Установка и администрирование сервера бота Штрих-Помощник

Версия 1.0

(редакция от 07.03.2022)

Содержание

Функционал

Версия 1.0

1. Управление пользователями (создание, изменение, удаление)
2. Управление складами (создание, изменение, удаление)
3. Управление кассам (создание, изменение, удаление)
4. Привязка нескольких менеджеров(пользователей) к одному складу
5. Привязка нескольких менеджеров(пользователей) к одной кассе
6. Привязка нескольких касс в одному складу
7. Привязка нескольких складов в одной кассе
8. Привязка одной кассы сразу к нескольким складам
9. Разграничение прав менеджеров(пользователей)
10. Запрос права на совершение операции на кассе через telegram
11. Отправка срочных оповещений от кассы в telegram
12. Запрос из telegram оперативных показателей с кассы

Версия 1.0.1

1. Исправлена ошибка при удалении менеджера кассы
2. Исправлена ошибка сохранения прав пользователя
3. Исправлено произвольное закрытие запроса права

Версия 1.0.2

1. Поставка сервера осуществляется сразу с необходимыми зависимостями
2. В панели администрирования на форму кассы добавлена кнопка “Очистить” для очистки текущего **cashbox-id** кассы

Установка

Требования к хосту

Для работы бота необходимо соблюсти следующие требования:

1. Машина, на которой будет разворачиваться сервер, должна иметь доступ в интернет (достаточным будет возможность взаимодействовать с серверами Telegram)
2. Машина, на которой будет разворачиваться сервер, должна быть доступна по сети всем кассами, с которыми надо будет взаимодействовать

Требования к программному обеспечению

NodeJS

Данный сервер может быть установлен как на машине с операционной системой (далее ОС) **Windows**, так и на машину с ОС **Linux**. Главное, чтобы ОС поддерживала установку среды выполнения **Node JS** версии **14.17.6 LTS**.

Обратите внимание!

Работоспособность сервера может гарантироваться только на основе среды выполнения **Node JS** версии **14.17.6 LTS**.

Для установки сервера и его работы в ОС **Windows** или **Linux** необходимо установить дополнительное программное обеспечение (далее ПО), а именно среду выполнения **Node JS.**

Установщик среды выполнения **Node JS** для ОС **Windows** можно получить бесплатно на официальном сайте https://nodejs.org/en/.

При установке обязательно надо проследить за тем, чтобы вместе со средой выполнения был установлен пакетный менеджер **NPM,** а также чтобы пути к бинарным файлам **“node”** и **“npm”** были добавлены в переменную **PATH**. Далее на скриншоте показаны опции, которые следует выбрать при установке **Node JS** (в данном случае, на скриншоте приведен пример установки **Node JS** версии **14.17.6 LTS**, поэтому выбраны все опции установки, в других версиях перечень компонентов для установки может быть расширен разработчиками **Node JS**)

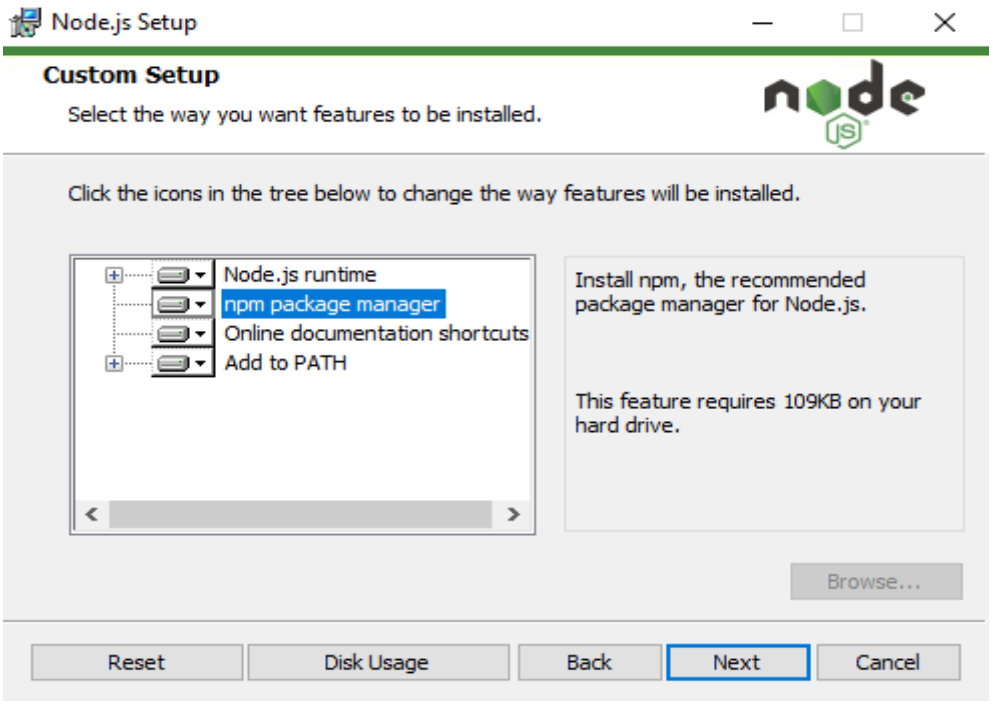


Рис 1. Выбранные опции установки среды выполнения **Node JS**

Установка среды выполнения **Node JS** на ОС **Linux** может отличаться в зависимости от конкретного дистрибутива, но информацию об установке также можно получить на официальном сайте https://nodejs.org/en/.

Система управления базами данных

Для работы бота необходимо также обеспечить средства для структурированного хранения данных.

Сервер бота может работать со следующими системами управления базами данных (далее СУБД): **PostgreSQL** (версии 10 и выше) и **SQLite** (версии 3).

Для работы бота с **PostgreSQL** необходимо самостоятельно установить соответствующую СУБД или быть готовым предоставить серверу доступ к удаленной СУБД.

Для работы с **SQLite** ничего дополнительно ставить не надо, так как при установке сервера все необходимое ПО для работы с **SQLite** будет автоматически установлено при выполнении скрипта установщика.

Конфигурационный файл

Перед установкой непосредственно самого сервера необходимо подготовить конфигурационный файл. Конфигурационный файл должен иметь следующее наименование **“.env”** (без кавычек и с точкой вначале) и его содержимое должно храниться с кодировкой **UTF-8**.

Конфигурационный файл размещается в директории установки сервера (точно в той же папке, где лежит пример конфигурационного файла **.env-example**).

Параметры конфигурационного файла

BOT_TOKEN

Значение данного параметра используется сервером для связи с Telegram-ботом и управление им. Без данного параметра работа бота **НЕВОЗМОЖНА**. Процесс получения такого токена описан в приложении **Регистрация бота в telegram и получение его токена**.

DEBUG

Параметр отвечающий за вывод логов. Если необходимо выводить логи работы сервера, то данному параметру необходимо задать значение “*****”(звездочка, без кавычек).

DATABASE_TYPE

Параметр отвечает за то с какой СУБД будет работать сервер бота. Возможные значения: **SQLITE** и **POSTGRE**. По умолчанию значение этого параметра - **SQLITE**.

DATABASE_PATH

Параметр отвечает за то, где сервер бота будет искать или разворачивать (при установке) файл базы данных **SQLite**. Поддерживается относительный и абсолютный формат описания пути, кроме сетевого, то есть файл базы данных должен находиться на той же машине, где будет работать сам сервер.

Если по указанному адресу файла базы данных нет, то при установке сервера он будет создан.

Если при работе по указанному адресу файла базы данных не будет, то сервер остановит выполнение с ошибкой.

Значение параметра игнорируется, если значение параметра **DATABASE_TYPE** не равно **SQLITE**.

По умолчанию значение этого параметра - “./main” (без кавычек)

DATABASE_URL

Параметр отвечает за доступ сервера бота к СУБД **PostgreSQL**. Значение параметра задается по следующему шаблону:
postgresql://[PGUSER[:PGPASSWORD]@][PGHOST][:PGPORT][/PGDATABASE], где

PGUSER - имя пользователя на сервере **PostgreSQL**

PGPASSWORD - соответствующий пароль данного пользователя

PGHOST - сетевой адрес сервера (если сервер запущен на той же машине, где будет работать сервер бота, то подставляется значение **localhost**)

PGPORT - порт на котором сервер **PostgreSQL** ожидает подключение

PGDATABASE - наименование базы выделенной для работы сервера бота. Перед установкой сервера бота база с указанным наименованием должна существовать на сервере и быть пустой - без таблиц, иначе сервер бота не сможет развернуть свои таблицы.

Пример значения: **postgres://postgres_user:postgres_pass@localhost:5432/test**

PORT

Параметр задает порт, который сервер бота будет прослушивать.

По умолчанию, порт 4000.

REQUEST_TIMEOUT

Параметр задает время жизни запросов от кассы.

Если в очередь запросов к менеджеру добавляется новый запрос, при наличии запроса ожидающего решения менеджера, то проверяется сколько времени ожидающий запрос находится в очереди и, если это время превысило значение заданное этим параметром, то ожидающий запрос из очереди удаляется с присвоением ему результата по умолчанию, после чего менеджеру в чат отправляется следующий запрос.

Значение задается в секундах. По умолчанию - 300.

ADMINPASSWORD

Параметр задает пароль администратора для доступа к серверу бота через панель администратора.

По умолчанию пароль - 123456

Пример содержимого заполненного конфигурационного файла

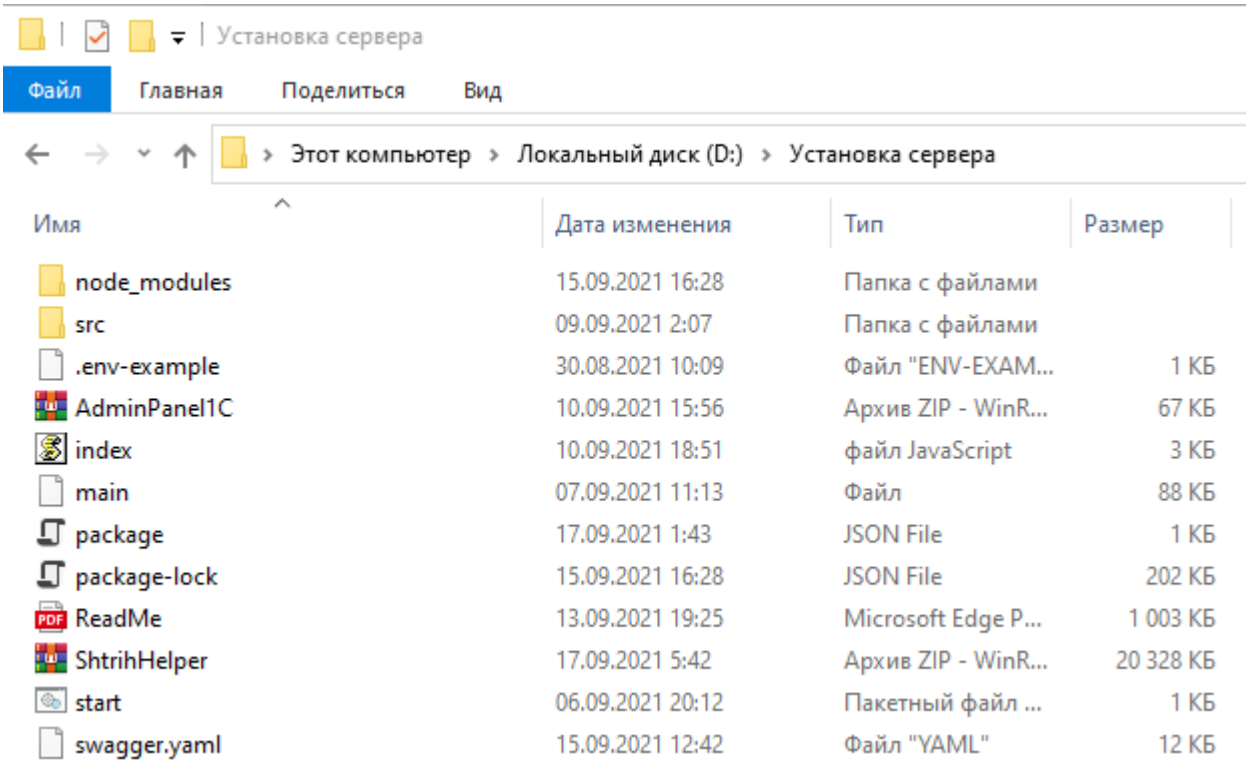
Процесс установки

Пошаговая инструкция

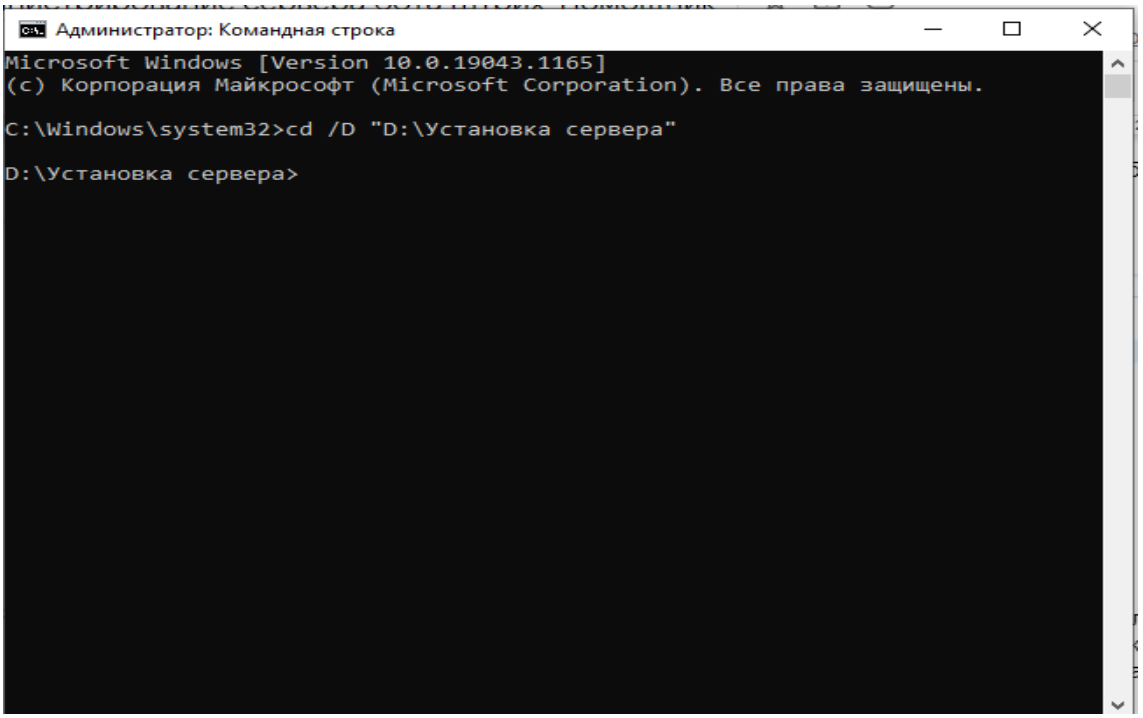
Обратите внимание!

1. Перед началом установки необходимо установить среду выполнения **Node JS** и пакетный менеджер **NPM** (подробно об этом говорится в разделе “**Требования к программному обеспечения. NodeJS**”).
2. Для управления **ОДНИМ** ботом необходимо запустить **ОДИН** и только **ОДИН** экземпляр сервера. Если будет запущено несколько экземпляром сервера, при этом настроенных на управление одного и того же бота, то управление этим ботом будет осуществляться только одним экземпляром - запущенным позже всех остальных.
3. Если имеется необходимость управления несколькими кассами, находящимися в разных сетях, то придется:
    a. либо настраивать между всеми этими кассами и сервером **VPN**
    b. либо осуществлять развертывание сервера на машине с публичным (“белым”) IP-адресом
    c. либо воспользоваться хостингом, который поддерживает развертывание приложений на базе **NodeJS**. Например, **Heroku (**при этом использовать желательно будет только вариант развертывания на основе **PostgreSQL).**

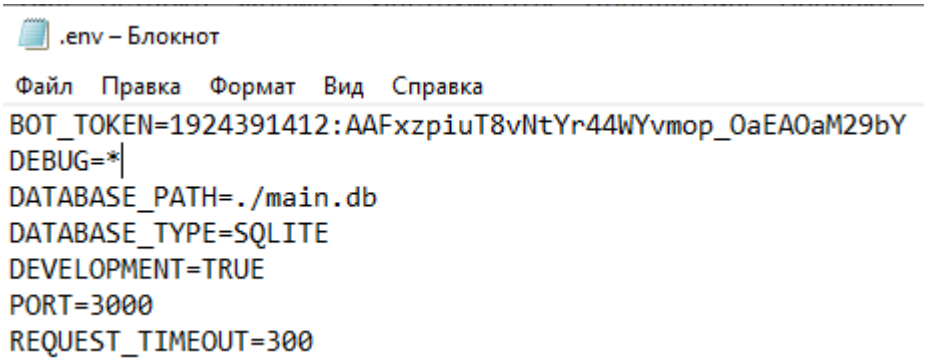
1. Распаковываем архив **ShtrihHelper.zip** в любую удобную папку для установки. Должно получиться как на скриншоте далее.



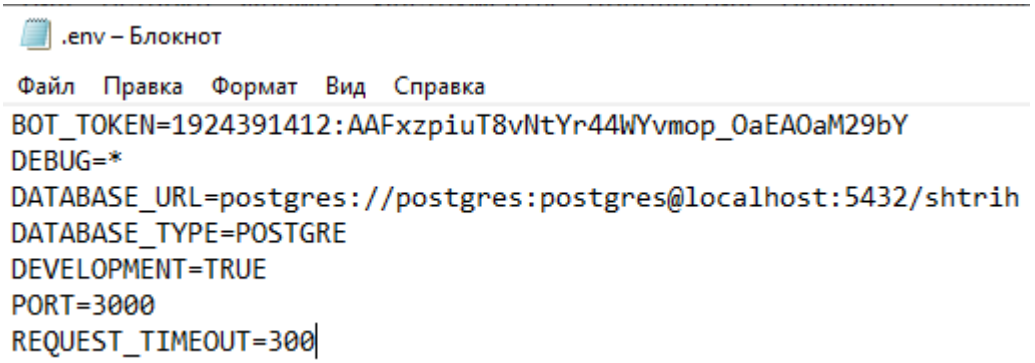
2. Запускаем командную строку от имени Администратора (или терминал, если установка производится на **Linux**) и переходим в папку, куда ранее распаковали архив **ShtrihHelper.zip**. В примере распаковка производилась в директорию “**D:\Установка сервера**”.



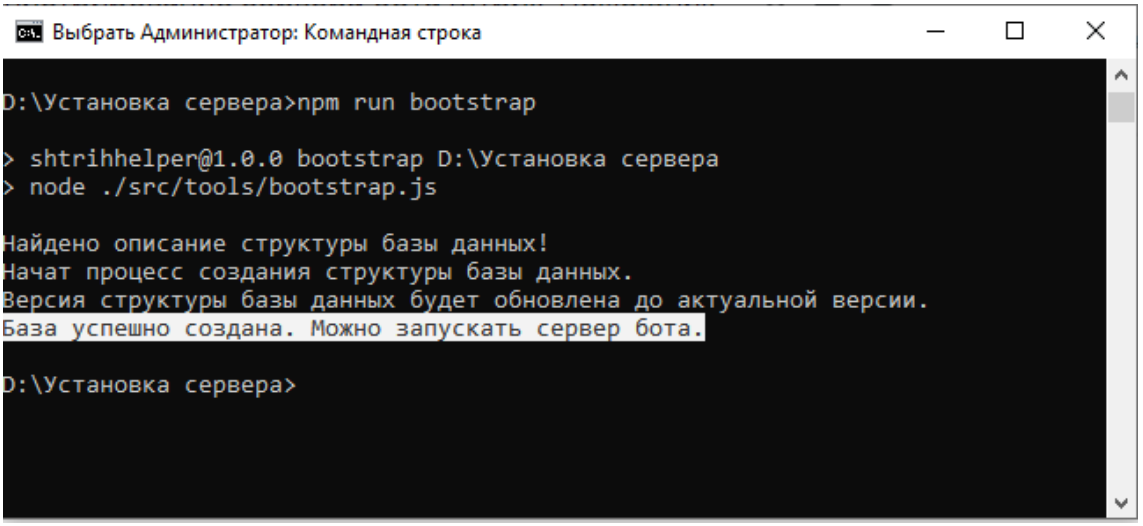
3. Перед следующим шагом надо создать конфигурационный файл, о котором говорилось в предыдущем разделе. Для установки из данного примера конфигурационный файл будет иметь следующее содержание:



так как установка и работа сервера в данном примере будет производиться на основе базы данных **SQLite**, то параметр **DATABASE_URL** нет необходимости настраивать. Если работа сервера будет основана на **PostgreSQL**, то пустая база уже должна быть создана на сервере **PostgreSQL**, путь к ней должен быть настроен параметром **DATABASE_URL**, а значение параметра **DATABASE_TYPE** должен быть заполнен значением **POSTGRE**, как в примере далее



4. Запускаем команду создания структуры базы данных - “**npm run bootstrap**” (без кавычек). Если создание структуры базы данных прошло успешно, то вы должны увидеть сообщение об этом, как на скриншоте далее:



5. Можно считать, что сервер установлен и остается его только запустить.

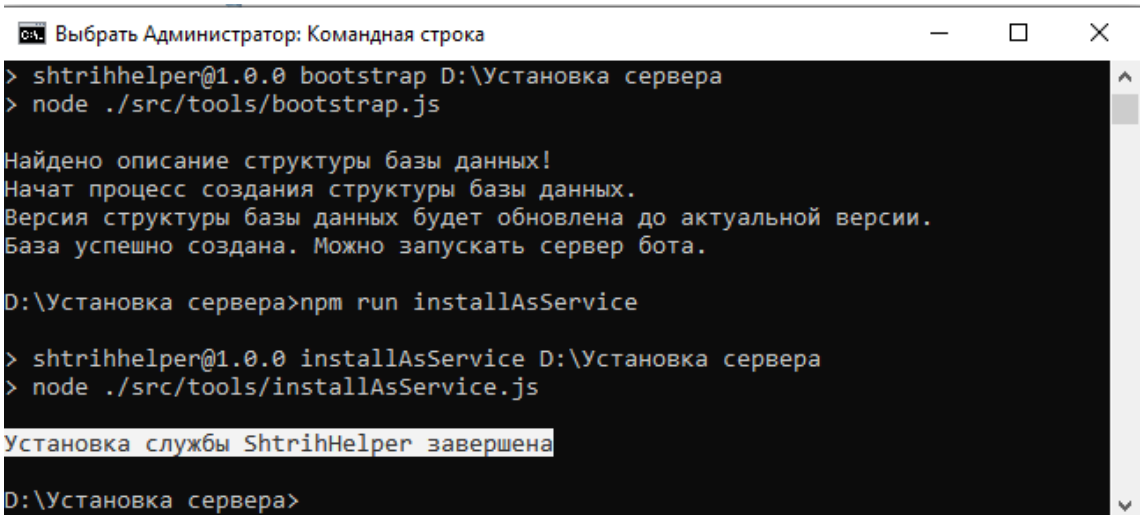
Установка в виде службы Windows

При установке сервера на **Windows** возможна установка сервера в виде службы **Windows**. Для того, чтобы установить сервер в виде службы необходимо после его установки, выполнить команду **“npm run installAsService”** из командной строки, запущенной от имени администратора, находясь в той директории, в которой был установлен сервер.

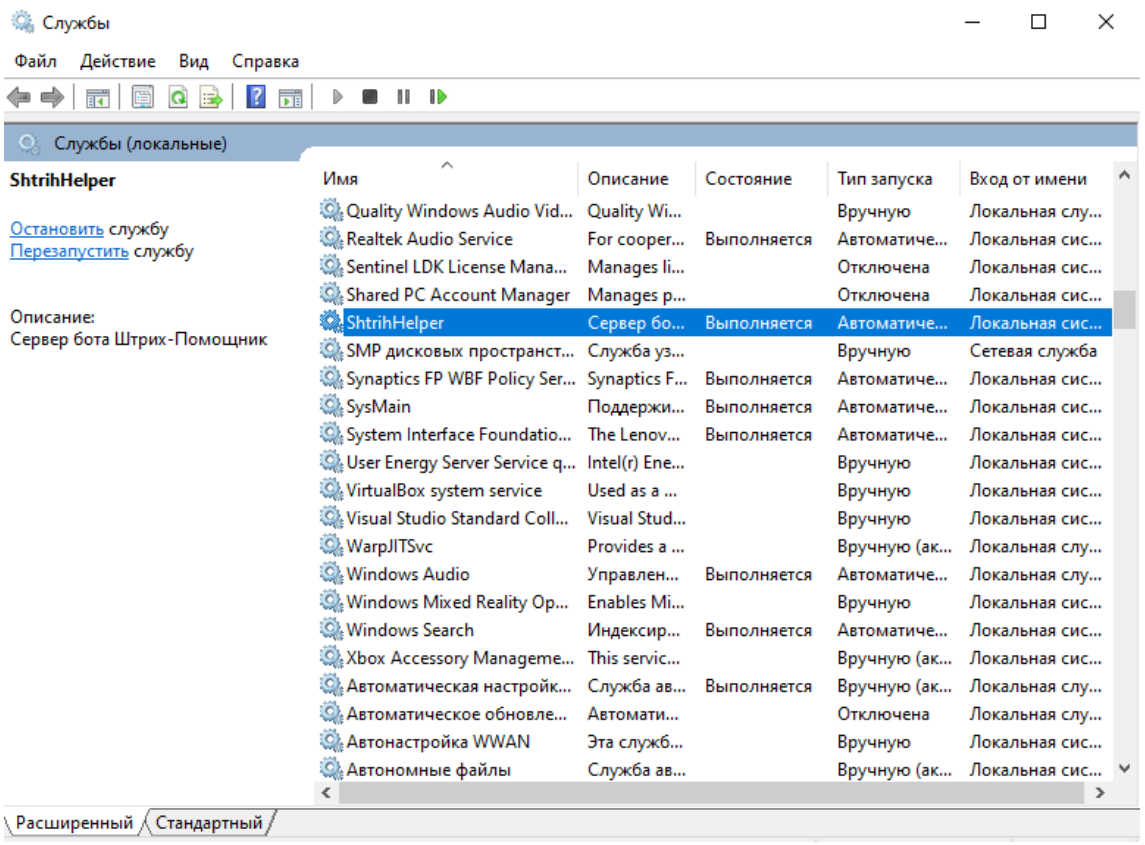
Обратите внимание!

Установка сервера в качестве службы выполняется с использованием программы **WinSW** версии **2**. Ручная установка этой программы не требуется, так как она имеется в пакете поставки этого сервера, однако для работы этой программы необходимо удовлетворить требования для ее работы с которыми можно ознакомится по ссылке **https://www.nuget.org/packages/WinSW/**. Ознакомьтесь с требованиями этой программы и обеспечьте возможность для ее работы на той машине, на которой вы собираетесь разворачивать сервер в виде службы.

Если установка пройдет успешно, то в консоле вы увидите сообщение, выделенное на скриншоте далее:



Также в панели управления службами Windows появится служба “ShtrihHelper”, как на скрине далее:



Обратите внимание!

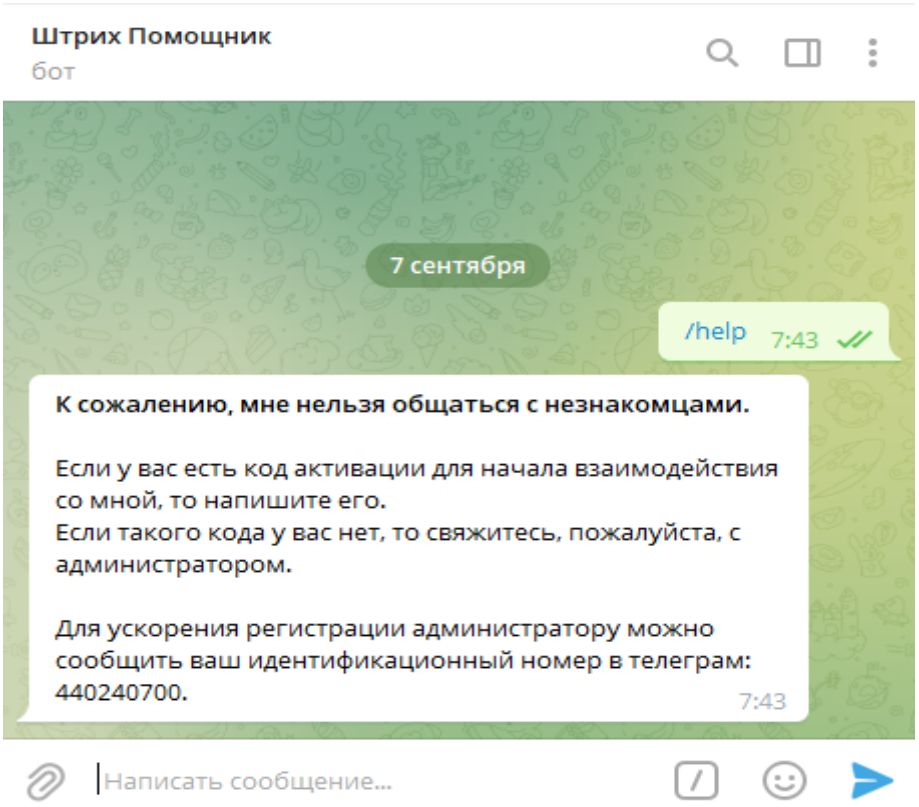
После установки сервера в качестве службы изменять расположение директории установки нельзя, иначе служба перестанет работать. Если есть необходимость в том, чтобы перенести директорию установки сервера, то сначала надо выполнить деинсталляцию службы (это выполняется командой **“npm run uninstallService”** из командной строки, запущенной от имени администратора, находясь в той директории, в которой в данный момент установлен сервер), затем перенести **ВСЕ** содержимое директории установки сервера в новую директорию и запустить установку службы заново.

Запуск сервера

Если сервера установлен в виде службы **Windows**, то управлять запуском и остановкой сервера можно прямо из панели управления службами **Windows**.

Если же сервер не был установлен в виде службы, то запускать его надо вручную командной строки с правами администратора командой **“npm start”** из директории установки сервера. Закрытие окна командной строки, в которой была введена команда запуска сервера, приведет к завершению работы сервера.

В случае, если весь процесс установки прошел успешно и конфигурационный файл был правильно настроен (особое внимание надо уделить токену бота, который должен быть получен из **telegram**), то бот, к которому должен быть привязан сервер начнет отвечать на сообщения, как на скрине далее:



Далее остается в панели администрирования бота зарегистрировать пользователей telegram, с которыми бот может взаимодействовать. Об это подробно будет описано в разделе **Администрирование**.

Запуск сервера под ОС Linux

Запуск сервера на ОС **Linux** осуществляется выполнением команды “**npm start**” (без кавычек) в терминале из директории, в которой установлен сервер.

Обновление сервера

Для обновления сервера необходимо распаковать содержимое обновленного архива с файлами сервера в директории, ранее уже установленного сервера, при этом работа сервера должна быть приостановлена.

Перед запуском обновленного сервера также необходимо запустить обновление структуры базы данных.

Делается это следующим образом: запускаем команду создания(эта команда отвечает также и за обновление) структуры базы данных - “**npm run bootstrap**” (без кавычек) из командной строки, запущенной от имени администратора, находясь в той директории, в которой в данный момент установлен сервер. Если обновление структуры базы данных прошло успешно, то вы должны увидеть сообщение об этом - “**Версия структуры базы данных актуальна. Обновление не требуется.**”.

Администрирование

Для администрирования сервера бота используется “**Панель администрирования**” выполненная в виде конфигурации **1С**. ZIP-архив с конфигурацией панели администрирования **AdminPanel1C.zip** хранится в архиве с установочными файлами сервера **ShtrihHelper.zip**. Для начала работы с панелью администрирования необходимо распаковать содержимое архива **AdminPanel1C.zip** в любую директорию, а затем добавить существующую базу в **1С** используя “**Добавление в список существующей информационной базы**” указав путь до директории, в которую вы распаковали содержимое архива **AdminPanel1C.zip**.

Обновление панели администрирования

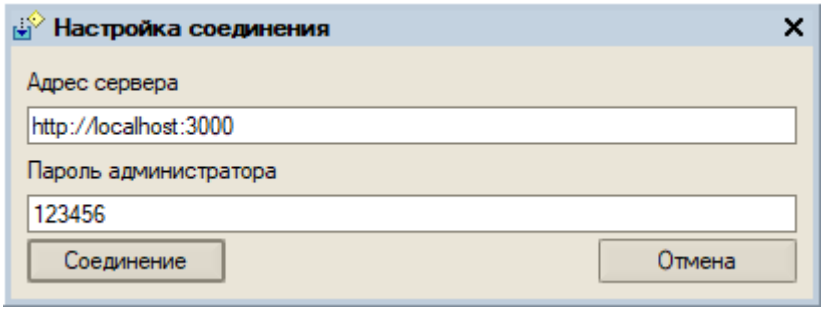
Из нового архива поставки сервера **ShtrihHelper.zip** надо взять содержимое архива **AdminPanel1C.zip** и заменить им содержимое папки, в которой в данный момент установлена конфигурация панели администрирования.

Обратите внимание!

Никаких данных в самой конфигурации не хранится, это просто интерфейс для взаимодействия с базой сервера бота, поэтому можно смело заменять файл **1Cv8.1CD**.

Запуск панели администрирования

При запуске конфигурации панели администрирования первое что, вы увидите - это форма настройки соединения (см. скриншот далее), в котором вам необходимо указать адрес и порт машины, на которой **ЗАПУЩЕН** сервер бота, а также пароль администратора, который был указан в конфигурационном файле (если пароль не будет установлен вручную в конфигурационном файле, то пароль по умолчанию - 123456). Так как в рамках примера сервер и панель администрирования работают на одной и той же машине, то параметры подключения будут выглядеть так, как изображено на скриншоте далее:

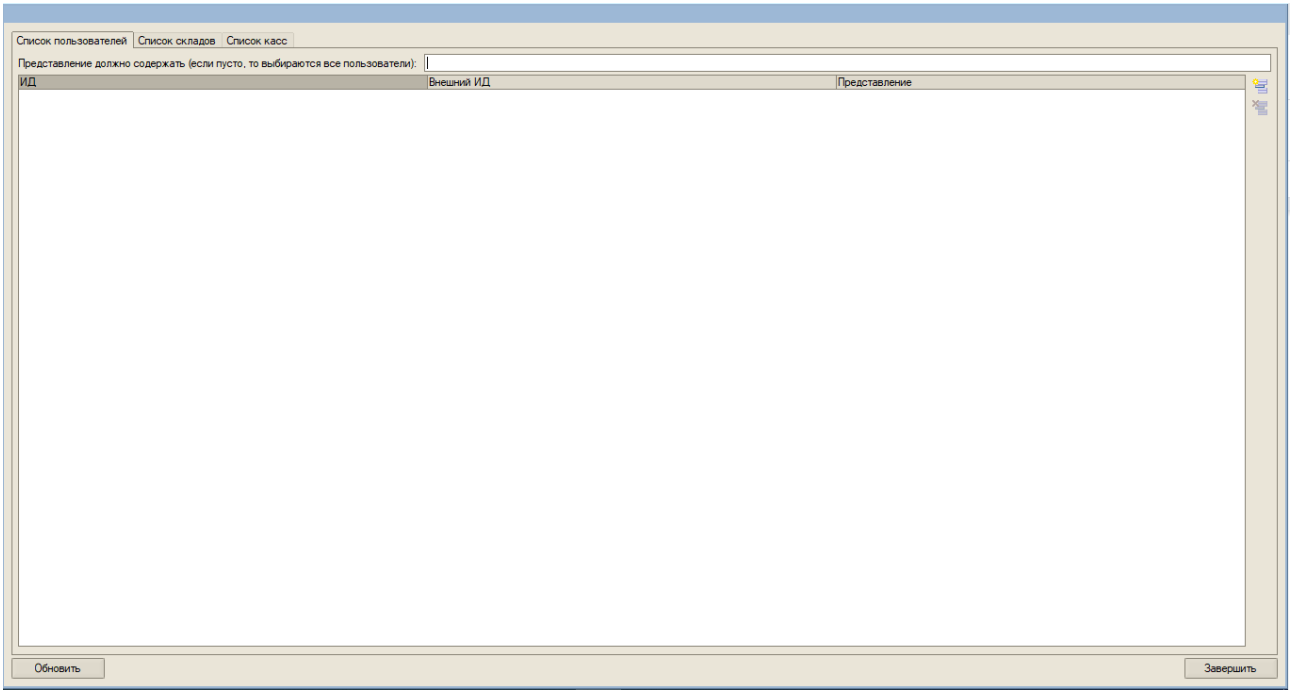


Обратите внимание!

Указание протокола **http:// ОБЯЗАТЕЛЬНО!**

При нажатии на кнопку **Отмена** будет предложено завершение работы системы, так как без прохождения аутентификации работа в панели администрирования невозможна!

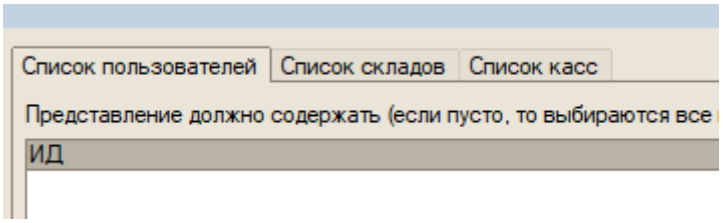
При нажатии **Соединение** проверяется содержимое полей ввода (все поля ввода на этой форме должны быть заполнены, желательно, адекватно), затем проверяется связь с сервером и если пароль верный и связь с сервером удалось установить, то вам сообщают об этом соответствующим сообщением “**Аутентификация прошла успешно**” и далее запускается сама панель администрирования во весь экран (см. скриншот далее).



На этой форме имеются три вкладки (см. скриншот далее) - “Список пользователей”, “Список складов”, “Список касс”, а также кнопки **Обновить** и **Завершить.**

По нажатии кнопки **Обновить** списки будут принудительно обновлены.

По нажатии кнопки **Завершить** будет предложено завершить работу программы панели администрирования.



Управление пользователями

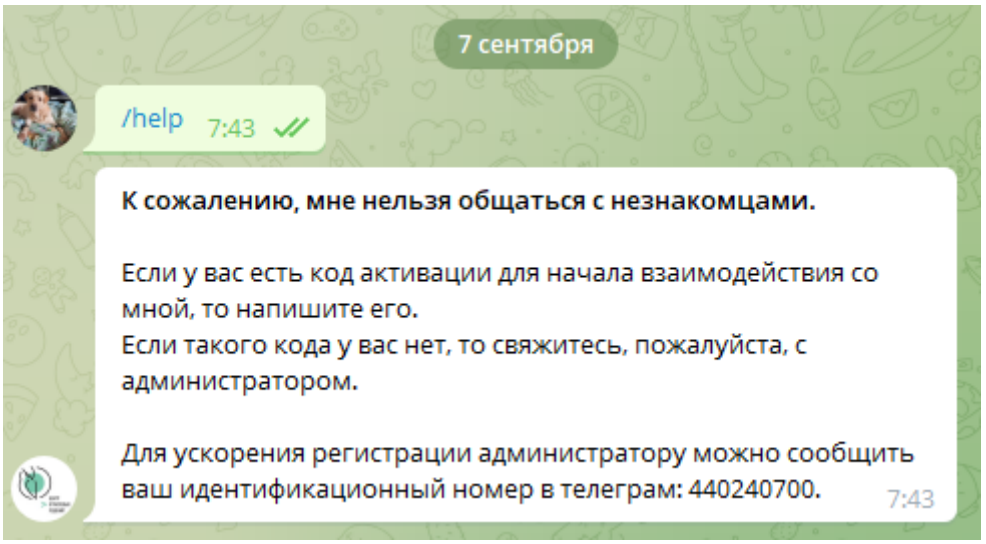
Из панели администрирования можно регистрировать новых пользователей и удалять текущих.

Регистрация пользователя

Регистрация нового пользователя может быть осуществлена двумя путями.

Первый путь доступен если вам известен уникальный идентификатор telegram-пользователя. Тогда при регистрации пользователя в панели администрирования необходимо указать этот уникальный идентификатор. После регистрации пользователя подобным путем, последний сможет сразу начать взаимодействовать с вашим ботом.

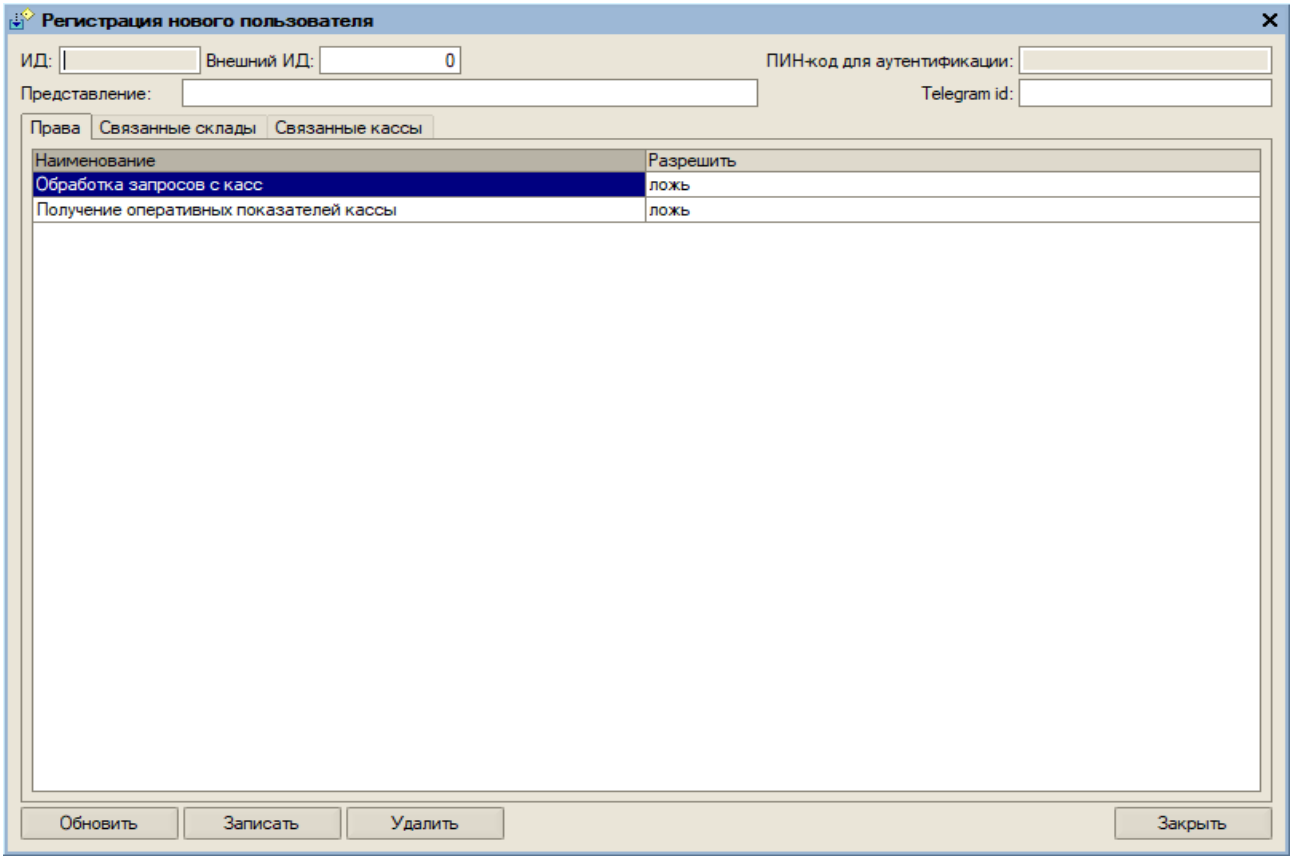
Узнать уникальный идентификатор telegram-пользователя можно обратившись к вашему боту, будучи не зарегистрированным, как на примере из скриншота далее:



Второй путь доступен всегда и не требует знания уникального идентификатора telegram-пользователя: необходимо зарегистрировать нового пользователя без заполнения соответствующего поля уникального идентификатора telegram-пользователя. В таком случае будет сгенерирован специальный PIN-код, который зарегистрированный ранее пользователь должен будет отправить боту в личном чате, после чего бот сможет связать пользователя из своей базы данных с telegram-пользователем.

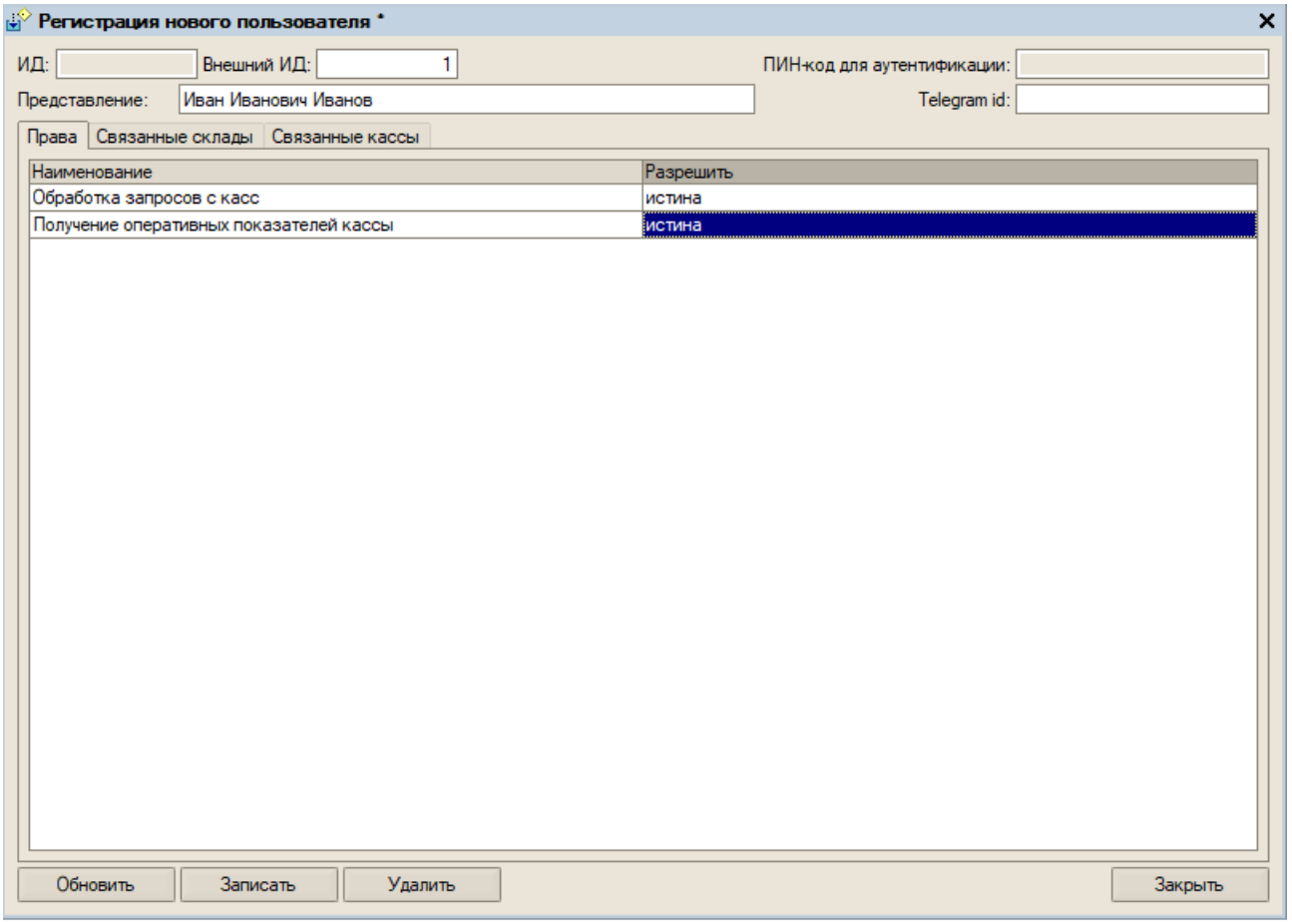
Рассмотрим второй путь регистрации пользователя, без информации об уникальном идентификаторе telegram-пользователя.

Для регистрации нового пользователя необходимо нажать соответствующую иконку добавления элемента в списке пользователей. после этого откроется форма пользователя (см. скриншот далее).

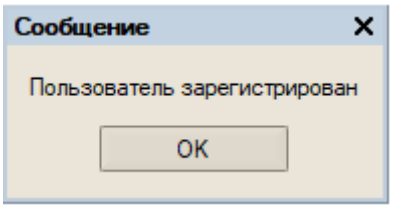


Для регистрации пользователя необходимо заполнить поля: **Представление** и “**Внешний ИД**” и указать какие права у регистрируемого пользователя. Поле **“Внешний ИД”** необходимо для того, чтобы можно было связать зарегистрированного пользователя с пользователем из **Торгового предприятия** или **Кассира**, то есть значение этого поля должно соответствовать коду элемента справочника **Пользователи** в **Торговом предприятии** или **Кассире**, при этом за достоверность этого значения отвечает администратор, регистрирующий пользователя.

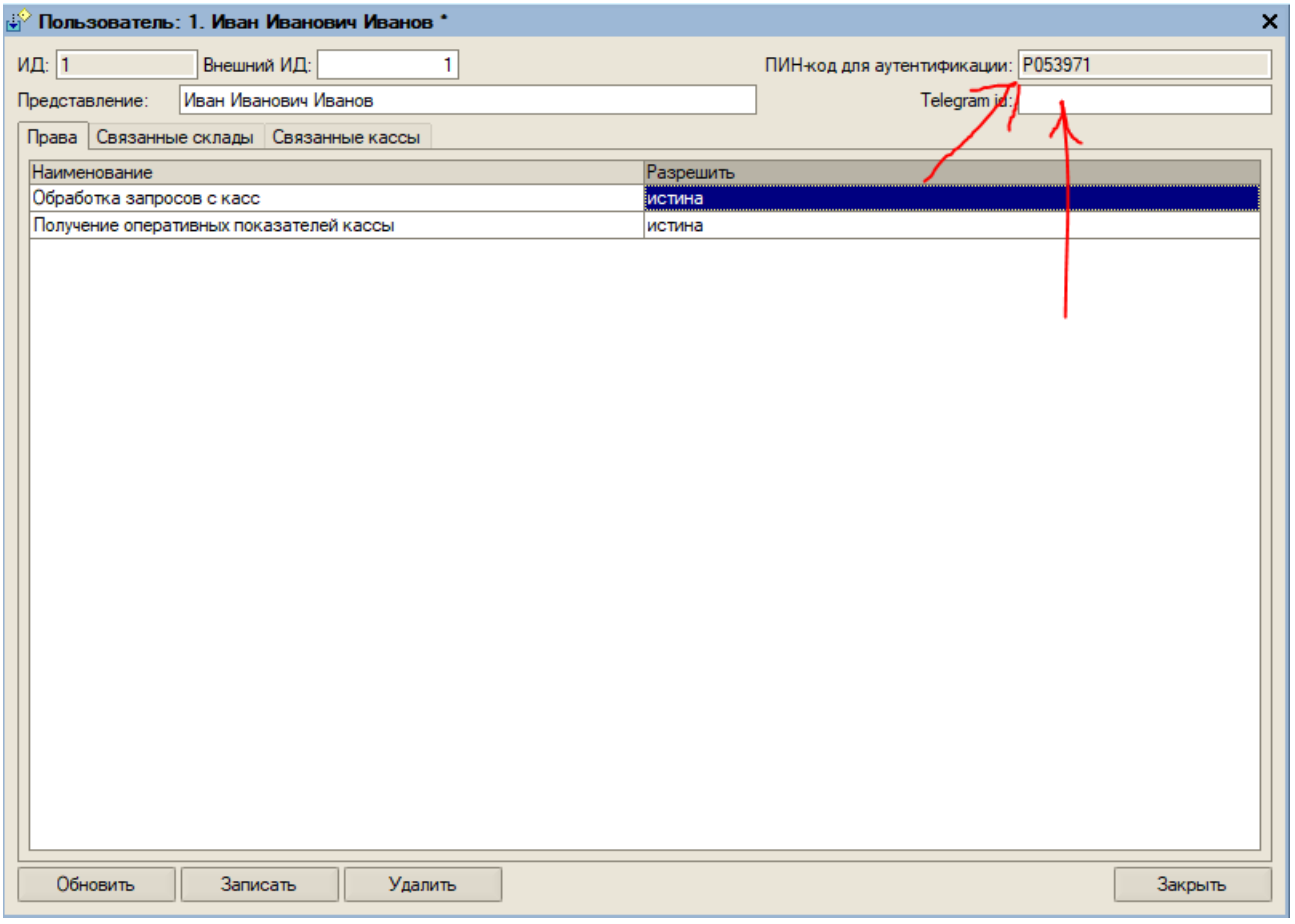
Продолжим регистрацию пользователя. После заполнения обязательных полей (см. скриншот далее) следует нажать кнопку **Записать.**



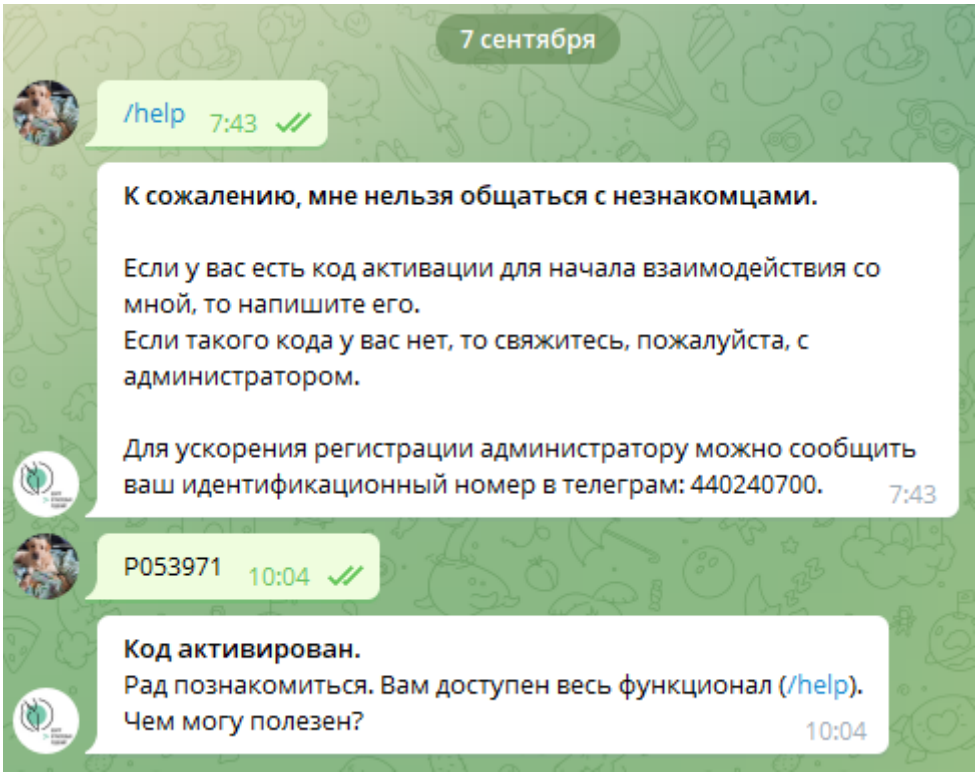
Если регистрация прошла успешно, то выведется соответствующее сообщение (см. скриншот далее).



После успешной регистрации нового пользователя форма обновится и в поле “ПИН-код для аутентификации” появится специальный ПИН-код, который зарегистрированному пользователю необходимо отправить вашему боту (см. скриншот далее).



Когда бот получит этот ПИН-код от незарегистрированного пользователя он свяжет зарегистрированного через панель администрирования пользователя с telegram-пользователем о чем и сообщит в чате в ответ на присланный ПИН-код, как это продемонстрировано на скриншоте далее:

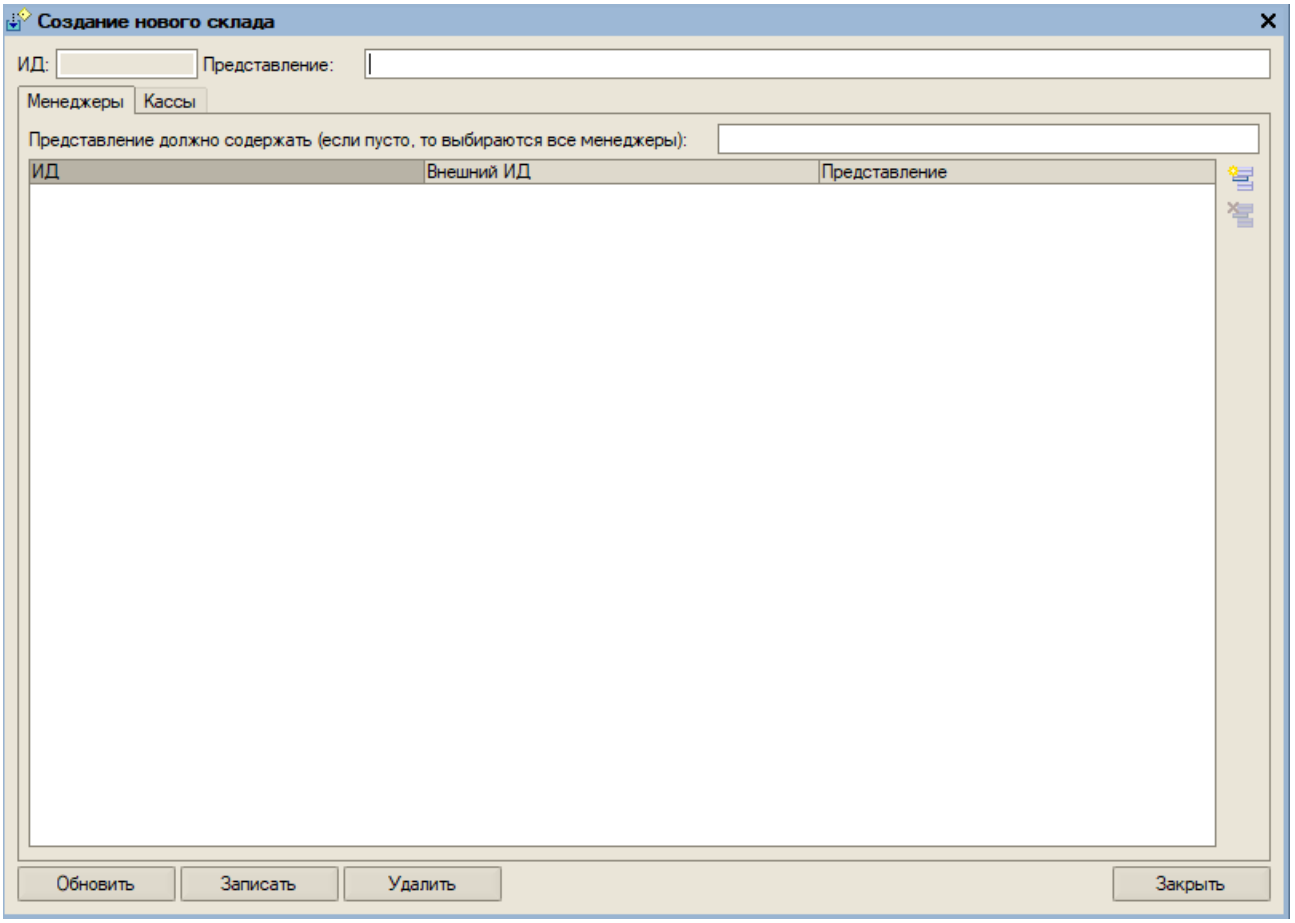


Удаление пользователя возможно из формы пользователя нажатием кнопки **Удалить,** а также из основной формы со списком пользователей нажатием соответствующей иконки удаления элемента списка.

Управление складами

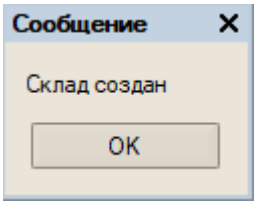
Создание нового склада

Для создания нового склада необходимо нажать соответствующую иконку добавления элемента в списке складов, после этого откроется форма склада (см. скриншот далее).



Для создания склада необходимо заполнить поле **Представление** и нажать кнопку **Записать**.

Если создание прошло успешно, то выведется соответствующее сообщение (см. скриншот далее).



Привязка менеджера

Привязка менеджера осуществляется нажатием соответствующей иконки добавления нового элемента в списке **Менеджеры** на форме склада, к которому необходимо добавить

нового менеджера. По нажатию на соответствующую иконку будет предложено создать нового пользователя: если вы хотите зарегистрировать нового пользователя и сразу привязать его к текущему складу, в роли менеджера, то можете смело соглашаться не предложение “Создать нового пользователя?”, если же вы хотите выбрать пользователя из уже зарегистрированных пользователей, то необходимо нажать кнопку **Нет** и тогда вам будет предложен список текущих пользователей на выбор.

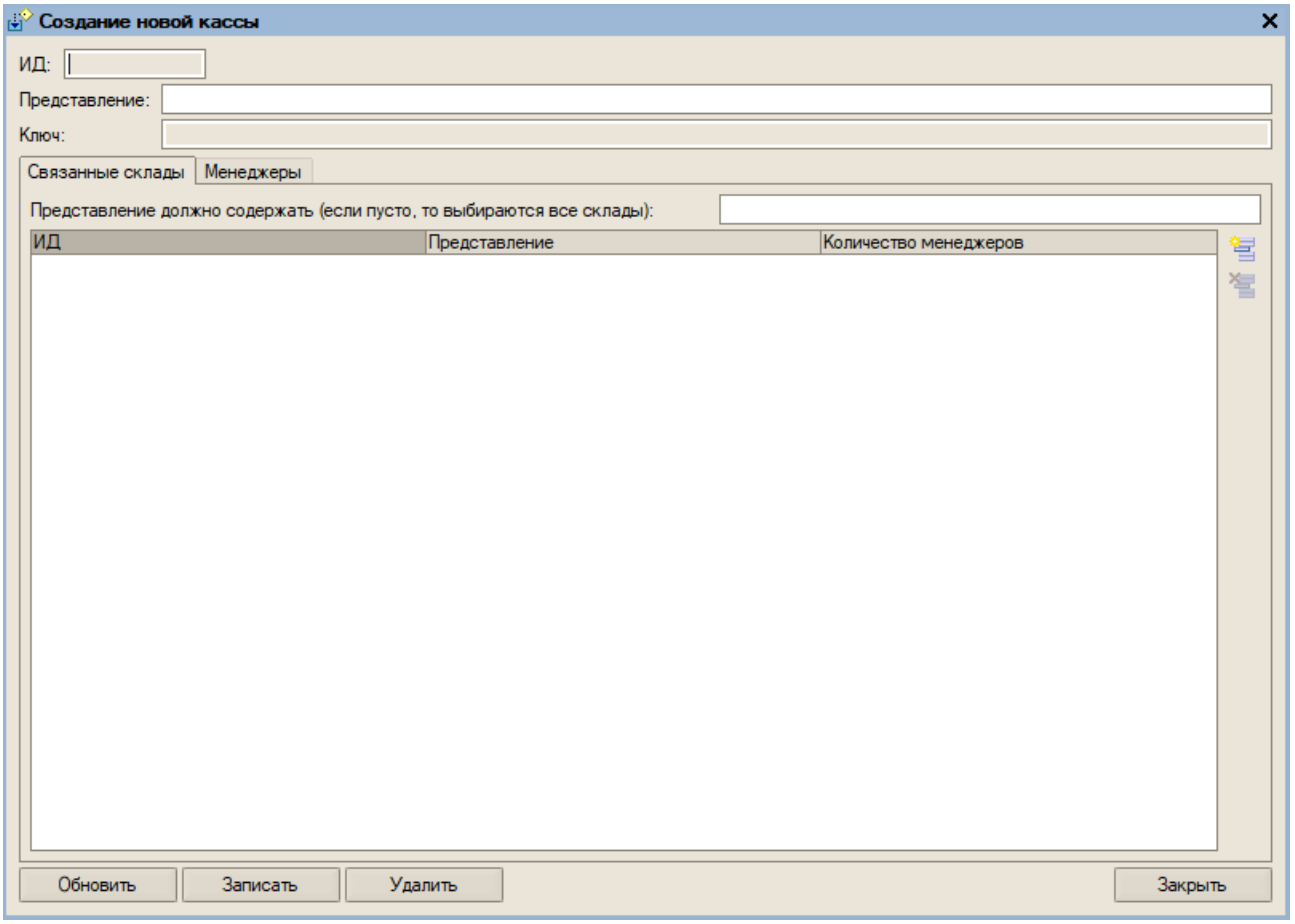
Привязка кассы

Привязка кассы осуществляется нажатием соответствующей иконки добавления нового элемента в списке **Кассы** на форме склада, к которому необходимо привязать новую кассу. По нажатию на соответствующую иконку будет предложено создать новую кассу: если вы хотите создать новую кассу и сразу привязать ее к текущему складу, то можете смело соглашаться не предложение “Создать новую кассу?”, если же вы хотите выбрать кассу из уже имеющихся касс, то необходимо нажать кнопку **Нет** и тогда вам будет предложен список текущих касс на выбор.

Управление кассами

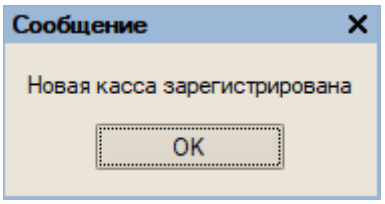
Создание новой кассы

Для создания новой кассы необходимо нажать соответствующую иконку добавления элемента в списке касс, после этого откроется форма кассы(см. скриншот далее).

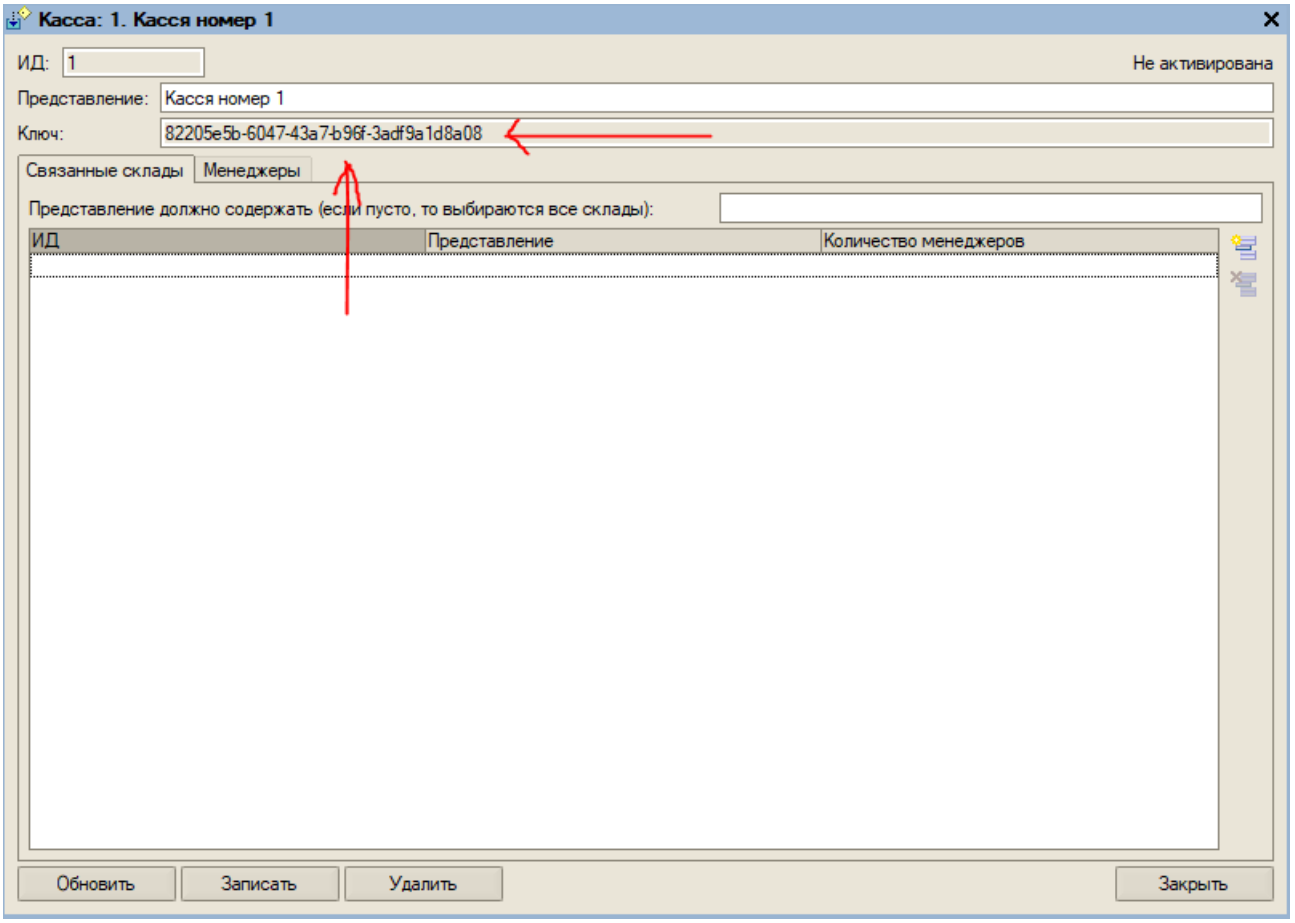


Для создания кассы необходимо заполнить поле **Представление** и нажать кнопку **Записать**.

Если создание прошло успешно, то выведется соответствующее сообщение (см. скриншот далее).



После успешного создания новой кассы форма обновится и в поле **Ключ** появится специальный ключ, который необходимо будет указать при настройке кассира (см. скриншот далее).



Привязка к складу

Привязка кассы к складу осуществляется нажатием соответствующей иконки добавления нового элемента в списке “**Связанные склады”** на кассы, которую необходимо привязать к какому-нибудь складу. По нажатию на соответствующую иконку будет предложено создать новый склады: если вы хотите создать новый склад и сразу привязать к нему текущую кассу, то можете смело соглашаться не предложение “Создать новый склад?”, если же вы хотите выбрать кассу из уже имеющихся касс, то необходимо нажать кнопку **Нет** и тогда вам будет предложен список текущих складов на выбор.

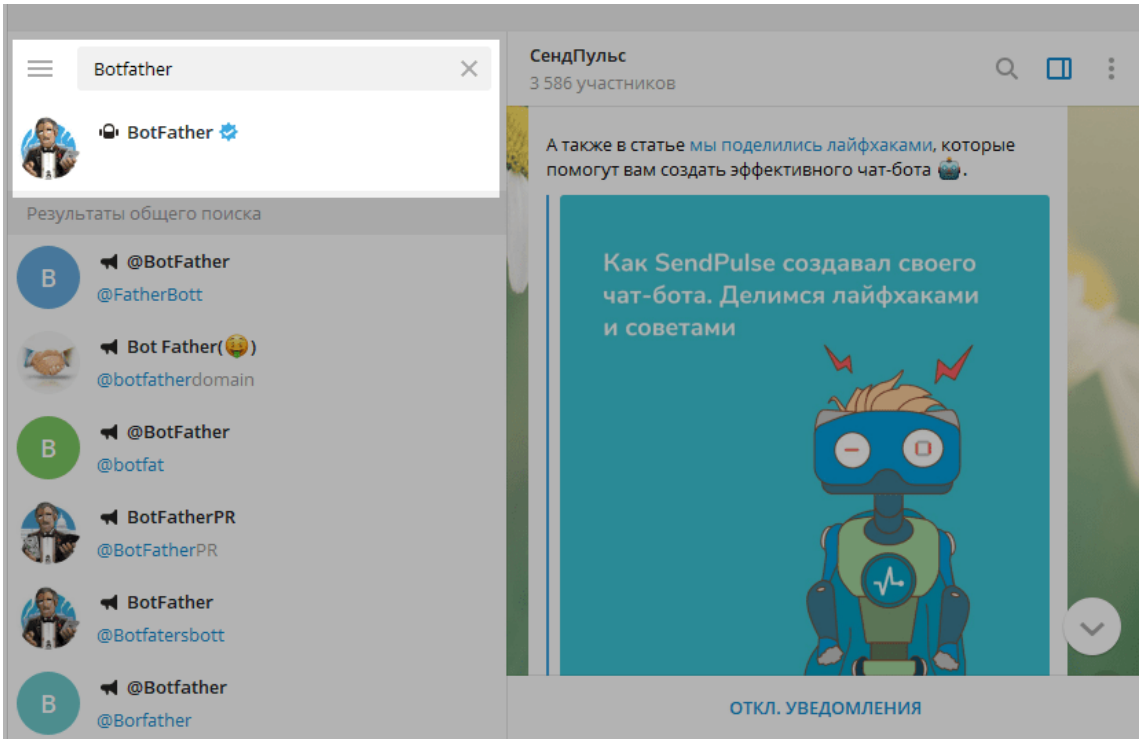
Привязка менеджера

Привязка менеджера осуществляется нажатием соответствующей иконки добавления нового элемента в списке **Менеджеры** на форме кассы, к которой необходимо привязать менеджера. По нажатию на соответствующую иконку будет предложено создать нового пользователя: если вы хотите зарегистрировать нового пользователя и сразу привязать его к текущей кассу, в роли менеджера, то можете смело соглашаться не предложение “Создать нового пользователя?”, если же вы хотите выбрать пользователя из уже зарегистрированных пользователей, то необходимо нажать кнопку **Нет** и тогда вам будет предложен список текущих пользователей на выбор.

Приложения

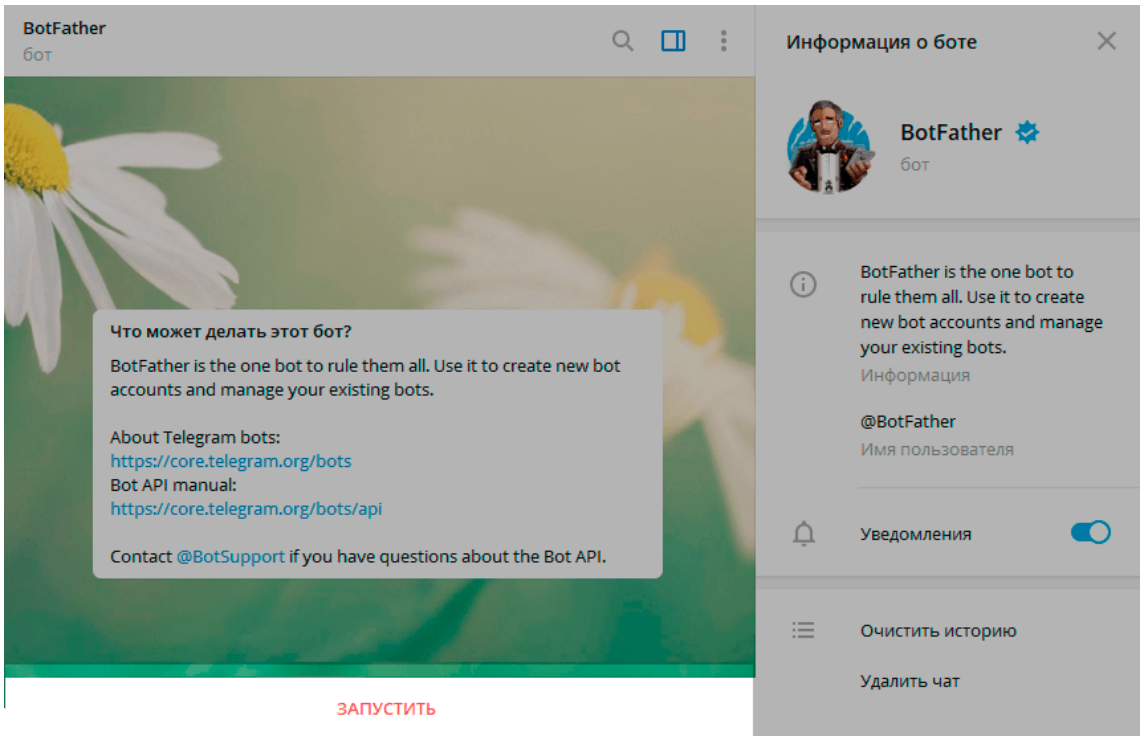
Регистрация бота в telegram и получение его токена

Шаг 1. Введите в поле поиска @BotFather и выберите бота.



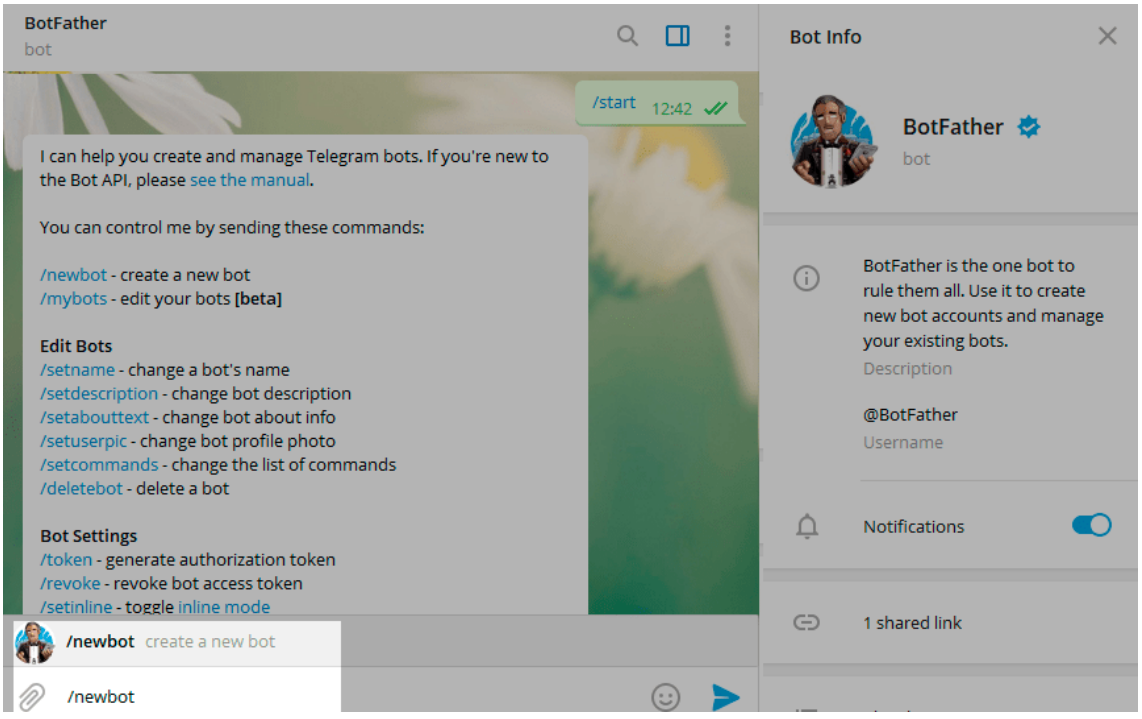
Обратите внимание, что у официального бота Telegram будет стоять синий подтверждающий знак возле имени в виде галочки.

Шаг 2. Нажмите «Запустить» для активации бота BotFather.

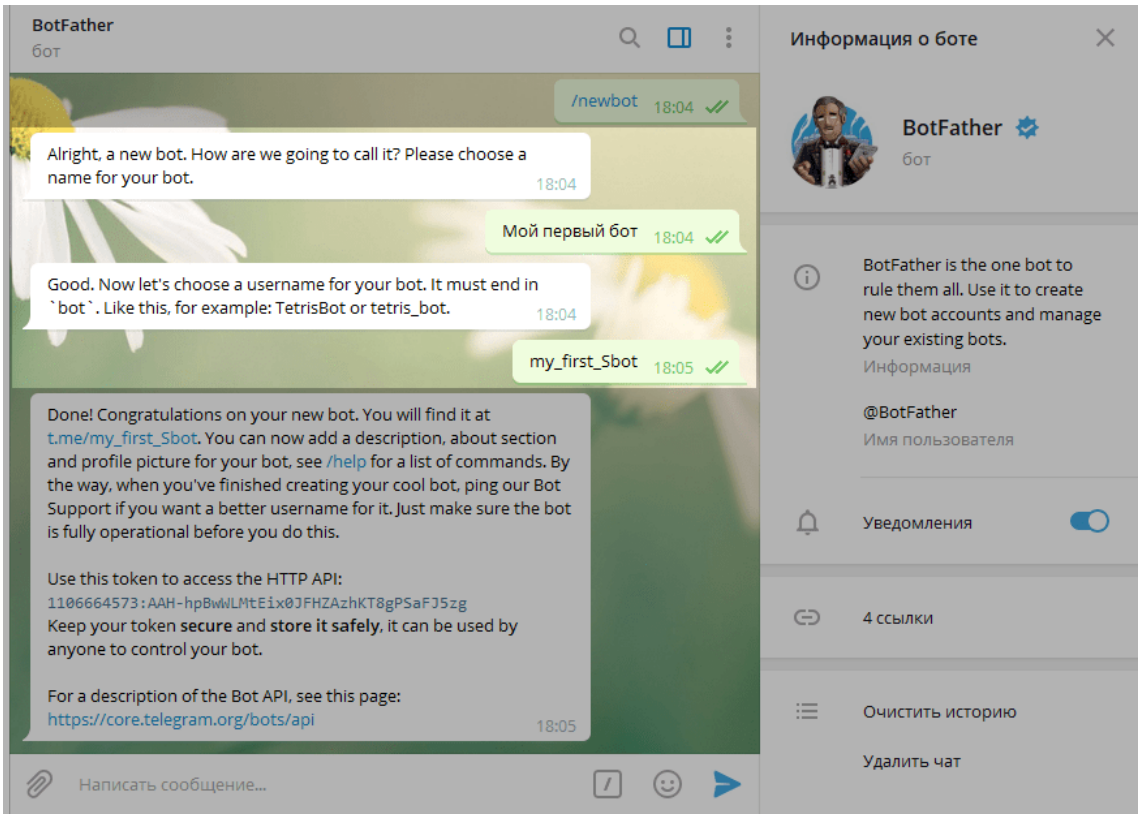


В ответ вы получите список команд по управлению ботов.

Шаг 3. Выберите или напечатайте и отправьте команду **/newbot**.

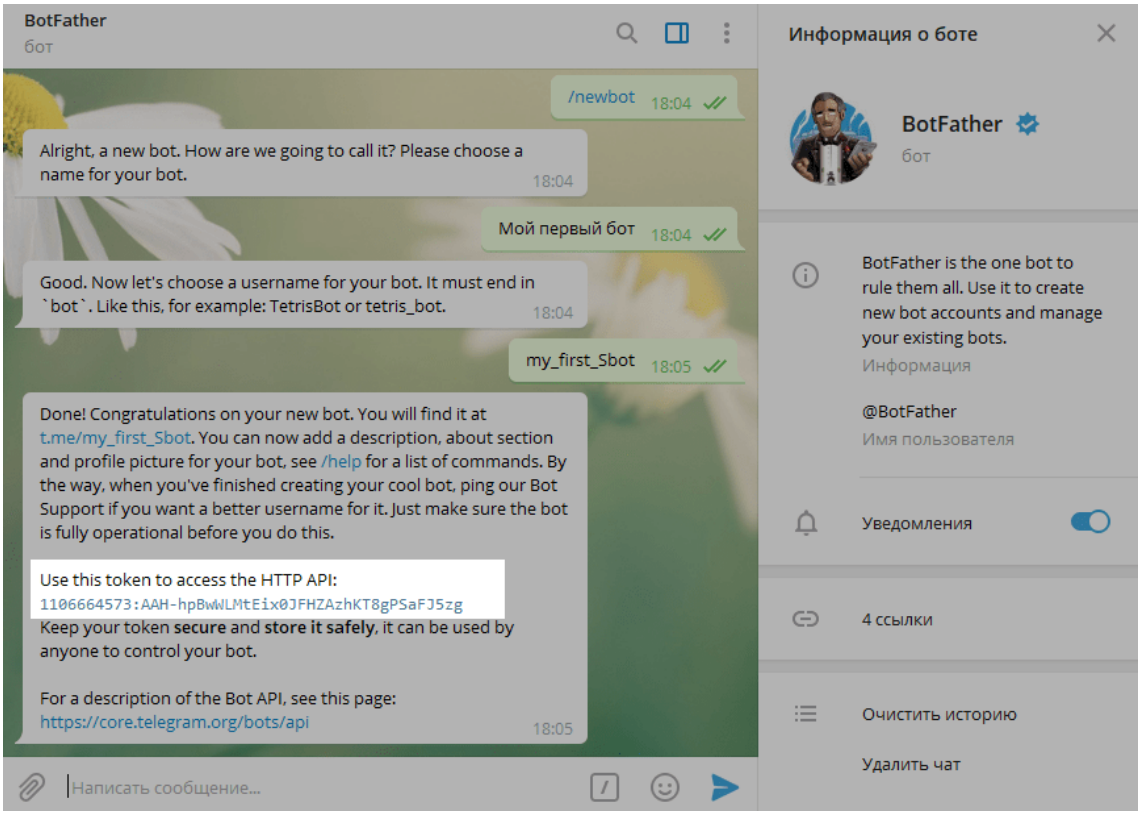


Шаг 4. Дайте имя боту — сотрудники увидят это имя при общении с ботом. И никнейм бота — по нему можно будет найти бота в Telegram. Никнейм должен быть уникальным, не повторять существующие в базе и заканчиваться на слово «bot».



После того как вы выберите подходящее имя бот будет создан. Вы получите сообщение со ссылкой на бота t.me/<никнейм_бота>, рекомендации по настройке аватарки, описание бота и список команд для настройки бота.

Для работы сервера бота вам понадобится токен. Скопируйте значение токена и вставьте его в конфигурационный файл как значение параметра **BOT_TOKEN**.



Список прав и их назначение

Обработка запросов с касс

Наличие этого права у пользователя дает этому пользователю возможность обрабатывать запросы с касс, к которым он привязан как менеджер, а также дает возможность получить срочные оповещения от касс.

Если касса инициирует запрос права на совершение операции (или инициирует отправку срочного оповещения), то бот отправляет запрос на совершение операции (или срочное оповещение) только тем, привязанным к данной кассе, менеджерам которые обладают данным правом - **Обработка запросов с касс**.

Получение оперативных показателей кассы

Наличие данного права у пользователя дает возможность этому пользователю запрашивать оперативные показатели (команда **/getOperationalIndicators**) с любой кассы, к которой он привязан в роли менеджера.

Если пользователя нет этого права, то команда /**getOperationalIndicators** не будет выполняться ботом и даже не будет отображаться в списке доступных команд (**/help**).