

## Создание бота в Telegram на базе node.js

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье описан процесс создания бота в telegram на базе интерпретатора языка программирования JavaScript и node.js. Для создания не потребуется ничего кроме доступа в интернет и номера мобильного телефона. Созданный бот позволяет определять настроение пользователя на основании его сообщений, и классифицировать их как положительные или негативные.

**Ключевые слова:** Телеграм, бот

## Creating a Telegram bot based on node.js

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University*

*student*

### **Abstract**

This article describes the process of creating a bot in telegram based on the interpreter of the JavaScript programming language and node.js. To create, you do not need anything other than Internet access and a mobile phone number. The created bot allows you to determine the mood of the user based on his messages, and classify them as positive or negative.

**Keywords:** Telegram, bot

Telegram - бесплатное облачное приложение для обмена сообщениями для мобильных и настольных компьютеров. У Telegram есть клиенты для платформ, включая Windows, OS X, 64-разрядную версию Linux и 32-разрядную версию Linux. Приложение Telegram также доступно для использования в веб-версии. Что касается мобильных устройств, у Telegram есть собственные приложения для Android, iOS и Windows Phone.

С Telegram можно отправлять сообщения, фотографии, видео и файлы любого типа (doc, zip, mp3 и многие другие); и можно создавать группы до 5000 человек или каналы для трансляции сообщений и медиа.

Одна из замечательных особенностей Telegram заключается в том, что сообщения Telegram зашифрованы и могут быть настроены на самоуничтожение. Во время чата или обмена сообщениями для тех, кто хочет большей конфиденциальности и секретности, в Telegram есть

секретные чаты. Это означает, что эти сообщения можете видеть только участники переписки; никто другой не может их видеть, включая Telegram.

Цель исследования – создать бота в Telegram на базе node.js.

Ранее этим вопросом интересовались В.Л. Колосков, И.Ю. Павлов, Е.Б. Иванов развивали тему «Создание ботов для telegram: с использованием языка программирования python и облачной операционной системы corezoid» [1] в которой описывается процесс создания ботов для Telegram с помощью языка программирования Python и операционной системы Corezoid. В начале работы рассматривается процесс регистрации бота и некоторые простейшие команды. Заканчивается статья рассмотрением преимуществ и недостатков двух этих способов разработки. Р.В. Баранов, А.А. Зинин, А.А. Антонов с темой «Реализация сервиса управления рассылками бота telegram» [2], а подробнее про описание разработки сервиса управления рассылками бота Telegram. Проектируется и разрабатывается серверное приложение, осуществляющее работу с Telegrambot API. Управление рассматриваемым сервисом рассылки сообщений производится посредством REST API, что позволяет использовать его как службу в любой информационной системе. Ш.Р. Миннивалиев, А.Н. Карамышев, Е.В. Абросимова опубликовали статью «Приложение telegram-bot: “система контроля состояния корпоративной сети”» [3] описали программный продукт предназначен для отправки уведомлений системным администраторам об ошибках корпоративной сети, которые он получает из системы мониторинга Nagios. Применяется IT-аутсорсинговой компанией для повышения скорости реагирования на инциденты, а, следовательно, повышения качества деятельности. К основным функциональным возможностям программы можно отнести: получение информации от CM NAGIOS; распределение уведомлений; отправка уведомлений.

Бот Telegram — это специальная учетная запись, для которой не требуется настраивать дополнительный номер телефона. Пользователи могут взаимодействовать с этими ботами двумя способами:

1. Отправляйте сообщения и команды ботам, открывая с ними чат или добавляя их в группы. Этот способ общения используется типичными чат-ботами.

2. Отправляйте запросы прямо из поля ввода, вводя имя пользователя бота и запрос. Это так называемые встроенные боты. Такие боты могут улучшать чаты Telegram с помощью контента из внешних источников, могут предупреждать или уведомлять о новостях и переводах, а также могут предоставлять соответствующую информацию.

Технически боты Telegram — это сторонние приложения, работающие внутри Telegram. Когда пользователь отправляет сообщение боту Telegram, промежуточный сервер Telegram обеспечивает шифрование и связь с помощью API ботов Telegram.

Сначала нужно завести свою учетную запись. Для этого нужно открыть окно браузера и ввести URL-адрес <https://web.telegram.org/#/login> . Это страница регистрации как на рисунке 1.

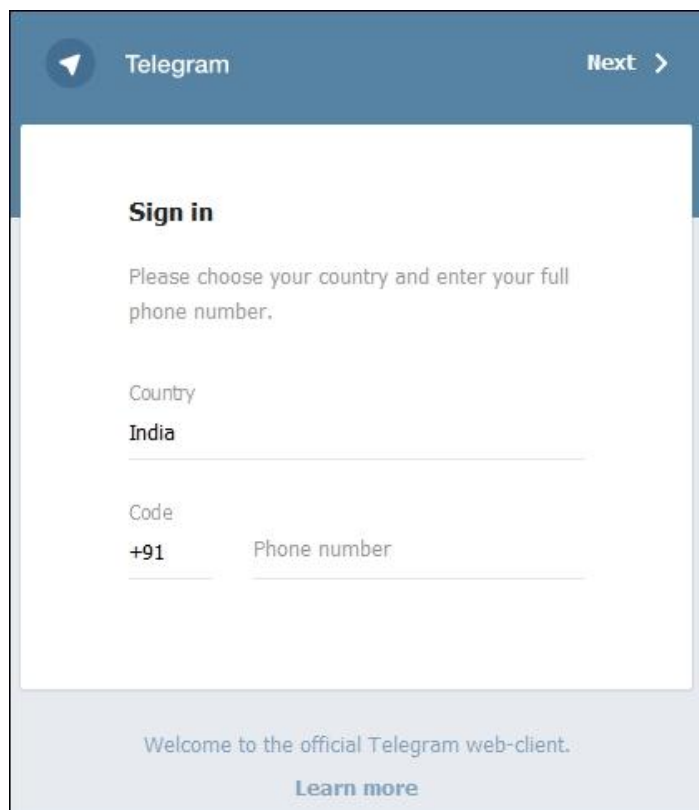
The image shows a screenshot of the Telegram web-client registration page. At the top, there is a blue header with the Telegram logo on the left and the word "Next" with a right-pointing arrow on the right. Below the header, the main content area is white and contains the following elements: the heading "Sign in", the instruction "Please choose your country and enter your full phone number.", a "Country" label with a dropdown menu showing "India", a "Code" label with a dropdown menu showing "+91", and a "Phone number" label with an input field. At the bottom of the page, there is a light blue footer with the text "Welcome to the official Telegram web-client." and a "Learn more" link.

Рис. 1 Окно регистрации

В форме нужно указать свою страну и номер телефона. Следующим шагом будет нажать ссылку next и перейти на следующую страницу, как на рисунке 2.

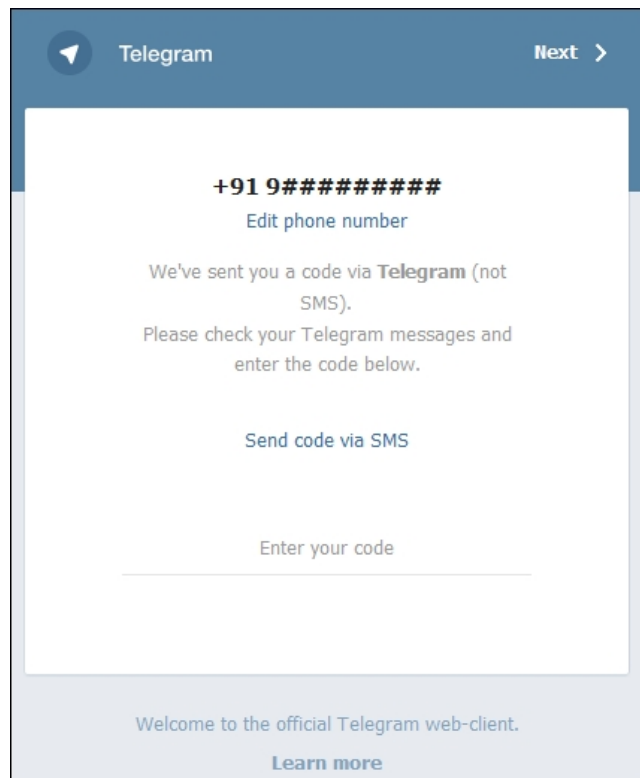


Рис 2. Следующее окно регистрации

Последним шагом будет ввести полученный через смс код в форму. После этого можно начать обмен сообщениями как на рисунке 3.

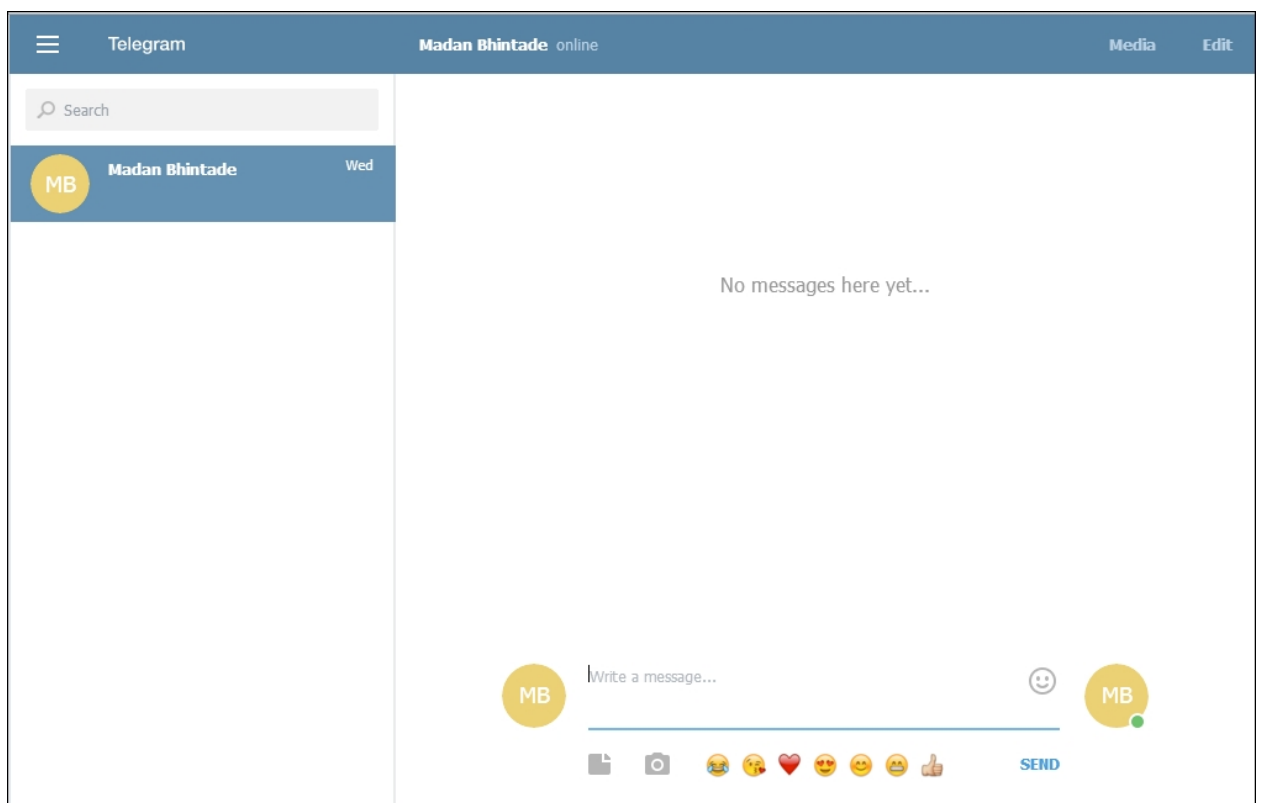


Рис. 3 Главное окно Telegram

На этом этапе учетная запись в telegram успешно создана.

Для создания первого бота нужно в окне поиска ввести @BotFather это самый простой способ начать работу с ботами Telegram.

Это потрясающий метод, который Telegram специально предоставил разработчикам для создания собственных ботов. Здесь можно наблюдать возможности одного бота, который помогает создавать других ботов.

Для начала BotFather представится и отобразит кнопку Start внизу для пользователя как на рисунке 4.

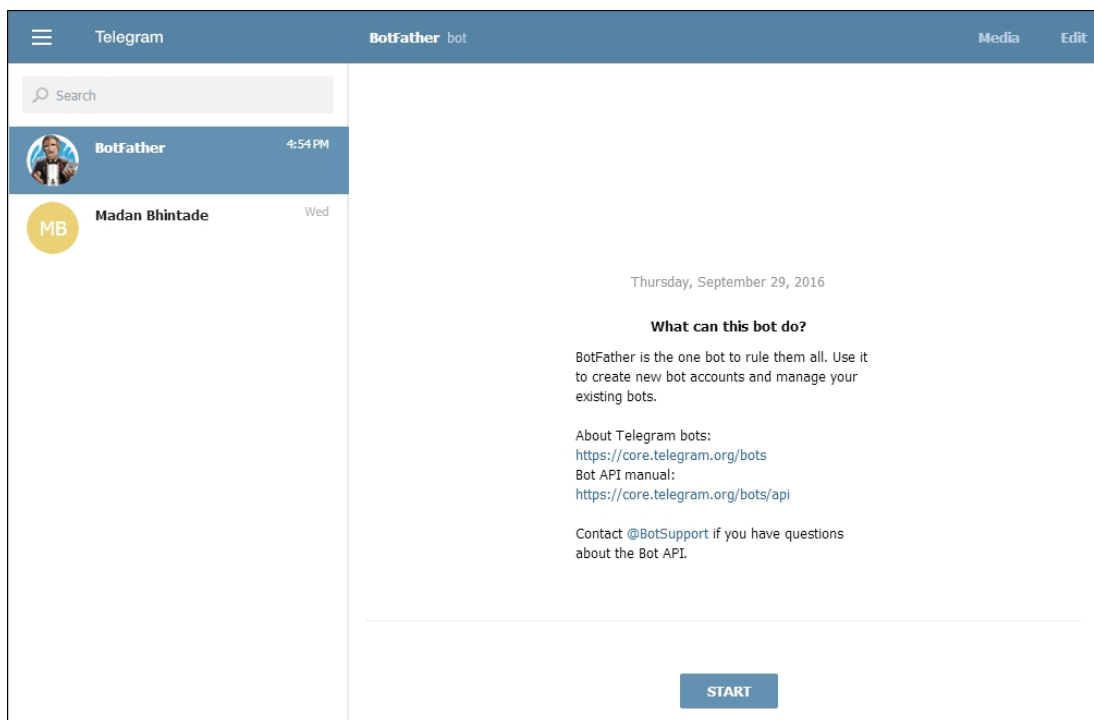


Рис. 4 Окно приветствия BotFather

После нажатия на кнопку Start, BotFather предоставит все команды, которые можно использовать для создания нового бота, как показано на рисунке 5.

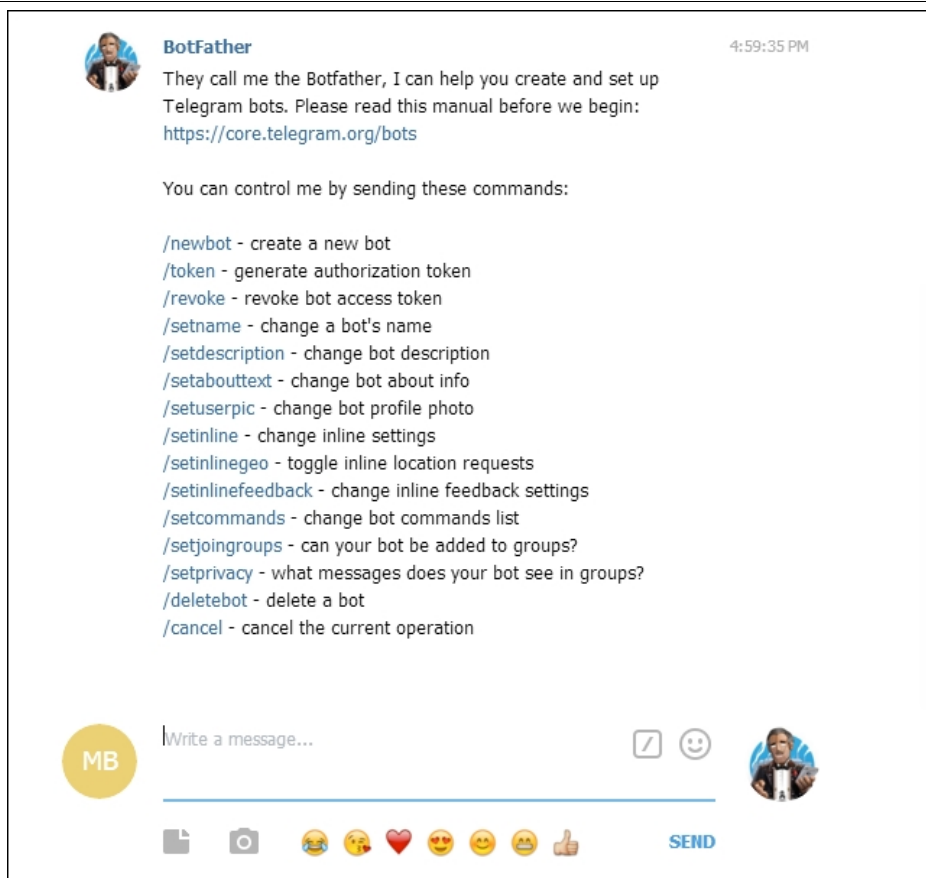


Рис. 5 Меню BotFather

Теперь нужно нажать на ссылке `/newbot` из диалога с BotFather. С помощью этой команды BotFather попросит выбрать имя для бота.

Для демонстрации будет использовано имя MadansNewTelegramBot. BotFather внутренне проверяет, доступно ли имя. Если он доступен, BotFather запрашивает имя пользователя для вновь созданного бота, что можно наблюдать на рисунке 6.

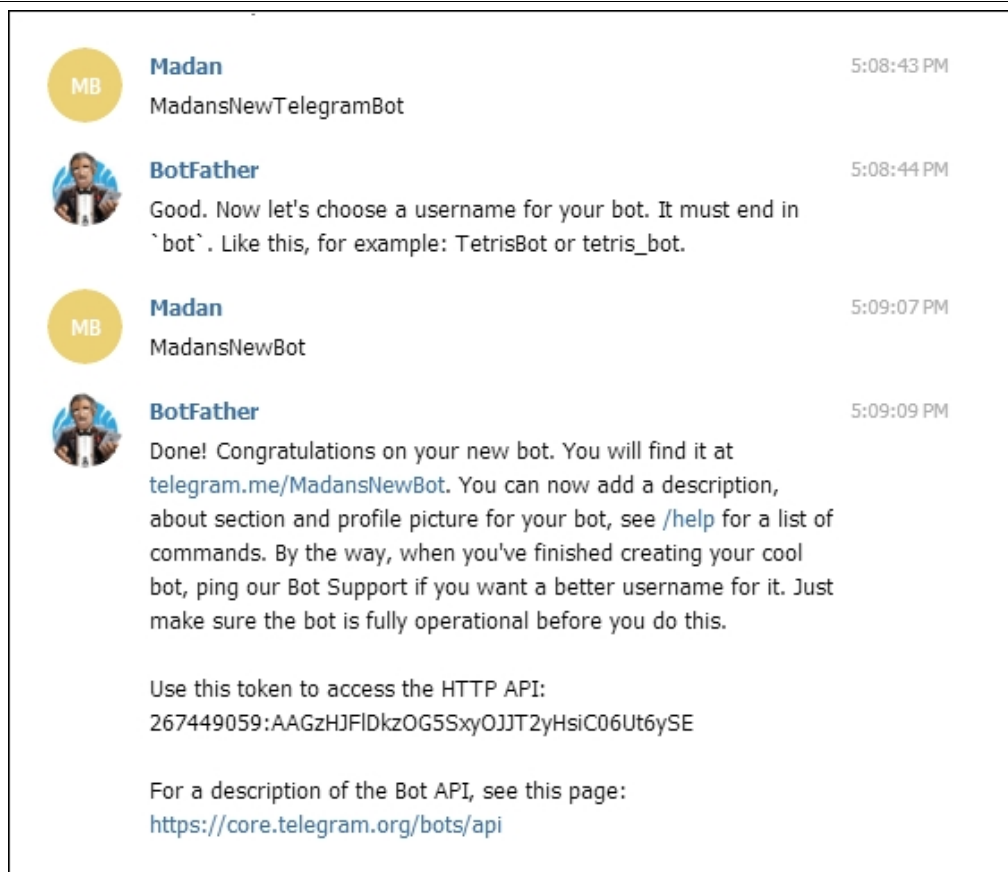


Рис. 6 Процесс создания бота в Botfather

Таким образом, BotFather создал бота, а также предоставил токен для него. Этот токен будет использоваться при подключении бота к API ботов Telegram.

Теперь бот доступен для поиска поле поиска, как показано на рисунке 7.

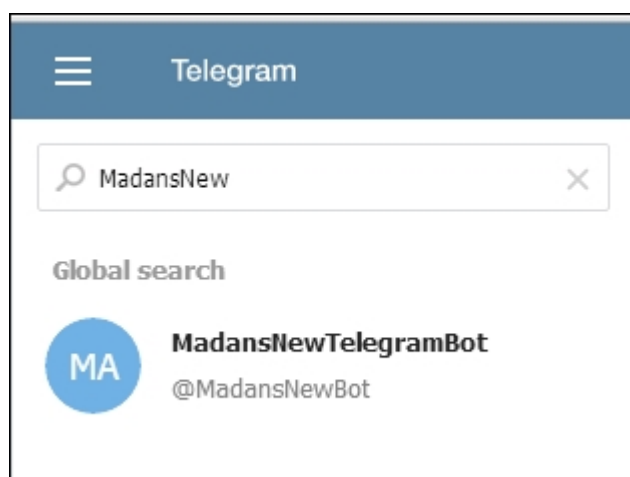


Рис. 7 Поиск бота

На данный момент была создана учетная запись в Telegram и создан базовый бот Telegram с помощью BotFather. Дальше нужно разобраться с термином сентиментальный анализ.

Проще говоря, анализ настроений — это просто классификация данного слова или предложения как положительного, отрицательного или нейтрального. Это также известно как анализ мнения или определение отношения человека, который пишет или говорит.

Что касается Telegram, анализ настроений может быть чрезвычайно полезен для мониторинга СМИ и получения мнений по некоторым публичным темам.

Анализ настроений может быть выполнен с помощью таких подходов, как методы, основанные на знаниях, статистические методы и их комбинация. Техники, основанные на знаниях, классифицируют текст на основе близости слов к определенным эмоциям, таким как счастье, грусть и т. д.

При представлении настроений термина или предложения словам, имеющим к ним отрицательное, нейтральное или положительное отношение, присваивается соответствующий номер по шкале от -10 до +10, а уровень тональности или оценка определяется в термине или на уровне предложения.

Для дальнейшей работы потребуется интерпретатор языка программирования JavaScript, nodejs. Его нужно скачать с официального сайта и установить.

Теперь непосредственно, как можно использовать API-интерфейсы Node.js и Telegram для создания базового бота Telegram.

Сначала нужно создать рабочую папку локальном диске из командной строки для хранения настроек бота. Нужно создать папку telegrambot и перейти в неё.

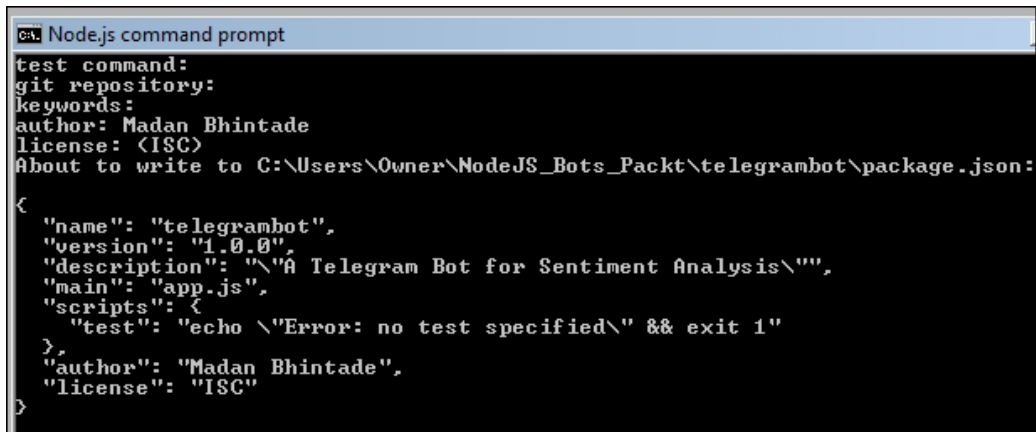
```
mkdir telegrambot  
cd telegrambot
```

Дальше нужно инициализировать директорию для хранения пакетов и зависимостей nodejs командой npm init.

```
npm init
```

При правильной установке nodejs должны быть следующие строки как на рисунке 8.





```

C:\> Node.js command prompt
test command:
git repository:
keywords:
author: Madan Bhintade
license: <ISC>
About to write to C:\Users\Owner\NodeJS_Bots_Packt\telegrambot\package.json:
{
  "name": "telegrambot",
  "version": "1.0.0",
  "description": "\<\/>A Telegram Bot for Sentiment Analysis\<\/>",
  "main": "app.js",
  "scripts": {
    "test": "echo \<\/>Error: no test specified\<\/>" && exit 1"
  },
  "author": "Madan Bhintade",
  "license": "ISC"
}

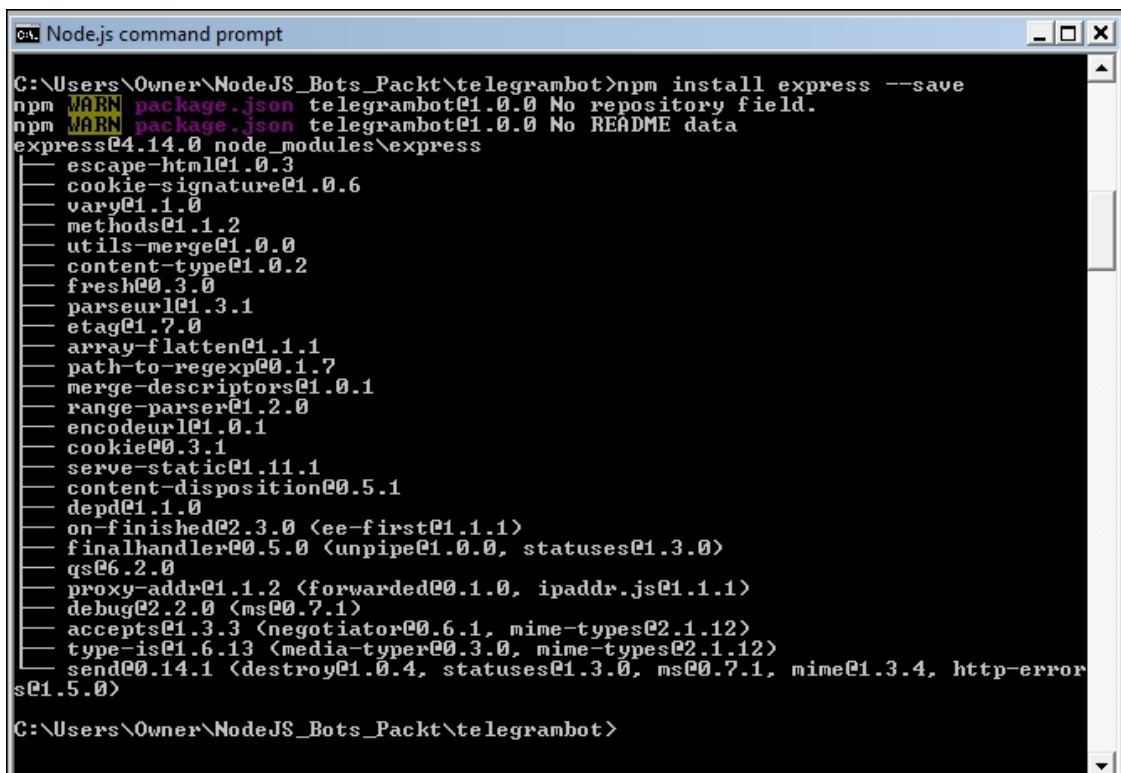
```

Рис. 8 Окно инициализации npm init

В папке проекта должен появиться документ package.json. Для взаимодействия с API Telegram нужно установить библиотеку Express ( <http://expressjs.com> ) в качестве инфраструктуры REST Node.js. Следующая команда установит нужную библиотеку.

```
npm install express --save
```

После установки Express в консоли сообщения как на рисунке 9.



```

C:\Users\Owner\NodeJS_Bots_Packt\telegrambot>npm install express --save
npm WARN package.json telegrambot@1.0.0 No repository field.
npm WARN package.json telegrambot@1.0.0 No README data
express@4.14.0 node_modules\express
├── escape-html@1.0.3
├── cookie-signature@1.0.6
├── vary@1.1.0
├── methods@1.1.2
├── utils-merge@1.0.0
├── content-type@1.0.2
├── fresh@0.3.0
├── parseurl@1.3.1
├── etag@1.7.0
├── array-flatten@1.1.1
├── path-to-regexp@0.1.7
├── merge-descriptors@1.0.1
├── range-parser@1.2.0
├── encodeurl@1.0.1
├── cookie@0.3.1
├── serve-static@1.11.1
├── content-disposition@0.5.1
├── depd@1.1.0
├── on-finished@2.3.0 (ee-first@1.1.1)
├── finalhandler@0.5.0 (unpipe@1.0.0, statuses@1.3.0)
├── qs@6.2.0
├── proxy-addr@1.1.2 (forwarded@0.1.0, ipaddr.js@1.1.1)
├── debug@2.2.0 (ms@0.7.1)
├── accepts@1.3.3 (negotiator@0.6.1, mime-types@2.1.12)
├── type-is@1.6.13 (media-typer@0.3.0, mime-types@2.1.12)
└── send@0.14.1 (destroy@1.0.4, statuses@1.3.0, ms@0.7.1, mime@1.3.4, http-error
s@1.5.0)
C:\Users\Owner\NodeJS_Bots_Packt\telegrambot>


```

Рис. 9 Установка Express

Следующая библиотека нахывается node-telegram-bot-api следующей командой.

```
npm install node-telegram-bot-api --save
```

Вывод должен быть как на рисунке 9.



```
C:\Users\Owner\NodeJS_Bots_Packt\telegrambot>npm install node-telegram-bot-api
npm WARN package.json telegrambot@1.0.0 No repository field.
npm WARN package.json telegrambot@1.0.0 No README data
node-telegram-bot-api@0.23.3 node_modules\node-telegram-bot-api
├── eventemitter3@1.2.0
├── file-type@3.8.0
├── mime@1.3.4
├── debug@2.2.0 (ms@0.7.1)
├── pump@1.0.1 (once@1.4.0, end-of-stream@1.1.0)
├── bl@1.1.2 (readable-stream@2.0.6)
├── bluebird@3.4.6
├── request@2.75.0 (is-typedarray@1.0.0, oauth-sign@0.8.2, aws-sign@0.6.0, tunnel-agent@0.4.3, forever-agent@0.6.1, stringstream@0.0.5, caseless@0.11.0, isstream@0.1.2, json-stringify-safe@5.0.1, aws4@1.4.1, extend@3.0.0, node-uuid@1.4.7, qs@6.2.1, combined-stream@1.0.5, tough-cookie@2.3.1, form-data@2.0.0, mime-types@2.1.12, har-validator@2.0.6, hawk@3.1.3, http-signature@1.1.1)
└── request-promise@2.0.1 (bluebird@2.11.0, lodash@4.16.2)
C:\Users\Owner\NodeJS_Bots_Packt\telegrambot>
```

Рис. 9. Установка node-telegram-bot-api

Теперь можно сосредоточиться на построении базовой логики для работы с ботом. Нужно создать файл app.js, он будет точкой входа бота. Файл app.js должен содержать следующее.

```
var telegramBot = require('node-telegram-bot-api');
var token = '267449059:AAGzHJF1Dkz0G5Sxy0JTT2yHsiC06Ut6ySE';
var api = new telegramBot(token, {polling: true});
api.onText(/\/help/, function(msg, match) {
  var fromId = msg.from.id;
  api.sendMessage(fromId, "I can help you in getting the sentiments of any text you send to me.");
});
api.onText(/\/start/, function(msg, match) {
  var fromId = msg.from.id;
  api.sendMessage(fromId, "They call me MadansFirstTelegramBot. " +
    "I can help you in getting the sentiments of any text you send to me."+
    "To help you i just have few commands.\n/help\n/start\n/sentiments");
});
console.log("MadansFirstTelegramBot has started. Start conversations in your Telegram.");
```

Первой строчкой идет подключение библиотеки node-telegram-bot-api.

```
var telegramBot = require('node-telegram-bot-api');
```

Следующие 3 строчки представляют собой токен, который дал BotFather и применение его в библиотеке node-telegram-bot-api.

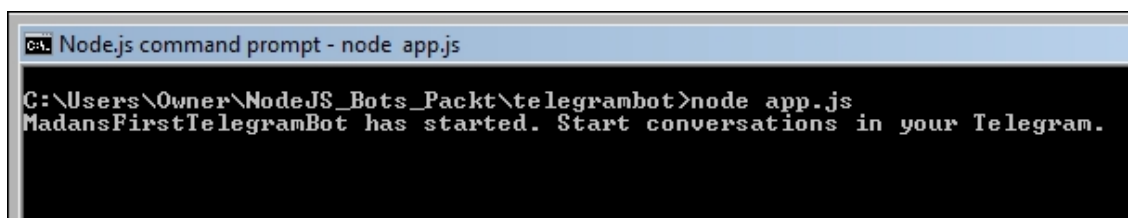
```
var token = '267449059:AAGzHJF1Dkz0G5Sxy0JTT2yHsiC06Ut6ySE';
var api = new telegramBot(token, {polling: true});
});
```

Для запуска бота, боты Telegram используют команду /start. При вводе команды запуска бот должен представиться и спросить, чем он может помочь. Это достигается с помощью следующего фрагмента кода:

```
api.onText(/\/start/, function(msg, match) {  
  var fromId = msg.from.id;  
  api.sendMessage(fromId, "They call me MadansFirstTelegramBot. " +  
    "I can help you in getting the sentiments of any text you send to me." +  
    "To help you i just have few commands. \n/help\n/start\n/sentiments");  
});
```

Это в основном говорит боту, что если пользователь отправляет команду /start, бот отправит сообщение в ответ на это с помощью api.onText() метода.

Теперь нужно запустить команду node app.js для начала работы с ботом как на рисунке 10.



```
C:\Users\Owner\NodeJS_Bots_Packt\telegrambot>node app.js  
MadansFirstTelegramBot has started. Start conversations in your Telegram.
```

Рис. 10. Запуск app.js

Теперь нужно перейти во вкладку переписки с ботом в telegram и нажать кнопку Start, и боту будет отправлена команда для начала разговора. Как только это будет сделано, бот ответит в ответ на команду Start как на рисунке 11.

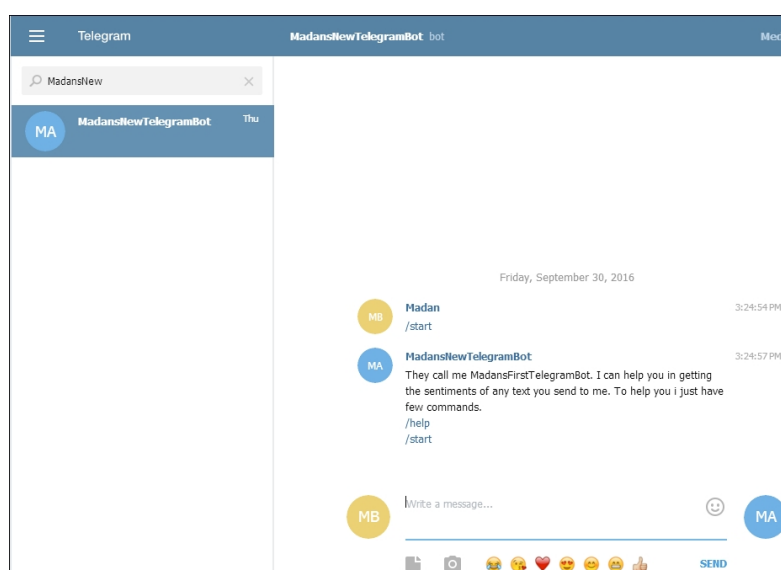


Рис. 11 Ответ на команду Start

На команду /help бот ответит следующим ответом как на рисунке 12.

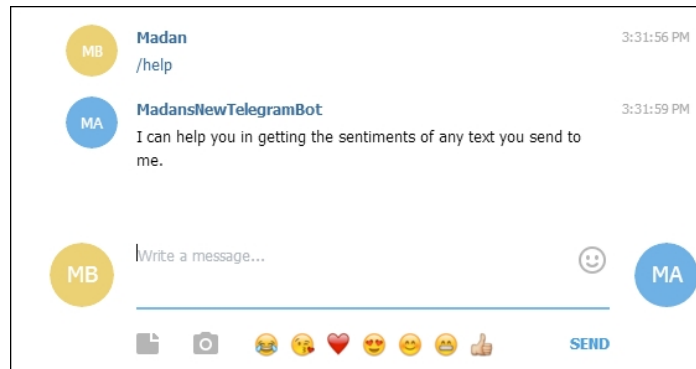


Рис. 12 Ответ на /help

Поскольку программа `app.js` выполняется за кулисами, бот отвечает на команды в зависимости от того, что было запрограммировано в `app.js`.

Для анализа настроений нужно использовать пакет для Node.js `Sentiment`. `Sentiment` — это модуль Node.js, который использует список слов AFINN-111 для анализа тональности входящего текста.

Чтобы установить его, нужно ввести команду.

```
npm install sentiment --save
```

В консоли будет вывод как на рисунке 13.

Рис. 13 Вывод установки sentiment

Теперь нужно изменить код `app.js`. А точнее добавить в файл библиотеку как написано ниже.

```
var sentiment = require('sentiment');
```

Идея здесь в том, что, как только вы отправите команду `/sentiments` боту, бот подтвердит ваши намерения и попросит отправить термин или предложение. Получив термин или предложение, бот проведет анализ настроений с помощью пакета настроений, подключенного к программе Node.js. Затем бот ответит с оценкой анализа настроений.

Обновленный `app.js` должен выглядеть так:

```
var telegramBot = require('node-telegram-bot-api');
var sentiment = require('sentiment');
```

```
var token = '267449059:AAGzHJF1DkzOG5SxyOJTT2yHsiC06Ut6ySE';
var api = new telegramBot(token, {polling: true});
api.onText(/\/help/, function(msg, match) {
  var fromId = msg.from.id;
  api.sendMessage(fromId, "I can help you in getting the sentiments of any text you
send to me.");
});
api.onText(/\/start/, function(msg, match) {
  var fromId = msg.from.id;
  api.sendMessage(fromId, "They call me MadansFirstTelegramBot. " +
  "I can help you in getting the sentiments of any text you send to me."+
  "To help you i just have few commands.\n/help\n/start\n/sentiments");
});
var opts = {
  reply_markup: JSON.stringify(
    {
      force_reply: true
    }
  )
});
api.onText(/\/sentiments/, function(msg, match) {
  var fromId = msg.from.id;
  api.sendMessage(fromId, "Alright! So you need sentiments of a text from me. "+
  "I can help you in that. Just send me the text.", opts)
  .then(function (sended) {
    var chatId = sendId.chat.id;
    var messageId = sendId.message_id;
    api.onReplyToMessage(chatId, messageId, function (message) {
      var sentival = sentiment(message.text);
      api.sendMessage(fromId, "So sentiments for your text are, Score:" +
sentival.score + " Comparative:"+sentival.comparative);
    });
  });
});
console.log("MadansFirstTelegramBot has started. Start conversations in your
Telegram.");
```

В приведенном выше коде, как только бот получает текст, выполняется вызов функции `sentiment()` и передается текст `message.text` для получения настроения.

Мнения возвращаются в объекте `sentival`. Этот `sentival` объект имеет как оценку, так и сравнительные значения тональности переданного текста. Эти значения возвращаются пользователю с использованием следующих строк кода:

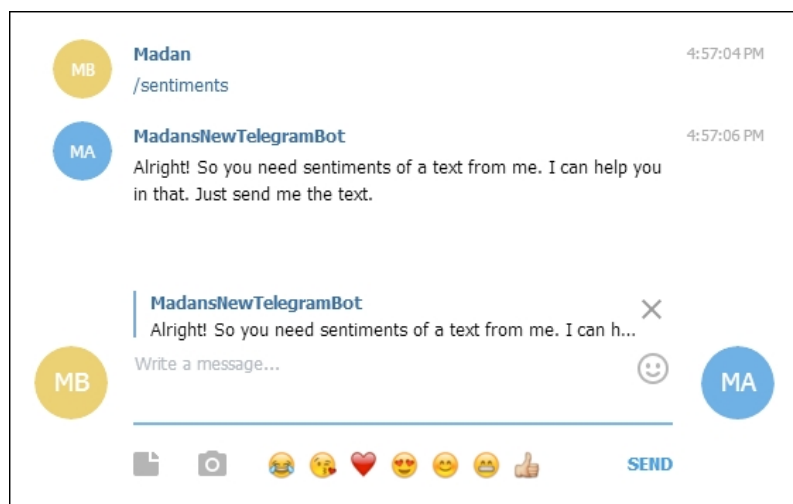
```
api.sendMessage(fromId, "So sentiments for your text are, Score:" + sentival.score
+ " Comparative:"+sentival.comparative);
```

Теперь нужно запустить программу Node.js, чтобы начать запустить ботом. Обновленный код работает, нужно пере зайти в переписки с ботов и снова ввести команду `/start`. Вывод должен быть как на рисунке 14.



Рис. 14 Вывод обновленного бота

При нажатии на ссылку `/sentiments` в ответе на `/start` команду бот ответит следующим образом как на рисунке 15.

Рис. 15 Ответ на `/sentiments`

Итак, здесь бот просит пользователя отправить текст в виде сообщения; в ответ бот поделится настроением сообщения, отправленного пользователем. Для примера можно написать сообщение «Bots are awesome!» в поле «Написать сообщение ...» и нажмем «Ввод».

Сразу же бот ответит с оценкой и сравнительной оценкой тональности вашего текста, как показано на рисунке 16.

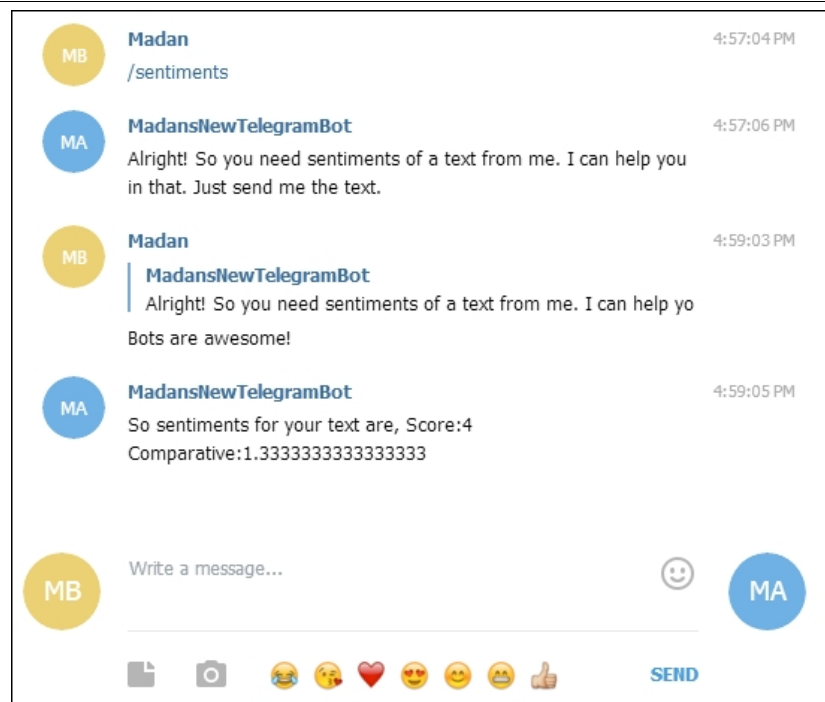


Рис. 16 Ответ бота на сообщение пользователя

Итак, пользователь ввел текст, «Bots are awesome!» и в ответ бот прислал оценку: 4 и сравнительная оценка: 1,33. Для текста, отправленного боту, с положительным значением, бот показывает нам положительное мнение.

Теперь, чтобы узнать мнение о новом предложении, нужно снова отправить /sentiments команду боту.

Бот снова подтвердит намерения и попросит поделиться ввести какой-то текст. Если пользователь введет текст «The food was very bad», то бот ответит как на рисунке 17.

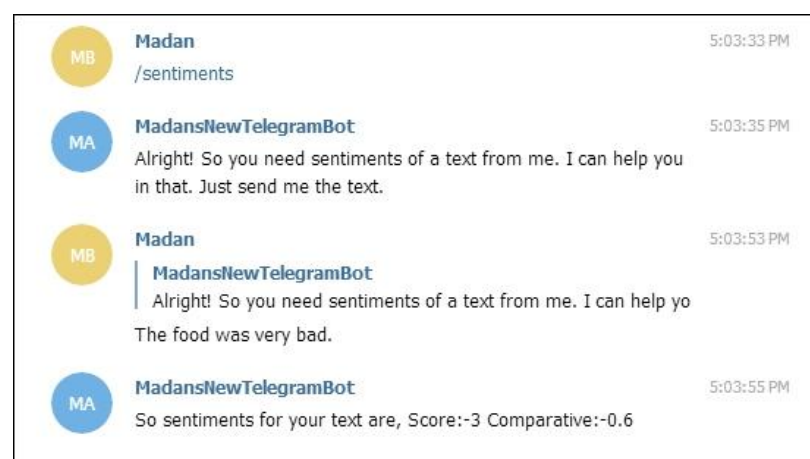


Рис. 17 Ответ бота на негативное предложение

В ответ бот ввел оценку: -3 и сравнительная оценка: -0,6. Таким образом, бот показывает, что предложение несет отрицательное значение.

### Вывод

Результатом статьи стал бот в telegram управляемый с помощью node.js и позволяющий определять настройку пользователя по его сообщениям. Пошаговая настройка создания бота через другого бота BotFather. Также расписаны команды для работы бота и возможности гибкого переделывания его под различные задачи.

### Библиографический список

1. Колосков В.Л., Павлов И.Ю., Иванов Е.Б. Создание ботов для telegram: с использованием языка программирования python и облачной операционной системы corezoid // Системный администратор. 2016. № 5 (162). С. 90-95. URL: <https://elibrary.ru/item.asp?id=26001762> (Дата обращения: 02.01.2021)
2. Баранов Р.В., Зинин А.А., Антонов А.А. Реализация сервиса управления рассылками бота telegram // Постулат. 2017. № 5-1 (19). С. 46. URL: <https://elibrary.ru/item.asp?id=26001762> (Дата обращения: 02.01.2021)
3. Миннивалиев Ш.Р., Карамышев А.Н., Абросимова Е.В. Приложение telegram-bot: “система контроля состояния корпоративной сети” // Хроники объединенного фонда электронных ресурсов Наука и образование. 2017. № 10 (101). С. 28. URL: <https://elibrary.ru/item.asp?id=26001762> (Дата обращения: 02.01.2021)