# Общие требования к фронтенду

1. Использовать **React или Angular**
2. Фронтенд должен быть **SPA** типа
3. Проект должен использовать АПИ для доступа к продуктам компании. **API URL:** [**http://api.idealtex.com.ua**](http://api.idealtex.com.ua)
4. Весь статичный текст должен быть локализирован и хранится в одном месте для легкого изменения. Локализация пока что только для Украинского языка.
5. В фронтенде используется верстка, которая идет вместе с ТЗ
6. Кешировать информацию из АПИ по минимуму. К примеру запретов на кеш, есть следующее: цену продуктов - нельзя, скидки - нельзя и т.д. Если делаете кеш, то или на сессию или на короткий промежуток времени.
7. В проекте нужно будет сохранять информацию о заказах пользователя на его стороне. Сохраняете **только** код продукта и кол-во каждой заказанной упаковки
8. Все ошибки от АПИ не должны крашить приложение или появлятся на странице. В случае ошибки показывать ошибку в попапе в нижнем правом углу. Также это касается ошибок при парсе ответов от АПИ или внутренней логики приложения. Ошибка это все что начинается на 4xx, 5xx
9. В случае если данные с АПИ уже не надо, но реквест к АПИ еще незакончился, то нужно отменить реквест
10. Все компоненты подразумевающие множество элементов (продукты, категории и т.д.) должны грузится инфинити лоадингом

# Пояснения

1. В проекте есть продукты и они делятся на обычные и аукционные. Отличие аукционных заключается в том, что в расчете итоговой цены за товар не участвует скидка, поскольку максимальная скидка уже была включена в товар. За подробностями **см. Страница: *Products*, Расчет: *Price* и см. Страница: *Cart*, Расчет: *Discount***

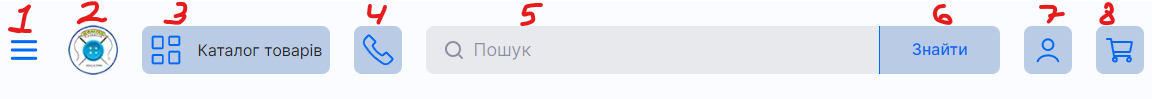
# Страница: Home

**URL: */***

## Компонент: Header

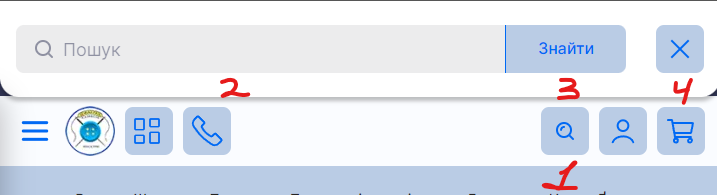
**Файл: *home.html***

### Версия: *PC*



1. Кнопка *открывающая* левый сайдбар. **см. Компонент: *Sidebar***
2. Лого компании в виде кнопки. *Открывающая* **Страница: *Home***
3. Кнопка *открывающая* каталог товаров. **см. Компонент: *Catalog***
4. Кнопка *открывающая* телефоны. **см. Компонент: *Phones***
5. Инпут для ввода слов для поиска.
6. Кнопка *открывающая* страницу поиска. **см. Страница: *Search***
7. Кнопка *открывающая* страницу с логинов, если пользователь не логинился. **см. Страница: *Login***. Если пользователь уже зашел в аккаунт, то *переходим* на страницу аккаунт. **см. Страница: *Account***
8. Кнопка *открывающая* страницу корзина. **см. Страница: *Cart***

### Версия: Tablet



1. Кнопка открывает попап сверху экрана.
2. Инпут для ввода слов для поиска.
3. Кнопка *открывающая* страницу поиска. **см. Страница: *Search***
4. Кнопка закрывающая попап.

### Версия: *Phone*

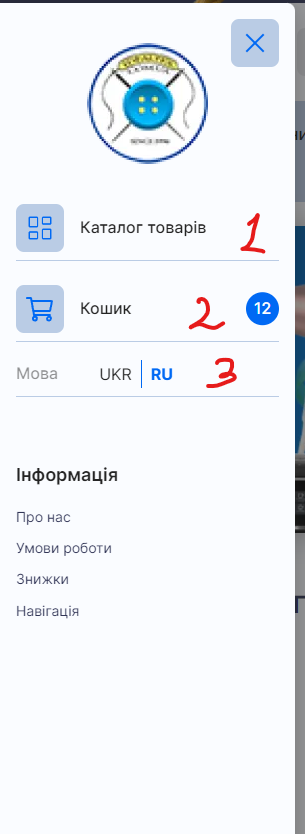
Тоже самое что Tablet версия



## Компонент: Sidebar

### Состояние: Пользователь не зашел в аккаунт

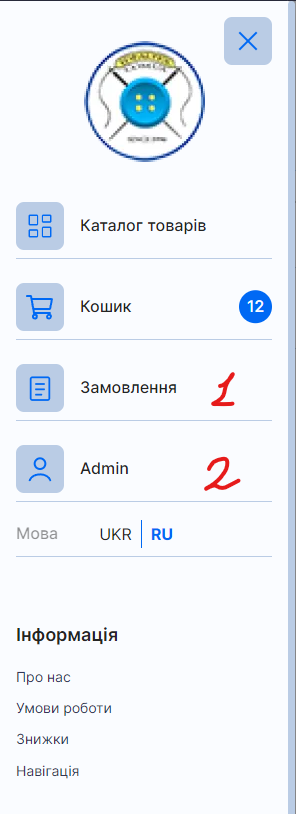
**Файл: *home.html***



1. Кнопка *открывающая* каталог товаров. **см. Компонент: *Catalog***
2. Кнопка *открывающая* страницу корзина. **см. Страница: *Cart***
3. Кнопка *меняющий* локализацию.

### Состояние: Пользователь зашел в аккаунт

**Файл:orders*.html***

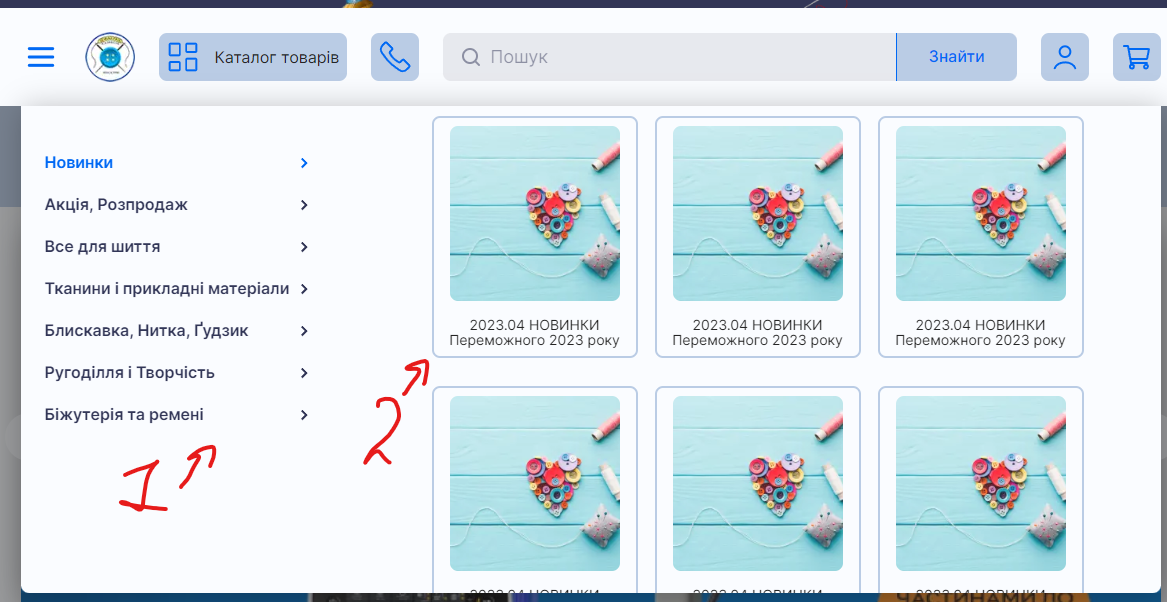


1. Кнопка *открывающая* страницу с заказами пользователя. **см. Страница: *Orders***
2. Кнопка для перехода в админку (http://admin.idealtex.com.ua). *Отображаем* кнопку, если **role = *administrator***из запроса *GET /v1/public/users/{userId}.* Для реквеста нужен токен

## Компонент: Catalog

**Файл: *home.html***

### Версия: PC

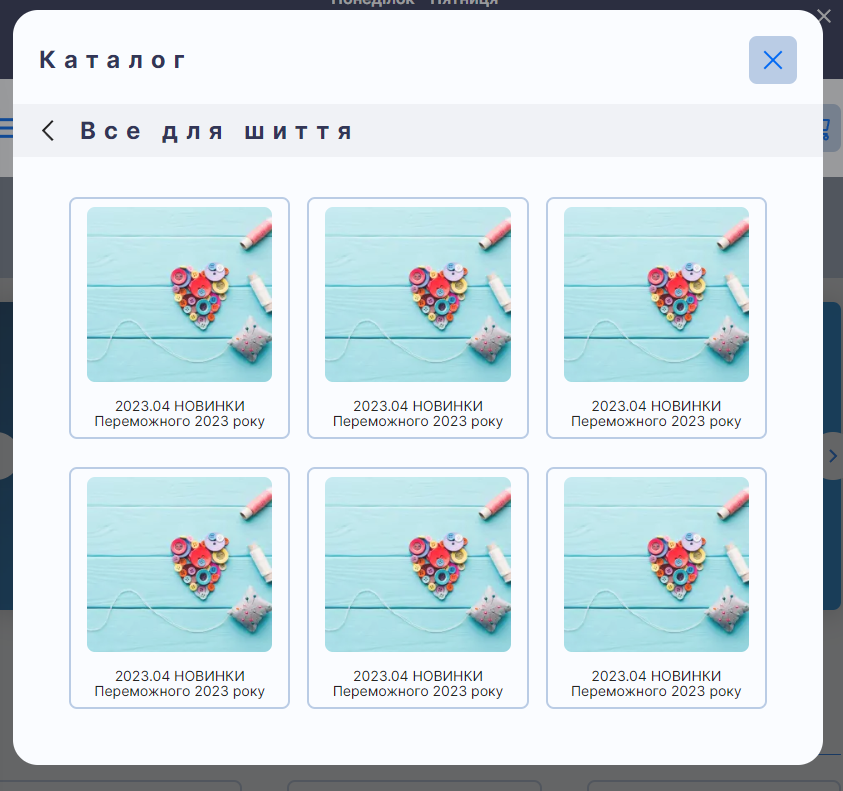


1. Список категорий. Берется из *GET /v1/public/categories*. При нажатии *меняется* пункт 2, если **hasChildren = *true.*** Если **hasChildren = *false,*** то *открывается* страница продуктов. **см. Страница: *Products***
2. Список подкатегорий. Берется из *GET /v1/public/categories/{alias}/children.* При нажатии *открывается* страница подкатегорий, если **hasChildren = *true***. **см. Страница: *Categories.*** Если **hasChildren = *false,*** то *открывается* страница продуктов. **см. Страница: *Products***

### Версия: Tablet, Phone

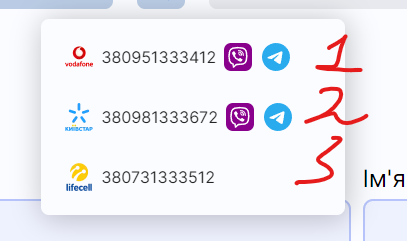


Категории. Берется из *GET /v1/public/categories*. При нажатии *открываются* подкатегории. **см. Ниже**, если **hasChildren = *true.*** Если **hasChildren = *false,*** то *открывается* страница продуктов. **см. Страница: *Products***



Подкатегории. Берется из *GET /v1/public/categories/{alias}/children*. При нажатии *открывается* страница категорий, если **hasChildren = *true***. **см. Страница: *Categories.*** Если **hasChildren = *false,*** то *открывается* страница продуктов. **см. Страница: *Products***

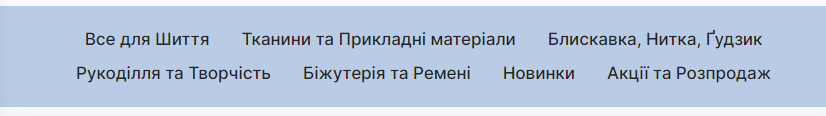
## Компонент: Phones



1. Vodafone телефон. Берется с *GET /v1/public/settings/companyPhoneVodafone*
2. Киевстар телефон. Берется с *GET /v1/public/settings/companyPhoneKyivstar*
3. lifecell телефон. Берется с *GET /v1/public/settings/companyPhoneLifecell*

## Компонент: HeaderCategories

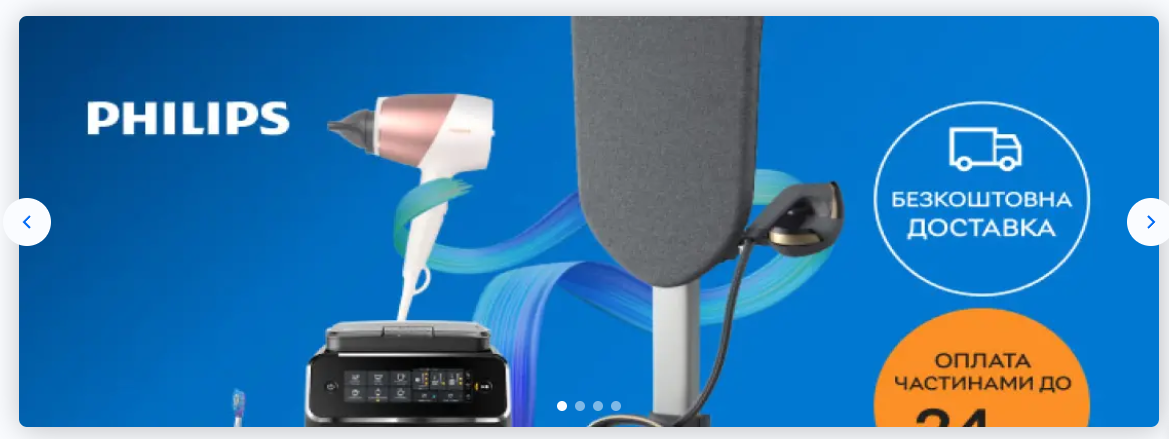
**Файл: *home.html***



Категории. Берется из *GET /v1/public/categories*. При нажатии *открываются* подкатегории, если **hasChildren = *true.* см. Страница: *Categories***. **см. Страница: *Categories.*** Если **hasChildren = *false,*** то *открывается* страница продуктов. **см. Страница: *Products***

## Компонент: FirstBanner

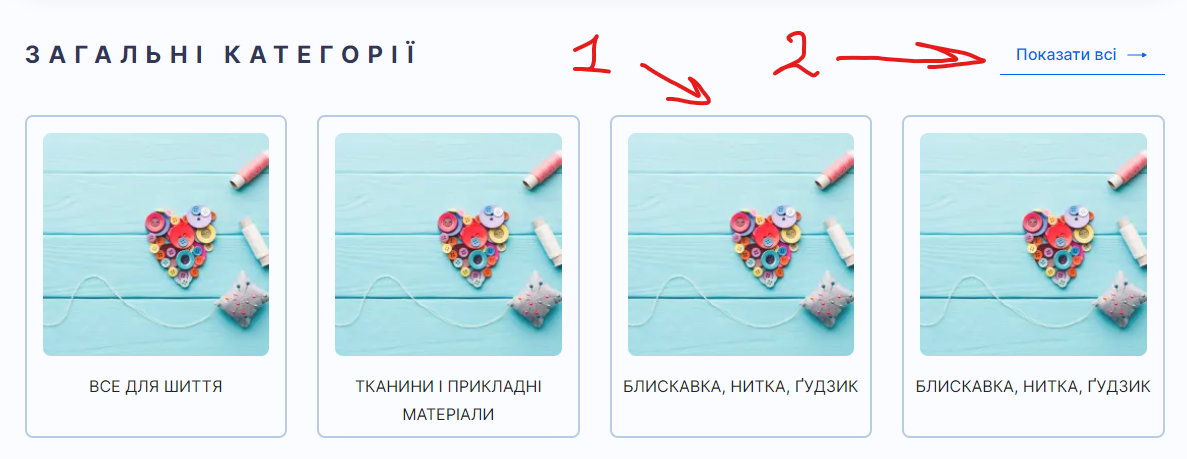
**Файл: *home.html***



Баннер с картинками. При нажатии на картинку *переход* на определенный урл. Берется с *GET /v1/public/settings/homeFirstBannersAd.*

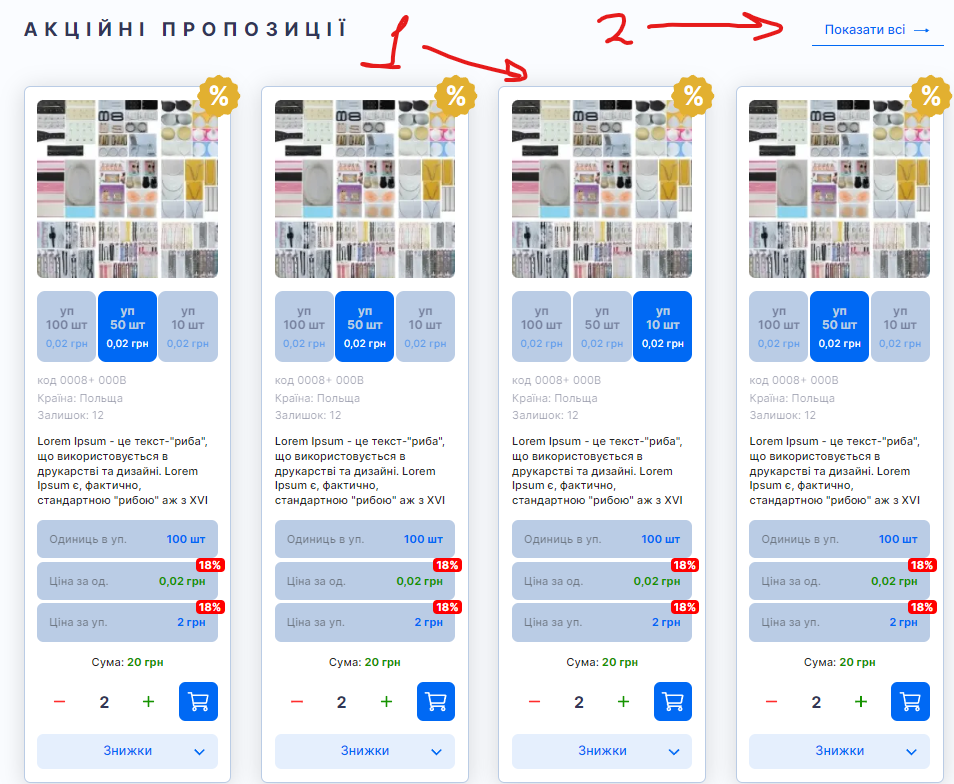
## Компонент: GeneralCategories

**Файл: *home.html***



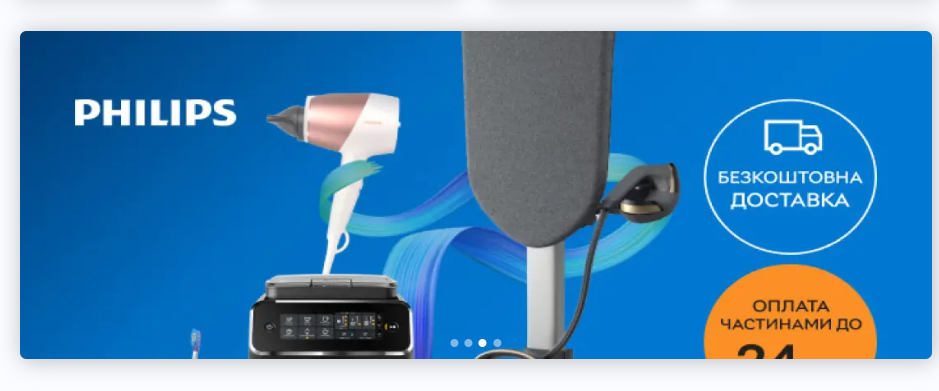
1. Категории. Берется из *GET /v1/public/categories.* При нажатии *открываются* подкатегории. **см. Страница: *Categories***
2. *Переход* на страницу категории. **см. Страница: *Categories***

## Компонент: AuctionProducts



1. Акционные продукты. *Берется* из *GET /v1/public/products.* Как работать с продуктом. **см. Страница: *Products*, Компонент: *Product***
2. *Переход* на страницу с аукционными товарами. **см. Страница: *Auction***

## Компонент: SecondBanner



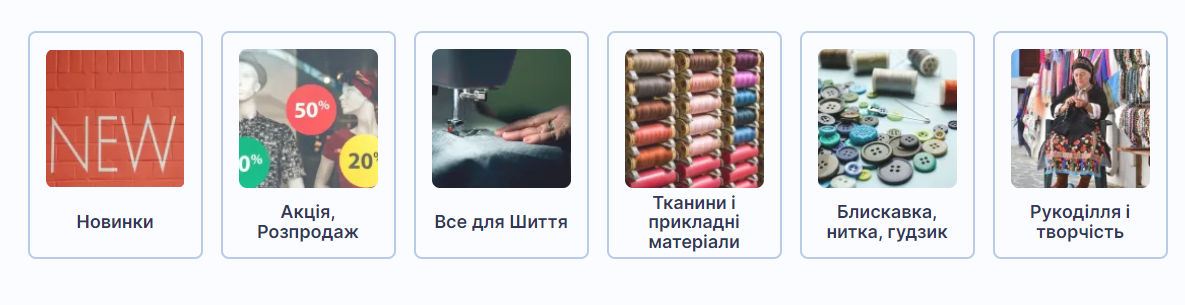
Баннер с картинками. При *нажатии* на картинку *переход* на определенный урл. Берется с *GET /v1/public/settings/homeSecondBannersAd.*

# Страница: Categories

**URL: */categories или /categories/{aliasName}***

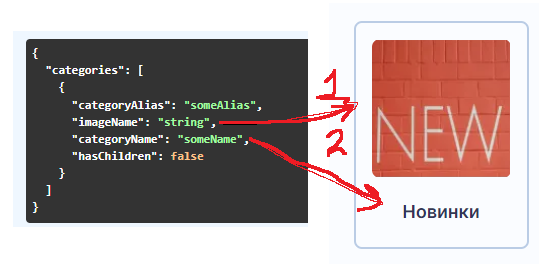
**Файл: *category.html***

## Компонент: CategoryList



Категории или подкатегории. Дозагрузка должна быть инфинити лоадингом. Берется с *GET /v1/public/categories* или *GET /v1/public/categories/{alias}/children* соответственно. При нажатии на категорию *переходим* на подкатегории, если **hasChildren = *true*.** Переход на страницу с продуктами если **hasChildren = *false*.**

## Компонент: Category



1. Картинка категории. Берем с ресурса *GET /v1/public/images/{imageName}*.
2. Название категории

# Страница: Products

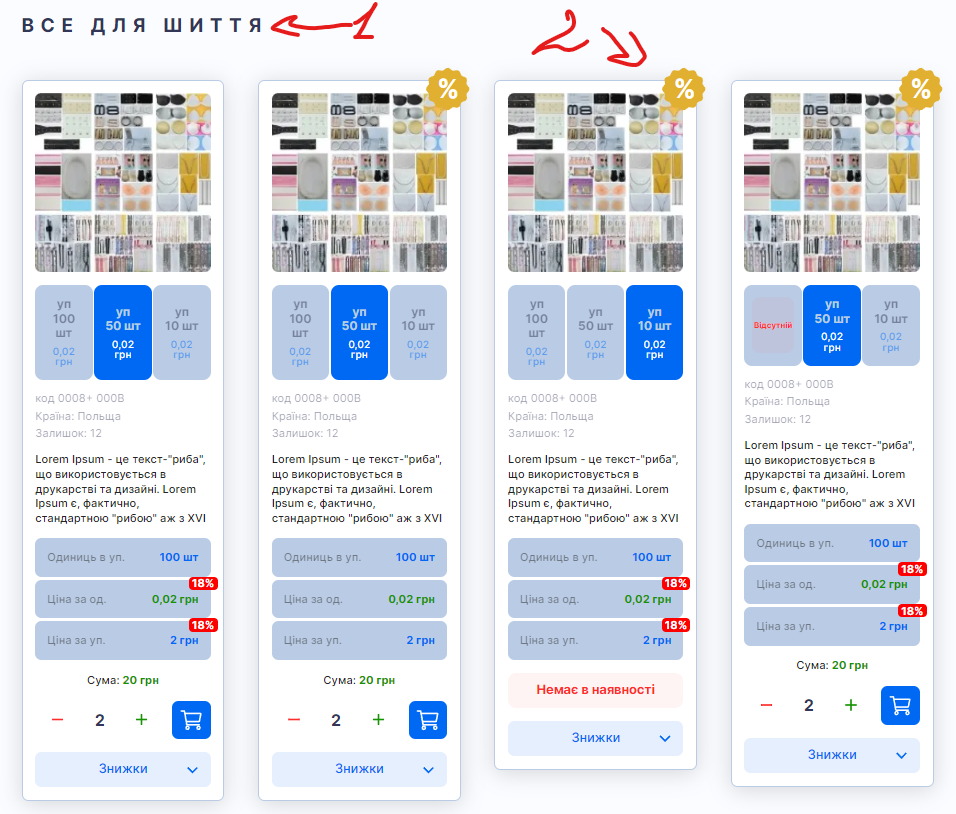
**URL: */categories/{aliasName}/products***

**Файл: *products.html***

## Расчет: Price

Эта страница где показываются цены. Поскольку пользователь может получать скидки во время набора продуктов в корзину, эти скидки должны *учитываться*. Как получить скидки с АПИ и когда пользователь получает их **см. Страница: *Cart*, Расчет: *Discounts.*** Если продукт не аукционный (productPackagesSizes.{package}.productAuction = false), то расчет идет по следующей формуле: *цена -* *цена \* (текущая скидка пользователя / 100)*. Если аукционный скидку не используем на продукте, то есть расчет идет по следующей формуле: *цена*

## Компонент: ProductList

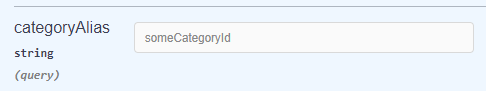


1. Название категории. Берется с *GET /v1/public/categories/{alias}.*
2. Продукты. Дозагрузка должна быть инфинити лоадингом. Берутся с *GET /v1/public/products*

## Компонент: Product

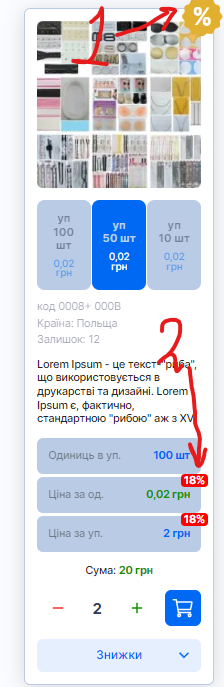
### Версия: Не акционный продукт



Данные для продукта берем с *GET /v1/public/products*. Категорию передаем в query параметр  


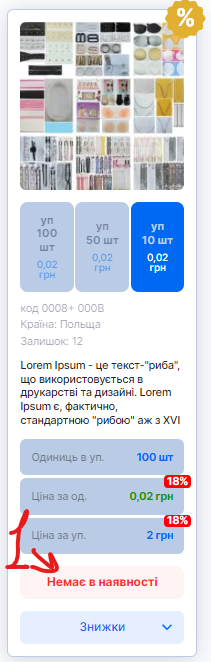
1. Картинка продукта. С АПИ будет массив, берем первую. *imageNames*  в АПИ. При нажатии *переходим* на страницу одного продукта. **см Страница: *Product***.
2. Разные упаковки. Слева на право big, mid, small. productPackagesSizes.big, productPackagesSizes.mid, productPackagesSizes.small в АПИ соответственно.   
   При нажатии *меняется* вся информация в продукте которая связана с упаковкой.   
   Надпись внутри *формируется* так: *уп + productPackagesSizes.{package}.productCountInPackage + productMetric* и *productPackagesSizes.{package}.productPrice + грн.*Как считать цену **см. Расчет: *Price***.
3. Код продукта. productCode в Апи.
4. Страна производитель. productCountry в АПИ.
5. Сколько осталось товара. productPackagesSizes*.{package}.displayPackageCount* в АПИ
6. Название продукта. productName в АПИ
7. Кол-во продукта в уп. productPackagesSizes*.{package}.productCountInPackage* в АПИ
8. Цена за единицу товара. *productPackagesSizes.{package}.productPrice* в АПИ. Как считать цену **см. Расчет: *Price***.
9. Цена за уп. товара. *productPackagesSizes.{package}.productPackagePrice* в АПИ. Как считать цену **см. Расчет: *Price***.
10. Скидка которая применяется к товару, если она есть. Больше инфы **см. Страница: *Cart*, Расчет: *Discount.***
11. Сумма выбранного товара. Считается как *productPackagesSizes.{package}.productPackagePrice \* (кол-во заказанных товаров)*. Как считать цену **см. Расчет: *Price.***
12. Кол-во заказанного товара.
13. Добавить в корзину
14. Скидки.   
    Первая строка это цены без скидок. Слева на право: Кол-во продукта в уп. *productPackagesSizes.{package}.productCountInPackage + productMetric, Цена за единицу productPackagesSizes.{package}.productPrice.* Цена за уп. *productPackagesSizes.{package}.productPackagePrice.*Все остальные строки это цены единицы и уп. товара умноженные на скидку. Где достать скидки **см. Страница: *Cart*, Расчет: *Discount.***

### Версия: Акционный продукт



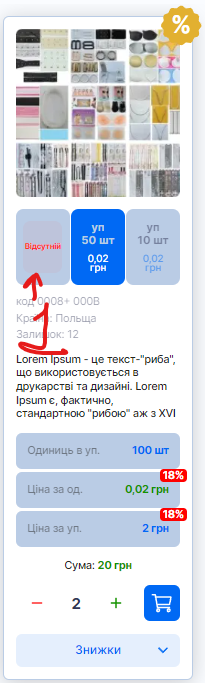
1. *Добавляется* если продукт акционный. *productPackagesSizes.{package}.productAuction* в АПИ
2. Скидка ставится максимальная, но она *не применяется* в цене. По поводу цен **см. Расчет: *Price***. Где достать скидки **см. Страница: *Cart*, Расчет: *Discount***

### Версия: Нет в наличии



1. Показывать нет в наличии. Показывать когда *productPackagesSizes.{package}.displayPackageCount = “0”* в АПИ

### Версия: Нет упаковки



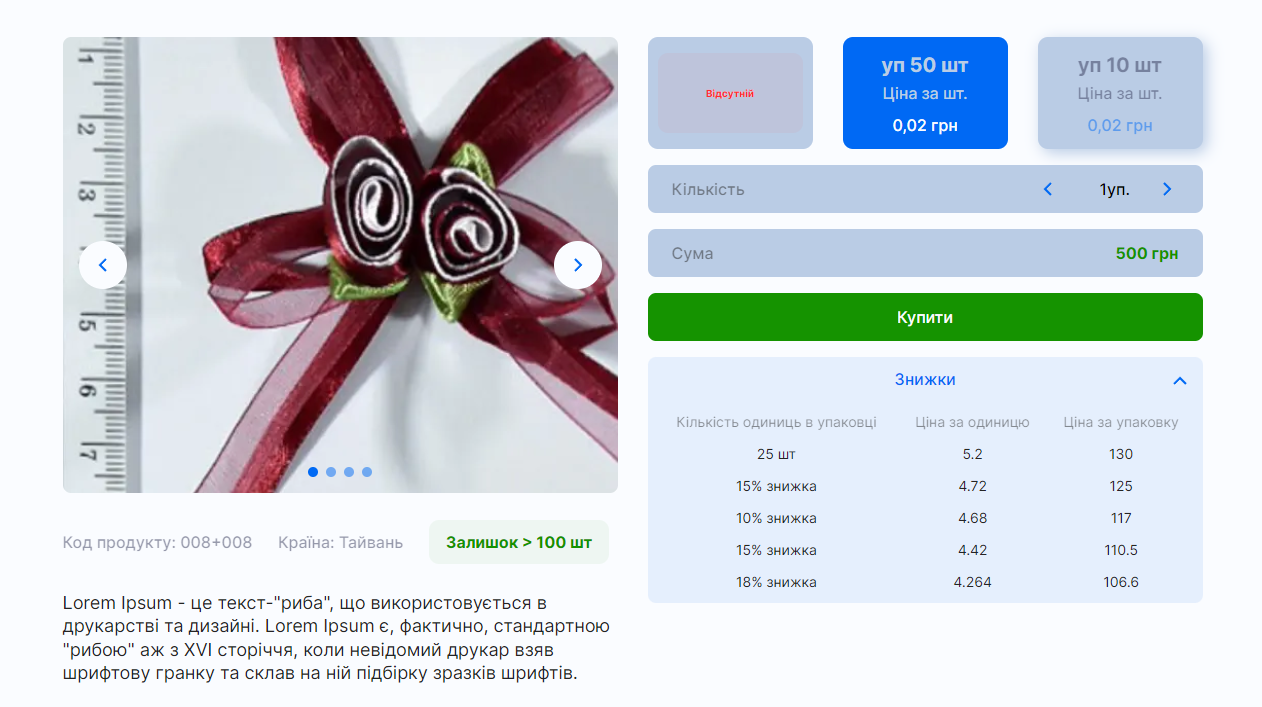
1. Упаковка отсутствует. Если *productPackagesSizes.{package} = null* в АПИ

# Страница: Product

**URL: */product/{productCode}***

**Файл: *product2.html или product.html***

## Компонент: Product



Это тот-же продукт что и в **Страница: *Products*, Компонент: *Product.*** Единственное *отличие* тут - это карусель со всеми картинками. Инфу о продукте берем с GET /v1/public/products/{productCode}

# Страница: Cart

**URL: */cart***

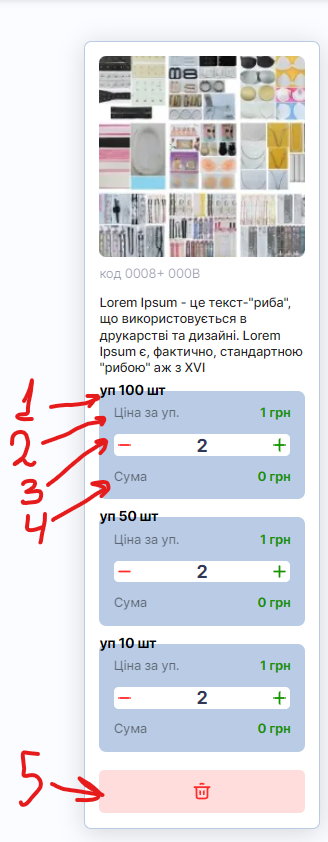
**Файл: *basket.html***

## Расчет: Discount

Скидки берутся из *GET /v1/public/settings/discounts*. У покупателя может быть только одна скидка, какая у него сейчас скидка *определяется* по полю *price* из АПИ. Если покупатель набрал достаточно сумму продуктов в корзине, то применяем соответствующую скидку на все цены, если только в описании не написано что скидка в цене не участвует, к примеру акционные товары. Как рассчитать цену со скидкой для продукта **см. Страница: *Products*, Расчет: *Price***

## Компонент: CartProduct

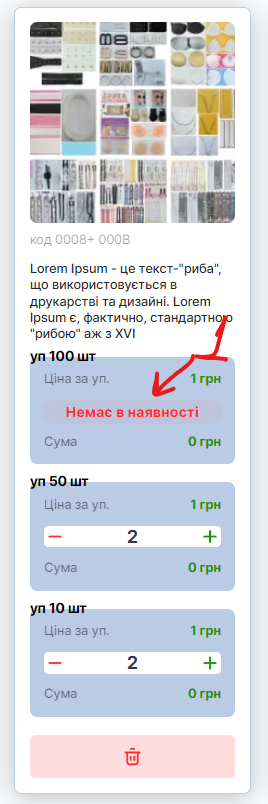
### Версия: Стандарт



Как сказано в требованиях, на стороне клиента сохраняется только код продукта и сколько он заказал каждой уп. продукта. Так что нужно получить инфу о продукте для расчетов цен и отображения продукта. Инфу берем с *GET /v1/public/products* используя параметр productsCodes. Максимальное кол-во продуктов которое может вернуть АПИ в одном запросе = 50, поэтому нужно будет сделать больше одного реквеста, если продуктов > 50.

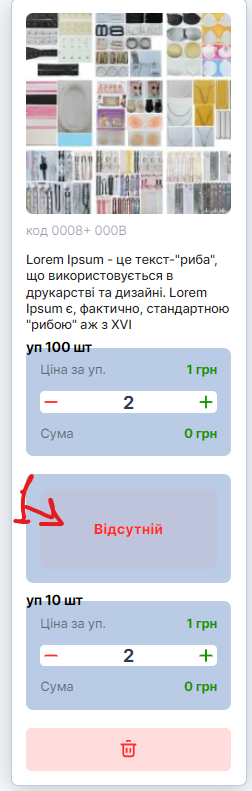
1. Кол-во продуктов в уп. *уп + productPackagesSizes.{package}.productCountInPackage + productMetric.*
2. Цена за уп. *productPackagesSizes.{package}.productPackagePrice* в АПИ
3. Кол-во заказанного продукта.
4. Сумма заказанного продукта. *productPackagesSizes.{package}.productPackagePrice \* (кол-во заказанного продукта)*
5. Удалить продукт с корзины. Также продукт удаляется если кол-во всех заказанных упаковок во всех типах упаковок = 0.

### Версия: Нет в наличии



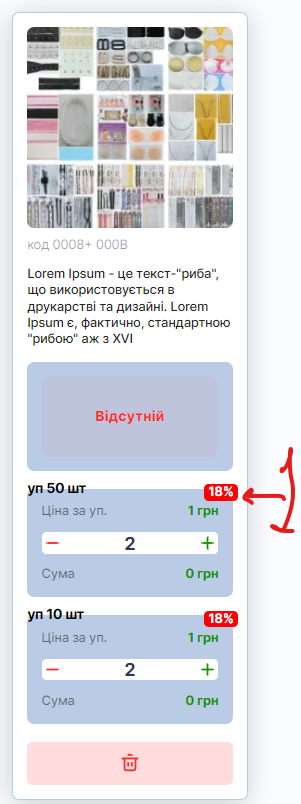
1. Показывать когда *productPackagesSizes.{package}.displayPackageCount = “0”* в АПИ

### Версия: Отсутствует упаковка



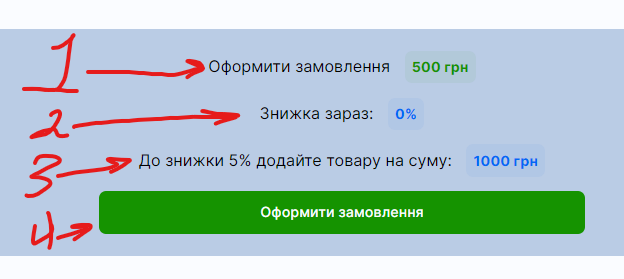
1. Упаковка отсутствует. Если *productPackagesSizes.{package} = null* в АПИ

### Версия: Когда есть скидка или это аукционный продукт



1. Если пользователь получил скидку достигнув нужной цены то применяем скидку и ставим плашку со скидкой. Если продукт аукционный *productPackagesSizes.{package}.productAuction* в АПИ, то ставим максимальную скидку, но не применяем скидку на цену.

## Компонент: CartInfo



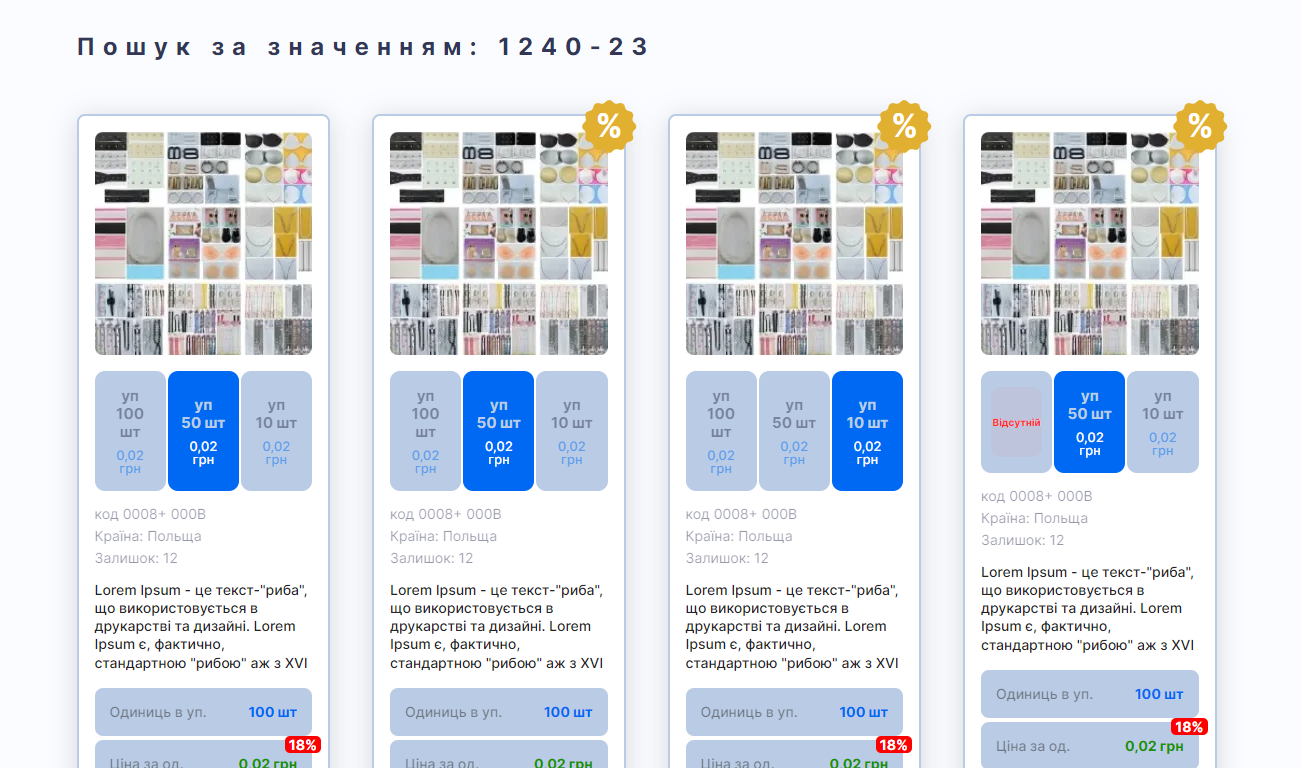
1. Сумма всех продуктов.
2. Действующая скидка
3. Сколько нужно набрать для следующей скидки
4. Кнопка для *перехода* на страницу оформления заказа. **см. Страница: *Order***

# Страница: Search

**URL: */search/{searchWords}***

**Файл: *search.html***

## Компонент: Products



Продукты с поиска. Инфу берем с *GET /v1/public/products*  c параметром *search.*

Как работать с продуктом **см. Страница: *Products*, Компонент: *Product***

# Страница: Login

**URL: */login***

**Файл: *login.html***

## Пояснение: JwtToken

Для некоторых ресурсов в АПИ нужна авторизация. После логина у пользователя будет его jwt токен с него берется инфа о айди пользователя (uid клейм), айди нужно будет для некоторых запросов в АПИ, также нужен сам токен.

## Пояснение: JwtRefreshToken

Токен для рефреша токена доступа. Его можно получить при логине. Если срок годности токена доступа истек, то идем с рефреш токеном на *POST /v1/public/auth/login#refreshToken* и получаем новый токен.

## Компонент: LoginForm



Форма для входа пользователя. Используем *POST /v1/public/auth/login#basic*

1. Почта. Не должно быть пустым. *email* в АПИ
2. Пароль. Не должно быть пустым. *password* в АПИ
3. Забыл пароль. Кнопка ведет на страницу сброса пароля. **см. Страница: *RestPassword***
4. Галочка отвечающая за то будет refreshToken в ответе или нет. rememberMe в АПИ
5. Кнопка отправляющая форму.
6. Переход на страницу создания аккаунта. **см. Страница: *Signup***

# Страница: Signup

**URL: */signup***

**Файл: *create.html***

## Компонент: SignupForm



Форма для создание пользователя. Используем *POST /v1/public/auth/signup*

1. Имя пользователя. Обязательно. firstName в АПИ
2. Фамилия пользователя. Обязательно. secondName в АПИ
3. Телефон. Обязательно. phone в АПИ
4. Пароль. Обязательно. password в АПИ
5. Подтвердить пароль. Обязательно. confirmPassword в АПИ

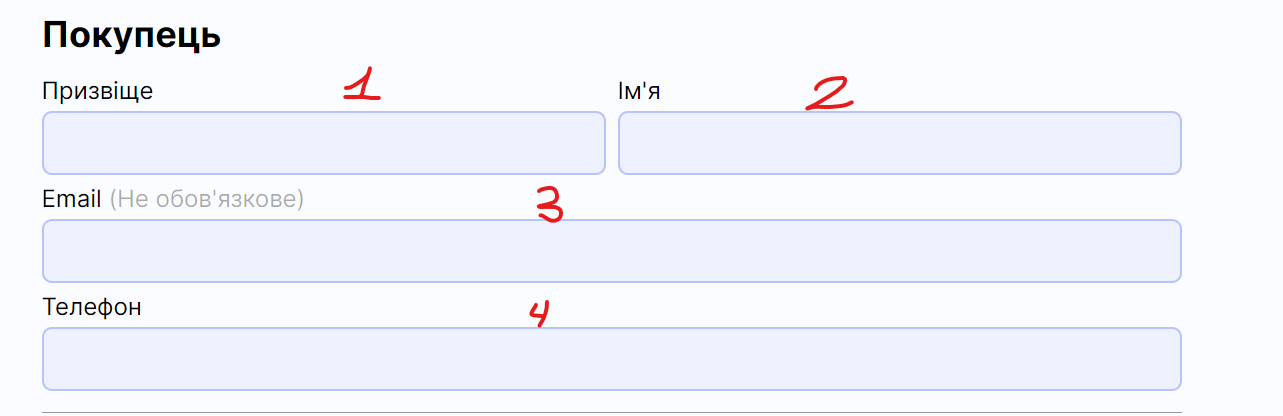
# Страница: Order

**URL: */order***

**Файл: *user.html***

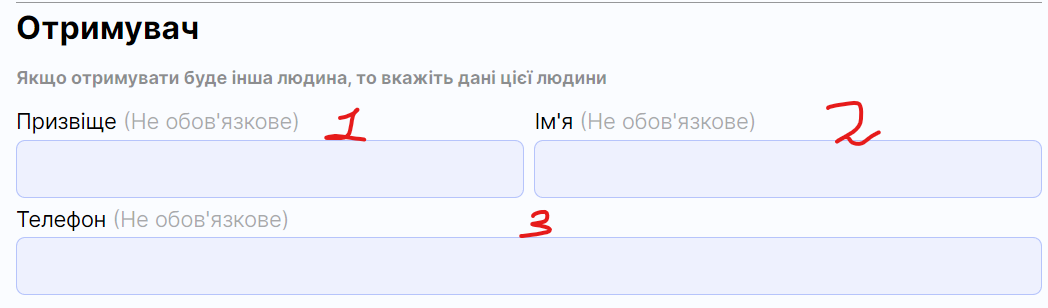
Эта страница нужна для оформления заказа. Используем следующий ресурс *POST /v1/public/orders.*

## Компонент: CustomerInfo



1. Фамилия заказчика. *customer.secondName* в АПИ
2. Имя заказчика. *customer.secondName* в АПИ
3. Эмеил заказчика. *customer.email* в АПИ
4. Телефон. *customer.phone* в АПИ

## Компонент: RecipientInfo



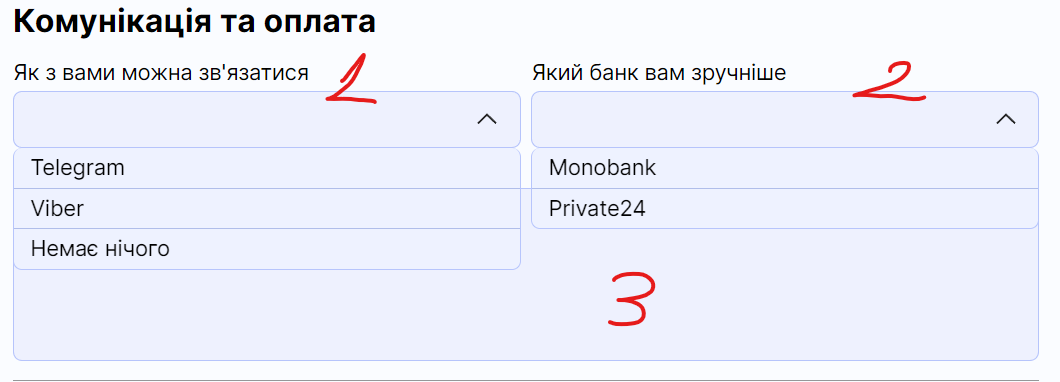
1. Фамилия получателя. Если не введено, то берем фамилию заказчика. *recipient.secondName* в АПИ
2. Имя получателя. Если не введено, то берем имя заказчика. *recipient.firstName* в АПИ
3. Телефон получателя. Если не введено, то берем телефон заказчика. *recipient.phone* в АПИ

## Компонент: PostOfficeForm



1. Название почты. *postOffice.name* в АПИ
2. Область почты. *postOffice.area* в АПИ
3. Город почты. *postOffice.city* в АПИ
4. Номер почты. *postOffice.office* в АПИ

## Компонент: CommunicationAndBankFrom



1. Селектор. Через что можно связатся. Варианты берутся с *GET /v1/public/settings/communicationMethods*. *communicationMethod* в АПИ
2. Селектор. Через что можно связатся. Варианты берутся с *GET /v1/public/settings/paymentMethods*. *paymentMethod* в АПИ
3. Сообщение от клиента. *customerMessage* в АПИ

## Компонент: ConfirmOrder



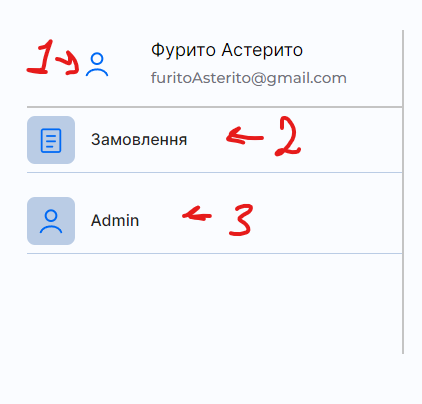
1. Сумма всех продуктов без скидок.
2. Текущая скидка клиента.
3. Сумма всех продуктов со скидками.
4. Отправляет запрос. cartId из АПИ должен быть сгенерирован. Новый айди должен будет генерировать после успешной отправки заказа.

# Страница: User

**URL: */user***

**Файл: *orders.html и orders2.html***

## Компонент: UserMenu



1. Информация о пользователе. Инфа берется с *GET /v1/public/users/{userId}*.
2. Кнопка *открывающая* страницу с заказами пользователя. **см. Страница: *Orders***
3. Кнопка для перехода в админку (http://admin.idealtex.com.ua). *Отображаем* кнопку, если **role = *administrator***из запроса *GET /v1/public/users/{userId}.* Для реквеста нужен токен

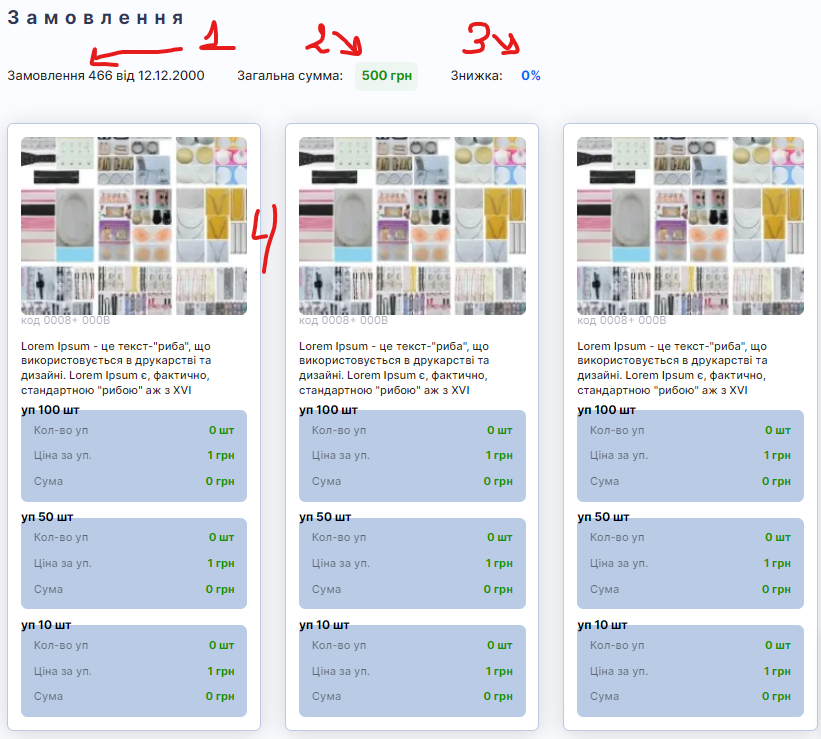
## Компонент: UserOrders



Продукты пользователя. Инфа по запросу *GET /v1/public/users/{userId}/orders.*

Формируется так: Замовлення + orderNumber. При нажатии *открывается* заказ **см. Ниже**

## Компонент: UserOrder



1. Продукты пользователя. Инфа по запросу *GET /v1/public/users/{userId}/orders.* Формируется так: Замовлення + orderNumber.
2. Сумма всего заказа. totalSumOfProducts из АПИ.
3. Скидка. discount из АПИ
4. Продукт почти такой же как в корзине