

Telegram Bot

- [Настройка БД](#)
- [Создание бота](#)
- [Настройка](#)
- [Интерфейс бота](#)
- [Начало работы с ботом](#)
- [Авторизация через кабинет пользователя с помощью телеграмм](#)

Настройка БД

Настройка работы с базой данных и создание таблиц:

```
mysql -D abills < /usr/abills/Abills/modules/Telegram/Telegram.sql
```

Создание бота

(если у вас уже есть бот, эту часть можно пропустить)

Эту операцию нужно делать с устройства где установлен Telegram.

1. Заходим <https://t.me/botfather> .
Если есть аккаунт нажимаем кнопку “*Send message*”.
Если нету, то регистрируемся в Telegram-е, затем снова заходим и нажимаем “*Send message*”, или можно сразу зайти в Telegram, в поиске вбить **BotFather** и перейти к диалогу с ним.
2. На главном экране бота нажимаем “*Start*”.
3. Покажется список возможных команд для обращения к **BotFather**.
Нам нужно ввести команду **/newbot**.
4. **BotFather** спросит имя нашего бота.
Это имя будет отображаться в деталях контакта. Вводим имя бота, например, *Testbot*.
5. Следующий шаг нас спрашивают Имя пользователя бота (username).
Этот параметр будет использоваться для запросов к боту.
От 5 до 32 символов, регистр-чувствительная латиница, цифры и нижние подчеркивания.
Имя пользователя бота **должно** заканчиваться на bot. Вводим имя пользователя (username), например **TestAbills_bot**.
6. Вы зарегистрировали вашего бота – внимательно осмотрите финальное сообщение на предмет параметров:
Линк Телеграм-бота
Например: https://t.me/TestAbills_bot
а так же очень важный параметр - **токен**.
Например: 167953392:AAHVUmXdE0TzLbdzSuV23QuEBTmDMjzg
7. Бот успешно создан!

Настройка

1. Для нормальной работы через webhook вам необходимо открыть порт **443**, или **80**, **88**, **8443** для входящих POST запросов.
Важно: Телеграм не работает с портом **9443**. Обязательно включить шифрование.
2. Необходимо создать конфигурацию доступа веб-сервера Apache для работы Telegram-бота.
Заходим в директорию, где находится конфиг Apache.

Создание конфигурации веб-сервера для Telegram-бота

```
cd /etc/apache2/sites-enabled/  
touch telegram_httpd.conf  
cat abills_httpd.conf >> telegram_httpd.conf
```

3. В **telegram_httpd.conf** меняем порт с 9443 на 8443.
4. Создайте директорию со сложным названием:

Создание каталога для Telegram-бота

```
mkdir /usr/abills/cgi-bin/TelegramAAHVUmXdE0TzLbdzSuV23QuEBTmDMjzg/
```

Название директории *желательно* сделать сложным и уникальным, для этого можно использовать токен (или его часть).

5. Скопируйте в эту директорию файл **telegram_bot.cgi**

Привязка телеграм-бота

```
ln -s /usr/abills/Abills/modules/Telegram/telegram_bot.cgi /usr/abills/cgi-bin/TelegramAAHVUmXdE0TzLbdzSuV23QuEBTmDMjzg/
```

6. Необходимо прописать в **config.pl** обязательные параметры (ниже в таблице).

В **config.pl** нужно добавить:

Параметр	Описание
<code>\$conf{TELEGRAM_TOKEN}=' ';</code>	Обязательный параметр. Токен доступа до вашего Telegram-бота. Пример: 167953392:AAHVUmXdE0TzLbdzSuV23QuEBTmDMjzg
<code>\$conf{TELEGRAM_BOT_NAME}=' ';</code>	Обязательный параметр. Должен строго совпадать с "username", что вы указали боту BotFather. Пример: TestAbills_bot
<code>\$conf{TELEGRAM_SUPPORT_PHONE}=' ';</code>	Подпись и номер телефона разделяются знаком ' ' а каждый следующий номер знаком ';' ;' Пример: 'МТС 096000000;Lifecell 123456789'
<code>\$conf{TELEGRAM_LINK_URL}=' ';</code>	Подпись и url разделяются знаком ' ' а каждый следующий url знаком ';' ;' Пример: ' https://test.ua; https://shop.ua;'
<code>\$conf{TELEGRAM_CHANEL}=' ';</code>	Ссылка на произвольный телеграм канал. Пример: 'https://t.me/test_abills'
<code>\$conf{TELEGRAM_SITE}=' ';</code>	Ссылка на сайт провайдера. Пример: 'http://abills.net.ua'
<code>\$conf{TELEGRAM_LANG}=' ';</code>	Локализация бота. По умолчанию, доступны три значения: "english", "russian", "ukrainian".
<code>\$conf{TELEGRAM_MSGS_BOT_ENABLE} = 1;</code>	Позволяет пользователю отвечать на сообщения тех. поддержки.
<code>\$conf{TELEGRAM_USER_PORTAL} = 'url';</code>	Прописать URL к личному кабинету пользователя (Портал абонента).
<code>\$conf{TELEGRAM_TERMINAL} = 'url';</code>	Прописать URL к сервису с картой где будут отображаться терминалы
<code>\$conf{TELEGRAM_PAYMENT_CREDENTIAL} = 'string'</code>	Здесь можно указать информацию для платежных реквизитов по оплате
<code>\$conf{MONEY_UNIT_NAMES}='.'</code>	Название валюты, используется ботом
<code>\$conf{TELEGRAM_NUMBER_EXPR} = '^380/0';</code>	Регулярное выражение для номера телефона. В примере '^380/0' позволяет удалять код страны из номера
<code>\$conf{TELEGRAM_CARDS_MESSAGE} = ' ';</code>	Текст, который будет отображаться для ввода карт пополнения.

Проверьте появился ли доступ к файлу (https://YOUR.URL:YOURPORT/TelegramAAHVUmXdE0TzLbdzSuV23QuEBTmDMjzg/telegram_bot.cgi).

Для подключения телеграм-бота нажмите **Подтвердить телефон** или кнопку **Подписаться** в кабинете пользователя.

Подтвердить телефон

Если доступ есть – можно установить вебхук, соединение с Bot API Telegram.

Если используется действующий SSL-сертификат:

```
curl -F "url=https://YOUR.URL:YOURPORT/TelegramAAHVumXdE0TzLbdzSuV23QuEBTmDMjzg/telegram_bot.cgi"
https://api.telegram.org/bot<YOURTOKEN>/setWebhook
```

Если используется самоподписанный SSL-сертификат:

Установка WebHook

```
curl -F "url=https://YOUR.URL:YOURPORT/TelegramAAHVumXdE0TzLbdzSuV23QuEBTmDMjzg/telegram_bot.cgi" -F
"certificate=@/usr/abills/Certs/server.crt" https://api.telegram.org/bot<YOURTOKEN>/setWebhook
```

Примечание:

- в поле CN (Common Name) сертификата должен быть ваш **домен** или **IP**, иначе Telegram его не примет.
- при любом изменении/обновлении самоподписанного сертификата, его необходимо обновить с помощью команды выше (сертификат указываем явно).

Успешное выполнение команды установки вебхука выглядит примерно так:

```
{"ok":true,"result":true,"description":"Webhook was set"}
```



После настройки телеграм бота нужно перезапустить бота "Остановить и блокировать/Остановить бота", а затем нажать кнопку "Перезапустить бота".

Интерфейс бота

Бот загружает файлы из папки **/usr/abills/Abills/modules/Telegram/buttons-enabled/**

Каждый файл - это отдельная кнопка, которая добавляется в меню.

Для подключения новых функций, необходимо скопировать нужные вам кнопки (или создать ссылку) из папки **/usr/abills/Abills/modules/Telegram/buttons-avaiable/**

Пример

```
mkdir /usr/abills/Abills/modules/Telegram/buttons-enabled/
ln -s /usr/abills/Abills/modules/Telegram/buttons-avaiable/Send_message.pm /usr/abills/Abills/modules
/Telegram/buttons-enabled/
```

На сегодняшний день доступны следующие функции.

Название файла	Кнопка	Описание
Balance.pm	Баланс	Выводит состояние счета, информацию о плановых списания. При недостатке денег появляется кнопка "Пополнить счет". Обязательно добавить параметр с ссылкой на кабинет пользователя: <code>\$conf{TELEGRAM_BALANCE_URL} = 'https://localhost:9443/index.cgi';</code>
Internet_info.pm	Интернет	Выводит информацию о подключенных услугах.
lsp_info.pm	О компании	Выводит информацию о компании (Данные заполняются в Настройка>Другое>Параметры организации)
Make_payment.pm	Пополнить счет	Подсказка о том как пополнить счет (текст нужно заполнять самому в файле)
Send_message.pm	Отправить сообщение	Позволяет пользователю создать сообщение в службу поддержки.
Services_and_account.pm	Услуги и счет	Расширенная версия кнопки Интернет, включает в себя также информацию о счете. Есть возможность взять кредит или отменить приостановление.
Msgs_reply.pm		Позволяет пользователю отвечать на сообщения тех. поддержки.
Privat_payment.pm	Оплата PrivatBank	Позволяет перейти на страницу оплаты PrivatBank для совершения оплаты (Должен быть подключен модуль Privat_terminal.pm)
Easypay_payment.pm	Оплата Easypay	Позволяет перейти на страницу оплаты Easypay для совершения оплаты (Должен быть подключен модуль Easypay.pm)
City24_payment.pm	Оплата City24	Позволяет перейти на страницу оплаты City24 для совершения оплаты (Должен быть подключен модуль City24.pm)
Liqpay_payment.pm	Оплата Liqpay	Позволяет перейти на страницу оплаты Liqpay для совершения оплаты (Должен быть подключен модуль Liqpay.pm)
Crm_new_lead.pm	Оставить заявку	Позволяет добавлять потенциальных клиентов через Телеграм. Если пользователя не найдено по номеру телефона, ему будет предложено оставить заявку на подключение.
Set_credit.pm	Кредит	Возможность взятия кредита.

Equipment_info.pm	Проверка связи Информация о состоянии подключения оптического терминала (в сети или выключен терминал) и состоянии порта оптического терминала, подключен ли к нему роутер.  Для работы кнопки потребуется включить API, в том числе следующие параметры: <pre>\$conf{BOT_API} = '192.20.208.1, 217.116.76.0/24, 104.81.60.171, 91.108.4.0/22, 149.154.160.0/20'; - IP API Telegram (91.108.4.0/22, 149.154.160.0/20)</pre> <pre>\$conf{API_URL} = 'https://your.server.com:9443'</pre> <pre>\$conf{API_ENABLE} = 1;</pre>  Шаблон для отображения действий при ошибке: <code>modules/Telegram/templates/telegram_equipment_info_error.tpl</code>  The screenshot shows a Telegram chat interface. At the top, a blue header bar says 'Проверка связи' (Check connection) with a timestamp of 15:23 and a checkmark. Below it, a grey bubble says 'Подождите, выполняется проверка' (Wait, check is being performed) with a timestamp of 15:23. Then, two green checkmark bubbles appear: 'Оптический терминал работает' (Optical terminal works) and 'Роутер работает' (Router works), both with a timestamp of 15:23. A red X bubble says 'Роутер не работает' (Router does not work) with a timestamp of 15:23. Below these, a large grey bubble contains a list of instructions: 1. Check if all optical cables are correctly connected to your equipment (primary PON port, optical terminal, etc.). 2. Check LEDs on your optical terminal. 3. Check if there is an active network connection on your computer or other device using the internet. Check that network settings are configured correctly. Below the list, a pink bubble with the initials 'AU' says 'После выполнения всех пунктов запустите проверку снова' (After completing all points, run the check again) with a timestamp of 15:23. The chat continues with another blue header bar 'Проверка связи' (15:23, checkmark), followed by a grey bubble 'Подождите, выполняется проверка' (15:23), and then two green checkmark bubbles: 'Оптический терминал работает' (15:23) and 'Роутер работает' (15:23). Finally, a pink bubble with 'AU' says 'Проверка завершена' (Check completed) with a timestamp of 15:23.
-------------------	---

Начало работы с ботом

Для того чтобы начать работать с ботом абонент должен подписаться на него. Сделать это можно двумя способами.

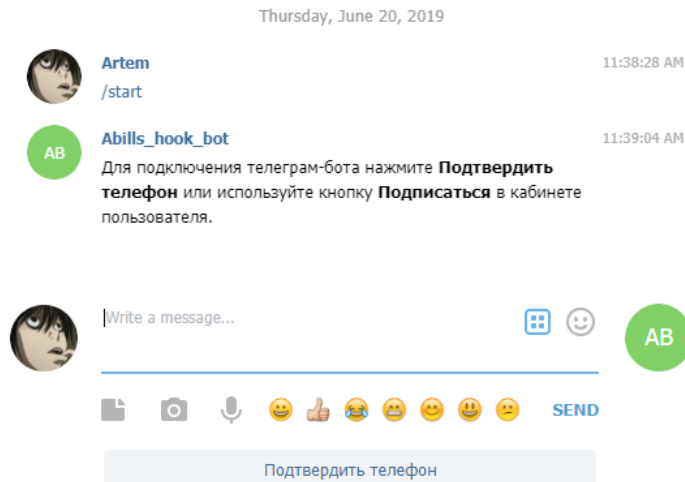
1. Пользователь может нажать кнопку в личном кабинете.



Дополнительно, при нажатии на правую подкнопку может открыться QR-код на подписку:



2. Пользователь может открыть чат с ботом, и там подтвердить свой телефон. (Необходимо чтобы его телефон был занесен в базу биллинга).



⚠ Для корректной работы с ботом нужно, чтобы номер телефона абонента в биллинге был уникален.
Подписаться на бота можно **только** с одного логина.

Для проверки результата после установки бота, используйте эту команду:

Проверка работы Telegram-бота со стороны Bot API

```
curl https://api.telegram.org/bot<TELEGRAM_TOKEN>/getWebhookInfo
```

Разработчикам - [Создание кнопок для бота](#) [Тестирование бота](#)

Авторизация через кабинет пользователя с помощью телеграмм

Пользователь должен быть подписан на бота, тогда будет работать авторизация.

[Как настроить?](#)