

РАЗРАБОТКА ЧАТ-БОТА, ЕГО ВОЗМОЖНОСТИ И ОРГАНИЗАЦИЯ ПОЛНОЦЕННОГО ЧАТА С ОПЕРАТОРОМ

Авторы: Еськин Егор Ильич (Нижевартовский государственный университет)
Катермина Татьяна Сергеевна (Нижевартовский государственный университет)

Аннотация: Статья посвящена разработке чат-бота для любых коммерческих или информативных компаний. В статье пошагово описывается процесс разработки с наглядным результатом. Научная новизна работы заключается в применении языка C# для программирования чат-бота для оповещения пользователей. В результате реализован чат-бот, организовано подключение к тестовой базе данных, организовано распределение свободных операторов с пользователями.

Ключевые слова: чат-бот, чат для сайта, чат-консультант, техническая поддержка, vkNet.

Annotation: The article is devoted to the development of a chatbot for any commercial or informative companies. The article describes the development process step by step with a clear result. The scientific novelty of the work is the use of the C# language for programming a chatbot to alert users. As a result, a chatbot is implemented, connection to the test database is organized, and distribution of free operators with users is organized.

Keywords: Chat bot, website chat, chat consultant, technical support, vkNet.

Чат-бот – виртуальный помощник, созданный для привлечения и помощи клиентам, выполняющий оповестительные функции. С каждым годом увеличивается повсеместное использование чат-ботов для упрощения взаимодействия с пользователями на сайтах. Согласно данным компании GfK, которая фиксирует эволюцию потребительских ценностей, назвала этот тренд «моментальная доступность».

Для реализации подобного чат-бота необходимо:

- Среда разработки Visual Studio;
- Учетная запись в Вконтакте;
- Система управления пакетами NuGet.

Учетная запись в Вконтакте необходима для использования VK API, который содержится в пакете VkNet. Чтобы установить VkNet, должна быть произведена интеграция среды разработки Visual Studio и NuGet.

Первым шагом для реализации чат-бота является создание и настройка сообщества, цель которого получить ключ доступа и осуществить выбор событий для сервера-слушателя. Далее производится установка пакета VkNet через NuGet. NuGet — система управления пакетами для платформ разработки Microsoft, в первую очередь библиотек .NET Framework. Управляется .NET Foundation [1], а VkNet - библиотека для работы с API Вконтакте [2].

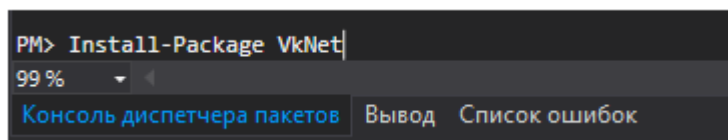


Рис. 1. Установка пакета VkNet

Второй шаг заключается в создании сервера и встраивании виджета на сайт. Это требует знаний начинающего уровня в языке C# и html.

Для начала необходимо создать публичное сообщество (для получения ключа доступа). Ключ-токен находится в разделе Работа с API в управлении сообщества.

Следующим этапом является включение Long Poll API и настройка типов обрабатываемых событий, который будет получать сервер-слушатель.

Во втором шаге создания сервера необходимо установить пакет VkNet через NuGet для работы API и объявить новый объект VkApi в классе консольного приложения:

```
public static VkApi api = new VkApi();
```

Далее авторизоваться используя ключ-токен группы который получили на странице Работы с API:

```
api.Authorize(new ApiAuthParams() {  
    AccessToken = Ключ-токен });
```

После авторизации необходимо в бесконечном цикле отправлять запросы на сервер Вконтакте. Для этого используется цикл while.

```
while (true)  
{  
    var s = api.Groups.GetLongPollServer(ID);  
    var poll = api.Groups.GetBotsLongPollHistory(  
        new BotsLongPollHistoryParams()  
        {  
            Server = s.Server,  
            Ts = s.Ts,  
            Key = s.Key,  
            Wait = 25  
        });  
    if (poll?.Updates == null) continue;  
}
```

Рис. 2. Запрос на сервер Вконтакте

В ID содержится id вашей группы, в которой вы авторизованы. Проверка `poll?.Updates == null` проверяет на наличие новых событий, которые произошли в группе. Если их нет, то продолжается ожидание и запускается вновь. В переменную `s` записываются данные для подключения к серверу: сервер, ключ.

Wait – время ожидания и её рекомендовано назначать 25 секунд, то есть каждые 25 секунд бот отправляет запрос на сервер Вконтакте и начинает ждать событие. Ts – номер последнего события, начиная с которого нужно получать данные. Это позволяет не обрабатывать сообщения одной и той же команды.

Далее необходимо создать цикл который будет соотносить новые события с получением нового сообщения от пользователя, для этого создаем цикл foreach и ставим проверку на новое сообщение.

```
foreach (var a in poll.Updates)
{
    if (a.Type == GroupUpdateType.MessageNew)
    {
        //Здесь происходит реализация ответов на новые сообщения
    }
}
```

Рис. 3. Проверка на новое сообщение

После получения нового сообщения нужно изъять текст сообщения и id человека, который его отправил. Для этого необходимо обратиться к данным, полученным от a.MessageNew.Message.Text и a.MessageNew.Message.FromId; в них содержатся текст и id соответственно.

Далее необходимо написать статический метод, который будет отправлять сообщение (рис. 4).

```
public static void SendMessage(string message, long? userID)
{
    Random rnd = new Random();
    api.Messages.Send(new MessagesSendParams
    {
        RandomId = rnd.Next(),
        UserId = userID,
        Message = message
    });
}
```

Рис. 4. Статический метод отправки сообщения пользователю

Random необходим для уникальности сообщения и избежания дублирований сообщений от бота, UserId является Id пользователя, который отправил сообщение, Message — это сообщение, которое будет отправляться.

Поскольку бот должен быть интерактивный, делаем ветки условий сравнения сообщений, на которые бот должен отвечать. Для этого вначале изменим приветственное сообщение на список команд для бота, так как при начале первого диалога у бота имеется кнопка с отправляющимся текстом «начать». Затем создадим условие на данную фразу — для этого в цикле foreach, где выполняется условие a.Type == GroupUpdateType.MessageNew добавим следующие строчки кода:

```
if (userMessage == "начать" )
```

```
{
SendMessage("Информация необходимая пользователю.", userID);

break;

}
```

Таким образом происходит оповещение пользователя о возможностях нашего бота с помощью метода SendMessage. Break в данном случае необходим, чтобы запустить цикл ожидания заново и работы остальных условий.

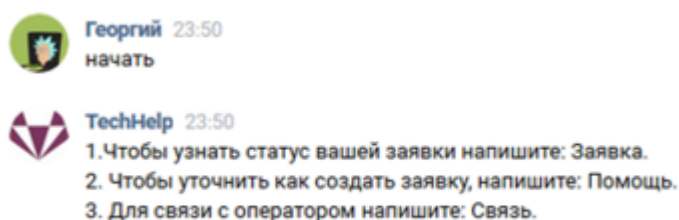


Рис. 5. Пример ответа чат-бота на слово начать

Связь с оператором происходит при вводе пользователем сообщения “связь”. Наличие операторов проверяется с помощью списка ID пользователей, которые вступили в группу и разрешили получение сообщений от неё. Список в данном примере хранится в папке с исполняемым файлом. При наличии операторов пользователь оповещается об ожидании, а оператор из списка получает сообщение о наличии нового пользователя и ссылку на диалог с пользователем от имени группы. Далее происходит связь непосредственно с оператором.

```
if (userMessage == "связь")
{
    ModerList = (long)(ModerListStr[rand.Next(0, m3.Length)]);
    SendMessage("У вас новый пользователь https://vk.com/gim\*\*\*?sel=" + userID + " ", ModerList);
    // Вместо *** необходимо написать id вашего сообщества
    SendMessage("Ожидайте ответа оператора.", userID);
    break;
}
```

Рис. 6. Пример организации связи с операторами.

После настройки программной части и её запуска необходимо настроить виджет для сайта. Для этого необходимо перейти на сайт www.vk.com/dev/CommunityMessages, выбрать сообщество и настроить оформление виджета, далее вставить его на сайте с помощью уже предоставленного готового кода.

Использование чат-бота благоприятно влияет на статистику пользователей, хотя он ещё редкость на российском рынке. Он выступает как отличный гид с удобным чатом, который ответит на часто задаваемые вопросы, а при возникновении

неизвестных ему, переключает на операторов. Это должно снизить количество звонков и распределить нагрузку операторам.

Литература

1. NuGet — Wikipedija URL: <http://www.consultant.ru/> (data obrashhenija: 10.02.2018).
2. [C#] VKNET dlja C# URL: [https://pawno-info.ru/threads/309386-VKNET-dlja-C data](https://pawno-info.ru/threads/309386-VKNET-dlja-C-data) (data obrashhenija: 10.02.2018).