Оглавление

[Общие 2](#_Toc177055455)

[*Запросить ассортимент товаров по определенному типу* 2](#_Toc177055456)

[Начнем проектирование с интерфейса для пользователя. 2](#_Toc177055457)

[Просмотр своей корзины: 2](#_Toc177055458)

[Добавление экземпляра товара в корзину, если его до этого не было и корзина пуста: 3](#_Toc177055459)

[Добавление экземпляра товара в корзину, если он первый и если в корзине уже были товары, а также изменение количества товаров (не на 0): 3](#_Toc177055460)

[Удаление товара из корзины 4](#_Toc177055461)

[Просмотр своих желаемых товаров: 4](#_Toc177055462)

[Добавление и удаление товара в желаемое: 4](#_Toc177055463)

[Оплата корзины: 4](#_Toc177055464)

[Просмотр своих заказов: 4](#_Toc177055465)

[Продолжим проектированием интерфейса для администратора 5](#_Toc177055466)

[Редактирование количества отображаемого товара: 5](#_Toc177055467)

[Удаление товара: 5](#_Toc177055468)

[Добавления товара: 5](#_Toc177055469)

[Выгрузка данных о заказах (в том числе не оплаченных): 6](#_Toc177055470)

[Выгрузка данных о товарах, добавленных в желаемое: 6](#_Toc177055471)

## Общие

#### Запросить ассортимент товаров по определенному типу:

Метод GET

URL: …/api/types/{type\_id}

Параметры запроса:

type\_id (string): Идентификатор типа товара, если равен «All», то мы просим вернуть типы товара, чтобы потом выбрать. Если не равен «All», то просим вернуть список этого товара с базы, в соответствии с переданным идентификатором

Ответы:

200 Ок Успешно переданы типы или список товаров:

Для id\_type == All:

[

{

type\_name: Тип товара,

count\_prod: Кол-во разных товаров

avg\_value\_price: Средняя стоимость товара в группе,

all\_value\_count: Общее количество товара в группе

}

{

…

}

]

Для id\_type == «type»:

[

{

id\_product: Индивидуальный номер для каждого товара

wish\_count: Кол-во людей, которые добавили в желаемое этот товар

type: Тип товара

name: Название товара,

country: Страна производитель,

price: Цена за штуку товара,  
count: Количество товара на складе магазина,

Дополнительная, специфичная для каждого товара информация

}

{

…

}

]

400 Если передан несуществующий тип товара

## Начнем проектирование с интерфейса для пользователя.

### Просмотр своей корзины:

Метод GET

URL: …/api/carts/{id\_customers}

Параметры запроса:

id\_customer (string): Передаём свой id, по которому смотрим что в корзине

Ответ:

200 OK Успешно передаем корзину с товарами от пользователя

[

{

\_id\_orders: Идентификатор заказа

goods: Расписываем все товары в корзине

{

count: Цена

\_id\_one\_goods: Идентификатор товара

}

{

…

}

}

{

\_id\_products: Идентификатор товара: {

type: Тип товара

count\_wish\_customers: Кол-во добавлений в желаемое

name: Имя товара

price: Цена

count: Оставшееся количество на складе

}

\_id\_products: {

…

}

}

]

400 Error Если ошибка в запросе

404 Not Found Если корзины не существует, так и писать: «Корзина пуста»

### Добавление экземпляра товара в корзину, если его до этого не было и корзина пуста:

Метод POST

URL: …/api/carts/{id\_customers}

Параметры запроса:

id\_customer (string): Передаём свой id

id\_product (string): Передаём id номер каждого товара

count (int): Количество товара, которое добавляем в корзину

Ответ: 200 ОК При успешном добавление

400 Если корзина уже создана

### Добавление экземпляра товара в корзину, если он первый и если в корзине уже были товары, а также изменение количества товаров (не на 0):

Метод PATCH

URL: …/api/carts/{id\_customers}

Параметры запроса:

id\_customer (string): Передаём свой id

id\_product (string): Передаём индивидуальный номер каждого товара

count (int): Количество товара, которое добавляем в корзину

Ответ: 200 ОК При успешном добавление

400 При всех ошибках (Передано не число или ноль, попытка не изменения содержимого корзины, а её создание)

### Удаление товара из корзины

Метод DELETE

URL …/api/carts/{id\_customers}

Параметры запроса:

id\_customer (string): Передаём свой id

id\_product (string): Индивидуальный номер каждого товара

Ответ:

200 ОК При успешном удалении из корзины

410 Gone Если объекта в корзине нет

### Просмотр своих желаемых товаров:

Метод GET

URL: …/api/wish/{id\_customers}

Параметры запроса:

id\_customers (string) (Обязательный): Передаём свой id

Ответы:

200 Ок Возвращается словарик с id понравившихся товаров

[

66afd0daf1dc34a2ba077480

66bc78543d02a9a2055dbb74

id\_product

]

400 Нет товаров в желаемом

### Добавление и удаление товара в желаемое:

Метод PATCH

URL: …/api/wish/{id\_customers}

Параметры запроса:

id\_customer (string): Передаём свой id

append (bool): Добавляем или удаляем

id\_product (string): id товара

Ответ:

200 ОК При успешном добавление/удаление в желаемое нового товара

400 Bad Request Если этот товар/покупателя уже есть/отсутствует в желаемом

### Оплата корзины:

Метод PATCH

URL: …/api/orders/{id\_customers}

Параметры запроса:

id\_customers (string): Передаём id пользователя, который взаимодействует с оплатой

paid (bool): Передаём статус, удалось ли оплатить

Ответ:

200 ОК удалось оплатить

400 Не удалось оплатить

### Просмотр своих заказов:

Метод GET

URL: …/api/orders/{id\_customers}

Параметры запроса:

id\_customers (string): Передаём айди пользователя

Ответ:

200 ОК Смогли передать данные

[

{

id\_orders: 66bc7043f0c595a32768620d

goods: [

{

count: 6

id\_product: 66afd0daf1dc34a2ba077480

}

{

count:

id\_product

}

]

}

{

…

}

]

404 Если нет заказов

## Продолжим проектированием интерфейса для администратора

### Редактирование количества отображаемого товара:

Метод PATCH

URL: …/api/admin/goods

Параметры запроса:

id\_product (string): Индивидуальный номер каждого товара

count (int): Новое количество товара

Ответы:

200 ОК Изменения приняты

404 Not Found Не найден товар

422 Unprocessable Entity Передано не число в count

### Удаление товара:

Метод DELETE

URL: …/api/admin/goods

Параметры запроса:

id\_product (string): Индивидуальный номер каждого товара

Ответы:

200 ОК При успешном удалении объекта

410 Gone Если объекта в ассортименте нет

### Добавления товара:

Метод POST

URL: …/api/admin/goods

Параметры запроса всё в json:

type (string): Тип товара,

name (string): Название товара,

country (string): Страна производитель,

price (int): Цена за штуку товара,  
 count (int): Количество товара на складе магазина,

Дополнительная, специфичная для каждого товара информация

Ответы:

201 Created Создана новая карточка товара

400 Bad Request При некорректных данных в запросе

422 Unprocessable Entity Отсутсвует нужный атрибут

### Выгрузка данных о заказах (в том числе не оплаченных):

Метод GET

URL: …/api/admin/orders

Параметры запроса:

id\_order (string): Индивидуальный номер каждого оформленного заказа (или 0 если нужны все заказы)

paid (bool): Флаг, если True то только оплаченные, и если False то оплаченные и не оплаченные

Ответы:

200 ОК Выгрузили данные

Формат ответа:

[

id\_order: Характеристики каждого заказа

id\_customer: Номер пользователя, который оформил заказ,

date\_of\_order: Дата заказа (Пустое если не оплачено)

paid: True/False Оплачено ли,

delivered: True/False Доставлено/вручено ли,

id\_goods: {

count: Количество определённого товара

\_id\_one\_goods: Номер товара

}

]

400 Bad Request Если некорректны параметры запроса

404 Not Found Если такого заказа нет

### Выгрузка данных о товарах, добавленных в желаемое:

Метод GET

URL: …/api/admin/wish/{id\_product}

Параметры запроса:

id\_product (string): Индивидуальный номер каждого товара (или 0 если нужны все товары)

Ответы:

200 ОК Выгрузили данные

Формат ответа:

[

{

id\_product: Для каждого товара

id\_wish:

{

n: id\_customer

}

}

]

400 Bad Request Не существует товара