

# Описание прототипа

## Документация по системе обработки заказов

### Введение

Этот документ представляет реализацию Системы обработки заказов, описывая модели, репозитории, контроллеры и операции CRUD для заказов. Кроме того, он упоминает создание моделей, совместимых с требованиями задачи.

### Модели

Система обработки заказов включает следующие модели:

**Order:** Представляет собой заказ, размещенный пользователем. Включает основные данные, такие как идентификатор пользователя, общая стоимость, адрес доставки и статус заказа. Кроме того, он связан с продуктами через отношение многие-ко-многим.

**User:** Представляет собой пользователя системы. Каждый заказ связан с пользователем.

**Product:** Представляет собой продукт, доступный для покупки. Заказы могут включать один или несколько продуктов.

**Group:** Представляет собой группу, к которой могут принадлежать категории. Эта модель обеспечивает категоризацию и организацию продуктов.

**Caregory:** Представляет собой категорию, к которой могут принадлежать подкатегории. Она обеспечивает организацию и группировку продуктов.

**Subcategory:** Представляет собой подкатеорию, дополнительно уточняющую классификацию продуктов в рамках категорий.

### Репозитории

Репозитории предоставляют абстрактный уровень между логикой приложения и базой данных. Они инкапсулируют логику взаимодействия с моделями и предоставляют последовательный интерфейс для доступа к данным. Система обработки заказов включает следующие репозитории:

Интерфейс репозитория заказов: Определяет интерфейс для репозитория заказов.

Репозиторий заказов: Реализует логику CRUD-операций над заказами. Он взаимодействует с моделью Заказа для выполнения операций с базой данных.

## Контроллер

Контроллер заказов обрабатывает HTTP-запросы, связанные с заказами, служа как точка входа для операций, связанных с заказами. Он делегирует бизнес-логику репозиториям и включает методы для CRUD-операций, а также вспомогательные методы для конкретных функциональностей, таких как получение статуса заказа.

## Тестирование с помощью Postman

Чтобы протестировать Систему обработки заказов с помощью Postman:

Убедитесь, что система работает локально или на сервере, доступном через HTTP-запросы.

Используйте Postman для отправки HTTP-запросов к соответствующим конечным точкам контроллера заказов (/api/orders).

Протестируйте каждую CRUD-операцию (Создание, Чтение, Обновление, Удаление), отправляя запросы к соответствующим конечным точкам с соответствующими телами запросов и параметрами.

Вот пример даты, которая будет использоваться для создания Order, Пожалуйста, обратите внимание на **Примечание о создании пользователей и тестировании**, которая написана внизу.

```
{
  "user_id": 1,
  "id": 1,
  "shipping_address": "Moscow, Russia",
  "order_status": "pending",
  "total_price": 10.00,
  "products": [
    {
      "product_id": 1,
      "quantity": 2,
      "attributes": [
        {
          "name": "Color",
          "value": "Red"
        }
      ],
    }
  ]
}
```

```
        "name": "Size",  
        "value": "Large"  
    }  
]  
}  
]  
}
```

### **Примечание о создании пользователей и тестировании:**

Перед тем как тестировать операции с заказами, пользователей необходимо создать вручную с помощью инструмента, такого как RHPMyAdmin. Кроме того, необходимо вручную создать группы, категории, подкатегории и продукты заранее. Это связано с тем, что системе требуются существующие идентификаторы пользователей и информация о продуктах для точного ассоциирования заказов. Как только пользователи и детали продуктов будут созданы, их соответствующие идентификаторы могут быть использованы в запросах для создания заказов.

### **Заключение**

С реализацией Системы обработки заказов, включая модели, репозитории, контроллеры и CRUD-операции для заказов, система предоставляет надежную платформу для эффективного управления заказами.