**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](#_Toc166284685)

[**Информация о проекте и команде разработчиков** 5](#_Toc166284686)

[**Назначение и структура документа** 5](#_Toc166284687)

[**Глоссарий** 5](#_Toc166284688)

[**ОБЩЕЕ ОПИСАНИЕ** 6](#_Toc166284689)

[**Общий взгляд на предлагаемое решение** 6](#_Toc166284690)

[**Ролевая модель** 8](#_Toc166284691)

[**ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ** 12](#_Toc166284692)

[**Концептуальная модель данных** 12](#_Toc166284693)

[**Логическая модель данных** 13](#_Toc166284694)

[**Сущности** 13](#_Toc166284695)

[**Описание API** 17](#_Toc166284696)

[***Формат ответа общих требований*** 17](#_Toc166284697)

[**Регистрация** 18](#_Toc166284698)

[**Авторизация** 19](#_Toc166284699)

[**Просмотр категорий товаров** 19](#_Toc166284700)

[**Просмотр товаров определенной категории** 20](#_Toc166284701)

[**Просмотр конкретного товара** 20](#_Toc166284702)

[**Просмотр всех товаров** 21](#_Toc166284703)

[**Просмотр отзывов у товара** 21](#_Toc166284704)

[**Просмотр всех отзывов** 22](#_Toc166284705)

[**Просмотр всех новостей** 23](#_Toc166284706)

[**Просмотр конкретной новости** 23](#_Toc166284707)

[**Просмотр способов оплаты** 24](#_Toc166284708)

[**Просмотр статусов заказов** 24](#_Toc166284709)

[**Выход пользователя** 25](#_Toc166284710)

[**Просмотр профиля пользователя** 25](#_Toc166284711)

[**Изменение пароля в профиле** 26](#_Toc166284712)

[**Добавление товара в корзину** 26](#_Toc166284713)

[**Просмотр своей корзины** 27](#_Toc166284714)

[**Оформление заказа** 27](#_Toc166284715)

[**Редактирование своей корзины** 28](#_Toc166284716)

[**Удаление товара из корзины** 28](#_Toc166284717)

[**Просмотр заказов** 29](#_Toc166284718)

[**Создание отзыва для купленного товара** 29](#_Toc166284719)

[**Просмотр всех пользователей** 30](#_Toc166284720)

[**Просмотр конкретного пользователя** 31](#_Toc166284721)

[**Создание смены** 31](#_Toc166284722)

[**Редактирование смены** 32](#_Toc166284723)

[**Удаление сотрудника со смены** 33](#_Toc166284724)

[**Добавление сотрудника на смену** 33](#_Toc166284725)

[**Просмотр смен** 34](#_Toc166284726)

[**Просмотр конкретной смены** 34](#_Toc166284727)

[**Прикрепление к заказу** 35](#_Toc166284728)

[**Открепление от заказа** 35](#_Toc166284729)

[**Создание новости** 36](#_Toc166284730)

[**Редактирование новости** 37](#_Toc166284731)

[**Удаление новости** 38](#_Toc166284732)

[**Создание товара** 38](#_Toc166284733)

[**Редактирование товара** 39](#_Toc166284734)

[**Удаление товара** 40](#_Toc166284735)

[**Просмотр ролей** 40](#_Toc166284736)

[**Создание роли** 41](#_Toc166284737)

[**Редактирование роли** 41](#_Toc166284738)

[**Удаление роли** 42](#_Toc166284739)

[**Удаление смены** 42](#_Toc166284740)

[**Создание нового пользователя с присвоением роли** 42](#_Toc166284741)

[**Редактирование пользователя** 43](#_Toc166284742)

[**Удаление пользователя** 44](#_Toc166284743)

[**Редактирование отзыва** 44](#_Toc166284744)

[**Удаление отзыва** 45](#_Toc166284745)

[**Создание категории** 45](#_Toc166284746)

[**Обновление категории** 46](#_Toc166284747)

[**Удаление категории** 46](#_Toc166284748)

[**Создание статуса заказа** 47](#_Toc166284749)

[**Обновление статуса заказа** 47](#_Toc166284750)

[**Удаление статуса заказа** 48](#_Toc166284751)

[**Редактирование способа оплаты** 48](#_Toc166284752)

[**Создание способа оплаты** 49](#_Toc166284753)

[**Удаление способа оплаты** 49](#_Toc166284754)

[**Просмотр заказа** 50](#_Toc166284755)

[**Редактирование заказа** 50](#_Toc166284756)

[**Удаление заказа** 51](#_Toc166284757)

[**Просмотр состава заказа по ID заказа** 51](#_Toc166284758)

[**Просмотр состава заказа** 52](#_Toc166284759)

[**Создание состава заказа** 53](#_Toc166284760)

[**Обновление состава заказа** 53](#_Toc166284761)

[**Удаление товара из состава заказа** 54](#_Toc166284762)

[**Выручка товара за всё время** 54](#_Toc166284763)

[**Выручка заказов за период** 55](#_Toc166284764)

[**Выручка товара за период** 56](#_Toc166284765)

[**Выручка категории за период** 58](#_Toc166284766)

[**ТЕСТИРОВАНИЕ** 60](#_Toc166284767)

# **ВВЕДЕНИЕ**

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

|  |  |
| --- | --- |
| Название проекта | «Счастливая жизнь в ином мире» |
| Цель проекта | Создание API сервиса для похоронного бюро. |
| 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 «Счастливая жизнь в ином мире».

## **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. |
| Draw.io | Это бесплатный онлайн инструмент для создания диаграмм и схем. Он предоставляет широкий набор инструментов и элементов, которые позволяют пользователям создавать различные типы диаграмм, такие как организационные диаграммы, блок-схемы, ER-диаграммы, сетевые диаграммы, диаграммы потока процессов и многое другое. |
| Laravel | Laravel - это бесплатный, открытый фреймворк для создания веб-приложений на языке программирования PHP. Он предоставляет структуру и базовые инструменты для быстрого создания веб-приложений, используя принципы модели-представления-контроллер (MVC) и другие современные подходы к разработке. |

# **ОБЩЕЕ ОПИСАНИЕ**

## **Общий взгляд на предлагаемое решение**

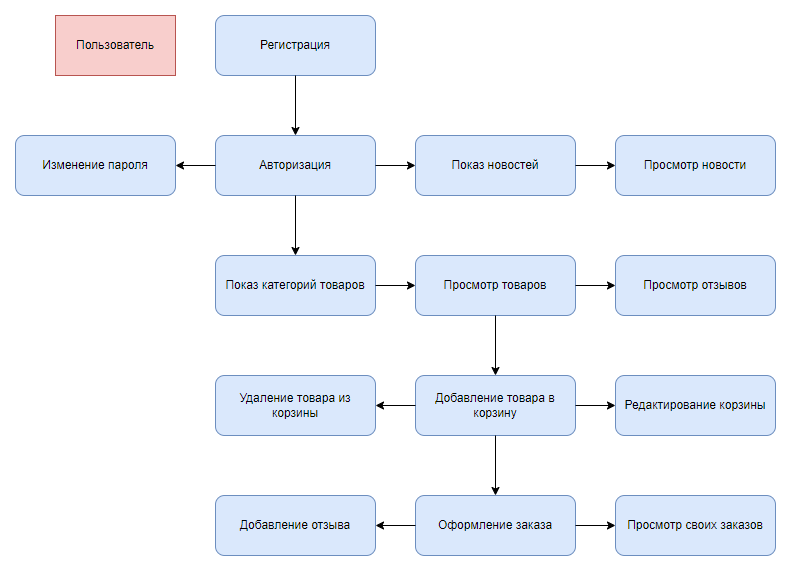


Рисунок 1 - Функционал пользователя

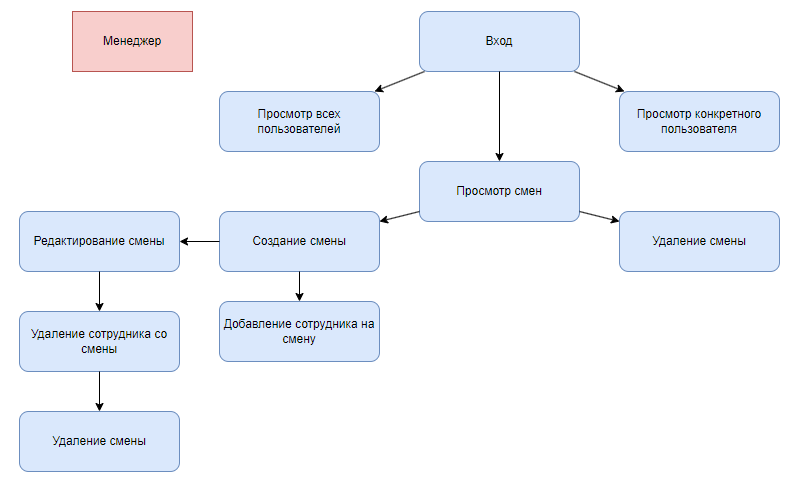


Рисунок 2 - Функционал менеджера

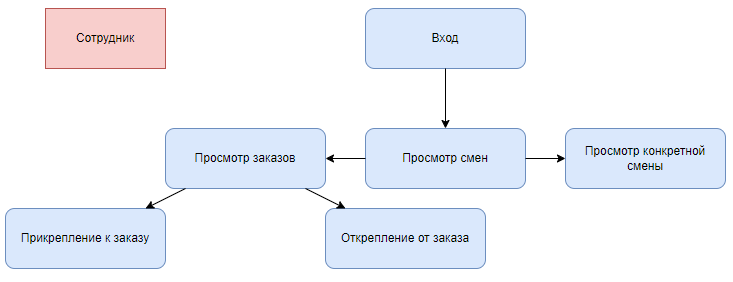


Рисунок 3 - Функционал сотрудника

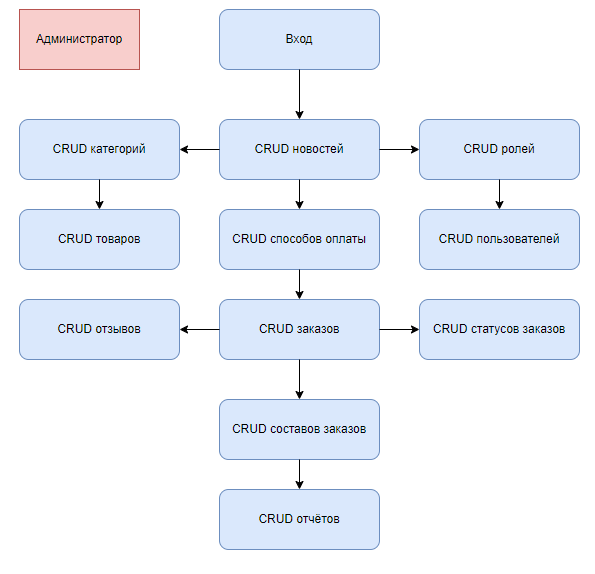


Рисунок 4 - Функционал администратора

## **Ролевая модель**

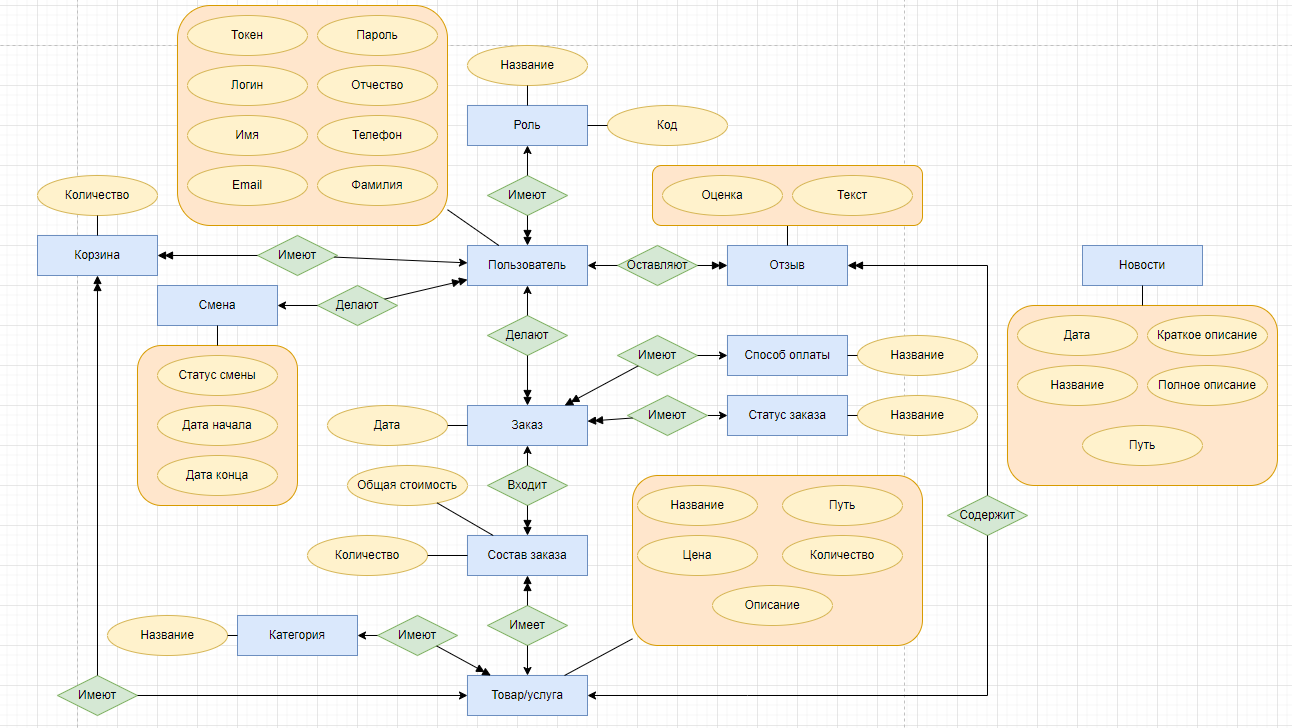
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Раздел** | **Фичи** | **Админ** | **Менеджер** | **Сотрудник** | **Клиент** |
| Инфопанель | Выход из системы | да | да | да | да |
| Пользователи | Просмотр списка пользователей | да | да | нет | нет |
| Просмотр своего профиля | да | да | 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 |
| 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**

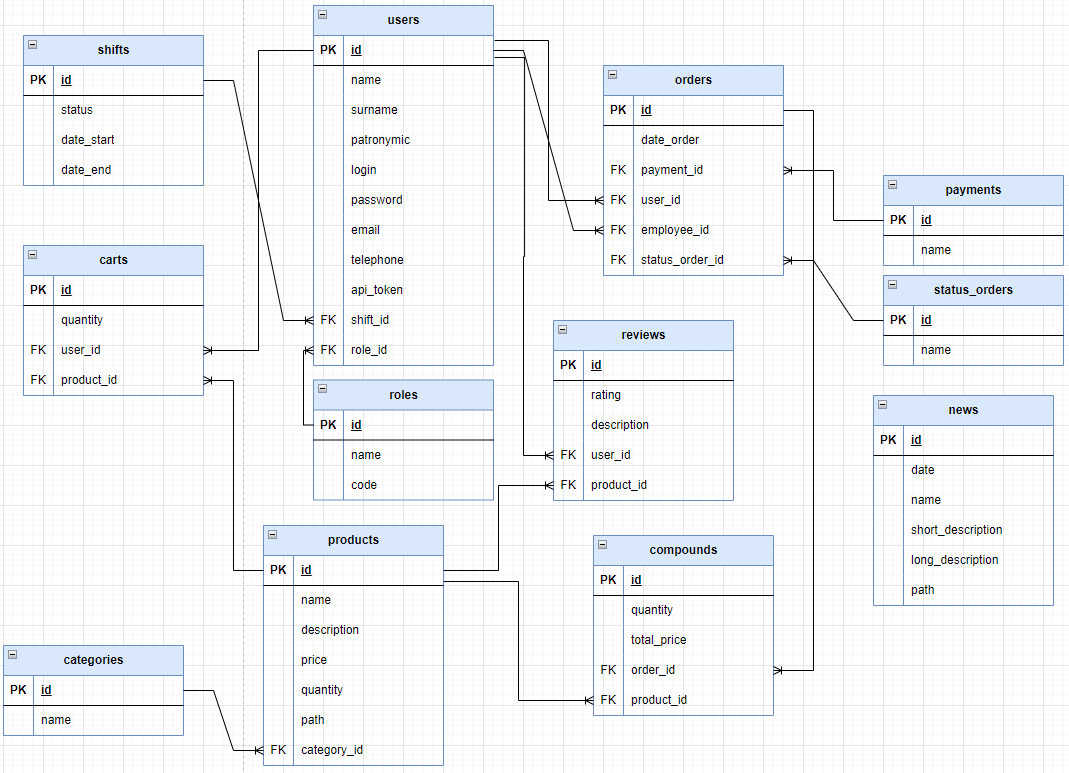
## **Концептуальная модель данных**

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**

1. Описание ресурса: Метод позволяет зарегистрировать нового пользователя.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/register
4. Заголовки запроса: Content-Type: application/json
5. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| login | string | Логин пользователя | До 128 символов | Да |
| password | string | Пароль пользователя | До 255 символов | Да |
| name | string | Имя пользователя | До 128 символов | Да |
| surname | string | Фамилия пользователя | До 128 символов | Да |
| patronymic | string | Отчество пользователя | До 128 символов | Нет |
| email | string | Эл. почта пользователя | До 128 символов | Да |
| telephone | integer | Номер телефона пользователя | От 1 до 20 цифр | Да |

1. Тело запроса:

{

"login": "example\_user",

"password": "example\_password",

"name": "Иван",

"surname": "Иванов",

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

"telephone": "1234567890"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа: []
3. Возможные статусы: 201, 422

### **Авторизация**

1. Описание ресурса: Метод позволяет авторизировать пользователя и получить токен доступа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/login
4. Заголовки запроса: Content-Type: application/json
5. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| login | string | Логин пользователя | До 128 символов | Да |
| password | string | Пароль пользователя | До 255 символов | Да |

1. Тело запроса:

{

"login": "example\_user",

"password": "example\_password"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

{

"token": "random\_token"

}

1. Возможные статусы: 200, 422

### **Просмотр категорий товаров**

1. Описание ресурса: Метод позволяет получить список всех категорий товаров.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/categories
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {},  ]  } |

1. Возможные статусы: 200

### **Просмотр товаров определенной категории**

1. Описание ресурса: Метод позволяет получить список всех товаров определенной категории.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/category/{id}
4. Заголовки запроса: отсутствуют
5. Параметры запроса:

id (integer): идентификатор категории товаров

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string",              "description": **string**,              "price": "decimal",              "quantity": int,              "category\_id": int,              "path": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {          },  ]  } |

1. Возможные статусы: 200, 404

### **Просмотр конкретного товара**

1. Описание ресурса: Метод позволяет получить информацию о конкретном товаре.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/product/{id}
4. Заголовки запроса: отсутствуют
5. Параметры запроса:

id (integer): идентификатор товара

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "name": "string",          "description": **string**,          "price": "decimal",          "quantity": int,          "category\_id": int,          "path": "string",          "created\_at": **null**,          "updated\_at": **null**      }  } |

1. Возможные статусы: 200, 404

### **Просмотр всех товаров**

1. Описание ресурса: Метод позволяет получить список всех товаров.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/products
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": 1,              "name": "string",              "description": "string",              "price": "decimal",              "quantity": int,              "category\_id": int,              "category\_name": "string",              "path": "string"          },  {},  ]  } |

1. Возможные статусы: 200

### **Просмотр отзывов у товара**

1. Описание ресурса: Метод позволяет получить список отзывов о конкретном товаре.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/product/{id}/reviews
4. Заголовки запроса: отсутствуют
5. Параметры запроса:

id (integer): идентификатор товара

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "rating": int,              "description": "string",              "user\_name": "string",              "product\_id": int          },          {          },      ]  } |

1. Возможные статусы: 200, 404

### **Просмотр всех отзывов**

1. Описание ресурса: Метод позволяет получить список всех отзывов.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/reviews
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "rating": int,              "description": "string",              "user\_id": int,              "product\_id": int,              "created\_at": **null**,              "updated\_at": **null**          },          {          },      ]  } |

1. Возможные статусы: 200

### **Просмотр всех новостей**

1. Описание ресурса: Метод позволяет получить список всех новостей.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/news
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "date": "datetime",              "name": "string",              "short\_description": "string",              "long\_description": "string              "path": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {              "id": int,              "date": "datetime",              "name": "string",              "short\_description": "string",              "long\_description": "string              "path": "string",              "created\_at": **null**,              "updated\_at": **null**          },      ]  } |

1. Возможные статусы: 200

### **Просмотр конкретной новости**

1. Описание ресурса: Метод позволяет получить информацию о конкретной новости.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/news/{id}
4. Заголовки запроса: отсутствуют
5. Параметры запроса:

id (integer): идентификатор новости

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "date": "datetime",          "name": "string",          "short\_description": "string",          "long\_description": "string          "path": "string",          "created\_at": **null**,          "updated\_at": **null**      }  } |

1. Возможные статусы: 200, 404

### **Просмотр способов оплаты**

1. Описание ресурса: Метод позволяет получить список всех доступных способов оплаты.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/payments
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа: массив всех способов оплаты

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {          },      ]  } |

1. Возможные статусы: 200

### **Просмотр статусов заказов**

1. Описание ресурса: Метод позволяет получить список всех доступных статусов заказов.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/status\_orders
4. Заголовки запроса: отсутствуют
5. Параметры запроса: отсутствуют
6. Тело запроса: отсутствует
7. Статус успешного ответа: 200.
8. Тело успешного ответа: массив всех статусов заказов

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {          },      ]  } |

1. Возможные статусы: 200

### **Выход пользователя**

1. Описание ресурса: Метод позволяет пользователю выйти из системы.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/logout
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 204.
4. Тело успешного ответа: отсутствует
5. Возможные статусы: 204, 401

### **Просмотр профиля пользователя**

1. Описание ресурса: Метод позволяет пользователю просмотреть свой профиль.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/profile
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "name": "string",          "surname": "string",          "patronymic": "string",          "login": "string",          "password": "string",          "email": "string",          "telephone": int,          "api\_token": "string",          "role\_id": int,          "shift\_id": int,          "created\_at": **null**,          "updated\_at": **null**,          "shifts": **null**      }  } |

1. Возможные статусы: 200

### **Изменение пароля в профиле**

1. Описание ресурса: Метод позволяет пользователю изменить свой пароль.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/profile/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| password | string | Пароль пользователя | От 5 до 255 символов | Нет |

1. Тело запроса:

{

"password": "новый\_пароль"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Пароль изменен"  } |

1. Возможные статусы: 200, 422, 401

### **Добавление товара в корзину**

1. Описание ресурса: Метод позволяет пользователю добавить товар в корзину.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/cart/product/{id}
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| quantity | integer | Количество товара | Число больше 0 | Нет |

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Продукт добавлен в корзину"  } |

1. Возможные статусы: 200, 422, 400, 401

### **Просмотр своей корзины**

1. Описание ресурса: Метод позволяет пользователю просмотреть свою корзину.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/cart
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "quantity": int,              "user\_id": int,              "product\_id": int,              "created\_at": "datetime",              "updated\_at": "datetime"          }      ]  } |

1. Возможные статусы: 200, 404, 401

### **Оформление заказа**

1. Описание ресурса: Метод позволяет пользователю оформить заказ на основе товаров в корзине.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/checkout
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| payment\_id | integer | Идентификатор способа оплаты | Существование способа оплаты | Да |

1. Тело запроса:

{

"payment\_id": "идентификатор\_способа\_оплаты",

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Заказ успешно оформлен"  } |

1. Возможные статусы: 200, 422, 404, 401

### **Редактирование своей корзины**

1. Описание ресурса: Метод позволяет пользователю редактировать свою корзину.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/cart/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| product\_id | integer | Идентификатор товара | Существование товара | Да |
| quantity | string | Количество товара | Число от 1 | Да |

1. Тело запроса:

{

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

"quantity": "новое\_количество"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Количество товара в корзине успешно обновлено"  } |

1. Возможные статусы: 200, 422, 400, 404, 401

### **Удаление товара из корзины**

1. Описание ресурса: Метод позволяет пользователю удалить товар из своей корзины.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/cart/product/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Товар успешно удален из корзины"  } |

1. Возможные статусы: 200, 422, 401, 404

### **Просмотр заказов**

1. Описание ресурса: Метод позволяет пользователю просмотреть список всех своих заказов.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/orders
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "date\_order": "datetime",              "payment\_id": int,              "user\_id": int,              "user": "string",              "employee": "string",              "status\_order": "string"          },          {          },      ]  } |

1. Возможные статусы: 200

### **Создание отзыва для купленного товара**

1. Описание ресурса: Метод позволяет пользователю создать отзыв для определенного купленного товара.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/product/{id}/review/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| rating | integer | Рейтинг отзыва | Число от 1 до 5 | Да |
| description | string | Описание отзыва | До 255 символов | Да |

1. Тело запроса:

{

"rating": "рейтинг\_отзыва",

"description": "текст\_отзыва"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Отзыв успешно сохранен"  } |

1. Возможные статусы: 200, 422, 401

### **Просмотр всех пользователей**

1. Описание ресурса: Метод позволяет менеджеру просмотреть список всех пользователей системы.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/users
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string ",              "surname": "string",              "patronymic": "string",              "login": "string",              "password": "string",              "email": "string              "telephone": int,              "api\_token": "string",              "role\_id": int,              "shift\_id": **null**,              "created\_at": **null**,              "updated\_at": **null**,              "shifts": **null**,              "role": "string"          },          {          },  ]  } |

1. Возможные статусы: 200, 401, 403

### **Просмотр конкретного пользователя**

1. Описание ресурса: Метод позволяет менеджеру просмотреть информацию о конкретном пользователе.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/user/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор пользователя

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "name": "string",          "surname": "string",          "patronymic": "string",          "login": "string",          "password": "string",          "email": "string",          "telephone": int,          "api\_token": "string",          "role\_id": int,          "shift\_id": **null**,          "created\_at": **null**,          "updated\_at": **null**,          "shifts": **null**,          "role": "string"      }  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание смены**

1. Описание ресурса: Метод позволяет менеджеру создать новую смену.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/shift/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| status | boolean | Статус смены | Число от 0 до 1 | Да |
| date\_start | datetime | Начало смены | Y-M-D H:I:S | Да |
| date\_end | datetime | Конец смены | Y-M-D H:I:S | Да |

1. Тело запроса:

{

"date\_start": "дата\_начала\_смены",

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

"status": "статус\_смены"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Смена создана",      "data": {          "date\_start": "datetime",          "date\_end": "datetime",          "status": "bool",          "updated\_at": "datetime",          "created\_at": "datetime",          "id": int      }  } |

1. Возможные статусы: 201, 422, 401, 403, 400

### **Редактирование смены**

1. Описание ресурса: Метод позволяет менеджеру изменить информацию о существующей смене.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/shift/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор смены

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| status | boolean | Статус смены | Число от 0 до 1 | Нет |
| date\_start | datetime | Начало смены | Y-M-D H:I:S | Нет |
| date\_end | datetime | Конец смены | Y-M-D H:I:S | Нет |

1. Тело запроса:

{

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

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

"status": "новый\_статус\_смены"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Смена {shift\_id} обновлена"  } |

1. Возможные статусы: 200, 404, 422, 403, 400, 401

### **Удаление сотрудника со смены**

1. Описание ресурса: Метод позволяет менеджеру удалить сотрудника из указанной смены.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/shift/{id}/user/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор смены

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| employee\_id | integer | Идентификатор сотрудника | Существование сотрудника | Да |

1. Тело запроса:

{

"employee\_id": "идентификатор\_сотрудника"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Пользователь {user\_id} удалён со смены {shift\_id}"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Добавление сотрудника на смену**

1. Описание ресурса: Метод позволяет менеджеру добавить сотрудника на указанную смену.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/shift/{id}/user/add
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор смены

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| employee\_id | integer | Идентификатор сотрудника | Существование сотрудника | Да |

1. Тело запроса:

{

"employee\_id": "идентификатор\_сотрудника"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Пользователь {user\_id} добавлен на смену {shift\_id}"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Просмотр смен**

1. Описание ресурса: Метод позволяет сотруднику просмотреть список всех доступных смен.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/shifts
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| [      {          "shift\_id": int,          "date\_start": "datetime",          "date\_end": "datetime",          "status": int,          "status\_name": "string",          "employees": [              {                  "employee\_id": int,                  "name": "string",                  "surname": "string",                  "patronymic": "string"              }          ]      },  {  },  ] |

1. Возможные статусы: 200, 401, 403

### **Просмотр конкретной смены**

1. Описание ресурса: Метод позволяет сотруднику просмотреть информацию о конкретной смене.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/shift/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор смены

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "shift\_id": int,          "date\_start": "datetime",          "date\_end": "datetime",          "status": bool,          "status\_name": "string",          "employees": [              {                  "employee\_id": int,                  "name": "string",                  "surname": "string",                  "patronymic": "string"              }          ]      }  } |

1. Возможные статусы: 200, 404, 401, 403

### **Прикрепление к заказу**

1. Описание ресурса: Метод позволяет сотруднику прикрепить себя к указанному заказу.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/order/{id}/employee/attach
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

1. Тело запроса:

employee\_id – обязательное, если эту функцию выполняет менеджер или администратор

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Сотрудник {user\_id} прикреплен к заказу {shift\_id}"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Открепление от заказа**

1. Описание ресурса: Метод позволяет сотруднику открепить себя от указанного заказа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/order/{id}/employee/detach
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

1. Тело запроса:

employee\_id – обязательное, если эту функцию выполняет менеджер или администратор

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Сотрудник {user\_id} откреплён от заказа {shift\_id}"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Создание новости**

1. Описание ресурса: Метод позволяет администратору создать новую новость.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/news/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date | datetime | Дата публикации новости | Нет | Да |
| name | string | Название новости | До 64 символов | Да |
| short\_description | string | Краткое описание новости | До 255 символов | Да |
| long\_description | text | Полное описание новости | Нет | Да |
| path | string | Путь до фото новости | Файл. Форматы: jpg, webp, png, jpeg | Нет |

1. Тело запроса:

{

"date": "2020-01-01 12:12:12",

"name": "Название новости",

"short\_description": "Короткое описание",

"long\_description": "Длинное описание",

"path": "Файл.webp",

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Новость создана",      "dataProduct": {          "date": "datetime",          "name": "string",          "short\_description": "string",          "long\_description": "string",          "updated\_at": "datetime",          "created\_at": "datetime",          "id": int      }  } |

1. Возможные статусы: 201, 422, 401, 403

### **Редактирование новости**

1. Описание ресурса: Метод позволяет администратору отредактировать новость.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/news/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор новости

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date | datetime | Дата публикации новости | Нет | Нет |
| name | string | Название новости | До 64 символов | Нет |
| short\_description | string | Краткое описание новости | До 255 символов | Нет |
| long\_description | text | Полное описание новости | Нет | Нет |
| path | file | Фото новости | Файл. Форматы: jpg, webp, png, jpeg | Нет |

1. Тело запроса:

{

"date": "2020-01-01 12:12:12",

"name": "Название новости",

"short\_description": "Короткое описание",

"long\_description": "Длинное описание",

"path": "Файл.webp",

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Новость обновлена",      "dataProduct": {          "date": "datetime",          "name": "string",          "short\_description": "string",          "long\_description": "string",          "updated\_at": "datetime",          "created\_at": "datetime",          "id": int      }  } |

1. Возможные статусы: 200, 422, 401, 403, 404

### **Удаление новости**

1. Описание ресурса: Метод позволяет администратору удалить существующую новость.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/news/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор новости

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Новость {news\_id} удалена"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание товара**

1. Описание ресурса: Метод позволяет администратору создать новый товар.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/product/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название товара | До 64 символов | Да |
| description | string | Описание товара | До 64 символов | Нет |
| price | string | Цена товара | До 255 символов | Да |
| quantity | integer | Количество товара | От 1 символа | Да |
| category\_id | biginteger | Идентификатор категории | Существующие категории | Да |
| path | file | Фото товара | Файл. Форматы: jpg, webp, png, jpeg | Нет |

1. Тело запроса:

{

"name": "Название товара",

"description": "Описание товара",

"price": 1000.00,

"quantity": 1000,

"category\_id": 1

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Товар создан",      "dataProduct": {          "name": "string",          "description": "string",          "price": "decimal",          "quantity": "int",          "category\_id": "int",          "updated\_at": "datetime",          "created\_at": "datetime",          "id": int      }  } |

1. Возможные статусы: 201, 422, 401, 403

### **Редактирование товара**

1. Описание ресурса: Метод позволяет администратору отредактировать существующий товар.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/product/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название товара | До 64 символов | Нет |
| description | string | Описание товара | До 64 символов | Нет |
| price | string | Цена товара | До 255 символов | Нет |
| quantity | integer | Количество товара | От 1 символа | Нет |
| category\_id | biginteger | Идентификатор категории | Существующие категории | Нет |
| path | file | Фото товара | Файл. Форматы: jpg, webp, png, jpeg | Нет |

1. Тело запроса:

{

"name": "Название товара",

"description": "Описание товара",

"price": 1000.00,

"quantity": 1000,

"category\_id": 1

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Товар {product\_id} обновлён"  } |

1. Возможные статусы: 200, 422, 401, 403

### **Удаление товара**

1. Описание ресурса: Метод позволяет администратору удалить существующий товар.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/product/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Товар {product\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Просмотр ролей**

1. Описание ресурса: Метод позволяет администратору просмотреть список всех ролей в системе.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/roles
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса: отсутствуют
2. Тело запроса: отсутствует
3. Статус успешного ответа: 200.
4. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "name": "string",              "code": "string",              "created\_at": **null**,              "updated\_at": **null**          },          {          },      ]  } |

1. Возможные статусы: 200, 401, 403

### **Создание роли**

1. Описание ресурса: Метод позволяет администратору создать новую роль.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/role/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название роли | От 3 символов до 64 символов | Да |
| code | string | Код роли | От 3 символов до 64 символов. Уникален. | Да |

1. Тело запроса:

{

"name": "Название роли",

"code": "Код роли"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Роль создана"  } |

1. Возможные статусы: 201, 422, 401, 403

### **Редактирование роли**

1. Описание ресурса: Метод позволяет администратору отредактировать существующую роль.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/role/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор роли

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название роли | От 3 символов до 64 символов | Нет |
| code | string | Код роли | От 3 символов до 64 символов. Уникален. | Нет |

1. Тело запроса:

{

"name": "Название роли",

"code": "Код роли"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Роль {role\_id} обновлена"  } |

1. Возможные статусы: 200, 422, 401, 403

### **Удаление роли**

1. Описание ресурса: Метод позволяет администратору удалить существующую роль.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/role/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор роли

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Роль {role\_id} удалена"  } |

1. Возможные статусы: 200, 404, 403, 401

### **Удаление смены**

1. Описание ресурса: Метод позволяет удалить существующую смену.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/shift/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор смены

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Смена {shift\_id} удалена"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание нового пользователя с присвоением роли**

1. Описание ресурса: Метод позволяет администратору создать нового пользователя и назначить ему роль.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/user/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| login | string | Логин пользователя | До 128 символов | Да |
| password | string | Пароль пользователя | До 255 символов | Да |
| name | string | Имя пользователя | До 128 символов | Да |
| surname | string | Фамилия пользователя | До 128 символов | Да |
| patronymic | string | Отчество пользователя | До 128 символов | Нет |
| email | string | Эл. почта пользователя | До 128 символов | Да |
| telephone | integer | Номер телефона пользователя | От 1 до 20 цифр | Да |
| role\_id | biginteger | Идентификатор роли | Существующая роль | Да |

1. Тело запроса:

{

"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"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа: []
3. Возможные статусы: 201, 422, 401, 403

### **Редактирование пользователя**

1. Описание ресурса: Метод позволяет администратору отредактировать существующего пользователя.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/user/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:
2. id (integer): идентификатор пользователя

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| login | string | Логин пользователя | До 128 символов | Нет |
| password | string | Пароль пользователя | До 255 символов | Нет |
| name | string | Имя пользователя | До 128 символов | Нет |
| surname | string | Фамилия пользователя | До 128 символов | Нет |
| patronymic | string | Отчество пользователя | До 128 символов | Нет |
| email | string | Эл. почта пользователя | До 128 символов | Нет |
| telephone | integer | Номер телефона пользователя | От 1 до 20 цифр | Нет |
| role\_id | biginteger | Идентификатор роли | Существующая роль | Нет |

1. Тело запроса:

{

"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"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Пользователь {user\_id} обновлён"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление пользователя**

1. Описание ресурса: Метод позволяет администратору удалить существующего пользователя.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/user/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор пользователя

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Пользователь {user\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Редактирование отзыва**

1. Описание ресурса: Метод позволяет администратору изменить существующий отзыв.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/review/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор отзыва

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| rating | string | Логин пользователя | От 1 до 5 цифр | Нет |
| description | string | Пароль пользователя | До 255 символов | Нет |
| user\_id | integer | Идентификатор пользователя | Существующий пользователь | Нет |
| product\_id | integer | Идентификатор товара | Существующий товар | Нет |

1. Тело запроса:

{

"rating": "Новый рейтинг отзыва",

"description": "Новое описание отзыва"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Отзыв {review\_id} обновлён"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление отзыва**

1. Описание ресурса: Метод позволяет администратору удалить существующий отзыв.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/review/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор отзыва

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Отзыв {review\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание категории**

1. Описание ресурса: Метод позволяет администратору создать новую категорию товаров.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/category/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название категории | До 64 символов | Да |

1. Тело запроса:

{

"name": "Название категории"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Категория успешно сохранена"  } |

1. Возможные статусы: 201, 422, 401, 403

### **Обновление категории**

1. Описание ресурса: Метод позволяет администратору изменить существующую категорию товаров.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/category/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор категории

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название категории | До 64 символов | Нет |

1. Тело запроса:

{

"name": "Новое название категории"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Категория {category\_id} обновлена"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление категории**

1. Описание ресурса: Метод позволяет администратору удалить существующую категорию товаров.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/category/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор категории

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Категория {category\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание статуса заказа**

1. Описание ресурса: Метод позволяет администратору создать новый статус заказа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/status\_order/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название статуса заказа | До 64 символов | Да |

1. Тело запроса:

{

"name": "Название статуса заказа"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Статус заказа успешно сохранён"  } |

1. Возможные статусы: 201, 422, 401, 403

### **Обновление статуса заказа**

1. Описание ресурса: Метод позволяет администратору изменить существующий статус заказа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/status\_order/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор статуса заказа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название статуса заказа | До 64 символов | Нет |

1. Тело запроса:

{

"name": "Новое название статуса заказа"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Статус заказа {status\_order\_id} обновлён"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление статуса заказа**

1. Описание ресурса: Метод позволяет администратору удалить существующий статус заказа.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/status\_order/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор статуса заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Статус заказа {status\_order\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Редактирование способа оплаты**

1. Описание ресурса: Метод позволяет администратору изменить существующий способ оплаты.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/payment/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор способа оплаты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название способа оплаты | До 64 символов | Нет |

1. Тело запроса:

{

"name": "Новое название способа оплаты"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Способ оплаты {payment\_id} обновлён"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Создание способа оплаты**

1. Описание ресурса: Метод позволяет администратору создать новый способ оплаты.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/payment/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| name | string | Название способа оплаты | До 64 символов | Да |

1. Тело запроса:

{

"name": "Название способа оплаты"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Способ оплаты успешно сохранен"  } |

1. Возможные статусы: 201, 422, 401, 403

### **Удаление способа оплаты**

1. Описание ресурса: Метод позволяет администратору удалить существующий способ оплаты.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/payment/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор способа оплаты

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Способ оплаты {payment\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Просмотр заказа**

1. Описание ресурса: Метод позволяет просмотреть информацию о конкретном заказе.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/order/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "date\_order": "datetime",          "payment\_id": int,          "user\_id": int,          "user": "string",          "products": [              {                  "compound\_id": int,                  "product\_id": int,                  "total\_price": "decimal",                  "quantity": int,                  "product": "string"              },              {              }          ],          "employee": "string",          "status\_order": "string"      }  } |

1. Возможные статусы: 200, 404, 401, 403

### **Редактирование заказа**

1. Описание ресурса: Метод позволяет администратору изменить информацию о конкретном заказе.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/order/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date\_order | datetime | Дата оформления | Нет | Нет |
| payment\_id | integer | Идентификатор способа оплаты | Существующий способ оплаты | Нет |
| user\_id | intger | Идентификатор пользователя | Существующий пользователь | Нет |
| employee\_id | integer | Идентификатор сотрудника | Существующий сотрудник | Нет |
| status\_order\_id | integer | Идентификатор статуса заказа | Существующий статус заказа | Нет |

1. Тело запроса:

{

"status\_order\_id": 1,

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Заказ {order\_id} обновлён"  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление заказа**

1. Описание ресурса: Метод позволяет администратору удалить конкретный заказ.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/order/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Удалено"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Просмотр состава заказа по ID заказа**

1. Описание ресурса: Метод позволяет просмотреть состав заказа по его идентификатору.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/compounds/order/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "quantity": int,              "total\_price": "decimal",              "order\_id": int,              "product\_id": int,              "created\_at": **null**,              "updated\_at": **null**          },          {          }      ]  } |

1. Возможные статусы: 200, 404, 401, 403

### **Просмотр состава заказа**

1. Описание ресурса: Метод позволяет просмотреть состав заказа по его идентификатору.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/order/compound/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор состава заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "quantity": int,          "total\_price": "decimal",          "order\_id": int,          "product\_id": int,          "created\_at": **null**,          "updated\_at": **null**      }  } |

1. Возможные статусы: 200, 404, 401, 403

### **Создание состава заказа**

1. Описание ресурса: Метод позволяет создать новый состав заказа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/order/{id}/compound/create
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор заказа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| product\_id | integer | Идентификатор товара | Существующий товар | Да |
| quantity | integer | Количество товара | От 1 символа | Да |
| total\_price | decimal | Итоговая цена | Нет | Нет |

1. Тело запроса:

{

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

"quantity": "количество\_товара"

}

1. Статус успешного ответа: 201.
2. Тело успешного ответа:

|  |
| --- |
| {      "message": "Состав заказа создан для заказа {order\_id}",      "data": {          "product\_id": "int",          "quantity": "int",          "total\_price": decimal,          "order\_id": int,          "updated\_at": "datetime",          "created\_at": "datetime",          "id": int      }  } |

1. Возможные статусы: 201, 404, 422, 401, 403

### **Обновление состава заказа**

1. Описание ресурса: Метод позволяет администратору обновить данные о составе заказа.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/order/compound/{id}/update
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор состава заказа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| product\_id | integer | Идентификатор товара | Существующий товар | Нет |
| quantity | integer | Количество товара | От 1 символа | Нет |
| total\_price | decimal | Итоговая цена | Нет | Нет |

1. Тело запроса:

{

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

"quantity": "новое\_количество\_товара"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "data": {          "id": int,          "quantity": "int",          "total\_price": decimal,          "order\_id": "int",          "product\_id": "int",          "product\_name": "string"      }  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Удаление товара из состава заказа**

1. Описание ресурса: Метод позволяет администратору удалить товар из состава заказа.
2. Метод запроса: DELETE.
3. URL-адрес: {{host}}/order/compound/{id}/delete
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор состава заказа

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "message": "Состав заказа {compound\_id} удалён"  } |

1. Возможные статусы: 200, 404, 401, 403

### **Выручка товара за всё время**

1. Описание ресурса: Метод позволяет получить суммарную выручку по конкретному товару за всё время.
2. Метод запроса: GET.
3. URL-адрес: {{host}}/orders/product/{id}
4. Заголовки запроса:

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

1. Тело запроса: отсутствует
2. Статус успешного ответа: 200.
3. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "quantity": int,              "total\_price": "decimal",              "order\_id": int,              "product\_id": int,              "created\_at": **null**,              "updated\_at": **null**,              "orders": {                  "id": int,                  "date\_order": "datetime                  "payment\_id": int,                  "user\_id": int,                  "employee\_id": **null**,                  "status\_order\_id": int,                  "created\_at": **null**,                  "updated\_at": **null**              }          },          {          }      ],      "total\_money": decimal,      "order\_count": int,      "product\_count": int  } |

1. Возможные статусы: 200, 404, 401, 403

### **Выручка заказов за период**

1. Описание ресурса: Метод позволяет получить суммарную выручку по всем заказам за определенный период.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/orders/between
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date\_start | date | Начало периода | YYYY-MM-DD | Да |
| date\_end | date | Конец периода | YYYY-MM-DD | Да |

1. Тело запроса:

{

"date\_start": "начальная\_дата",

"date\_end": "конечная\_дата"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "date\_order": "datetime",              "payment\_id": int,              "user\_id": int,              "user": "string              "employee": "string",              "status\_order": "string",              "compounds": [                  {                      "id": int,                      "order\_id": int,                      "quantity": int,                      "total\_price": "decimal",                      "product\_id": int,                      "product": {                          "id": int,                          "name": "string"                      }                  },              ]          },          {          },      ],      "total\_money": decimal,      "count\_order": int  } |

1. Возможные статусы: 200, 422, 401, 404, 403

### **Выручка товара за период**

1. Описание ресурса: Метод позволяет получить суммарную выручку по конкретному товару за определенный период.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/orders/product/{id}/between
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор товара

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date\_start | date | Начало периода | YYYY-MM-DD | Да |
| date\_end | date | Конец периода | YYYY-MM-DD | Да |

1. Тело запроса:

{

"date\_start": "начальная\_дата",

"date\_end": "конечная\_дата"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "quantity": int,              "total\_price": "decimal",              "order\_id": int,              "product\_id": int,              "created\_at": **null**,              "updated\_at": **null**,              "orders": {                  "id": int,                  "date\_order": "datetime",                  "payment\_id": int,                  "user\_id": int,                  "employee\_id": **null**,                  "status\_order\_id": int,                  "created\_at": **null**,                  "updated\_at": **null**              }          }      ],      "price": "decimal",      "total\_money": decimal,      "order\_count": int,      "product\_count": int,      "product\_name": "string",      "product\_description": "string",  } |

1. Возможные статусы: 200, 404, 422, 401, 403

### **Выручка категории за период**

1. Описание ресурса: Метод позволяет получить суммарную выручку по товарам из определенной категории за определенный период.
2. Метод запроса: POST.
3. URL-адрес: {{host}}/orders/category/{id}/between
4. Заголовки запроса:

Content-Type: application/json

Authorization: Bearer {токен}

1. Параметры запроса:

id (integer): идентификатор категории

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | **Ограничения** | **Обязательность** |
| date\_start | date | Начало периода | YYYY-MM-DD | Да |
| date\_end | date | Конец периода | YYYY-MM-DD | Да |

1. Тело запроса:

{

"date\_start": "начальная\_дата",

"date\_end": "конечная\_дата"

}

1. Статус успешного ответа: 200.
2. Тело успешного ответа:

|  |
| --- |
| {      "data": [          {              "id": int,              "quantity": int,              "total\_price": "decimal",              "order\_id": int,              "product\_id": int,              "created\_at": **null**,              "updated\_at": **null**,              "products": {                  "id": int,                  "name": "string",                  "description": **null**,                  "price": "decimal",                  "quantity": int,                  "category\_id": int,                  "path": "string",                  "created\_at": **null**,                  "updated\_at": **null**              }          },          {          },      ],      "total\_money": int,      "order\_count": int,      "product\_count": int,      "category\_name": "string",      "category\_description": **null**  } |

1. Возможные статусы: 200, 404, 422. 401, 403

# **ТЕСТИРОВАНИЕ**

**Название теста:** Проверка авторизации, статус код 401

**Описание:** Этот тест проверяет аутентификацию пользователя, он отправляет запрос без авторизации на ресурс, ожидая получить ответ с кодом состояния 401 и сообщением "Ошибка авторизации", а также проверяет время на ответ в 500мс.

**Ресурсы:**

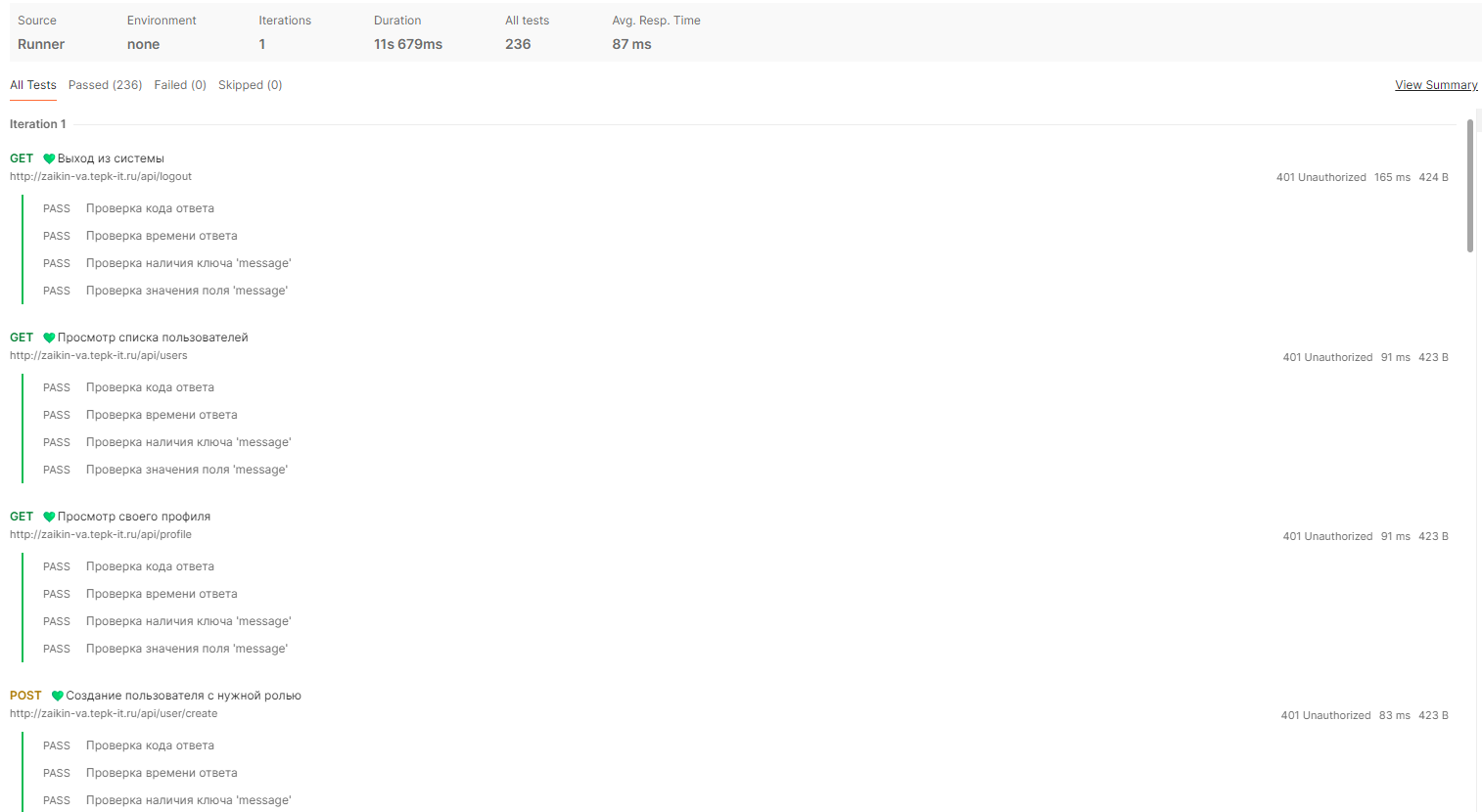
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| GET | {{host}}/logout |
| GET | {{host}}/profile |
| POST | {{host}}/profile/update |
| POST | {{host}}/cart/product/{id} |
| GET | {{host}}/cart |
| POST | {{host}}/checkout |
| POST | {{host}}/cart/update |
| DELETE | {{host}}/cart/product/{id} |
| GET | {{host}}/orders |
| POST | {{host}}/product/{id}/review/create |
| GET | {{host}}/users |
| GET | {{host}}/user/{id} |
| POST | {{host}}/shift/create |
| POST | {{host}}/shift/{id}/update |
| DELETE | {{host}}/shift/{id}/delete |
| DELETE | {{host}}/shift/{id}/user/delete |
| POST | {{host}}/shift/{id}/user/add |
| GET | {{host}}/shifts |
| GET | {{host}}/shift/{id} |
| POST | {{host}}/order/{id}/employee/attach |
| POST | {{host}}/order/{id}/employee/detach |
| POST | {{host}}/news/create |
| POST | {{host}}/news/{id}/update |
| DELETE | {{host}}/news/{id}/delete |
| POST | {{host}}/product/create |
| POST | {{host}}/product/{id}/update |
| DELETE | {{host}}/product/{id}/delete |
| GET | {{host}}/roles |
| POST | {{host}}/role/create |
| POST | {{host}}/role/{id}/update |
| DELETE | {{host}}/role/{id}/delete |
| POST | {{host}}/user/create |
| POST | {{host}}/user/{id}/update |
| DELETE | {{host}}/user/{id}/delete |
| POST | {{host}}/review/{id}/update |
| DELETE | {{host}}/review/{id}/delete |
| POST | {{host}}/category/create |
| POST | {{host}}/category/{id}/update |
| DELETE | {{host}}/category/{id}/delete |
| POST | {{host}}/status\_order/create |
| POST | {{host}}/status\_order/{id}/update |
| DELETE | {{host}}/status\_order/{id}/delete |
| POST | {{host}}/payment/{id}/update |
| POST | {{host}}/payment/create |
| DELETE | {{host}}/payment/{id}/delete |
| GET | {{host}}/order/{id} |
| POST | {{host}}/order/{id}/update |
| DELETE | {{host}}/order/{id}/delete |
| GET | {{host}}/compounds/order/{id} |
| GET | {{host}}/order/compound/{id} |
| POST | {{host}}/order/{id}/compound/create |
| POST | {{host}}/order/compound/{id}/update |
| DELETE | {{host}}/order/compound/{id}/delete |
| GET | {{host}}/orders/product/{id} |
| POST | {{host}}/orders/between |
| POST | {{host}}/orders/product/{id}/between |
| POST | {{host}}/orders/category/{id}/between |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(401);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  });  pm.**test**("Проверка значения поля 'message'", **function** () {      pm.expect(pm.response.json().message).to.eql("Ошибка авторизации");  }); |

**Результаты тестирования:**

Все пути успешны прошли проверку



**Название теста:** Проверка доступности функционала, статус код 403

**Описание:** Этот тест проверяет ошибку доступа для пользователя с недостаточными правами, он отправляет запрос с авторизацией простого пользователя на ресурс, ожидая получить ответ с кодом состояния 403 и сообщением "Доступ запрещён", а также проверяет время на ответ в 500мс.

**Ресурсы:**

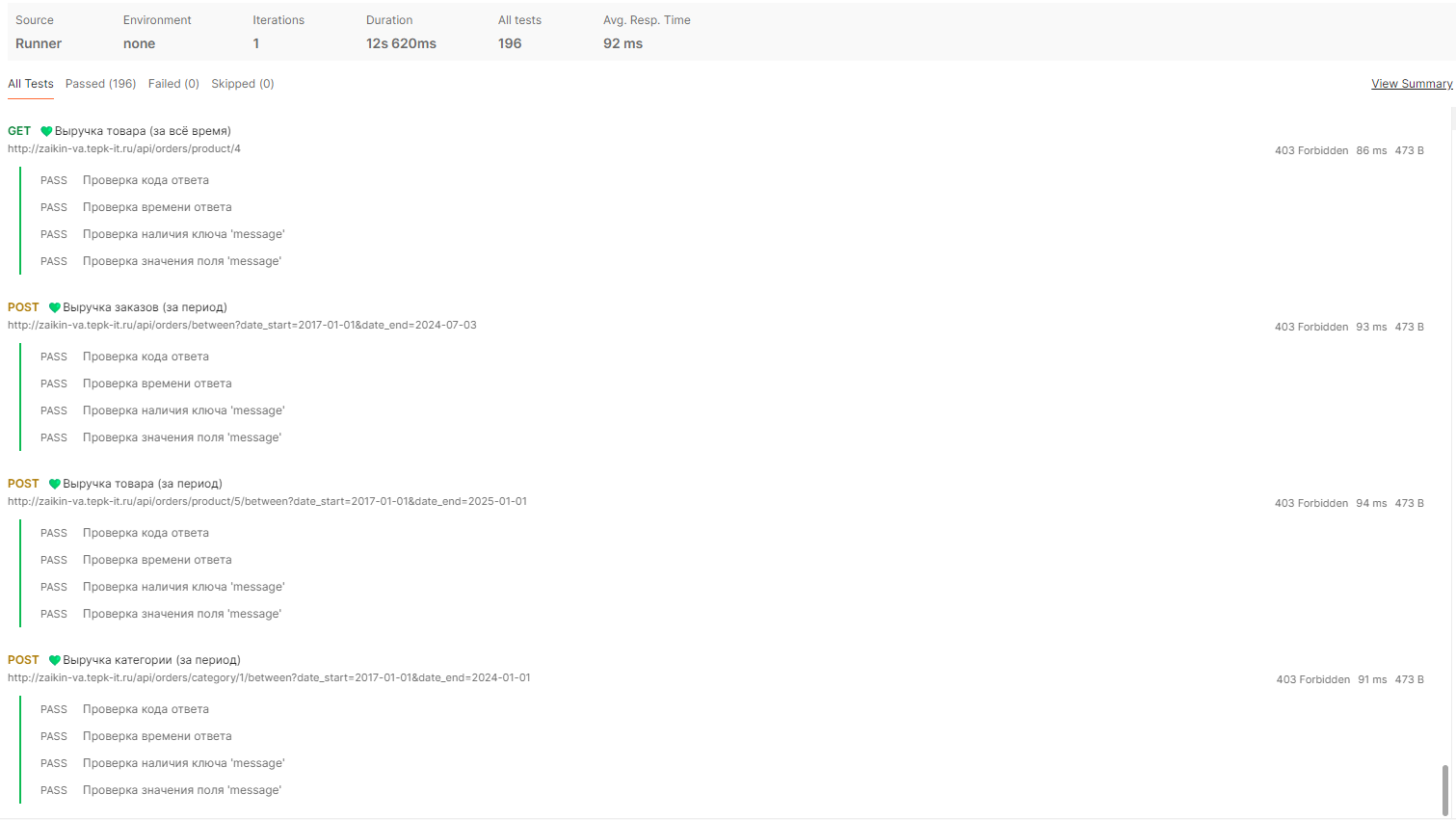
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| GET | {{host}}/users |
| GET | {{host}}/user/{id} |
| POST | {{host}}/shift/create |
| POST | {{host}}/shift/{id}/update |
| DELETE | {{host}}/shift/{id}/delete |
| DELETE | {{host}}/shift/{id}/user/delete |
| POST | {{host}}/shift/{id}/user/add |
| GET | {{host}}/shifts |
| GET | {{host}}/shift/{id} |
| POST | {{host}}/order/{id}/employee/attach |
| POST | {{host}}/order/{id}/employee/detach |
| POST | {{host}}/news/create |
| POST | {{host}}/news/{id}/update |
| DELETE | {{host}}/news/{id}/delete |
| POST | {{host}}/product/create |
| POST | {{host}}/product/{id}/update |
| DELETE | {{host}}/product/{id}/delete |
| GET | {{host}}/roles |
| POST | {{host}}/role/create |
| POST | {{host}}/role/{id}/update |
| DELETE | {{host}}/role/{id}/delete |
| POST | {{host}}/user/create |
| POST | {{host}}/user/{id}/update |
| DELETE | {{host}}/user/{id}/delete |
| POST | {{host}}/review/{id}/update |
| DELETE | {{host}}/review/{id}/delete |
| POST | {{host}}/category/create |
| POST | {{host}}/category/{id}/update |
| DELETE | {{host}}/category/{id}/delete |
| POST | {{host}}/status\_order/create |
| POST | {{host}}/status\_order/{id}/update |
| DELETE | {{host}}/status\_order/{id}/delete |
| POST | {{host}}/payment/{id}/update |
| POST | {{host}}/payment/create |
| DELETE | {{host}}/payment/{id}/delete |
| GET | {{host}}/order/{id} |
| POST | {{host}}/order/{id}/update |
| DELETE | {{host}}/order/{id}/delete |
| GET | {{host}}/compounds/order/{id} |
| GET | {{host}}/order/compound/{id} |
| POST | {{host}}/order/{id}/compound/create |
| POST | {{host}}/order/compound/{id}/update |
| DELETE | {{host}}/order/compound/{id}/delete |
| GET | {{host}}/orders/product/{id} |
| POST | {{host}}/orders/between |
| POST | {{host}}/orders/product/{id}/between |
| POST | {{host}}/orders/category/{id}/between |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(403);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  });  pm.**test**("Проверка значения поля 'message'", **function** () {      pm.expect(pm.response.json().message).to.eql("Доступ запрещён");  }); |

**Результаты тестирования:**

Все пути успешны прошли проверку



**Название теста:** Проверка корректности запроса, статус код 400

**Описание:** Этот тест проверяет ошибку корректности, он отправляет запрос с некорректными данными, ожидая получить ответ с кодом состояния 400 и сообщением "Некорректный запрос", а также проверяет время на ответ в 500мс.

**Ресурсы:**

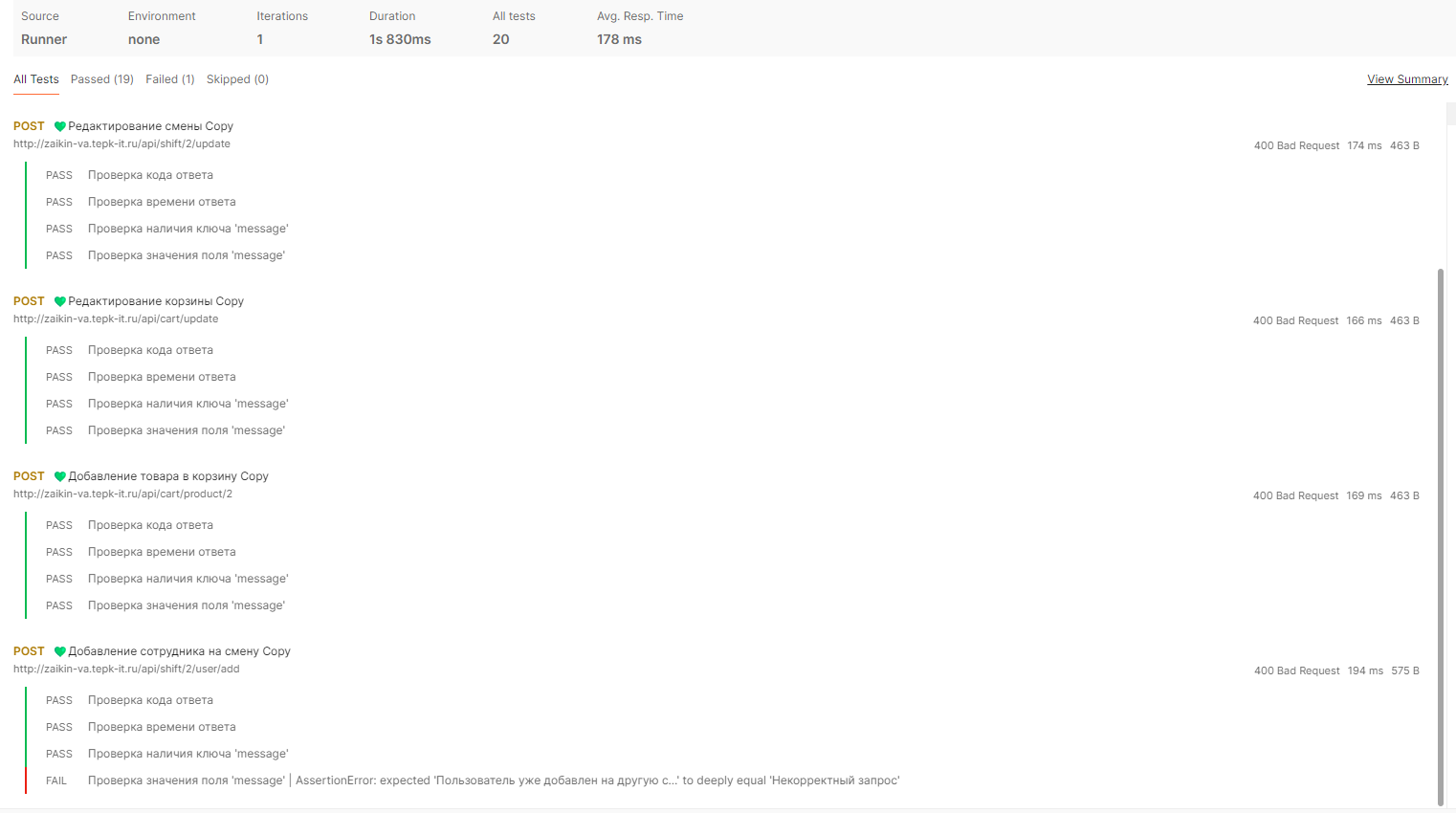
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| POST | {{host}}/shift/create |
| POST | {{host}}/shift/{id}/update |
| POST | {{host}}/cart/update |
| POST | {{host}}/cart/product/{id} |
| POST | {{host}}/shift/{id}/user/add |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(400);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  });  pm.**test**("Проверка значения поля 'message'", **function** () {      pm.expect(pm.response.json().message).to.eql("Некорректный запрос");  }); |

**Результаты тестирования:**

Практически все пути успешно прошли тест, за исключением одного {{host}}/shift/{id}/user/add, потому что в поле message выводится другое сообщение.



**Название теста:** Проверка существования, статус код 404

**Описание:** Этот тест проверяет обработку ошибки при запросе на несуществующий параметр, он отправляет запрос на ресурс, ожидая получить ответ с кодом состояния 404 и сообщением "Не найдено", а также проверяет время на ответ в 500мс.

**Ресурсы:**

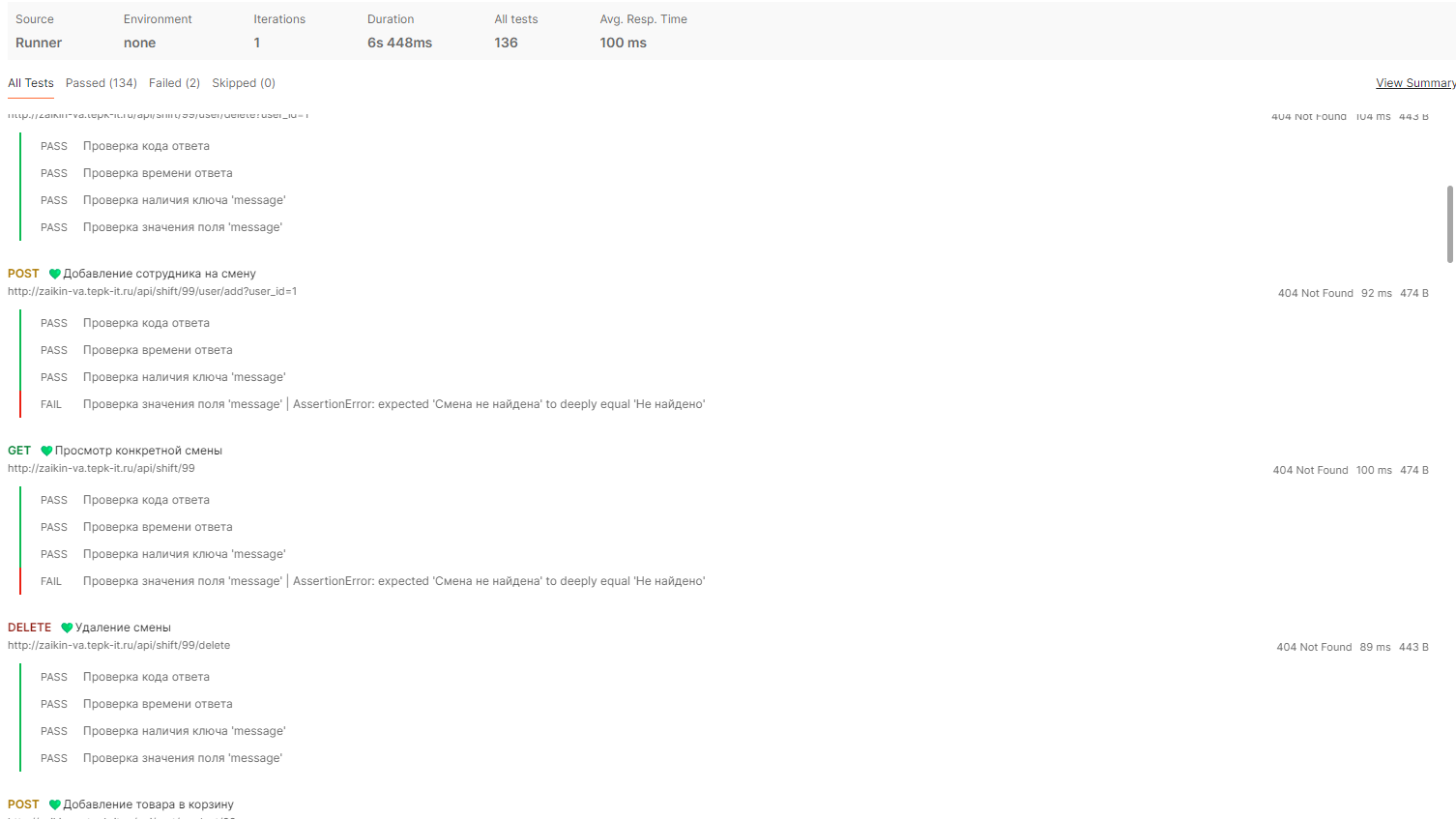
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| DELETE | {{host}}/user/{id}/delete |
| GET | {{host}}/user/{id} |
| DELETE | {{host}}/role/{id}/delete |
| DELETE | {{host}}/shift/{id}/user/delete |
| POST | {{host}}/shift/{id}/user/add |
| GET | {{host}}/shift/{id} |
| DELETE | {{host}}/shift/{id}/delete |
| POST | {{host}}/cart/product/{id} |
| DELETE | {{host}}/cart/product/{id} |
| GET | {{host}}/user/{id}/cart |
| GET | {{host}}/news/{id} |
| POST | {{host}}/news/{id}/update |
| DELETE | {{host}}/news/{id}/delete |
| GET | {{host}}/product/{id}/reviews |
| POST | {{host}}/review/{id}/update |
| DELETE | {{host}}/review/{id}/delete |
| POST | {{host}}/payment/{id}/update |
| DELETE | {{host}}/payment/{id}/delete |
| POST | {{host}}/status\_order/{id}/update |
| DELETE | {{host}}/status\_order/{id}/delete |
| POST | {{host}}/category/{id}/update |
| DELETE | {{host}}/category/{id}/delete |
| GET | {{host}}/product/{id} |
| DELETE | {{host}}/product/{id}/delete |
| POST | {{host}}/product/{id}/update |
| POST | {{host}}/checkout |
| POST | {{host}}/order/{id}/update |
| GET | {{host}}/order/{id} |
| DELETE | {{host}}/order/{id}/delete |
| GET | {{host}}/compounds/order/{id} |
| POST | {{host}}/order/compound/{id}/update |
| DELETE | {{host}}/order/compound/{id}/delete |
| GET | {{host}}/orders/product/{id} |
| POST | {{host}}/orders/product/{id}/between |
| POST | {{host}}/orders/category/{id]/between |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(404);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  });  pm.**test**("Проверка значения поля 'message'", **function** () {      pm.expect(pm.response.json().message).to.eql("Не найдено");  }); |

**Результаты тестирования:**

Некоторые пути не смогли пройти проверку из-за того, что значение в поле message у них отличалось.



**Название теста:** Проверка ошибки валидации данных, статус код 422

**Описание:** Этот тест проверяет ошибку валидации данных. Он отправляет запрос на русурс с неверными полями данных, ожидая получить ответ с кодом состояния 422 и наличием в ответе поля "message", а также проверяет время на ответ в 500мс.

**Ресурсы:**

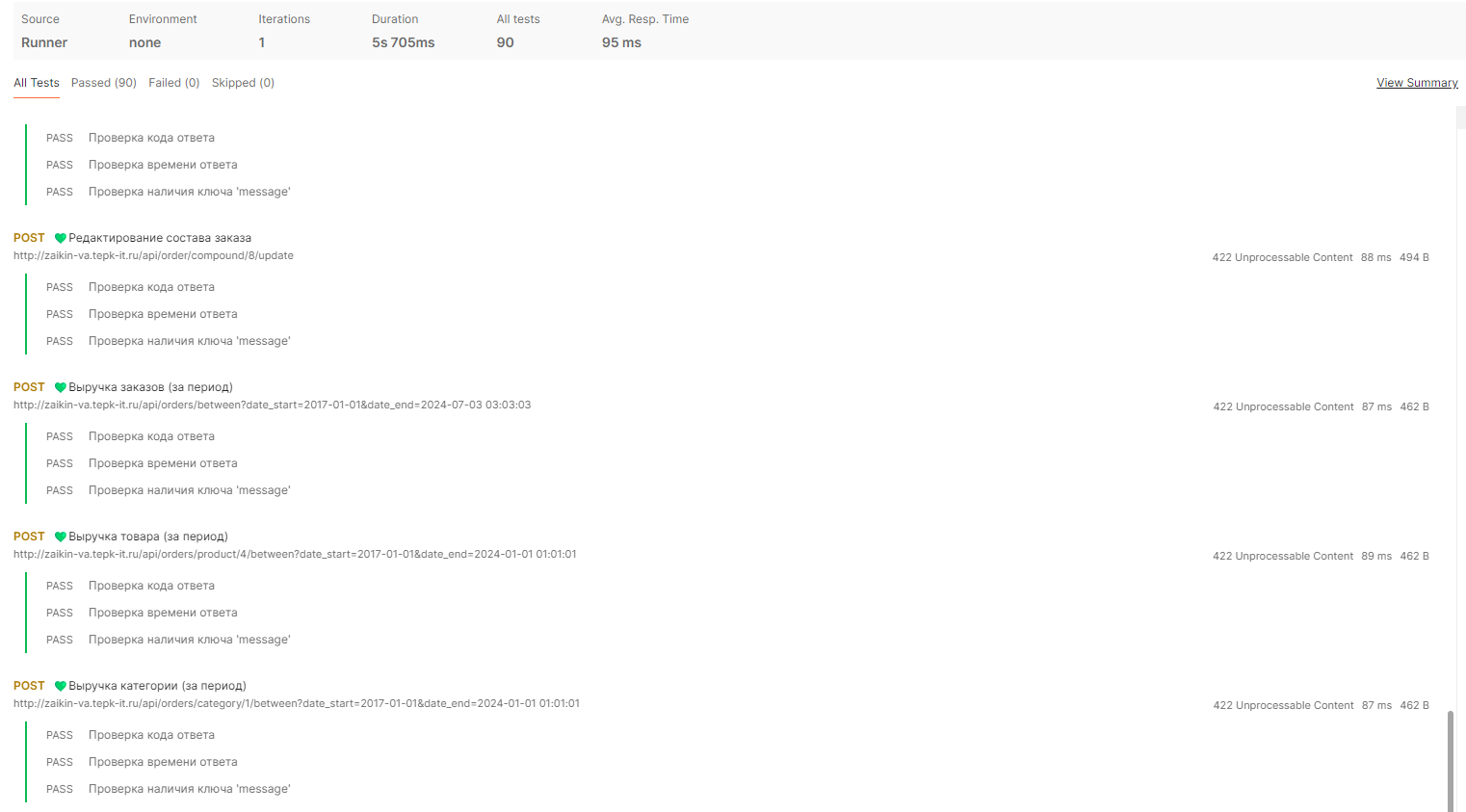
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| POST | {{host}}/login |
| POST | {{host}}/register |
| POST | {{host}}/user/create |
| POST | {{host}}/user/{id}/update |
| POST | {{host}}/profile/update |
| POST | {{host}}/role/create |
| POST | {{host}}/role/{id}/update |
| POST | {{host}}/shift/{id}/user/add |
| POST | {{host}}/shift/create |
| POST | {{host}}/shift/{id}/update |
| POST | {{host}}/news/{id}/update |
| POST | {{host}}/news/create |
| POST | {{host}}/product/{id}/review/create |
| POST | {{host}}/review/{id}/update |
| POST | {{host}}/payment/{id}/update |
| POST | {{host}}/payment/create |
| POST | {{host}}/status\_order/create |
| POST | {{host}}/status\_order/{id}/update |
| POST | {{host}}/category/create |
| POST | {{host}}/category/{id}/update |
| POST | {{host}}/checkout |
| POST | {{host}}/order/{id}/update |
| POST | {{host}}/order/{id}/employee/attach |
| POST | {{host}}/order/{id}/compound/create |
| POST | {{host}}/order/compound/{id}/update |
| POST | {{host}}/orders/between |
| POST | {{host}}/orders/product/{id}/between |
| POST | {{host}}/orders/category/{id}/between |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(422);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  }); |

**Результаты тестирования:**

Все пути успешны прошли проверку



**Название теста:** Проверка успешного создания, статус код 201

**Описание:** Этот тест проверяет успешное создание при выполнении запроса. Он отправляет запрос на русурс с верными данными, ожидая получить ответ с кодом состояния 201 и наличием в ответе поля "message", а также проверяет время на ответ в 500мс.

**Ресурсы:**

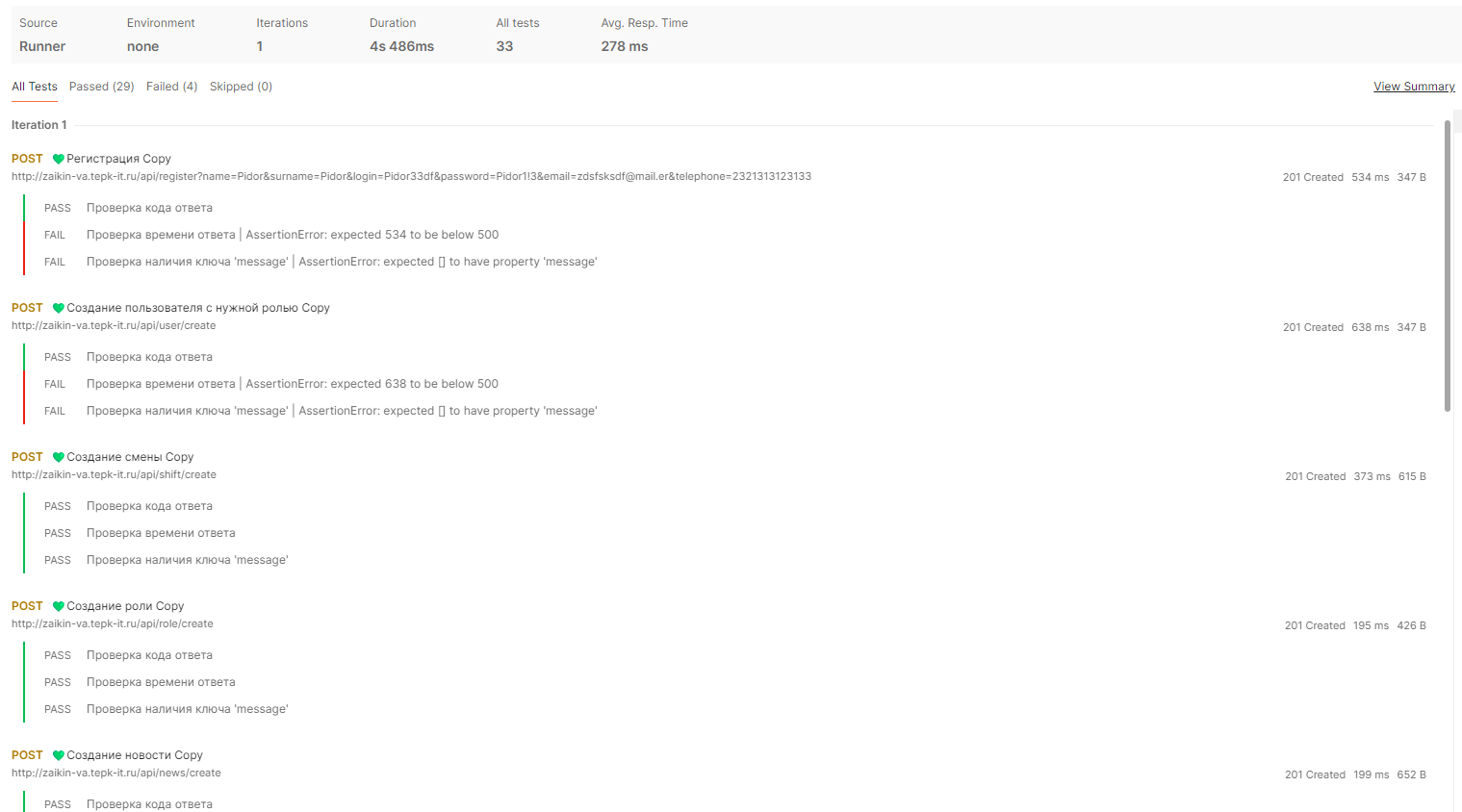
|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| POST | {{host}}/register |
| POST | {{host}}/user/create |
| POST | {{host}}/shift/create |
| POST | {{host}}/role/create |
| POST | {{host}}/news/create |
| POST | {{host}}/product/{id}/review/create |
| POST | {{host}}/payment/create |
| POST | {{host}}/status\_order/create |
| POST | {{host}}/category/create |
| POST | {{host}}/product/create |
| POST | {{host}}/order/{id}/compound/create |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(201);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  }); |

**Результаты тестирования:**

Не все пути прошли проверку, некоторые запросы выполнялись более 500мс, некоторые запросы не имели в себе поля "message".



**Название теста:** Проверка успешного выполнения, статус код 200

**Описание:** Этот тест проверяет успешное выполнение запроса. Он отправляет запрос на русурс с верными данными, ожидая получить ответ с кодом состояния 200 и наличием в ответе поля "message", а также проверяет время на ответ в 500мс.

**Ресурсы:**

|  |  |
| --- | --- |
| ***Метод*** | ***Путь*** |
| POST | {{host}}/login |
| POST | {{host}}/categories |
| GET | {{host}}/category/{id} |
| GET | {{host}}/product/{id} |
| GET | {{host}}/products |
| GET | {{host}}/product/{id}/review |
| GET | {{host}}/reviews |
| GET | {{host}}/news |
| GET | {{host}}/news/{id} |
| GET | {{host}}/payments |
| GET | {{host}}/status\_orders |
| GET | {{host}}/profile |
| POST | {{host}}/profile/update |
| POST | {{host}}/cart/product/{id} |
| GET | {{host}}/cart |
| POST | {{host}}/cart/update |
| DELETE | {{host}}/cart/product/{id} |
| GET | {{host}}/orders |
| POST | {{host}}/product/{id}/review/create |
| GET | {{host}}/users |
| GET | {{host}}/user/{id} |
| POST | {{host}}/shift/{id}/update |
| DELETE | {{host}}/shift/{id}/delete |
| DELETE | {{host}}/shift/{id}/user/delete |
| POST | {{host}}/shift/{id}/user/add |
| GET | {{host}}/shifts |
| GET | {{host}}/shift/{id} |
| POST | {{host}}/order/{id}/employee/attach |
| POST | {{host}}/order/{id}/employee/detach |
| POST | {{host}}/news/{id}/update |
| DELETE | {{host}}/news/{id}/delete |
| POST | {{host}}/product/{id}/update |
| DELETE | {{host}}/product/{id}/delete |
| GET | {{host}}/roles |
| POST | {{host}}/role/{id}/update |
| DELETE | {{host}}/role/{id}/delete |
| POST | {{host}}/user/{id}/update |
| DELETE | {{host}}/user/{id}/delete |
| POST | {{host}}/review/{id}/update |
| DELETE | {{host}}/review/{id}/delete |
| POST | {{host}}/category/{id}/update |
| DELETE | {{host}}/category/{id}/delete |
| POST | {{host}}/status\_order/{id}/update |
| DELETE | {{host}}/status\_order/{id}/delete |
| POST | {{host}}/payment/{id}/update |
| DELETE | {{host}}/payment/{id}/delete |
| GET | {{host}}/order/{id} |
| POST | {{host}}/order/{id}/update |
| DELETE | {{host}}/order/{id}/delete |
| GET | {{host}}/compounds/order/{id} |
| GET | {{host}}/order/compound/{id} |
| POST | {{host}}/order/compound/{id}/update |
| DELETE | {{host}}/order/compound/{id}/delete |
| GET | {{host}}/orders/product/{id} |
| POST | {{host}}/orders/between |
| POST | {{host}}/orders/product/{id}/between |
| POST | {{host}}/orders/category/{id}/between |

**Код теста:**

|  |
| --- |
| pm.**test**("Проверка кода ответа", **function** () {      pm.response.to.have.status(200);  });  pm.**test**("Проверка времени ответа", **function** () {      pm.expect(pm.response.responseTime).to.be.below(500);  });  pm.**test**("Проверка наличия ключа 'message'", **function** () {      pm.expect(pm.response.json()).to.have.property("message");  }); |

**Результаты тестирования:**

Не все пути смогли пройти проверку, некоторые запросы занималии более 500мс на ответ, некоторые не имели в себе поле "message".

