date (string): конечная дата периода (в формате "YYYY-MM-DD")

Тело запроса:

json

{

"start\_date": "начальная\_дата",

"end\_date": "конечная\_дата"

}

Статус успешного ответа: 200.

Тело успешного ответа: суммарная выручка по товару за период

Возможные статусы: 200, 404, 422

## Выручка категории за период

Описание ресурса: Метод позволяет получить суммарную выручку по товарам из определенной категории за определенный период.

Метод запроса: POST.

URL-адрес: {{host}}/orders/category/{id}/between

Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

Параметры запроса:

id (integer): идентификатор категории

start\_date (string): начальная дата периода (в формате "YYYY-MM-DD")

end\_date (string): конечная дата периода (в формате "YYYY-MM-DD")

Тело запроса:

json

{

"start\_date": "начальная\_дата",

"end\_date": "конечная\_дата"

}

Статус успешного ответа: 200.

Тело успешного ответа: суммарная выручка по категории за период

Возможные статусы: 200, 404, 422

# **ТЕСТИРОВАНИЕ**

## **Авторизация пользователя**

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**