

Урок 4: Бот-магазин — списание остатков и блокировки двойных продаж

В данном уроке вы узнаете как создать самый простой бот для интернет-магазина с контролем остатков на складе.

Мы создадим бот для небольшого магазина глиняных изделий, который даст возможность клиентам рассмотреть имеющиеся товары и заказать их. Все заказы будут сохраняться в базу данных, а на складе будет производиться списание остатков.

Интерфейс бота в мессенджере будет состоять из экранов выбора категории и просмотра товаров конкретной категории. На экране товара, будут доступны кнопки навигации по товарам текущей категории и кнопка оформить заказ.

Главной особенностью данного урока будет работа с блокировками. Мы можем столкнуться с случаем когда остался один экземпляр определенного товара. Для того чтобы исключить ошибки, когда несколько заказчиков в одно и то же время хотят заказать этот товар, мы используем блокировки.

При использовании блокировок вы сможете обеспечить последовательное выполнение операций, запрошенных одновременно, гарантируя таким образом более надежную работу вашего приложения.

Подробнее познакомиться с блокировками вы можете из инструкции [Блокировки как средство управления параллелизмом](#)

Подробнее изучить работу бота вы можете с помощью нашего примера — бота в Telegram: [@BlockingMetabot](#)

В ходе урока вы узнаете:

- Как работать с кастомными таблицами и создать с их помощью БД соответствующую вашим потребностям;
- Как контролировать параллельные процессы в ботах при помощи блокировок.

Отметим, что урок является одним из вариантов реализации данной задачи, целью которого является знакомство с функционалом для работы с блокировками.

Понятно, что можно придумать и другие варианты реализации, например задачу можно было бы решить с помощью бронирования товара, когда менеджер вручную проверяет остатки и одобряет продажу первому кто успел забронировать товар, но цель урока именно показать как используя блокировки, реализовать гарантию продажи не более чем есть на остатке и полностью в автономном режиме, без ручных действий по бронированию.

Пора перейти к практике!

Перед началом следует ознакомиться с предыдущими уроками [Hello Humans: ваше руководство по быстрому старту](#) и [Metabot 101: Вывод фото в боте по REST API](#)

Создание скриптов

1. Чтобы не отвлекаться на это в будущем, сразу создадим стартовый скрипт с приветствием, маршрут связанный с ним и канал Телеграм, при помощи которого будем тестировать бота, как мы делали это в предыдущих уроках.

Нам потребуются следующие скрипты:

- Стартовый скрипт;
- Меню;
- Скрипты категорий;
- Скрипты выбора товара;
- Скрипт оформления заказа.

Перейдем к их созданию

Стартовый скрипт и меню

Стартовый скрипт будет активироваться всего раз после начала лидом диалога с ботом. В нем должно быть всего две команды: отправка сообщения с приветствием и переход на скрипт меню.

КОМАНДЫ

Эти команды бот выполняет до того как покажет меню.

КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Отправить текст	Здравствуйте! Я бот помощник интернет магазина.	<div><div></div><div></div><div></div></div>
Выполнить скрипт	Меню	<div><div></div><div>Перейти к скрипту</div><div></div><div></div></div>
<div>+ Добавить команду</div>		

Скрипт меню будет выполнять роль навигатора для переключения между категориями товаров. Основным его наполнением будут кнопки, каждая из которых ведет в скрипт определенной категории товаров. Например, в нашем случае, для магазина глиняных изделий, будет создано две категории: тарелки и кружки.

2. Таким образом в скрипте меню нам нужно добавить команду отправки сообщения-подсказки, например: "Какие товары вас интересуют?", и кнопки меню соответствующие каждой категории.

КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Отправить текст	Какие товары вас интересуют?	<div><div></div><div></div></div>
<div>+ Добавить команду</div>		

МЕНЮ

Настройте здесь пункты вашего меню, которые бот будет присылать пользователю.
Код — это текст, который должен прислать пользователь, чтобы попасть в нужный раздел меню.
Меню опционально. Если меню нет, то бот покажет пользователю меню из раздела выше (кроме Старта).

КОД	НАДПИСЬ	СКРИПТ	УСЛОВИЕ ВЫВОДА	СКРЫТЬ КОД	СКРЫТЬ СТРОКУ	№ СТРОКИ	ОПЕРАЦИИ
1	Тарелки	Тарелки					<div><div></div><div></div><div></div><div></div></div>
2	Кружки	Кружки					<div><div></div><div></div><div></div><div></div></div>

Скрипты категорий

Для каждой категории мы создадим свой скрипт, на который будет ссылаться соответствующая кнопка из меню.

Для удобства все скрипты категорий можно занести в меню скриптов в один раздел. Разделы можно сворачивать, чтобы ненужные в данный момент скрипты не мешались перед глазами, а так же их можно сортировать между собой и организовывать свое рабочее пространство как вам удобно.

3. Чтобы это сделать создайте раздел при помощи кнопки **Создать раздел**. Введите название раздела и подтвердите действия кнопкой **Создать**.

Скрипт	Тип	Свойства	Операции 
Категории (2)			   
Кружки	Стандарт	Корневой скрипт: Да	   
Тарелки	Стандарт	Корневой скрипт: Да	   
Выбор товара (3)			   
Старт	Стандарт	Корневой скрипт: Да	   
Меню	Стандарт	Корневой скрипт: Да	   
 Создать скрипт		 Создать раздел	

4. Далее в редакторе свойств скрипта из выпадающего списка **Раздел** выберите созданный ранее раздел.

Свойства скрипта



Название:

Кружки

Код: ⓘ

Раздел:

Категории

Теперь ваше рабочее пространство будет лучше организовано и с ним будет удобнее работать.

5. Но вернемся к наполнению скриптов категорий.

Данные скрипты понадобятся нам для того, чтобы бот мог запомнить, товары из какой категории следует выводить боту. В каждом таком скрипте мы создадим две команды: JS код с созданием атрибутов и переход к скрипту вывода товаров.

КОМАНДЫ

Эти команды бот выполняет до того как покажет меню.

КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Выполнить JavaScript	<pre>lead.setAttr('category', 'mugs'); lead.setAttr('productNum', 0);</pre>	  
Выполнить скрипт	Вывод товара	 Перейти к скрипту  

В команде **Выполнить JavaScript** в атрибуты бота будут добавлены две переменные:

- **category** — переменная в которой будет храниться наименование выбранной лидом категории;
- **productNum** — переменная в которой будет храниться начальный номер выводимого продукта.

Ниже вы можете скопировать код с фото.

```
lead.setAttr('category', 'mugs');  
lead.setAttr('productNum', 0);
```

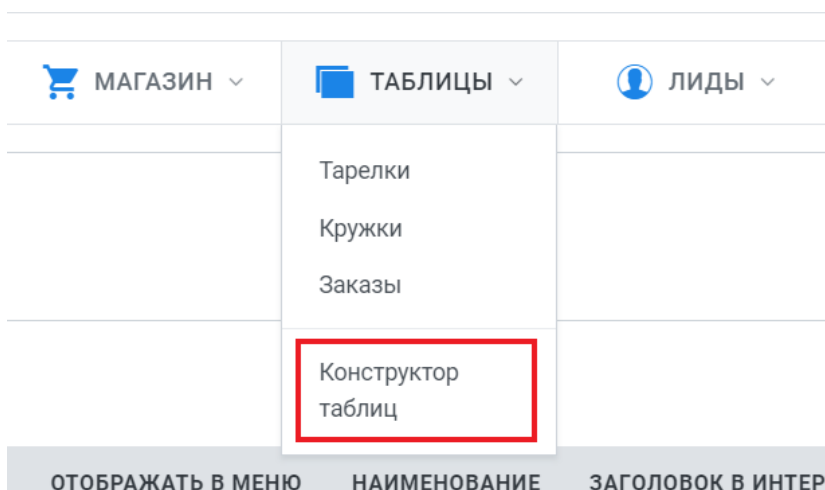
Перед созданием оставшихся скриптов следует реализовать БД для магазина.

Создание кастомных таблиц

Платформа Metabot предоставляет возможность создания своей БД при помощи кастомных таблиц.

Подробнее познакомиться с кастомными таблицами вы можете из инструкции [Кастомные таблицы](#)

1. Чтобы перейти в меню кастомных таблиц в верхнем меню бота из выпадающего списка **Таблицы** выберите **Конструктор таблиц**.



Для нашего бота нужны следующие таблицы:

- Таблица товаров;
- Таблица заказов.

Перейдем к их созданию.

Таблица товаров

2. Для создания новой таблицы нажмите на кнопку **Создать** в меню кастомных таблиц. В открывшемся окне заполните следующие поля:

- **Наименование** — название таблицы используемое в JS методах (должно состоять только из латинских букв без пробелов);
- **Заголовок в интерфейсе** — название таблицы отображаемое в интерфейсе (может состоять из любых символов).

3. Остальные поля оставьте без изменения и нажмите кнопку **Создать**.

☒ Активна

Если опция отключена, то просмотр и редактирование будут недоступны, также таблица будет скрыта из меню быстрого доступа.

☒ Синхронизация структуры

Если опция отключена, то синхронизация структуры таблицы с бд не будет выполняться.

Отключение опции используется когда все поля в БД создаются вручную.

Опция доступна только администраторам платформы.

☒ Отображать в меню

Наименование:

products

Заголовок в интерфейсе:

Продукты

Комментарий:

4. После создания перейдем в структуру таблицы по соответствующей кнопке в конструкторе таблиц.

ТАБЛИЦЫ										+ Создать
ID	АКТИВНА	СИХР. СТРУКТУРЫ	ОТОБРАЖАТЬ В МЕНЮ	НАИМЕНОВАНИЕ	ЗАГОЛОВОК В ИНТЕРФЕЙСЕ	СОЗДАНА	ОБНОВЛЕНА	ОПЕРАЦИИ		
97	Да	Да	Да	products	Продукты	2023-07-24 13:30:37	2023-07-27 11:33:07	✎	🗑	↑
									🔍 Структура	🔍 Данные
										🔄 Обновить все поля в БД

Здесь мы можем указать какие поля будет содержать наша таблица. Для примера создадим первое поле таблицы товаров **Название**, в котором будут храниться названия продающихся в магазине товаров.

5. Для этого в меню структуры таблицы нажимаем на кнопку **Создать**. В открывшемся окне заполняем следующие поля:

- **Наименование** — название поля используемое в JS методах (должно состоять только из латинских букв без пробелов);
- **Заголовок в интерфейсе** — название поля отображаемое в интерфейсе (может состоять из любых символов);
- **Тип** — выпадающий список с типами значений, которые может принимать поле.

☒ Активно

Если опция отключена, то поле не будет доступно для просмотра и редактирования.

☒ Синхронизация структуры

Если опция отключена, то синхронизация структуры поля с бд не будет выполняться.

Отключение опции используется когда поле в БД создается вручную.

Опция доступна только администраторам платформы.

Наименование:

Заголовок в интерфейсе:

Тип:

☐ Обязательное поле

Значение по умолчанию:

6. Таким же образом создаем остальные поля таблицы товаров:

- **Категория** — категория товара (в нашем случае **plates** или **mugs**);
- **Количество** — количество товаров данного типа оставшихся на складе;
- **Цена** — цена товара;
- **Фото** — ссылка на изображение товара.

ПОЛЯ ТАБЛИЦЫ "PRODUCTS"

ID	АКТИВНО	СИХР. СТРУКТУРЫ	НАИМЕНОВАНИЕ	ЗАГОЛОВОК В ИНТЕРФЕЙСЕ	ТИП	ОБЯЗАТЕЛЬНОЕ ПОЛЕ
580	Да	Да	id	ID	AUTOINCREMENT	Да
601	Да	Да	category	Категория	TEXT	Да
582	Да	Да	name	Название	TEXT	
583	Да	Да	qua	Количество	INT	
584	Да	Да	price	Цена	INT	
585	Да	Да	photo	Фото	TEXT	

Значения в поле **category** и названия категорий из скриптов категорий должны быть одинаковыми для исправной работы данного бота.

7. Затем таблицу нужно заполнить товарами для каждой категории, для исправной работы бота:

ID	КАТЕГОРИЯ	НАЗВАНИЕ	КОЛИЧЕСТВО	ЦЕНА	ФОТО	ОПЕРАЦИИ
1	plates	Набор мисок в форме сердца	1	760	https://unsplash.com/photos/JxLf4KYVT-A	 
2	plates	Декоративная тарелка для мелочей	0	500	https://unsplash.com/photos/a-white-and-blue-plate-sitting-on-top-of-a-wooden-table-47t-txFK80Y	 
3	plates	Набор из двух мисок	4	600	https://unsplash.com/photos/-Olinu1YDTk	 
5	mugs	Бело-красная кружка	3	400	https://unsplash.com/photos/e3B0j05lc20	 
6	mugs	Кофейная чашка	5	350	https://unsplash.com/photos/bpGr7defHKk	 
7	mugs	Кружка с листом	2	450	https://unsplash.com/photos/njGJ3RZOBS	 

Таблица заказов

8. По тому же принципу создаем таблицу заказов со следующими полями:

- **ФИО** — имя заказчика;
- **Адрес** — адрес на который нужно прислать товар;
- **Телефон** — номер телефона заказчика;
- **Категория** — категория из которой был заказан товар;
- **Товар** — id товара в таблице товаров.

100

Да

Да

Да

orders

Заказы

2023-07-25 13:24:59

2023-07-26 13:48:30

Структура таблицы (JSON словарь):

АКТИВНО	СИХР. СТРУКТУРЫ	НАИМЕНОВАНИЕ	ЗАГОЛОВОК	ТИП	ОБЯЗАТЕЛЬНОЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ	РАЗМЕР	ТОЧНОСТЬ	ПОДСКАЗКА	ВСПЛЫВАЮЩАЯ ПОДСКАЗКА
Да	Да	id	ID	AUTOINCREMENT	Да	NULL				
Да	Да	name	ФИО	TEXT						
Да	Да	address	Адрес	TEXT						
Да	Да	phone	Телефон	TEXT		NULL				
Да	Да	category	Категория	TEXT						
Да	Да	product_id	Товар	INT		NULL				

Таблица заказов будет заполняться автоматически при взаимодействии лидов с ботом. Теперь мы можем переходить к созданию оставшихся скриптов.

Работа с методами таблиц

Для просмотра товаров будет создано два скрипта: скрипт вывода товара и скрипт перехода на следующий товар.

Скрипт вывода товаров

В скрипте вывода товара при помощи обращения к таблице товаров и команды отправки текста лиду будет выведена вся информация о товаре.

1. Для начала добавим команду **Выполнить JavaScript** со следующим кодом:

```
let category = lead.getAttr('category');
let productNum = lead.getAttr('productNum')

let item = table.find("products", [], [{"category", "=", category}]);

lead.setAttr('productNumber', parseInt(productNum) + 1);
lead.setAttr('productName', item[productNum].name);
lead.setAttr('productQua', item[productNum].qua);
lead.setAttr('productPrice', item[productNum].price);
lead.setAttr('productPhoto', item[productNum].photo);
```

В первой и второй строке мы берем из атрибутов лида информацию о том, какая категория была выбрана пользователем и какой по счету товар следует выводить.

Затем в четвертой строке мы обращаемся к таблице с помощью команды нахождения записей и записываем результат в локальную переменную **item**. Для этого указываем наименование таблицы и условие по которому будет искаться запись. В нашем случае мы ищем запись сопоставляя категорию выбранную лидом с категорией товара в таблице.

Подробнее изучить методы кастомных таблиц вы можете из [Справочника функций JS](#)

В строках 5-8 из записи под номером **productNum** в массиве записей запоминаем в атрибутах лида все данные о товаре. Эти данные затем будут выведены в диалог с лидом при помощи команды отправки текста.

2. После JS команды добавляем команду **Отправить текст** со следующим содержимым:

Товар №{{productNumber}} — {{productName}}

Осталось на складе: {{productQua}}

Цена: {{productPrice}}

Фото товара: {{productPhoto}}

Таким образом у нас получается следующий набор команд:

КОМАНДЫ

Эти команды бот выполняет до того как покажет меню.

КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Выполнить JavaScript	<pre>let category = lead.getAttr('category'); let productNum = lead.getAttr('productNum') let item = table.find("products", [], [{"category", "=", category}]); lead.setAttr('productNumber', parseInt(productNum) + 1); lead.setAttr('productName', item[productNum].name); lead.setAttr('productQua', item[productNum].qua); lead.setAttr('productPrice', item[productNum].price); lead.setAttr('productPhoto', item[productNum].photo);</pre>	  
Отправить текст	<p>Товар №{{productNumber}} - {{productName}}</p> <p>Осталось на складе: {{productQua}}</p> <p>Цена: {{productPrice}}</p> <p>Фото товара: {{productPhoto}}</p>	  

3. В данном скрипте осталось только создать меню со следующими кнопками:

- **Следующий заказ** — будет ссылаться на скрипт для вывода следующего товара;
- **Оформить товар** — будет ссылаться на скрипт оформления заказа;
- **Выбрать другую категорию** — будет ссылаться на скрипт меню.

Скрипт перехода на следующий товар

Данный скрипт поможет нам выводить лиду следующие товары из таблицы. В нем нужно создать две команды: JS скрипт с изменением номера товара и переход в скрипт вывода товаров.

4. Для начала добавим команду **Выполнить JavaScript** со следующим кодом:

```
let category = lead.getAttr('category');
let productNum = lead.getAttr('productNum');

let item = table.find("products", [], [{"category", "=", category}]);
```

```
if (item.length > productNum + 1){
    lead.setAttr('productNum', parseInt(productNum) + 1);
} else {
    lead.setAttr('productNum', 0);
}
```

В данном скрипте происходит проверка больше ли количество записей в таблице товаров выбранной категории чем значение номера товара, который должен быть выведен лиду. Если да, то номер товара увеличивается на один, иначе счет номера товара начинается сначала (с нуля).

5. После JS команды добавляем команду **Выполнить скрипт — Вывод товара** и переходим к созданию следующего скрипта.

Работа с блокировками

Перейдем к созданию последнего скрипта в нашем боте — **Оформление заказа**.

В данном скрипте у лида будут запрошены контактные данные, а затем они и остальные данные о заказе будут добавлены в таблицу заказов.

1. Первым делом создадим три команды Запросить значение. Из них мы узнаем имя, адрес и телефон заказчика.

КОМАНДЫ

Эти команды бот выполняет до того как покажет меню.

КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Запросить значение	Напишите ваше ФИО: [customer]	  
Запросить значение	Укажите адрес на который следует отправить товар: [address]	  
Запросить значение	Укажите ваш номер телефона: [phone] Регулярное выражение:Российский мобильный, компактно 79xxxxxxxx ((7 8)9\d{9}), «Требуется ввести телефонный номер в формате "79xxxxxxxx" или "89xxxxxxxx"»)	  

2. Затем перейдем к созданию команды **Выполнить JavaScript**. Одной из основных ее частей будет являться работа с блокировками.

```
let customer = lead.getAttr('customer');
let address = lead.getAttr('address');
let phone = lead.getAttr('phone');
let category = lead.getAttr('category');
let productNum = lead.getAttr('productNum');

// Запись для таблицы заказов

let order = {
  "name": customer,
  "address": address,
  "phone": phone,
  "category": category,
  "product_id": productNum
}

//-----

let isLocked = false

// Получаем блокировку по боту

bot.sendText('Рассматриваем ваш заказ! Ожидайте подтверждения');

isLocked = bot.waitForBotLock('my_lock', '', 20, 300);

//-----

if (isLocked) {

  // Проверяем остались ли товары на складе

  let item = table.find("products", [], [{"category", "=", category}]);

  if (item[productNum].qua > 0){

    //-----

    // Обновляем данные на складе и создаем запись в таблице заказов
```

```

    item[productNum].update({"qua": item[productNum].qua - 1});

    table.createItem("orders", order);

//-----

    bot.sendText('Заказ оформлен');

} else {

    bot.sendText('К сожалению данного товара уже нет на складе');

}
} else {

    bot.sendText('Время ожидания истекло, попробуйте еще раз');

}

// Освобождаем блокировку полученную в данном скрипте

let hasLock = bot.hasLockForBot('my_lock');

if (hasLock) {

    bot.releaseCurrentLockForBot('my_lock');

}

//-----

```

Для начала скрипт узнает все необходимые переменные из атрибутов лида и заносит их в JSON переменную, которая понадобится при создании новой записи в таблице заказов.

Далее переходим к работе с блокировками.

Скрипт отправляет лиду сообщение о том что его заказ рассматривается и пытается перехватить блокировку. Если время ожидания захвата истекло, то пользователю отправляется сообщение с просьбой попробовать еще раз.

Если же блокировка была захвачена, то скрипт проверяет, точно ли остались товары на складе, и если и эта проверка прошла успешно, то из поле **Количество** в записи товара

вычитается единица, а данные заказа добавляются в таблицу заказов.

В конце скрипта происходит проверка на существование ранее захваченной блокировки и ее удаление.

3. После JS команды добавляем команду **Выполнить скрипт — Меню**.

Бот готов к тестированию.

Проверка работы бота

Проверим как работает наш бот.

1. Выберите категорию в меню. У вас должны вывестись данные о товаре.

Какие товары вас интересуют? 15:24

Товар №1 - Набор мисок в форме сердца

Осталось на складе: 2

Цена: 760

Фото товара: <https://unsplash.com/photos/JxLf4KYVT-A>

Unsplash

Photo by Jessica Ruscello on Unsplash

Red heart bowls – Download this photo by Jessica Ruscello on Unsplash



15:24

Следующий товар

Оформить заказ

Выбрать другую категорию

LB

2. Нажмите на кнопку **Следующий товар**. У вас должны вывестись данные о другом товаре.

Товар №2 - Декоративная тарелка для мелочей

Осталось на складе: 1

Цена: 500

Фото товара: <https://unsplash.com/photos/a-white-and-blue-plate-sitting-on-top-of-a-wooden-table-47t-txFK80Y>

Unsplash

Photo by Gloria Medlin on Unsplash

Download this photo by Gloria Medlin on Unsplash



15:27

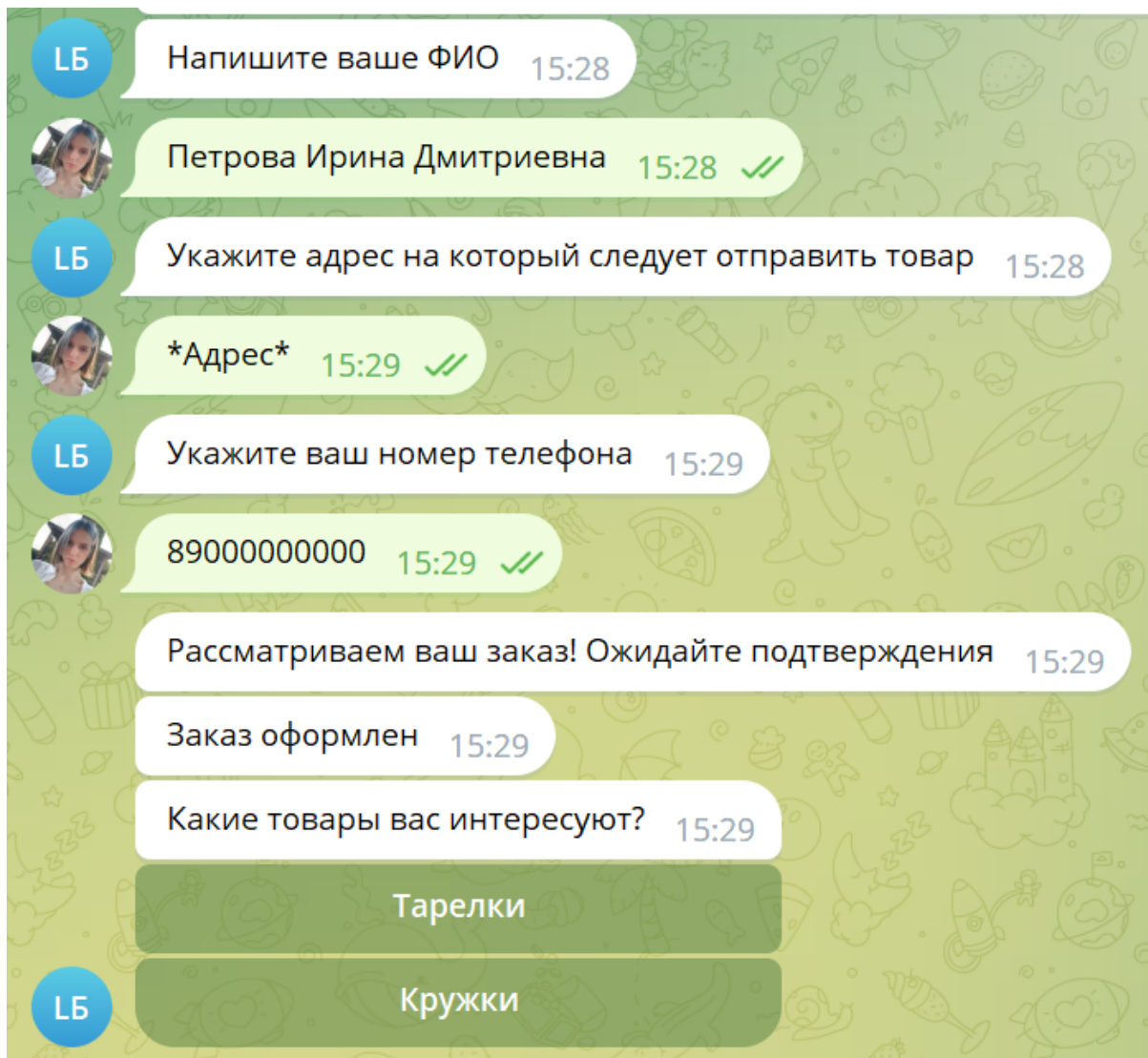
Следующий товар

Оформить заказ

Выбрать другую категорию

LB

3. Нажмите на кнопку **Оформить заказ**. Бот должен запросить у вас данные и прислать сообщение о принятии заказа.



В боте при этом должны обновиться данные в таблицах. В таблице заказов должен появиться новый заказ. А в таблице товаров должно уменьшиться количество выбранного товара.

ЗАКАЗЫ							Создать
ID	ФИО	АДРЕС	ТЕЛЕФОН	КАТЕГОРИЯ	ТОВАР	ОПЕРАЦИИ	
2	Петрова Ирина Дмитриевна	*Адрес*	890000000000	plates	2	 	

ПРОДУКТЫ							Создать
ID	КАТЕГОРИЯ	НАЗВАНИЕ	КОЛИЧЕСТВО	ЦЕНА	ФОТО	ОПЕРАЦИИ	
1	plates	Набор мисок в форме сердца	1	760	https://unsplash.com/photos/JxLf4KYVT-A	 	
2	plates	Декоративная тарелка для мелочей	0	500	https://unsplash.com/photos/a-white-and-blue-plate-sitting-on-top-of-a-wooden-table-47t-txFK80Y	 	

Если в каждом из пунктов все работает правильно, значит у вас получилось сделать свой бот-магазин.

Поздравляем вас с прохождением урока!

Версия #12

Ирина Петрова создал 2023-07-26 09:24:29 UTC

Ирина Петрова обновил 2024-03-20 08:57:29 UTC