

# Как настроить платежи через бота в Telegram

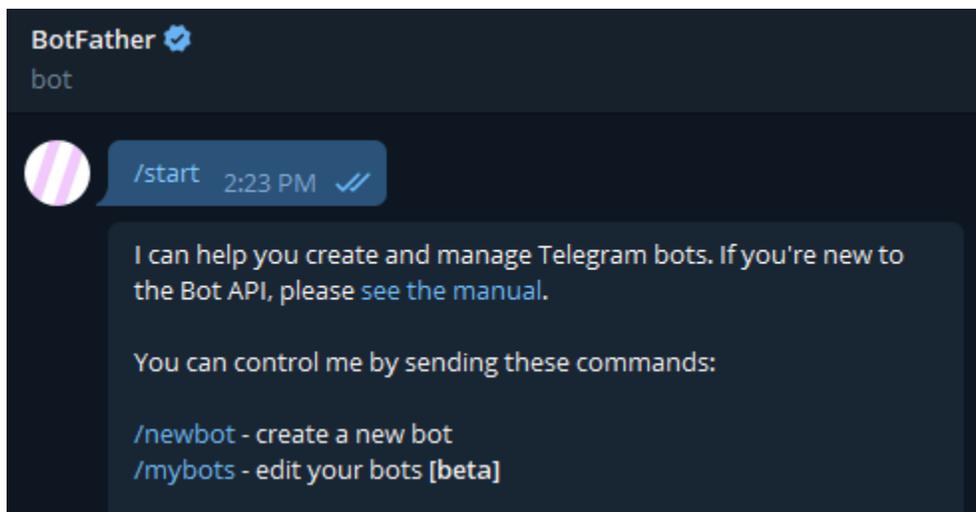
Чтобы принимать платежи в Telegram, нужно выполнить три действия:

1. Создать бота.
2. Подключить его к боту Робокассы.
3. Настроить своего бота для работы с платежами.

Для настройки платежей через бота потребуется идентификатор магазина, который вы указывали в Робокассе при создании магазина.

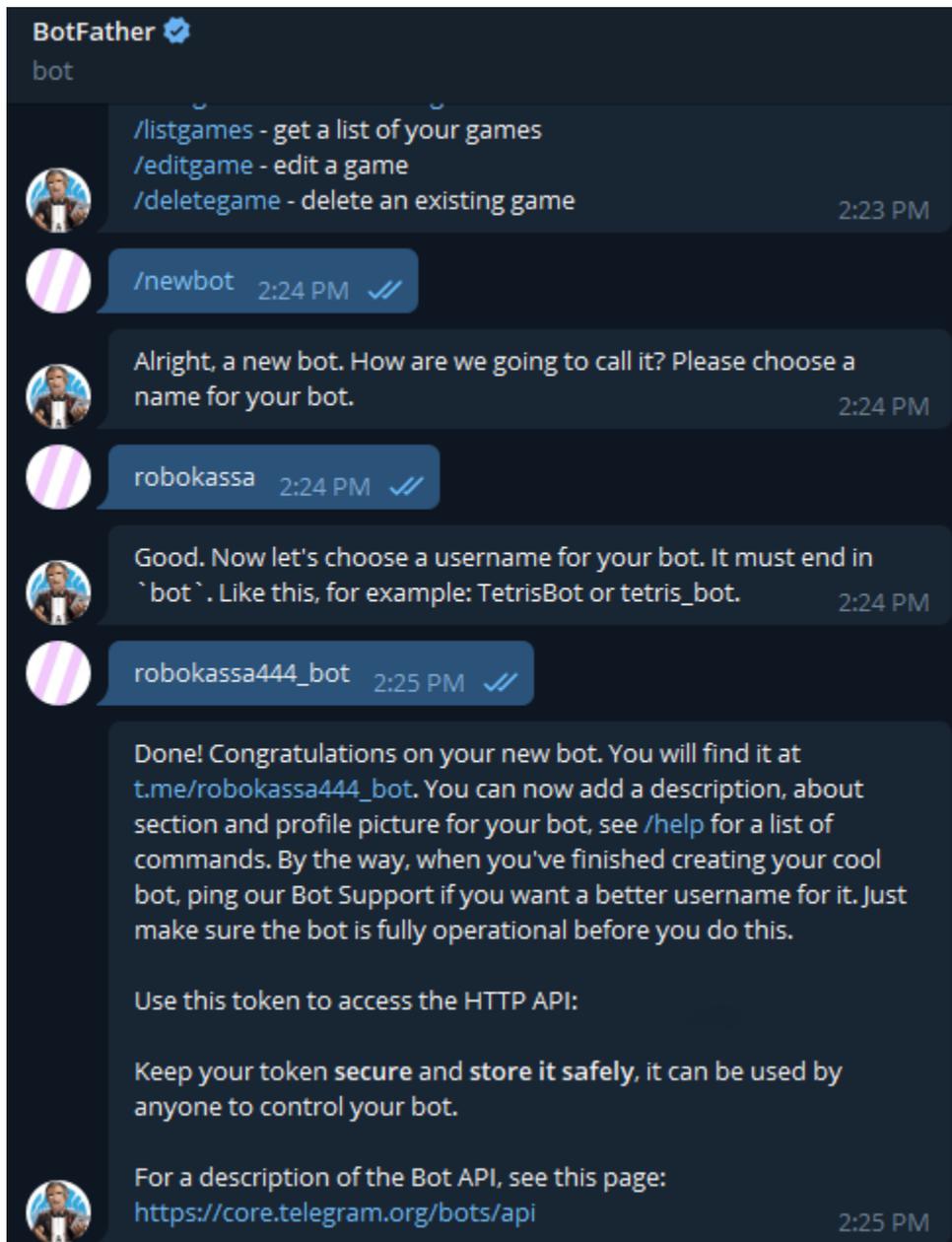
## Как создать бота

1. Зайдите в Telegram и найдите [@BotFather](#). Ему необходимо отправить 2 сообщения: сначала /start, а потом /newbot.



2. Теперь надо придумать имя и юзернейм. Обратите внимание: имя необязательно должно быть уникальным - его будут видеть пользователи. А вот юзернейм нужен уникальный - он будет отображаться в ссылке на вашего бота.

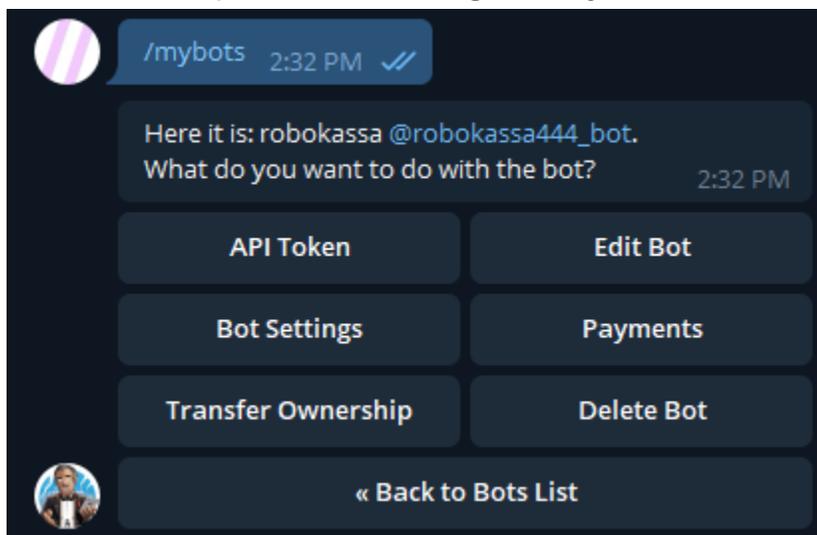
3. Если у вас получилось придумать уникальный юзернейм, вы получите токен, обеспечивающий доступ к вашему боту. Будьте осторожны и не присылайте его третьим лицам.



Вот и все - останется лишь произвести платежные настройки.

Как подключить прием платежей от Robokassa к своему боту

1. Зайдите в [@BotFather](#) и отправьте сообщение /mybots.
2. Кликните по имени вашего бота - вы увидите панель управления, в которой необходимо перейти в **BotSettings — Payments**.



3. Здесь нужно выбрать режимы платежей: за основные платежи отвечает **“Connect Robokassa Live”**, а за тестовые - **“Connect Robokassa Test”**.
4. Если все сделано верно, вы перейдете в “платежный” бот - **RobokassaPaymentBot**. Отправьте ему сообщение /start и введите идентификатор вашего магазина. Вам придет SMS для подтверждения.
5. Снова зайдите в [@BotFather](#). От него вы получите токен для выставления счетов. Если возникнет необходимость, вы всегда сможете посмотреть его в настройках: **BotSettings — Payments**.

## Как настроить бота для работы с платежами

Дальше речь пойдёт о настройке бота через API Telegram. Если вы используете конструктор ботов (Botmother или любой другой), следуйте его инструкциям. Если вы создали бота самостоятельно, следуйте нашей инструкции ниже. При необходимости вы можете посмотреть пример бота на PHP в нашем GitHub – <https://github.com/robokassa/telegram-pay>.

1. Используйте [метод sendinvoice](#) из платёжного API Telegram. По вашему запросу созданный вами бот будет присылать покупателю счёт. Обратите внимание: он должен содержать цену на ваш товар, его название, информацию о товаре и два параметра - [payload](#) и [provider\\_token](#).

```
{
  "chat_id": 000000,
  "title": "Оплата через Робокассу",
  "description": "Описание заказа",
  "payload": "test"
  "provider_token": "111111111:LIVE:637955761195921111",
  "currency": "RUB",
  "prices": [
    {
      "label": "Руб",
      "amount": 10000
    }
  ]
}
```

Chat\_id – уникальный идентификатор целевого чата или имя пользователя целевого канала.

Title – название продукта, 1-32 символа

Description – описание продукта, 1-255 знаков

Payload – полезные данные счета-фактуры, определенные ботом, 1–128 байт. Это не будет отображаться пользователю, используйте его для своих внутренних процессов.

Provider\_token – токен выданный через бот Robokassa.

Currency – трехбуквенный код валюты ISO 4217.

Prices – разбивка цен, сериализованный список компонентов в формате JSON. Сумма указывается в копейках: **"amount": 10000 == 100 руб**

Для передачи номера заказа и номенклатуры в **sendInvoice** необходимо добавить еще одно поле – **provider\_data**, который должен содержать в себе JSON с номером заказа(InvoiceId) и [номенклатурой\(Receipt\) согласно документации Robokassa](#).

Пример с использованием **provider\_data**:

```
{
  "InvoiceId": 100,
  "Receipt": {
    "sno": "osn",
```

```
"items": [  
  {  
    "name": "Товар 1,  
    "quantity": 1,  
    "sum": 50,  
    "tax": "vat10",  
    "payment_method": "full_payment",  
    "payment_object": "commodity"  
  },  
  {  
    "name": "Товар 2",  
    "quantity": 2,  
    "sum": 50,  
    "tax": "vat10",  
    "payment_method": "full_payment",  
    "payment_object": "commodity"  
  }  
]  
}  
}
```

2. После подтверждения платежа покупателем вы получите webhook с Update, содержащий объект [PreCheckoutQuery](#). На запрос необходимо ответить в течение 10 секунд, используя метод [answerPreCheckoutQuery](#).

3. Если все сделано правильно, вам придет сообщение с объектом [SuccessfulPayment](#). В боте есть удобная опция: если вы продаете услугу, можно настроить ее автоматическое подключение после успешной оплаты, если цифровой контент - его отправку покупателю.