Создание чат-бота для Discord

Цель: научиться создавать чат-бот на языке программирования JavaScript

# Ход работы

## Установка инструментов для разработки

### Среда разработки

До начала работы необходимо установить инструменты разработчика: IDE, библиотеку Node.js, пакет discord.js.

В качестве IDE (Integrated Development Environment – интегрированная среда разработки) можно использовать любой из программных продуктов, перечисленных ниже[[1]](#footnote-1):

* Sublime Text <https://www.sublimetext.com/>
* Notepad++ <https://notepad-plus-plus.org/downloads/>
* Atom <https://atom.io/>
* Brackets <http://brackets.io/>
* InteliJ Idea <https://www.jetbrains.com/idea/>
* Eclipse <https://eclipse.ru.uptodown.com/windows>
* Visual Studio Code <https://code.visualstudio.com/>

В данном примере используется IDE «Visual Studio Code». Инструкция по установке представлена ниже.

1. Для установки «Visual Studio Code» необходимо перейти по ссылке и скачать набор установочных файлов для вашей операционной системы.
2. Запустить загруженный файл с расширением \*.exe и, следуя указаниям мастера установки, установить программный продукт (все параметры и настройки, запрашиваемые в процессе установки, можно оставить «по умолчанию»).

### Среда выполнения

В качестве среды выполнения JavaScript выбрана Node.js. Инструкция по установке представлена ниже.

1. Перейти на сайт <https://nodejs.org/ru/> и загрузить набор установочных файлов (желательно использовать версию долгосрочной поддержи LTC, левая кнопка на Рисунок 1).
2. Запустить загруженный файл с расширением \*.msi и, следуя указаниям мастера установки, установить программный продукт (все параметры и настройки, запрашиваемые в процессе установки, можно оставить «по умолчанию»).

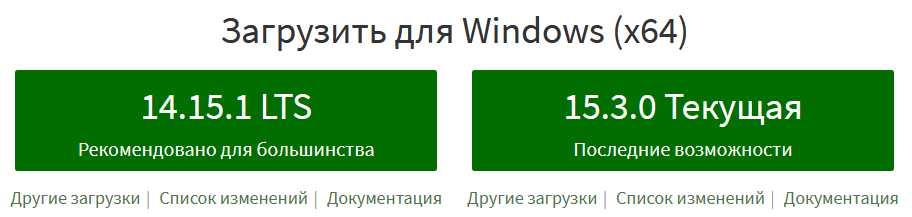


Рисунок . Выбор версии node.js для установки

## Создание аккаунта в Discord

Необходимо создать аккаунт в Discord и подтвердить его с использованием электронной почты

## Создание и настройка бота

1. Открыть браузер и выполнить переход на портал разработчиков Discord по ссылке <https://discord.com/login>

***Примечание****. Если при переходе по ссылке не происходит переход на портал разработчиков, а открывается привычный интерфейс Discord, то необходимо в приложении Discord включить режим разработчика. Для этого выполнить следующие действия:*

1. *вызвать окно «Настройки пользователя» щелчком по значку  в нижней левой части экрана, рядом с именем пользователя;*
2. *слева в списке команд найти команду «Внешний вид» в группе команд «Настройки приложения»;*
3. *пролистать вниз открывшуюся страницу и в разделе «Расширенные» включить переключатель «Режим разработчика»;*
4. *щелкнуть по кнопке  в верхнем правом углу окна;*
5. *еще раз выполнить п. 1 данного раздела.*
6. При необходимости перейти в панель приложений Discord (в области задач слева щелкнуть по ссылке «Applications»).
7. В верхнем правом углу страницы щелкнуть по кнопке «New Application».
8. В открывшемся окне «Create an application» ввести имя создаваемого бота и щелкнуть по кнопке «Create».

**!!Дополнительное задание: не используйте для бота имя, указанное в примере.**

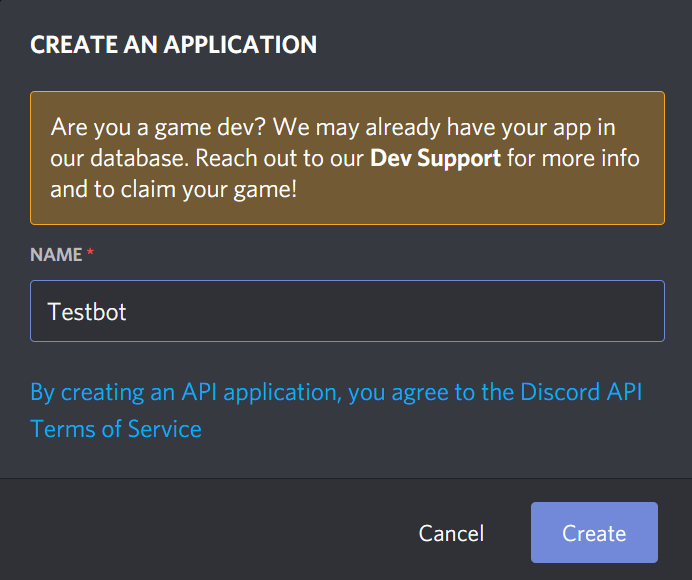


Рисунок . Ввод имени бота

1. Откроется страница «General information», на которой можно задать некоторые свойства создаваемого бота и увидеть его параметры: Client ID, Public Key и др.

**!!Дополнительное задание: добавьте иконку для бота (ссылка «App icon», выбрать любой рисунок), введите описание бота в поле «Description».**

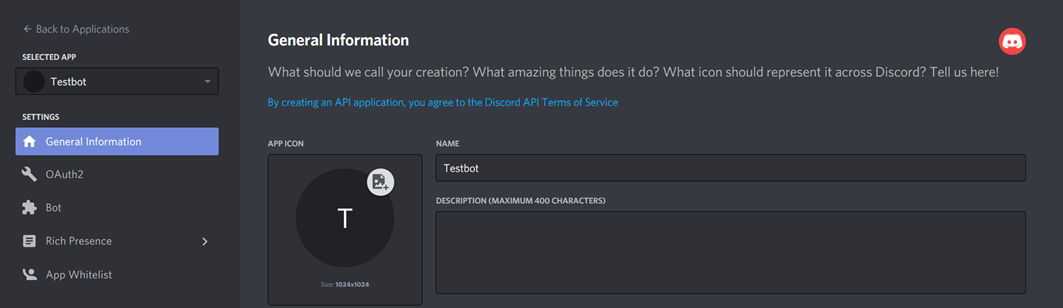


Рисунок . Страница «General information»

1. Перейти на вкладку «Bot» (см. список вкладок слева, Рисунок 4) и нажать на кнопку «Add Bot».

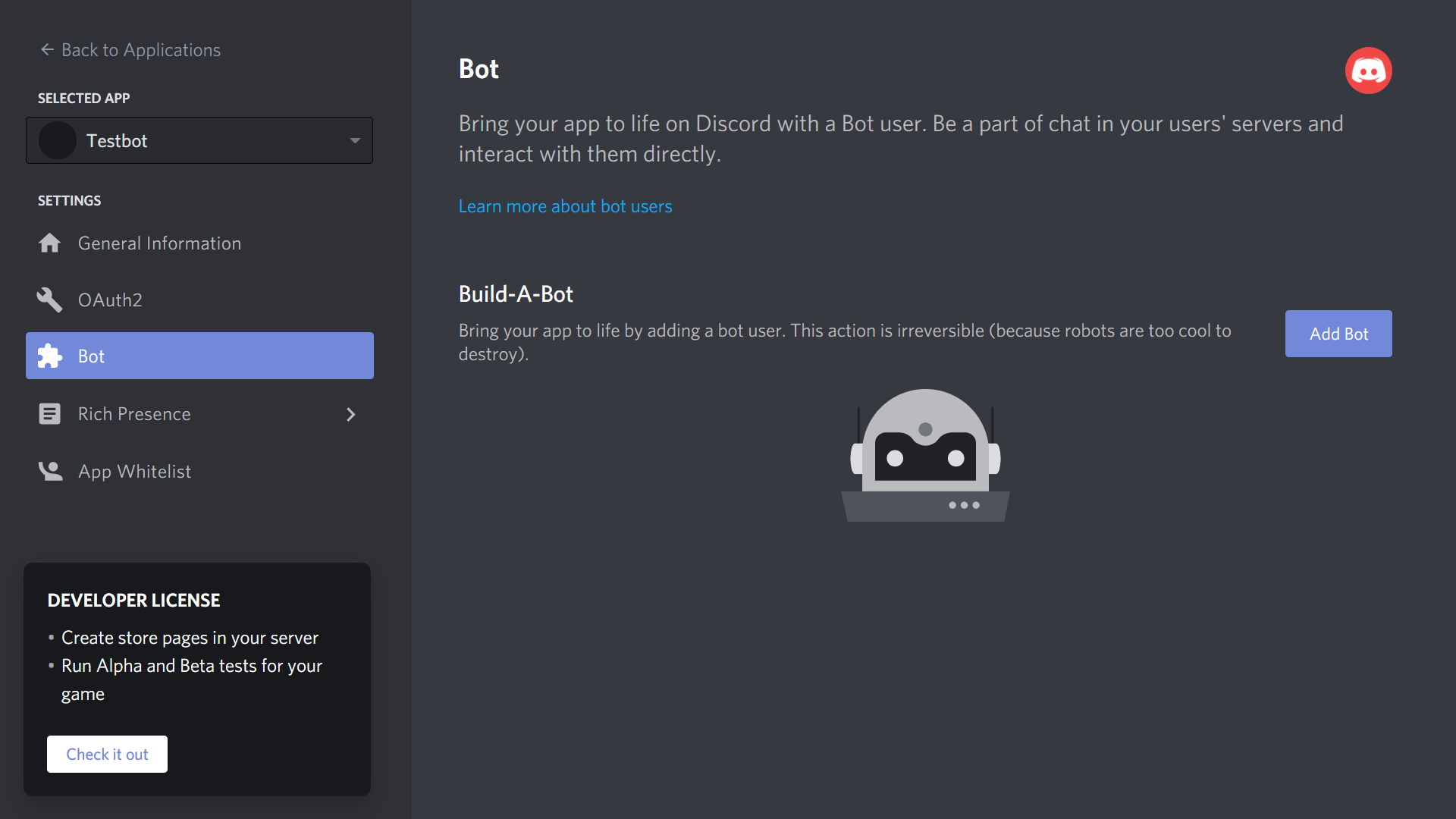


Рисунок . Вкладка «Bot»

1. В открывшемся окне запроса на подтверждение действия нажать кнопку «**Yes, do it!».** Бот будет добавлен в создаваемое приложение.
2. Откроется панель с подробными сведениями о созданном боте: имя, токен аутентификации, изображение в профиле и т.д. (Рисунок 5).

### 

Рисунок . Сведения о созданном боте

### Добавление бота в Discord

1. Перейти на вкладку «OAuth2» и в разделе «SCOPES» установить «флажок» в позицию «bot»
2. На той же странице ниже, в разделе «BOT PERMISSIONS» задать разрешения для бота на выполнение действий. Для данного учебного проекта можно выбрать разрешение «Administrator», что позволит боту выполнять практически любые действия.
3. Скопировать ссылку, сгенерированную в разделе «SCOPES», нажав на кнопку «Copy», открыть любой браузер и вставить скопированную ссылку в строку адреса, нажать клавишу <Enter>.
4. В открывшемся диалоговом окне (Рисунок 6) из раскрывающегося списка выбрать название сервера Discord, куда требуется добавить бота, и щелкнуть по кнопке «Продолжить».

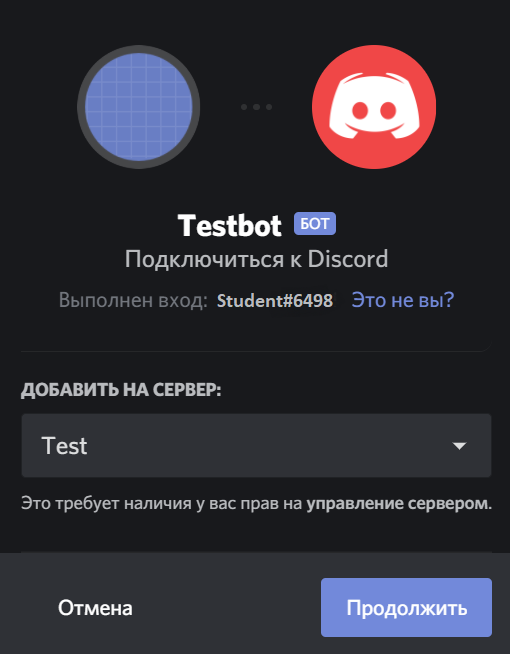


Рисунок . Добавление бота на сервер Discord

1. Бот будет добавлен на выбранный сервер, его имя появится в списке пользователей сервера в правой части окна приложения Discord (Рисунок 7) со статусом «OFFLINE» («НЕ В СЕТИ»).

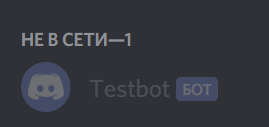


Рисунок . Бот, добавленный на сервер

## Создание проекта

Далее необходимо написать программу для входа в бот.

### Настройка папки и файлов проекта

1. На локальном диске компьютера (например, D:) создать папку с именем бота (в данном примере создана папка «Testbot» в корневом каталоге диска D:).
2. Запустить среду разработки «Visual Studio Code».
3. Открыть созданную папку в открытой среде разработке «Visual Studio Code», выполнив команду «File – Open Folder» («Файл – Открыть папку»).
4. В открывшемся окне перейти на вкладку «Terminal» и выбрать команду «New Terminal» для открытия окна терминала («Терминал – Создать терминал». В нижней части диалогового окна «Testbot – Visual Studio» откроется область терминала (Рисунок 8).

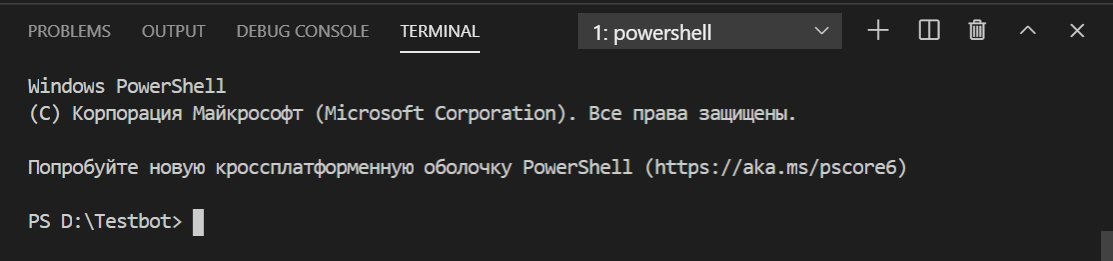


Рисунок . Область терминала

1. Создать файл «package.json», где будут храниться детали проекта и информация о зависимостях, используемых в проекте. Для создания файла «package.json» необходимо ввести в строку терминале команду «npm init» и нажать клавишу <Enter>:

npm init

1. В терминале появится следующий текст (Рисунок 9).

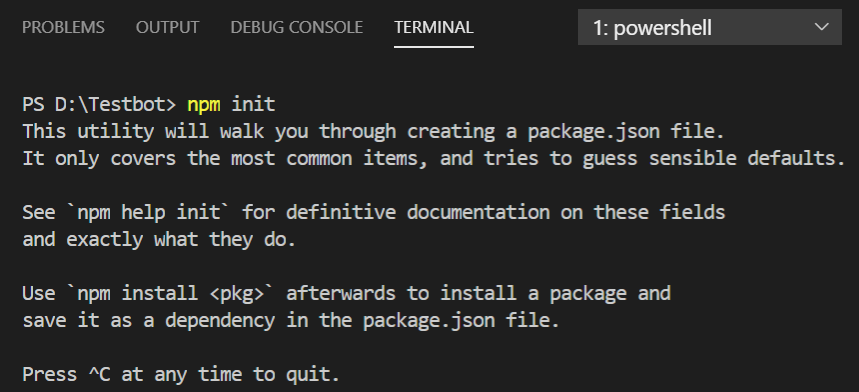


Рисунок . Начало выполнения команды «npm init»

1. Далее последовательно будут появляться строки с именами параметров бота. Можно оставить все параметры по умолчанию (можно указать свои значения) и нажимать после каждой строки клавишу «Enter».

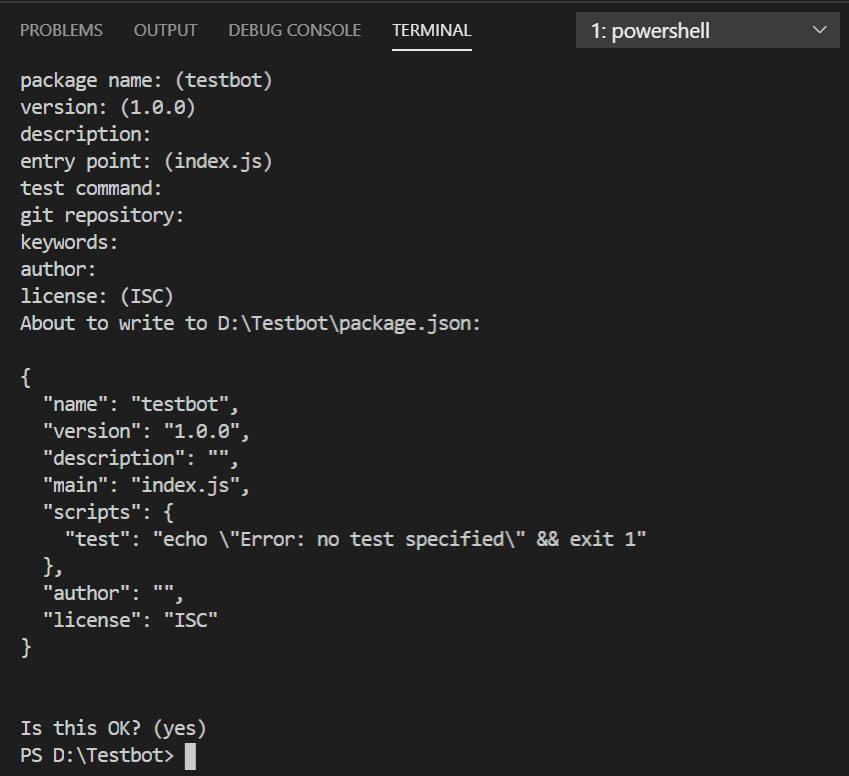


Рисунок . Результат выполнения команды «npm init»

1. Создать файл «package-lock.json». Для этого следует ввести в терминале команду «npm install» и нажать клавишу <Enter>:

npm install

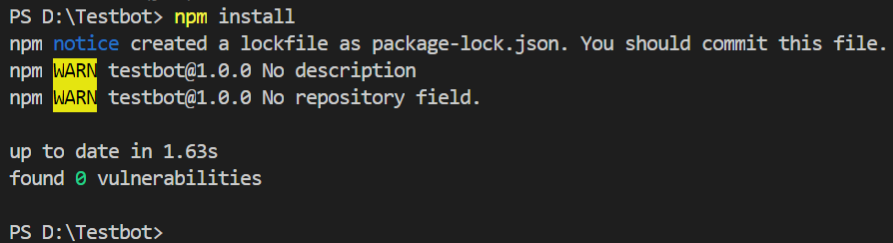
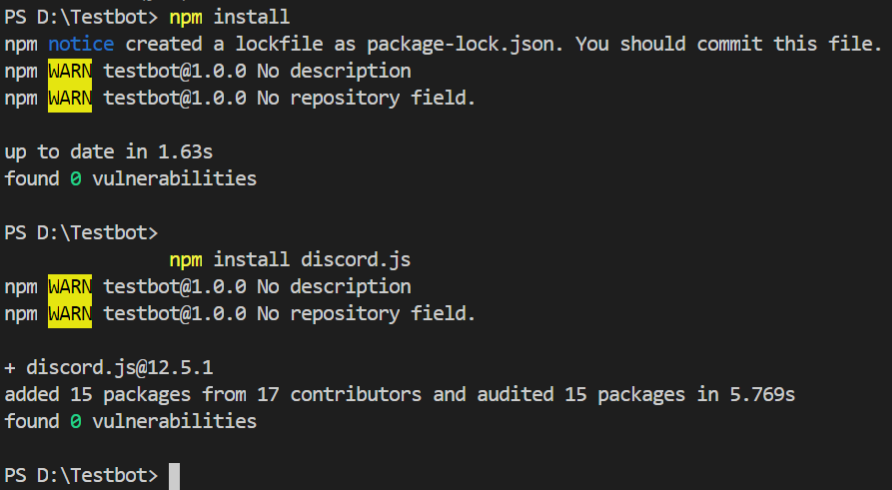


Рисунок . Создание файла «package-lock.json»

1. Установить пакет «discord.js», который будет использоваться для взаимодействия с Discord API. Для этого следует ввести в терминале команду «npm install discord.js» и нажать клавишу <Enter>:

npm install discord.js



1. Если все действия были выполнены верно, то в папке, созданной в подпункте 1 пункта 1.4.1, должны были появиться 3 объекта:

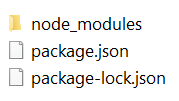


Рисунок . Созданные файлы и папки проекта

Таким образом, был настроен файл конфигурации и установлены необходимые зависимости.

### Программирование и сборка бота

В реальном приложении код бота делится на несколько файлов, но в данном обучающем примере весь код бота размещен в одном файле.

#### Программирование бота

1. Создать файл «config.json». Для этого в диалоговом окне «Testbot – Visual Studio» выполнить команду «File – New File». Откроется файл «Untitled-1».
2. Добавить в окно файла следующий код:

{

"token" : "Ваш\_токен",

"prefix" : "Ваш\_префикс"

}

где вместо параметров «Ваш\_токен» и «Ваш\_префикс» нужно ввести данные для созданного бота.

* 1. Для получения токена необходимо перейти на портал разработчиков Discord (см. пункт 1.3, подпункт 1), открыть вкладку «Bot» и щелкнуть по кнопке «Copy» (Рисунок 13).

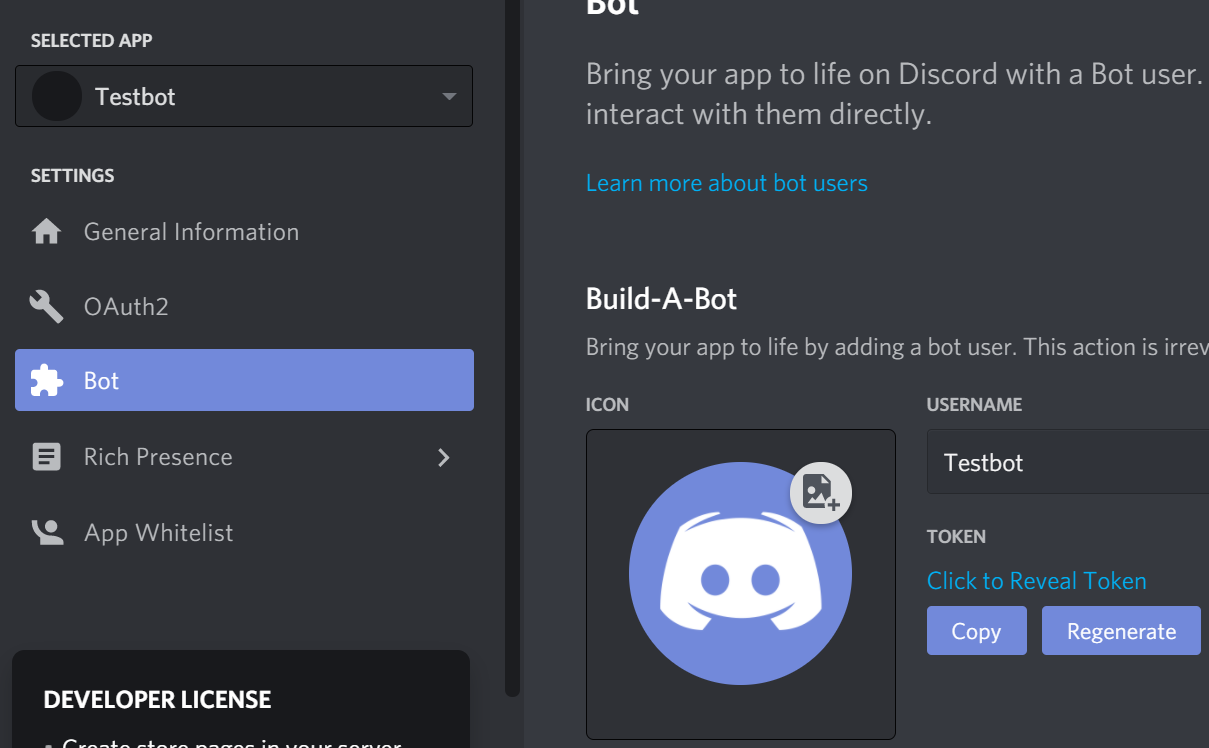


Рисунок . Копирование токена

* 1. В качестве префикса можно использовать любой символ. Самым распространённым среди разработчиков префиксом является символ «!».

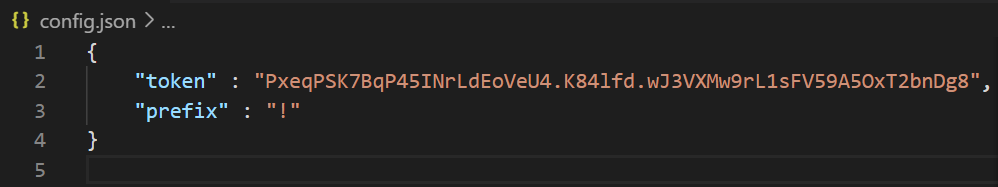


Рисунок . Код с добавленными параметрами бота

1. Сохранить созданный файл «Untitled-1» под именем «config.json». Для этого в диалоговом окне «Testbot – Visual Studio» выполнить команду «File – Save» (или нажать сочетание клавиш <Ctrl> + <s>) и в открывшемся диалоговом окне в поле «Имя файла» ввести «config» (без кавычек), а в раскрывающемся списке «Тип файла» выбрать вариант «JavaScript (\*.js; \*.es6; \*.mjs; \*.cjs; \*.pac)» (см. Рисунок 15).

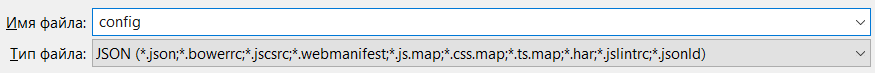


Рисунок . Сохранение файла «config.json»

1. Создать файл «bot.js». Для этого в диалоговом окне «Testbot – Visual Studio» выполнить команду «File – New File». Откроется файл «Untitled-1».
2. Добавить в окно файла следующий код:

const Discord = require('discord.js'); // Подключение библиотеки discord.js

const robot = new Discord.Client(); // Создание клиента-бота

const comms = require("./comms.js"); // Подключение файла с командами для бота

const fs = require('fs'); // Подключение модуля файловой системы node.js

let config = require('./config.json'); // Подключение файла с параметрами и информацией

let token = config.token; // Использование токена, указанного в файле

let prefix = config.prefix; // Использование префикса, указанного в файле

robot.on("ready", function() {

/\* Вывод в консоли сообщения «[Имя бота] запустился!», если запуск бота был успешен\*/

console.log(robot.user.username + " запустился!");

});

robot.on('message', (msg) => { // Реакция бота на сообщения

if (msg.author.username != robot.user.username && msg.author.discriminator != robot.user.discriminator) {

var comm = msg.content.trim() + " ";

var comm\_name = comm.slice(0, comm.indexOf(" "));

var messArr = comm.split(" ");

for (comm\_count in comms.comms) {

var comm2 = prefix + comms.comms[comm\_count].name;

if (comm2 == comm\_name) {

comms.comms[comm\_count].out(robot, msg, messArr);

}

}

}

});

robot.login(token); // Авторизация бота

1. Сохранить данный файл под именем «bot.jc». Для этого выполнить в диалоговом окне «Testbot – Visual Studio» команду «File – Save» (или нажать сочетание клавиш <Ctrl> + <s>) и в открывшемся диалоговом окне в поле «Имя файла» ввести «bot» (без кавычек), а в раскрывающемся списке «Тип файла» выбрать вариант «JavaScript (\*.js; \*.es6; \*.mjs; \*.cjs; \*.pac)» (см. Рисунок 16).

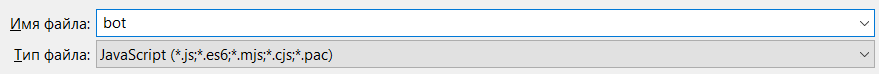


Рисунок . Сохранение файла «bot.jc»

#### Создание команд, которые будет выполнять бот

1. Создать файл «comms.js», в котором будет непосредственно реализован функционал бота за счет описания команд, выполняемых ботом. Для этого в диалоговом окне «Testbot – Visual Studio» выполнить команду «File – New File». Откроется файл «Untitled-1».
2. Добавить в окно файла следующий код:

const config = require('./config.json'); // Подключаем файл с параметрами и информацией

const Discord = require('discord.js'); // Подключаем библиотеку discord.js

const prefix = config.prefix; // «Вытаскиваем» префикс

// Команды //

function test(robot, mess, args) {

mess.channel.send('Test!')

}

// Список команд //

var comms\_list = [{

name: "test",

out: test,

about: "Тестовая команда"

}];

// Name - название команды, на которую будет реагировать бот

// Out - название функции с командой

// About - описание команды

module.exports.comms = comms\_list;

1. Сохранить данный файл под именем «comms.js». Для этого выполнить в диалоговом окне «Testbot – Visual Studio» команду «File – Save» (или нажать сочетание клавиш <Ctrl> + <s>) и в открывшемся диалоговом окне в поле «Имя файла» ввести «comms» (без кавычек), а в раскрывающемся списке «Тип файла» выбрать вариант «JavaScript (\*.js; \*.es6; \*.mjs; \*.cjs; \*.pac)» (см. ).

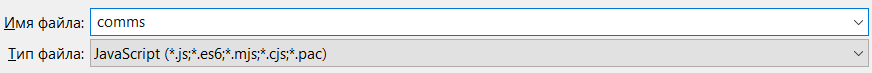


Рисунок . Сохранение файла «comms.js»

#### Запуск бота

1. В диалоговом окне «Testbot – Visual Studio» открыть новое окно терминала и ввести следующую команду:

node bot.js

1. Завершить ввод команды нажатием клавиши «Enter». Появится сообщение об успешном запуске бота (Рисунок 18).



Рисунок . Сообщение о запуске бота

## Проверка работы бота

1. Перейти в приложение Discord и убедиться, что созданный бот имеет статус «ONLINE» («В СЕТИ»).
2. Перейти в любой текстовый канал, ввести команду для бота в формате «префикс имя\_команды»:

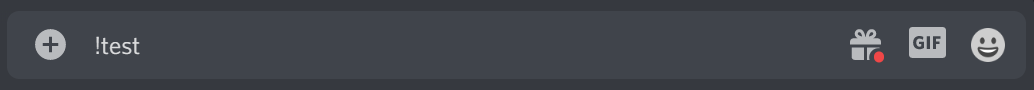


Рисунок . Ввод команды, которую может выполнить бот

1. Для выполнения введенной команды нажать клавишу <Enter>
2. Бот должен отреагировать на команду выводом сообщения «Test!» (см. Рисунок 20).

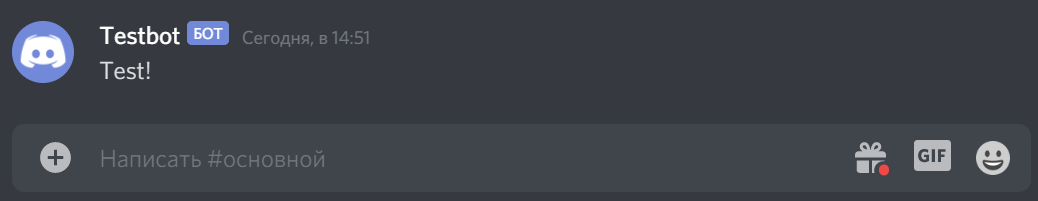


Рисунок . Результат выполнения команды

1. Для остановки всех процессов бота необходимо в окне «Testbot – Visual Studio» нажать сочетание клавиш <Ctrl> + <C>.

# Задание для самостоятельного выполнения

Самостоятельно реализовать дополнительную команду для бота. Описать процесс реализации команды и результат ее работы в отчете.

# Список литературы

1. Discord.js v12 documentation. URL: <https://discord.js.org/#/docs/main/stable/general/welcome>
2. Создание бота Discord Bot с помощью Node.js. URL: <https://www.digitalocean.com/community/tutorials/how-to-build-a-discord-bot-with-node-js-ru>
3. Документация API Discord. URL: [GitHub - discord/discord-api-docs: Official Discord API Documentation](https://github.com/discord/discord-api-docs)
4. Всё что вам нужно знать о Node.js URL: <https://habr.com/ru/post/460661/>
5. Создание Discord-бота, используя библиотеку discord.js | Часть №1. URL: <https://habr.com/ru/post/507948/>

1. Необходимо отметить, что некоторые из перечисленных продуктов не являются полноценными IDE, а представляют собой редакторы кода, для которых требуется устанавливать дополнительные плагины. [↑](#footnote-ref-1)