

Инструкция по установке,
подключению и использованию
Telegram-бота AgileBot

Для версии 1.6

Содержание

Введение в установку и подключение	3
Требования для установки и подключения	4
1. Установка и подключение программы бота	
1.1 Создание учетной записи бота в Telegram	5
1.2 Установка программы Telegram-бота	7
1.3 Настройка программы Telegram-бота	11
1.4 Подключение и запуск программы	16
2. Что дальше.	
2.1 Админ-панель бота	18
2.2 Модель бота или как работает основная программа бота	19
2.3 Создание и редактирование шагов бота и пользователя	20
2.4 Легкое и эффективное создание или обновление шагов	23
2.5 Использование бота	24
2.6 Логи программы бота	25
2.7 Роли менеджеров бота	25

Введение в установку и подключение

AgileBot 1.x - это программа или веб-приложение, которая подключается к учетной записи бота в Telegram и **состоит из 2 составляющих:**

- основной программы бота (программа бота и база данных бота)
- дополнительной программы бота (админ-панель бота).

Для установки, подключения и запуска программы AgileBot, необходимо:

1. Проверить сервер на требования работы программы Telegram-бота
2. Создать учетную запись бота в Telegram

После подключения к Telegram, бот AgileBot начинает работать. А дальше необходимо создавать или обновлять шаги бота и шаги пользователя.

Шаг бота - сообщение от бота пользователю или в канал, либо любое другое действие, выполняемое программой бота на сервере.

Шаг пользователя - это сообщение от пользователя боту или нажатие кнопки.

Требования для установки, подключения и запуска бота

1. Сервер должен быть веб-сервером.
2. К этому веб-серверу, подключена СУБД MySQL.
Например, основная СУБД для современных сайтов это MySQL.
3. Включен протокол HTTPS и есть SSL-сертификат для этого протокола.
Учетная запись бота в Telegram будет связываться с основной программой бота на веб-сервере по протоколу HTTPS.
Но для работы этого протокола, нужен официальный SSL-сертификат, полученный из центра сертификации.

Например: SSL-сертификат из центра GlobalSign.

Поэтому, чтобы программа Telegram-бота заработала, нужно чтобы был включен HTTPS-протокол и имелся SSL-сертификат!

Примечания:

1. *Веб-сервер - сервер предназначенный для работы сайтов или веб-приложений.*

Основная программа бота выполнена как веб-приложение, а база данных программы бота как база данных для СУБД MySQL.

- 2.. *Требования 2 и 3 заданы самим мессенджером Telegram.*

3. В качестве примера в инструкции берется новая папка на сайте, например папка на сайте компании.

4. *Описание работы, возможностей Telegram-ботов, их разработку, можно посмотреть на официальном сайте Telegram:*

<https://core.telegram.org/bots>

и на популярном неофициальном сайте Telegram:

<https://tlgrm.ru/docs/bots>

<https://tlgrm.ru/docs/bots/api>

5. *Подробно про SSL-сертификат и HTTPS-протокол можно прочитать:*

<https://help.reg.ru/support/ssl-sertifikaty/obshchaya-informatsiya-po-ssl>

1. Установка и подключение программы бота

1.1 Создание учетной записи бота в Telegram

Состоит из нескольких шагов, где **2 шага обязательные** (1.3 и 1.4):

1.1. Открыть специальный бот в Telegram, адрес бота - @BotFather, в котором и будет создаваться (регистрироваться) учетная запись бота:

1.2. Ввести или выбрать из меню команду: "/newbot".

Далее бот спросит (2 обязательных шага):

1.3. Имя бота в Telegram (поле "Имя" или "First Name")

1.4. Адрес бота, это то название, которое пишется после знака @

1.5. После задания адреса бота (1.4), присылается сообщение - что бот создан и даются идентификационные данные учетной записи (аккаунта) бота в Telegram (токен-ключ).

Примечание:

Этот токен-ключ надо обязательно записать, то есть шаг 1.5 - это запись токена-ключа.

Токен-ключ нужен для подключения программы бота к его учетной записи в Telegram!

Это как "логин и пароль", но только для подключения основной программы бота к его учетной записи в Telegram.

Имеет вид, пример: `110201543:AAHdqTcvCHlvGWJxfSeofSAsOK5PALDsaw`

Основная программа бота - это **файлы** бота в формате файлов для сайта (.php, .js, .css и др.) и **база данных** бота в СУБД MySQL.

Основная программа подключается к учетной записи бота через главный файл основной программы бота: bot.php.

Подключение главного файла в разделе 4 инструкции.

Дополнительное в программе бота - это **админ-панель** бота.

Визуальный веб-интерфейс для работы с ботом: создание и редактирование шагов бота и пользователей. Админ-панель работает через браузер.

Необязательные шаги, которые лучше сделать после создания аккаунта бота в Telegram:

1.6. Задать список команд:

1.6.1. Ввести команду: "/setcommands"

1.6.2. Написать список команд.

Введите следующий список команд, которые работают в конструкторе бота, написаны в формате записи команд:

start - старт бота

menu - меню бота

help - список команд

Формат записи одной команды:

название команды - краткое описание команды

1. Это **одна строка**,
2. Название команды пишется без слэша,
3. После записи строки, запятая не ставится (после краткого описания команды),

1.7. Задать краткое описание бота:

1.7.1. Ввести команду: "/setdescription"

1.7.2. Написать краткое описание бота - виде одной строки.

Например, оно будет показываться с названием бота (1.3) при открытии ссылки (URL-адреса) на бота в браузере, ссылка вида:

t.me/<адрес бота>, где <адрес бота> (1.4) в ссылке пишется без знака @.

Подробнее про создание учетной записи на неофициальном сайте:

<https://tlgrm.ru/docs/bots#botfather>

1.2. Установка программы Telegram-бота

Для установки программы и подключения Telegram-бота, необходимо::

1. **Комплект продукта** (zip-архив), из него:

- основная и дополнительная программа бота, файлы в папке *"/files"*,
- база данных бота, это файл *"basebot.sql"*.
(база данных составляющая основной программы)

2. **Доступы** (они доступны через админ-панель хостинга):

- к файлам и папкам на веб-сервере, где будет создаваться новая папка бота,
- через файловый менеджер (например программа "Проводник" в Windows), который предоставляется хостингом.
А также можно через программу файловый менеджер на компьютере по FTP-протоколу (то что называют FTP-доступом)
- к СУБД MySQL через программу "phpMyAdmin".

После установки (раздел 2), настройки (раздел 3) и подключения Telegram-бота (раздел 4), бот заработает и им можно будет пользоваться!

Шаги по установке программы Telegram-бота:

1.2.1. Установить основную программу бота на веб-сервер.

Установка основной программы - это просто копирование файлов.

1.2.1.1. Создать новую папку в корне сайта или другой вложенной папки на сайте

1.2.1.2. Скопировать файлы из папки *"/files"* в созданную новую папку.

На рисунке 1 показана корневая папка бота, после копирования.

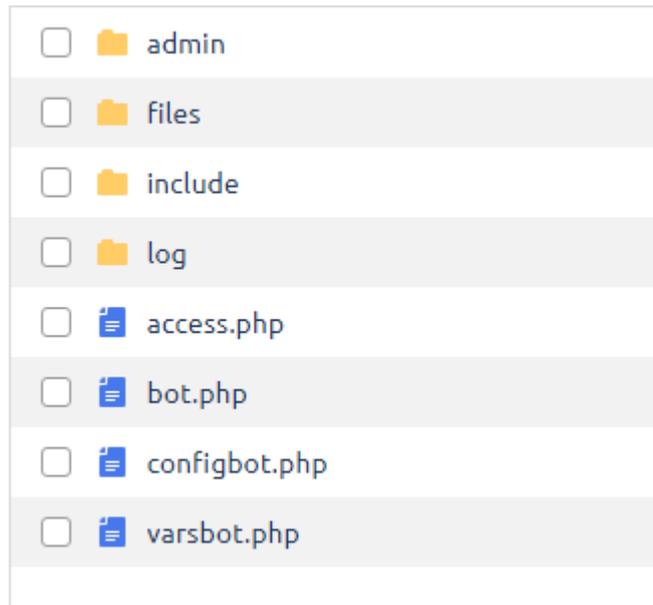


Рисунок 1. Файлы программы бота

1.2.2. Установка базы данных бота.

Установка базы данных - это установка дампа базы данных в виде одного файла в формате sql (*basebot.sql*).

1.2.2.1. Открыть панель хостинга, на котором находится сайт(открывается в браузере, например название "ISPmanager").

1.2.2.2. Создать новую базу данных: названия пишутся на английском языке.

1.2.2.2.1. Задать имя базы данных, имя любое,

1.2.2.2.2. Выбрать кодировку "utf8"!

Внимательно, выберите эту кодировку. Могут быть разные варианты с названием utf8, например "utf8mb4", надо выбрать просто "utf8"!

1.2.2.2.3. Задать имя пользователя, имя может быть любым.

1.2.2.2.4. Пароль пользователя любой (рекомендуется делать сложный от 9 символов). После выполнения шагов 2.2.2.1-2.2.2.4, создается пустая база данных.

1.2.2.3. Далее открыть веб-приложение phpMyAdmin.

Делается это через панель управления хостингом - найти пункт в меню

панели управления на веб-приложении phpMyAdmin или например такой пункт "Web-интерфейс БД"или если есть ссылка

(адрес) на phpMyAdmin например в письме при регистрации хостинга.

Скопировать адрес (ссылку) в адресную строку браузера и открыть.

Веб-приложение работает как админ-панель для работы с СУБД MySQL.

Примечание:

Админ-панель и панель управления - это все веб-приложения (две веб-программы).

Веб означает веб-формат предоставляемых данных - основная программа для получения данных в формате веб - браузер, например Chrome, Firefox, Yandex Browser, Safari, Edge.

1.2.2.4. Найти созданную базу данных. Для этого в веб-приложении в левой панели щелкнуть по названию базы данных.

В левой панели отображается список баз данных, которые есть у пользователя.

1.2.2.5. Открыть справа вкладку "Импорт",

1.2.2.5.1. Далее нажать кнопку "Выберите файл" и в появившемся окне выбрать файл дампа базы данных бота "*basebot.sql*" в папке на компьютере.

1.2.2.5.2. Проверить или выбрать кодировку "utf8". Остальные настройки импорта не трогаем.

1.2.2.5.3. Нажать кнопку "Вперед". После импорта дампа базы данных должна отобразиться фраза об успешном выполнении установки с зеленой подсветкой.

Зеленая подсветка как раз отражает успешное выполнение. Команда импорта выполняется как запросы к СУБД, множество запросов, которые после выполнения создают таблицы в этой базе данных.

Всё, **база данных** бота **установлена**.

Чтобы посмотреть щелкните слева снова по названию базы данных, откроются названия таблиц, первые 3:

"StepsBot" - таблица шагов бота,

"StepsChat" - таблица шагов пользователя,

"TableData" - таблица данных бота - данные, полученные от пользователей.

Ниже, на рисунке 2 представлены все таблицы базы данных бота, из веб-приложения (программы) phpMyAdmin.

<input type="checkbox"/>	StepsBot	★	Обзор	Структура
<input type="checkbox"/>	StepsChat	★	Обзор	Структура
<input type="checkbox"/>	TableData	★	Обзор	Структура
<input type="checkbox"/>	TableMode	★	Обзор	Структура
<input type="checkbox"/>	TableOptions	★	Обзор	Структура
<input type="checkbox"/>	TableSteps	★	Обзор	Структура
6 таблиц		Всего		

Рисунок 2. Таблицы база данных бота.

Примечание:

Создавать и редактировать шаги бота или пользователя также можно здесь! Админ-панель бота - это тоже работа с таблицами, но в простом и в более удобном виде для обычного пользователя и плюс есть справка как работать.

Всего таблиц в базе данных 6.

1.3. Настройка программы Telegram-бота

После копирования файлов на сервер и установки (импорта) дампа базы данных, надо:

- **обязательно** сделать настройки в файле настроек бота "*configbot.php*",
- не обязательно в файле "*varsbot.php*".

1.3.1. Открыть файл "*configbot.php*" через файловый менеджер в панели хостинга или перед копированием в папку бота на сервере, открыть файл на компьютере.

1.3.2. И заполнить значения переменных (переменная термин кода программы).

Формат заполнения:

- текстовые значения, пишутся в кавычках (это все значения, кроме одного),
- числовые значения, пишутся числом, без кавычек! (в файле одно такое значение - в переменной "***\$Chat_Id_AdminBot***")
- после написания значения обязательно стоит точка с запятой ";" (в файле они поставлены, просто проверьте наличие).

1.3.2.1. В переменную ***\$BotData*** после знака "=" и в кавычках написать - **токен (ключ)**, который получили после создания учетной записи бота в виде: *110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw*.

Далее проверяем стоит ли точка запятой ";" и в последующих ниже переменных также проверять ";" после написания значения.

3.2.2. В переменную ***\$DataBase*** написать - **название созданной базы данных** (пункт 2.2.2.1).

3.2.3. В переменную ***\$user*** - **имя пользователя** базы данных (2.2.2.3).

3.2.4. В переменную ***\$Password*** - **пароль пользователя** базы данных (2.2.2.4).

Примечание:

Также есть переменная ***\$serverbd*** - задает адрес до СУБД MySQL базы данных бота.

В основном на серверах где установлен веб-сервер, также установлена и СУБД, то есть на одном сервере, работает и веб-сервер и СУБД MySQL.

Поэтому в основном эту переменную не нужно задавать.

Значение задано по умолчанию: "127.0.0.1".

И веб-сервер на самом деле - это программа-служба.

3.2.5 В переменной **\$Users** задается логин и пароль пользователей (учетных записей пользователей), а также их роль.

Роль задает права доступа учетной записи к действиям в админ-панели и доступа к вкладкам и субвкладкам.

По умолчанию задано 3 учетные записи, с логинами: "Editor", "Manager", "Viewer".

Переменная **\$Users** представлена как массив массивов словарного типа, где каждый массив - это массив (данные учетной записи) для одного пользователя.

Поэтому значения для учетной записи пользователя задаются следующим образом:

в ключе массива с названием **Name** - пишется логин,
в ключе с названием **Password** - пароль,
в ключе с названием **Role** - роль.

Список ролей и их права доступа к действиям и к объектам админ-панели (вкладки и субвкладки) описаны в таблице "Роли и права доступа" подразделе 2.7.

Примечание:

Сами права доступа для ролей задаются в файле "access.php".

Пароли пользователей админ-панели бота, задавайте обязательно сложные (пароль длиной от 12 символов, буквы в нижнем и верхнем регистрах, с цифрами и спецсимволами).

Далее переменные, заполнение которых, может быть **необязательным**:

3.2.6. В переменную **\$PathBot** - URL-адрес до папки бота.

Начинается с "https://", заполнение необязательное, но если нужно отправлять изображения, тогда **обязательно!**

Это для типов шагов бота: "SendImage", "SendImageWithButtons".

3.2.7. В переменную **\$PathLog** - относительный путь до папки логов.

Задано значение по умолчанию, поэтому можно не заполнять.

Относительный путь - начинается от корневой папки бота.

Примечание:

Пишется вначале как точка со слэшем: *“./”*, например: *“./log”*.

Где точка означает путь начинать от текущей папки.

Корневая папка - это главная папка бота, в которой содержатся все остальные папки и файлы бота (или по-другому самая верхняя папка по уровню вложенности папок).

3.2.8. В переменную ***\$PathFile*** - **относительный путь до папки файлов**.

В этой папке будут файлы, которые нужны для записи данных от пользователей или любой заданной информации менеджером бота при выполнении типа шага бота: "SaveFile".

Задано значение по умолчанию, поэтому можно не заполнять.

3.2.9 В переменную ***\$FileLog*** - **относительный путь до файла лога** бота.

Задано значение по умолчанию.

Примечание:

До файла - это указание пути вместе самим файлом (пишется ещё название файла с расширением).

3.2.10 В переменную ***\$FileLogAdmin*** - **относительный путь до файла лога** пользователей админ панели.

3.2.11 В выражении в скобках, где строка:

“define(“ReportError”, true)” - задать разрешение на вывод сообщений об ошибках в чат бота в Телеграме при выполнении там шагов бота:

true - разрешить вывод,

false - не разрешить вывод.

Указываются значения *“true”*, *“false”* без кавычек!

По умолчанию, значение *“true”*.

Примечание:

Этот вывод (репорт) применяется при тестировании (отладки) шагов бота, после того как их (шаги бота) создали или редактировали (обновляли) в админ-панели бота.

Если точнее, то эти текстовые сообщения нужны для выявления ошибок в задании параметров шагов бота (которые задаются в поле “Дополнения” на субвкладках “Создание шага” и “Редактирование шага” во вкладке “Шаги бота”).

Если шаг выполняется верно, **то сообщение об ошибки не выводиться!**

Поэтому можно оставить **постоянно включенным этот параметр**.

3.2.12. В переменную **`$Chat_Id_AdminBot - Id`** главного менеджера бота. Является не обязательным. Сюда вводится Id пользователя Telegram, который будет главным менеджером ботом или просто менеджером бота. Менеджер бота - пользователь Telegram, который будет добавлять и редактировать шаги или просто менеджер, которому будут приходить данные от пользователей.

Для заполнения этой переменной нужно выполнить следующие шаги:

3.2.12.1. В строке где переменная убрать два слэша - `//`.

Это означает раскомментировать строку на языке PHP.

3.2.12.2. Написать значение в переменную **`$Chat_Id_AdminBot - Id`** **пользователя Telegram**, значение числовое, без кавычек (около 10 цифр).

Примечание:

Числовые значения в PHP задаются без кавычек!

Как узнать Id пользователя Telegram:

1. Через специального бота в Telegram: `@getmyid_bot`

2. Либо в "Таблице шагов" в столбце "Id пользователя".

Для этого этому пользователю надо сделать один шаг - написать сообщение боту или нажать на кнопку в чате, и этот шаг запишется в таблицу шагов, а также в лог-файл бота.

Все шаги пользователей и бота записываются в "Таблицу шагов" во вкладке "Информация" в адм и файл логов бота.

3. Посмотреть в файле лога бота:

найти массив с названием "from" и в нем найти параметр "id". Массив "from" находится в большой текстовой строке в лог-файле.

Примечание:

В этой строке данные, которые передаются из Telegram в программу бота: каждое сообщение от пользователя или действие, передается из Telegram в программу в виде большой текстовой строки (в виде несколько массивов данных, один из них массив "from" - данные от пользователя, который написал сообщение или например нажал на кнопку).

В терминах Telegram оно называется как "Update" (Обновления).

Этот термин для разработчиков Telegram-ботов.

4. Снова посмотреть в файле лога бота, только другое: найти параметр с названием "Chat Id", который расположен ниже за большой текстовой строкой.

3.2.13. Закрывать файл с сохранением изменений.

3.2.14 Если нужно отключить возможность записи шагов бота в таблицу шагов (субвкладка "Таблица шагов" во вкладке "Информация"), например

чтобы не записывались данные шагов пользователей (примечание 1.), такие как **Id, имя и фамилия пользователя**, то нужно выполнить 2 простых шага - закомментировать 2 раза одну строку кода в 2 местах файла "bot.php".

Подробнее:

3.2.14.1 Открыть файл "bot.php".

3.2.14.2 Найти 1-ую строку, в файле 507 строка:

```
$ModelBot->saveNextStepBot($Bot, $Data);
```

3.2.14.3 Поставить 2 слэша: "//" в начале этой строки, чтобы получилось:

```
//$ModelBot->saveNextStepBot($Bot, $Data);
```

3.2.14.4 Далее найти 2 такую же строку, в файле 556 строка.

3.2.14.5 И также поставить 2 слэша в начале строки, чтобы получилось:

```
//$ModelBot->saveNextStepBot($Bot, $Data);
```

3.2.14.6 Закрывать файл с сохранением изменений.

Примечания:

1. При записи шага бота в "Таблицу шагов", которое делается при каждом выполнении шага бота, также записываются в эту таблицу и данные шага пользователя, который запустил этот шаг бота.

2. Постановка 2 слэшей перед строкой кода - это называется комментирование кода в языке PHP. Закомментированный код не работает.

Файл *varsbot.php*.

Файл *varsbot.php* содержит:

- **текстовые сообщения ошибок** менеджера бота **при неверном вводе параметров типа шага бота** (параметры, которые задаются в поле "Дополнения").

Эти сообщения выводятся в чате бота с пользователем, если менеджер бота не правильно записал параметр, например неправильное имя параметра или значение параметра.

То есть как сказано было выше применяются при тестировании шагов бота и пользователя - когда шаги бота и пользователя через админ-панели были введены, и затем тестируются на правильную работу.

- **текстовые сообщения ошибок** пользователя **при неверном написании сообщений в самом чате** Телеграм бота, например вместо числа пользователь пишет ещё и буквы.

Примечание:

Особенность этих сообщений об ошибках, в том что они выводятся в напрямую чат при выполнении шага бота, что позволит сразу отследить

неверный заданный шаг бота.

Если шаг задан верно, то сообщения не будут появляться.

1.4. Подключение и запуск программы

Запуск Telegram-бота заключается просто в подключение главного файла бота (*bot.php*) на веб-сервере к учетной записи бота в Telegram:

1.4.1. Открыть браузер,

1.4.2. Написать:

<https://api.telegram.org/bot<token>/setWebHook?url=<addressbota>>

Подробнее на странице документации:

<https://tlgrm.ru/docs/bots/api#setwebhook>

<token> - токен(ключ) учетной записи бота (пункт 1.5),

<addressbota> - адрес до главного файла бота, это в папки бота файл "bot.php".

Пример: токен возьмем, который как пример, указан выше:

110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw.

Отдельная папка бота на сервере пусть будет иметь название "/bot", а название сайта компании "company.ru", тогда полный URL-адрес: (открытие этого URL-адреса в браузере - это как выполнение запроса на подключение):

<https://api.telegram.org/bot110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw/setWebHook?url=https://company.ru/bot/bot.php>

Пишем этот URL-адрес (2 выше строки это 1 строка) в браузере и открываем. На самом деле при открытии этого URL-адреса, на сайте Telegram выполняется скрипт, который подключает главный файл основной программы бота с сервером мессенджера Telegram!

При успешном подключении, бот (программа бота) **начинает работать!**

И теперь бот может принимать и отправлять сообщения через свою учетную запись в Telegram!

После открытия этого адреса в браузере (выполнения скрипта) откроется страница, в которой в одну строку будет написано сообщение на английском языке - (на самом деле это JSON массив), что WebHook установлен:

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

Это сообщение надо обязательно проверить! - Так оно не означает что бот работает и файл правильно подключен!

Для проверки подключения:

1. Надо перейти в чат бота в Телеграме и написать или выбрать команду: `"/start"`.
2. Если появиться сообщение от бота, начинающиеся со слов: "Добрый день! Это бот .." - значит подключение произведено!

Если сообщение в Телеграме не появилось, то нужно проверить, основное:

- **правильно заполнены переменные** в файле `"configbot.php"`, в особенности токен-ключ бота и переменные для подключения к БД бота;
- **правильно написан** в URL-запросе **полный путь до главного файла** и сам главный файл (**название файла**) в параметре `url`
- **установлен (импортирован) дамп базы данных** в СУБД MySQL (то есть сделаны и полностью шаги, которые в пункте 1.2.2 этого руководства).

2. Что дальше.

2.1 Админ-панель бота.

После установки и подключения бота к Телеграму, дальше будет создание или обновление (редактирование) шагов бота и шагов пользователя и через админ-панель бота в браузере.

Админ-панель доступна по адресу:

`<https://><название сайта>/<путь до главной папки бота>/login.php`

или

`<https://><название сайта>/<путь до главной папки бота>/adminpanel.php`

Например: если главная папка бота на сайте `"/bot"` и эта папка размещается на сайте компании `"https://company.ru/"`, то адрес то админ-панели будет следующий:

`https://company.ru/bot/login.php`

или

`https://company.ru/bot/adminpanel.php`

На рисунке 3 представлена часть админ-панели бота - вкладки, субвкладки и меню. Где вкладки - зеленого цвета, субвкладки - синего цвета.



Рисунок 3. Админ-панель бота. Вкладки и субвкладки

2.2 Модель бота или как работает основная программа бота

Основная программа бота работает следующим образом:

1) Пользователь Телеграма отправляет сообщения (определенного типа) в чат боту.

3 типа сообщений:

1. Текстовые (текстовое сообщение – текстовая команда).

2. Кнопочный (нажатие кнопки)

Режим кнопки включается, если написать текстовые команды:

"режим кнопки" или "бот кнопки".

3. Командный (текстовое сообщение - виде стандартной команды Телеграм-бота на английском языке через знак слэш "/", например: "/start").

2) Программа бота принимает их и обрабатывает (определяет что ему нужно сделать).

3) И далее выполняет заданное определенное действие на сообщение от пользователя.

Например:

- отправляет сообщение с определенной информацией обратно пользователю в чат.
- или выполняет определенное действие на стороне сервере, где работает сама программа бота, например:
 - записывает сообщение полученное от пользователя в БД бота
 - или проводит математический расчет по заданной формуле,
 - или отправляет обычное письмо на E-mail с содержимым полученным от пользователя.

То, что пишет пользователь, называется в модели программы бота - **шаг пользователя**.

То, что выполняет бот на шаг пользователя, называется в модели программы бота - **шаг бота**.

Поэтому работа с ботом заключается просто в создании шагов бота и пользователя и в создании связи между ними!

Где связь между ними - определенная логика и последовательность выполнения шагов бота на шаги пользователей, что в стандартном обычном виде для ботов выглядит как **ветки** по которым работает

бот и пользователь.

2.3 Создание и редактирование шагов бота и пользователя

Создание шага бота - это просто заполнение одной строки в таблице "Шагов бота".

В зависимости от типа шага бота заполняются **3** или **4 поля** строки.

Где, главные первые 2 поля:

“Шаг” - задается номер нового шага бота,

“Тип шага” - задается тип выполняемого нового шага бота (простая функция, которую выполняет бот).

Можно на шаг пользователя - создать цепочку из нескольких последовательных шагов бота, тогда получится более сложная функция, выполняемая ботом на 1 шаг пользователя.

Примеры:

Для записи данного от получаемого от пользователя надо использовать 2 шага построенных в последовательную цепочку:

сначала шаг бота с типом “Получить данное от пользователя”, затем шаг бота с типом “Записать данное в БД”.

Но можно использовать и 1 шаг вместо 2-х, это использование типа шага бота: “Получение данного от кнопки с сообщением”.

В таблице шагов как цепочка из 2 шагов это примеры шагов под номерами 102-103, как 1 шаг это пример шага под номером 118.

Создание шага пользователя - это заполнение одной строки в таблице "Шаги пользователей".

Заполняются **все 3 поля** строки шага пользователя.

2.4 Легкое и эффективное создание или обновление шагов

Для **более легкого, удобного и эффективного** создания или обновления (редактирования) шагов бота или пользователя и просто работы в админ-панели применяется или сделано следующие:

1. Примеры шагов бота и пользователя:
шаги бота под номерами 1-4 и 100-124,
шаги пользователей с Id 1-16 и 43-44.

Просматривая которые, можно **легко создавать** свои.

2. Вкладок в админ-панели всего 4 и для создания и редактирования нужны всего 2, которые соответственно так и называются: “Шаги бота” и “Шаги пользователя”.

3. В каждой из них есть 3 субвкладке, которые соответствуют необходимым действиям при работе с шагами и имеют соответствующие названия этим действиям, субвкладки:
“Шаги ..” - субвкладка для просмотра шагов бота или пользователей.
“Редактирование шага” - для редактирования или удаления одного шага.
“Создание шага” - субвкладка для создания одного шага.

4. Небольшое количество полей для создания шага бота или пользователя:

4.1 для создания или редактирования шага бота нужно заполнить 3 или 4 поля

4.2 для создания или редактирования шага пользователя нужно заполнить обязательно 3 поля.

5. Выделение области удаления шага бота или пользователя от области редактирования в субвкладке “Редактирование шага” через линию красного цвета.

6. Вывод сообщений:

6.1 При создании или редактировании шагов бота и пользователя, а также при неверном вводе значений полей шагов и параметров поля “Дополнения” в шаге бота виде всплывающих сообщений зеленого цвета. Где зеленый цвет означает - удачное выполнение действия в админ-панели.

6.2 При не успешном выполнении действия в админ-панели, вывод всплывающих сообщений красного цвета.

Которые отображают подробно (**указывают дополнительно**) из-за чего не получается выполнить. Например: не заполнено обязательное поле и название соответствующего поля.

Примечание:

Вывод сообщений сделан почти ко всем полям шагов бота и пользователя и параметрам типа шага бота.

Параметры типа шага бота - это просто дополнительные поля шага бота, но для более легкого заполнения при создании или изменении при редактировании шага бота с определенным типом шага, переведены в отдельное поле "Дополнения" и названы как **параметры** типа шага бота.

7. Разграничение доступа для учетных записей по выполняемым действиям (создание, редактирование и удаление шагов) и по доступу к объектам (вкладкам и субвкладкам).

Разграничение **определяется по назначаемой роли** для учетной записи пользователя админ-панели.

По умолчанию в админ-панели задано 3 учетной записи: "Editor", "Manager", "Viewer", с назначенными соответственно со следующими ролями: "editor1", "manager2", "viewer2".

Подробно и **таблица ролей и с наделенными правами** приведена в подразделе 2.2 "Настройка подключения программы бота".

Примечание:

Названия или по другому логины пользователей здесь совпадают с названиями ролей. Сделано для того чтобы показать наделенную роль.

2.5 Использование бота

После создания шагов бота и пользователей, ботом можно пользоваться в Телеграме.

Дальнейшая работа с ботом в админ-панели заключается:

1. В редактировании (обновлении) шагов бота и пользователя или в создании новых шагов,
2. В просмотре в работе с информацией, которую собирает бот через вкладку “Информация”:

2.1 Данные получаемые от пользователей - субвкл. “Таблица данных”, например при создании опроса или заявки в Телеграм-боте, записывается текстовый ответ или название нажатой кнопки пользователя (*примечание 1*).

2.2 Данные записанных шагов бота и пользователей - субвкл. “Таблица шагов” (*примечание 3*):

- **Время, Номер шага** и **Тип** выполняемого шага бота,
- Текст **сообщения или нажатой кнопки** шага пользователя, который вызвал этот шаг бота, а также **Имя и Фамилия** пользователя.

2.3 Режимы, которые включают пользователи - субвкл. “Таблица режимов” (*примечание 3*).

Примечания:

1. Данное получаемого от пользователя можно сделать не только как текстовое небольшое сообщение, которое например напишет пользователь на вопрос от бота, но и через нажатие кнопки, где получаемое данное будет название кнопки.

Чтобы это сделать, нужно применить тип шага бота: “Получение данного от кнопки с сообщением”, в примерах шагов бота, этот шаг под номером 118.

2. Так как в субвкладке “Таблица шагов” записываются данные шагов пользователей, такие как **Id, имя и фамилия** пользователя в Телеграме, то возможно будет необходимо **отключить возможность записи данных пользователей** - для этого нужно **закомментировать 2 повторяющиеся строки кода** в файле “bot.php”.

Подробно как это сделать написано в подразделе в пункте настройки .

3. Субвкладка. “Таблицы режимов” является менее используемой для менеджеров бота, так как включение режимов: “текстовый” и “кнопочный” нужны показать пользователям, объяснить пользователям.

А так как пользователи Телеграма используют кнопки или стандартные команды Телеграм-бота (пишутся на английском и через знак слэша “/”, в

основном будет использоваться кнопочный режим и таблица режимов будет не нужна в работе менеджеру бота.

По умолчанию в боте включен постоянно “кнопочный” режим.

Текстовый - подразумевает что пользователь шагает в боте (или пользуется ботом) через текстовые команды (слова или выражения переходы), а не через кнопки или стандартные команды Телеграм бота.

Слова или **выражения переходы** (или по другому это **есть шаги пользователя**), повторяют названия (слова или выражения) написанные на кнопках.

2.6 Логи бота

В программе бота ведется **2 файла логов**:

- логи **работы основной программы** бота (шаги бота и шаги пользователей).
- логи **работы в админ-панели** менеджеров бота (действий: открытия админ-панели, создания или обновлении шага и авторизация менеджеров).

Админ-панель - это дополнительная часть программы бота.

2.7 Роли менеджеров бота

В таблице “Роли и права доступа” представлены роли и их права на действия и на доступ к объектам (вкладкам и субвкладкам).

Таблица “Роли и права доступа”

№	Роль	Права				
		Создавать и редактировать шаги бота и пользователей	Удалять шаги бота и пользователей	Доступ к вкладкам “Шаги бота” и “Шаги пользователей”	Доступ к вкладке “Информация”, кроме субвкладке “Таблица шагов”	Доступ к субвкладке “Таблица шагов” и вкладке “Параметры бота”
1	viewer1	-	-	+	+	-
2	viewer2	-	-	+	-	-
3	manager1	+	-	+	+	+
4	manager2	+	-	+	-	-
5	editor1	+	+	+	+	+
6	editor2	+	+	+	-	-