

РАЗРАБОТКА ЧАТ-БОТА ДЛЯ ПЛАТФОРМЫ TWITCH.TV

В.А. Щукин

Научный руководитель: А.О. Савельев

Томский политехнический университет

vas74@tpu.ru

Введение

К 2018-му году уже более 1,8 млрд. людей используют мессенджеры [1]. Данные сервисы дали толчок к развитию целой отрасли в разработке программного обеспечения – чат-ботам. Во многих случаях удобнее автоматизировать общение с пользователем, что снижает потребность в человеческих ресурсах и позволяет обрабатывать десятки тысяч запросов в сутки, используя всего одну программу. Такие «собеседники», имитирующие поведение человека, называются чат-ботами и становятся все более универсальным решением для огромного спектра задач. Посредством текста или голоса, чат-бот может принимать заказы на доставку продуктов, создавать электронную очередь, осуществлять контроль сообщений (фильтрация спама), быть когнитивным ассистентом.

Описание предметной области

Стриминговая платформа Twitch.tv – мировой лидер по предоставлению услуг прямой трансляции пользователям через сеть Интернет. За трансляцией может наблюдать неограниченное число зрителей (во время масштабных спортивных событий данные цифры достигают сотен тысяч). При просмотре трансляции у пользователей есть возможность использовать чат. Автоматизация общения с людьми в чате и внедрение ботов – один из векторов развития данной платформы и сообщества, т.к. благодаря данным программам человек, ведущий эфир, не вынужден отвлекаться и многократно отвечать на похожие запросы пользователей. Наиболее популярные функции чат-ботов: модерация сообщений, предоставление дополнительной информации (время эфира, краткая информация о ведущем). Для взаимодействия программы с серверами Twitch, платформа предоставляет открытый для разработчиков API – TwitchLib. Данный API представляет собой фреймворк для языка C# и находится в открытом доступе – достаточно загрузить данный фреймворк через модуль Nuget в среде MS Visual Studio.

Постановка задачи

Разработка чат-бота для стримингового сервиса Twitch, позволяющего автоматизировать ответ на ряд частых вопросов пользователей и разнообразить развлекательный контент трансляции. Индивидуальность чат-бота заключается в спектре выполняемых задач. Для данного проекта поставлены следующие функциональные требования к конечному продукту:

- Ответ на приветствия пользователей в чате;

- Отправка по команде статьи из новостной RSS ленты сайта, указанного в файле конфигурации (по умолчанию – Habr.com);

Список нефункциональных требований:

- В конце краткого содержания статьи содержится короткая ссылка на источник;
- Необходимо форматирование текста новости по тегам;
- Консольное приложение с отправкой отчетов о процессах авторизации и отправки сообщения в консоль;
- Ответ на запрос содержит имя пользователя, отправившего этот запрос;

После анализа технологий, требующихся для реализации продукта, была создана следующая схема разработки:

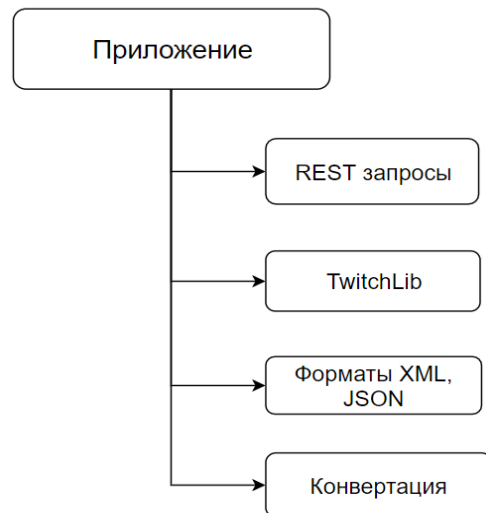


Рис. 2. Диаграмма используемых приложением технологий

Разработка приложения

Для программной реализации используется библиотека TwitchLib, находящаяся в открытом доступе, документация к которой располагается на сайте dev.twitch.tv [2]. Для использования данной библиотеки необходимо зарегистрировать приложение в системе dev.twitch, что позволяет получить уникальные токен и client-id для соединения с сервером.

Изучение документации TwitchLib позволило определить необходимые для разработки компоненты библиотеки, в числе которых классы: TwitchClient, ConnectionCredentials. После успешного соединения с сервером, приложение обрабатывает все поступающие от пользователей сообщения и в случае наличия команды вызывает её обработчик.

REST запросы, short URL

На начальном этапе создания бота было обнаружено, что у каждого отправляемого в чат сообщения существует лимит символов в размере 500. Одно из нефункциональных требований предусматривает, что в конце краткого описания статьи будет прилагаться ссылка на источник, однако полный URL страницы часто занимает большое количество символов. Решение данной проблемы – использование API сайта to.click [3], который позволяет с помощью REST запросов отправлять исходный URL на сервер и получать его короткий аналог для вставки в сообщение. Данная процедура позволяет сэкономить десятки символов и сделать результат визуально легким для восприятия. После сокращения ссылки, приложение делает повторную проверку длины сообщения. В случае выхода за лимит, последнее слово заменяется троеточием. Согласно описанным решениям, была создана диаграмма классов:

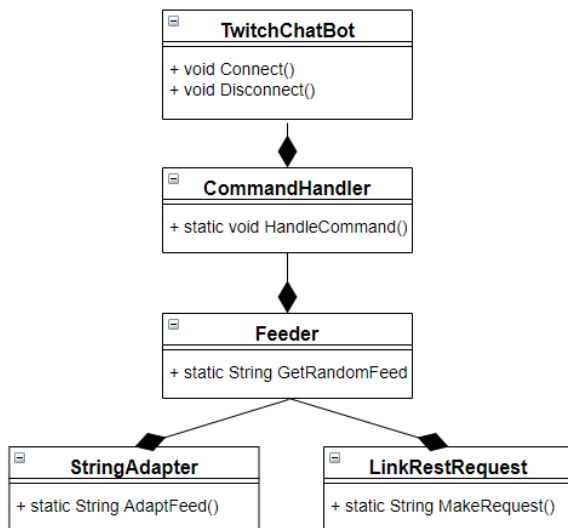


Рис. 2. Диаграмма классов

Ответ сервер to.click присылает в формате JSON и для него требуется конвертация. Самым популярным решением по работе с JSON форматом на языке C# является фреймворк от Newtonsoft. Данный пакет позволяет десериализовать файл в объект. Так как структура класса известна и не меняется со временем, при десериализации и работе с объектом используется класс dynamic (соответствие типов проверяется на этапе выполнения, а не компиляции).

Новость извлекается из RSS ленты сайта (по

умолчанию – Habr.com), данные предоставляются в формате XML. Файл данного формата содержит большое количество служебных символов, поэтому при обработке полученных данных строка фильтруется с помощью регулярных выражений (класс Regex).

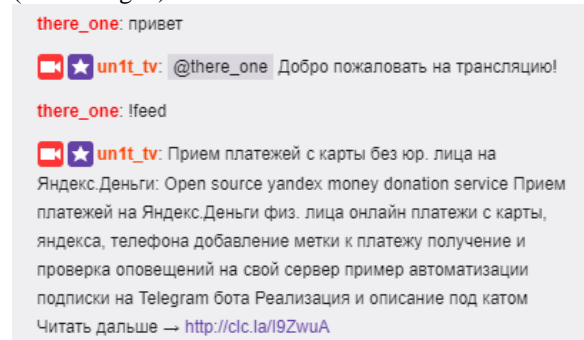


Рис. 3. Пример работы приложения

Заключение

В результате проведенной работы, была решена задача автоматизации приветствия пользователей в чате трансляции на платформе Twitch.tv, программный продукт нашел практическое применение и был протестирован в реальных условиях. Изучены механизмы обмена данными между сервером и клиентом и работа с форматами XML, JSON. Следующий этап разработки – реализация алгоритма идентификации спам-ботов среди других пользователей на основе алгоритма Байеса.

Список использованных источников

1. Что такое чат-боты и зачем они нужны? [Электронный ресурс] / Новостной ресурс informБИОРО. – URL: <https://informburo.kz/cards/chto-takoe-chat-boty-i-zachem-oni-nuzhny.html> (дата обращения 12.09.2018).
2. Twitch Developer Documentation [Электронный ресурс] / Twitch Developers Web-site – URL: <https://dev.twitch.tv/docs/> (дата обращения 26.08.2018).
3. Описание методов API [Электронный ресурс] / API документация сервиса to.click – URL: <https://to-click.gitbooks.io/api-ru/content/opisanie-metodov-api.html> (дата обращения 28.08.2018).
4. JObjectParse Method [Электронный ресурс] / Документация фреймворка Newtonsoft – URL: <https://www.newtonsoft.com/json/help/html/> (дата обращения 29.08.2018).