

Создание кнопочного телеграмм бота

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье содержится описание разработки кнопочного телеграмм бота помощника. Программа написана на языке программирования php. В результате исследования будут разобраны ключевые моменты, связанные с созданием установкой и логикой работы бота на сервере, а также методы хранения и автоматического изменения данных.

Ключевые слова: телеграмм бот, php, webhook, json.

Creating a push-button telegram bot

Vikhlyaev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

В данной статье содержится описание разработки кнопочного телеграмм бота помощника. Программа написана на языке программирования php. В результате исследования будут разобраны ключевые моменты, связанные с созданием установкой и логикой работы бота на сервере, а также методы хранения и автоматического изменения данных.

Keywords: telegram bot, php, webhook, json.

1 Введение

1.1 Актуальность

Кнопочный чат-бот или бот-клавиатура позволяет пользователям отвечать на вопросы через выбор конкретных ответов, которые представлены в виде кнопок с различными вариантами ответов. Такие боты используются в тех случаях, когда нужно предоставить пользователю доступ к определенным функциям или информации, и часто встречаются в мессенджерах. Удобство использования кнопочных ботов в том, что они упрощают процесс коммуникации между пользователем и ботом, что делает использование более удобным для людей, которые не уверены в своих технических навыках. Использование кнопок повышает эффективность, может помочь в быстром получении ответов на определенные вопросы, ускорить процесс выполнения задач или получения информации.

1.2 Обзор исследований

Н.Ю.Юрков применил библиотеки классов платформы.net и пакета telegram.bot при создании чат бота в telegram [1]. Р.И.Шакиров, А.С.Татаурова автоматизировали учебное расписание через telegram-bot [2]. А.С.Еринская, А.Р.Бакаева, Д.С.Скворцов использовали telegram-бота при обучении биологии [3]. М.И.Гурский создал чат-бота красноярского ГАУ в telegram [4]. М.А.Селина построила базу данных для бота telegram [5]. А.Г.Навасардян, Н.А.Зубач, Н.А.Хромова, А.В. Некрасов спроектировали базы данных для телеграмм-бота [6].

1.3 Цель исследования

Цель исследования – описать создание, логику работы и методы взаимодействия с пользователем кнопочного телеграмм бота.

2 Материалы и методы

Для реализации post запроса используется библиотека Curl. В качестве формата файлов для хранения данных применяется json. Взаимодействие между пользователем и ботом происходит через webhook.

3 Результаты и обсуждения

В качестве примера выбран телеграмм-бот, который оказывает помощь быстро и удобно получать информацию о банке и его предоставляемых услугах. Бот работает на основе кнопок, где каждая кнопка соответствует определенному разделу: услуги, контакты, карты, депозиты, отделения, кредиты и т.д. Таким образом, клиент сможет быстро получить информацию о том, что его интересует, без необходимости набирать длинные и сложные запросы.

Для создания и начала работы с телеграмм-ботом необходимо:

1. Зарегистрироваться в Telegram и создать свой аккаунт.
2. Создать нового бота через BotFather, который является официальным ботом Telegram для создания и управления телеграмм-ботами. Для этого нужно найти BotFather в Telegram и следовать инструкциям.
3. Задать имя и уникальный идентификатор для бота, который будет использоваться в дальнейшем для его поиска.
4. Получить токен от BotFather - это уникальный ключ доступа, который будет привязан к боту и требуется для возможности взаимодействия с ботом.

В данном примере взаимодействие между клиентом и ботом осуществляется через Webhook. Обработанный ответ клиенту бот отправляет через post запрос, с помощью модуля Curl. Запрос отправляется на сервер телеграмма с указанием токена бота, метода и самого запроса (рис.1).

```
private function post_query($getQuery, $method="/sendMessage?")
{
    $ch = curl_init("https://api.telegram.org/bot". TG_TOKEN . $method . http_build_query($getQuery));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HEADER, false);

    $resultQuery = curl_exec($ch);
    curl_close($ch);
    return $resultQuery;
}
```

Рис. 1. Реализация post запроса на сервер телеграмма

Основным методом для отправки сообщений пользователя служит «send/Message». Чтобы отправить простой текст нужно в параметрах запроса указать идентификатор чата пользователя и сам текст, дополнительно есть параметр осуществляющий парсинг некоторых html тегов в отправляемом тексте и реализующий их действия (рис.2).

```
public function send_message($message,$chat_id)
{
    $getQuery = array(
        "chat_id" => $chat_id,
        "text" => $message,
        "parse_mode" => "html",
    );

    return $this->post_query($getQuery);
}
```

Рис. 2. Запрос сообщения пользователю

Помимо простого текста, телеграмм позволяет отправлять клавиатуру в сообщении вместе с текстом. В таком случае к запросу добавляется параметр, в который необходимо указать массив в состоящий из строк и столбцов кнопок в формате json (рис.3).

```
public function send_button($message,$list_button,$chat_id)
{
    $getQuery = array(
        "chat_id" => $chat_id,
        "text" => $message,
        "parse_mode" => "html",
        'reply_markup' => json_encode(
            array(
                //массив клавиатуры
                'inline_keyboard' => $list_button
            )
        )
    );
    return $this->post_query($getQuery);
}
```

Рис. 3. Запрос клавиатуры в сообщении пользователю

Существует также возможность создать клавиатуру непосредственно в текстовой строке телеграмм бота. Для этого придется лишь изменить название клавиатуры (рис.4).

```
public function set_keyboard($message,$keyboard,$chat_id)
{
    $data = [
        'keyboard' => $keyboard,
        'resize_keyboard' => true,
        'one_time_keyboard' => true,
    ];

    $getQuery = array(
        "chat_id" => $chat_id,
        "text" => $message,
        "parse_mode" => "html",
        'reply_markup' => json_encode($data)
    );
    return $this->post_query($getQuery);
}
```

Рис. 4. Запрос на установку клавиатуры для телеграмм бота

Вся информация о кнопках и тексте хранятся в json-файлах. Такое хранение позволяет легко добавлять новые разделы и редактировать имеющиеся.

Когда пользователь впервые заходит в телеграмм-бота ему предлагается начать, нажав кнопку старт. В результате бот отправляет приветственное сообщение и создаёт клавиатуру у пользователя для более удобного общения (рис.5,6,7).

```
{
  "text": "Добро пожаловать в Помощник банка!\n\nЯ всегда готов помочь вам ответить на ваши вопросы и решить\n"
  "keyboard":
  {
    {
      "text": " Главное меню"
```

Рис. 5. Хранение текста и клавиатуры в json-файле

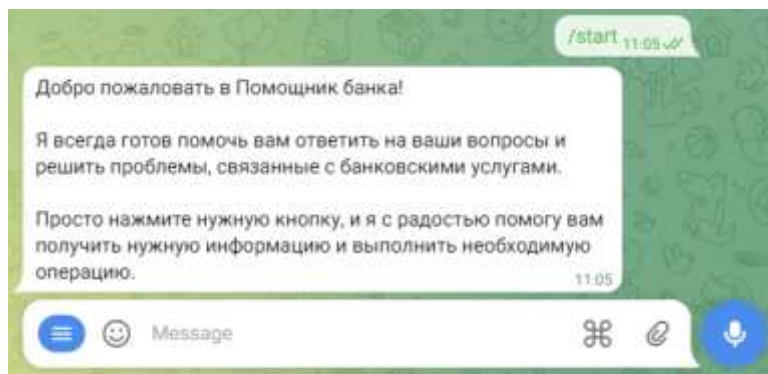


Рис. 6. Ответ бота на сообщение /start

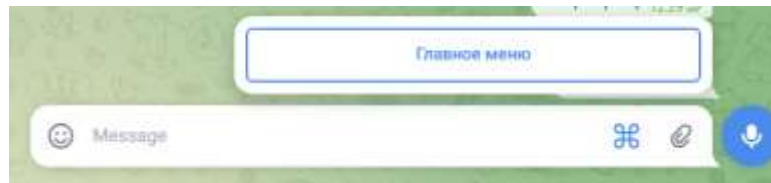


Рис. 7. Установка клавиатуры

При отправке команды меню, бот находит соответствующее имя файла и отправляет его содержимое пользователю. Клавиатуру представляет из себя трёхмерный массив, хранящий столбцы, строки и данные соответственно. Кнопка имеет текст с названием и возвращаемое значение, приходящее боту, при нажатии на кнопку (рис.8,9).

```
{
  "text": "О чем ты хочешь узнать подробнее?",
  "list_button": [
    [
      {
        "text": "📅 Счета и услуги",
        "callback_data": "create_a_schedule"
      }
    ],
    [
      {
        "text": "🏨 Подача заявления",
        "callback_data": "hotel_booking"
      }
    ],
    [
      {
        "text": "☎ Контакты и поддержка",
        "callback_data": "restaurant_reservations"
      }
    ],
    [
      {
        "text": "🎁 Блокировка карты",
        "callback_data": "selection_of_gifts"
      }
    ],
    [
      {
        "text": "👤 Курсы валют",
        "callback_data": "create_a_guest_list"
      }
    ]
  ]
}
```

Рис. 8. Содержание json данных списка меню

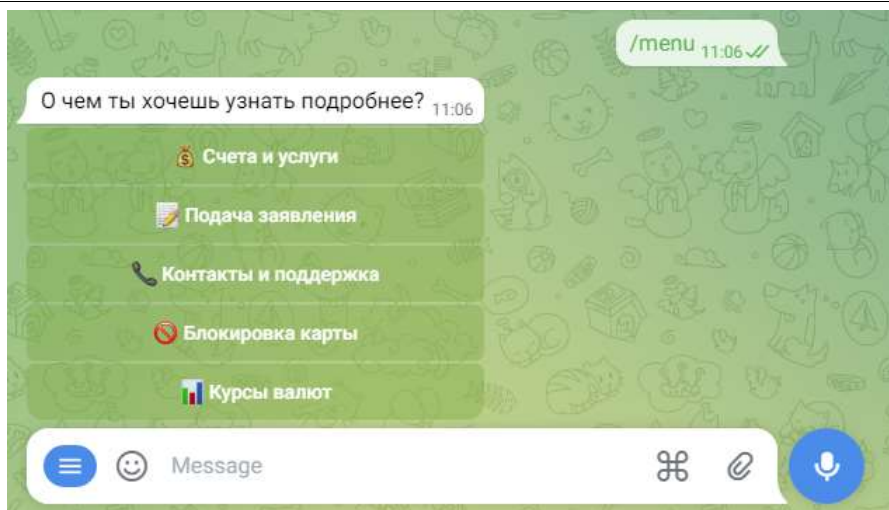


Рис. 9. Ответ бота на сообщение /menu

В начале работы php скрипт прослушивает присылаемые запросы. Затем проверяет json данные на наличие определённых полей и извлекается идентификатор чата, на который затем отправится ответ.

Если вернулось значение нажатой кнопки, то проверяется наличие данного файла на сервере и в случае успеха, их содержание отправляется как ответ пользователю.

Если же пришло сообщение, осуществляется проверка псевдонимов команд. Чтобы клавиатура была читаемой, кнопки написаны на русском языке. В таком случае возникает необходимость использовать отдельный файл, где хранятся несколько разных вариаций одной и той же команды. Далее проверяется наличие данного файла на сервере и в случае успеха, их содержание отправляется как ответ пользователю (рис.10,11).

```
$data = file_get_contents('php://input');
$data = json_decode($data, true);
writeLogFile($data, true);

if (!$data)
{
    // receive wrong update, must not happen
    exit;
}

if($data["callback_query"])
{
    $chat_id=$data["callback_query"]["message"]["chat"]["id"];
    $filename = "callback/".$data["callback_query"]["data"].".json";
    $response=file_processing($filename);
    answer_bot($response,$chat_id);
}

else if($data["message"])
{
    $chat_id=$data["message"]["chat"]["id"];
    $aliases = json_validation("aliases.json");
    $text=$data["message"]["text"];
    $message = isset($aliases["alias"][$text]) ? $aliases["alias"][$text] : $text;
    $filename="command".$message.".json";
    $response=file_processing($filename);
    answer_bot($response,$chat_id);
}

else
{
    exit;
}
```

Рис. 10. Логика обработки данных пришедших от пользователя

```
{
  "alias":
  {
    "Главное меню": "/menu",
    "меню": "/menu",
    "Начать": "/start",
    "старт": "/start",
    "помощь": "/help"
  }
}
```

Рис. 11. Содержание json данных псевдоним команд

В целях безопасности необходима проверка наличия и валидности файлов, чтобы возможно было получить и отправить ответ. В случае некорректной команды, пользователю отправляется ответ с исключением (рис.12,13).

```
function json_validation($file)
{
    //конвертация json в ассоциативный массив
    $json = json_decode(file_get_contents($file),true);
    //проверка файла json
    if($json)
    {
        return $json;
    }
    else
    {
        $error = "Файл ".$file." не определен или не является корректным форматом json!";
        writeLogFile($error, true);
        return null;
    }
}

function file_processing($file)
{
    $response=file_exists($file) ? json_validation($file) : json_validation("exception.json");
    //значение "text" обязательно должно присутствовать в json файле
    if($response["text"])
    {
        return $response;
    }
    else
    {
        return ["text"=> "ошибка"];
    }
}
```

Рис. 12. Проверка корректности файлов

```
{
    "text": "Прости, я тебя не понял.\nПопробуй еще! Нажми «Главное меню»."
}
```

Рис. 13. Содержание json файла исключения

В зависимости от наличия того или другого поля в файле отправляется команда на исполнение (рис.14).

```
function answer_bot($response,$chat_id)
{
    $bot = new TgBot();

    if($response["list_button"])
    {
        $bot->send_button($response["text"],$response["list_button"],$chat_id);
    }
    else if ($response["keyboard"])
    {
        $bot->set_keyboard($response["text"],$response["keyboard"],$chat_id);
    }
    else
    {
        $bot->send_message($response["text"],$chat_id);
    }
}
```

Рис. 14. Вызов метода для ответа бота

Ниже представлены содержания некоторых файлов и результат их исполнения в чате телеграмма (рис.15,16,17,18).


```
"text": "Чтобы найти нужную тебе информацию, выбери интересующий раздел:",  
"list_button":  
[  
  {  
    "text": "Кредиты",  
    "callback_data": "conferences"  
  },  
  {  
    "text": "Специальных предложения",  
    "callback_data": "seminars"  
  },  
  {  
    "text": "Информация о тарифах на банковские услуги",  
    "callback_data": "trainings"  
  },  
  {  
    "text": "Сведения об услугах по карточкам",  
    "callback_data": "meetings_with_clients"  
  },  
  {  
    "text": "Онлайн-банкинг",  
    "callback_data": "training_sessions"  
  },  
  {  
    "text": "Назад",  
    "callback_data": "menu"  
  }  
]
```

Рис. 15. Содержание json данных из списка меню «счета и услуги»

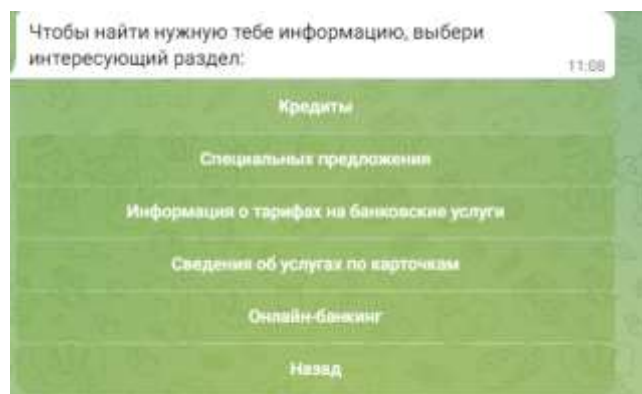


Рис. 16. Ответ бота на нажатие кнопки «счета и услуги»

```
{  
  "text": "Хотите управлять своими финансами быстро и удобно? Вам поможет наш онлайн-банкинг!",  
  "list_button":  
  [  
    {  
      "text": "Назад",  
      "callback_data": "entrance"  
    }  
  ]  
}
```

Рис. 17. Содержание json данных информации про «онлайн-банкинг»

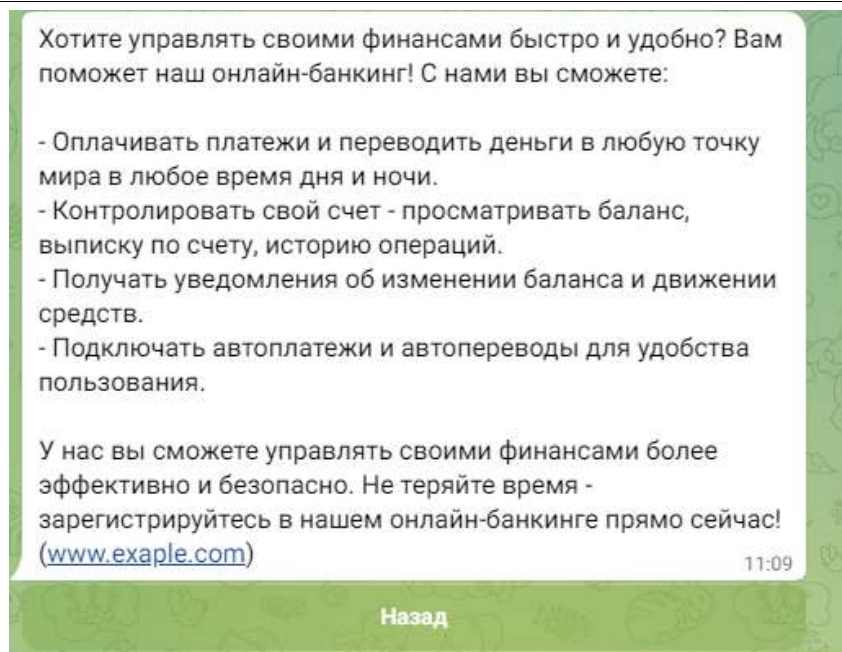


Рис. 16. Ответ бота на нажатие кнопки «онлайн-банкинг»

Таким образом, была рассмотрена программная реализация кнопочного телеграмм-бота помощника банка. В результате были описаны шаги по созданию телеграмм-бота, способу получения запросов от пользователя, хранению заранее заготовленной информации и отправлению ответов в чат пользователя.

Библиографический список

1. Юрков Н.Ю. Применение библиотеки классов платформы.net и пакета telegram.bot при создании чатбота в telegram // В книге: Новые информационные технологии в научных исследованиях НИТ-2021. материалы XXVI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязань, 2021. С. 9-10.
2. Шакиров Р.И., Татаурова А.С. Автоматизация учебного расписания через telegram-bot // Символ науки: международный научный журнал. 2020. № 1. С. 37-41.
3. Еринская А.С., Бакаева А.Р., Скворцов Д.С. Использование telegram-бота при обучении биологии // В сборнике: Проблемы биологического и экологического образования школьников и студентов. Сборник статей Всероссийских с международным участием студенческих Герценовских чтений, посвященных 100-летию кафедры методики обучения биологии и экологии РГПУ им. А.И. Герцена. Санкт-Петербург, 2022. С. 67-69.
4. Гурский М.И. Создание чат-бота красноярского ГАУ в telegram // В сборнике: Science start up: students' meeting in Siberia. Материалы сибирского международного студенческого аграрного форума. Красноярск, 2023. С. 203-205.
5. Селина М.А. Создание базы данных для бота telegram // В сборнике: IX

- Международная научно-практическая конференция имени академика А.Г. Шипунова. Материалы IX Международной научно-практической конференции . Орел, 2022. С. 269-273.
6. Навасардян А.Г., Зубач Н.А., Хромова Н.А., Некрасов А.В. Разработка базы данных для телеграмм-бота // В сборнике: Технологии 2022: основные проблемы и направления развития. Сборник статей II Международной научно-практической конференции. Пенза, 2022. С. 67-69.
 7. Шакиров Р.И., Татаурова А.С. Автоматизация учебного расписания через telegram-bot // Символ науки: международный научный журнал. 2020. № 1. С. 37-41.
 8. Харевич А.А., Кумратова А.М. Telegram bot api как средство маркетинга и менеджмента // В сборнике: информационное общество: современное состояние и перспективы развития. сборник материалов X международного студенческого форума. 2018. С. 201-202.