

Как работает бот

Что такое чат-бот

Чат-боты встречаются везде: виртуальные ассистенты есть на сайтах и в мессенджерах (Viber, WhatsApp, Telegram, VK), а некоторые крупные компании разработали своих голосовых помощников (Алиса, Маруся и другие). Они экономят проектам много ресурсов благодаря автоматизации рутинных дел, которые до этого выполняли операторы поддержки или диспетчеры. Чат-боты — это очень удобно, ведь пользователю не нужно куда-то звонить и ничего скачивать. Он может получить мгновенный ответ, зайдя в мессенджер, на сайт или задав вопрос голосовому ассистенту.

В этом материале мы рассмотрим внутреннее устройство чат-ботов и принципы их разработки, которые позволят сделать общение с ними приятным и продуктивным.

Цикл взаимодействия «клиент — сервер»

Прежде чем говорить о внутреннем устройстве ботов, давайте обозначим их роль в цепочке обмена информацией.

1. Пользователь взаимодействует с приложением телеграм-клиентом на своём устройстве (смартфон, персональный компьютер или веб-приложение).
2. Взаимодействие пользователя с сервисами происходит через серверы Telegram.
3. Бот, а точнее его бэкенд-программа, обрабатывает действия пользователей.

Когда пользователь взаимодействует с ботом через приложение Telegram (клиент), например пишет сообщение, нажимает на кнопки или выбирает команду в меню, приложение отправляет запрос телеграм-серверу. И он понимает, что это обращение к боту. Затем в зависимости от настроек сервер передаёт запрос боту или ждёт запрос от него, чтобы отправить ему данные о действиях пользователя.

Есть две основные технологии, с помощью которых бот получает информацию от сервера: **Long polling** и **Webhook**.

Long polling используется для постоянной проверки сервера на наличие обновлений для бота. Этот процесс можно представить в виде диалога между ботом и сервером.

Бот: «Есть ли для меня какие-то обновления?»

Сервер Telegram: «Пока нет».

Бот: «Есть ли для меня какие-то обновления?»

Сервер Telegram: «Ты получил сообщение! Пользователь с ID 12567 отправил команду /start».

Webhook — это противоположная Long polling технология. В этом случае сервер Telegram сам отправляет уведомление боту о наличии обновлений: «Ты получил сообщение!»

Вне зависимости от выбранной технологии бот получает обновления от сервера, обрабатывает их и отправляет ответ обратно серверу. Последний определяет, кому из пользователей адресовано сообщение, и передаёт ему в приложение. Весь процесс взаимодействия между ботом и сервером осуществляется с помощью HTTP-запросов.

Теперь когда вы понимаете, как устроен бот, давайте разберём, как между ним и пользователем происходят диалоги.

Лучшие принципы построения диалогов с ботом

Первый контакт с ботом

Первым делом нужно рассказать пользователю, что делает бот. Если на реплику «Привет!» бот отвечает так же, это, конечно, дружелюбно, но неинформативно — после него пользователю непонятно, что делать дальше. Именно поэтому важно рассказать ему о предназначении бота и его функциональности: обозначить, зачем он нужен и что умеет делать.

Хорошие примеры знакомства пользователя с ботом

- Пользователь: «Привет!»
Бот: «Привет! Я виртуальный ассистент компании XXX. Я отвечу на ваши вопросы о нашей компании, продукции и помогу сделать заказ. Какой у вас вопрос?»
- П: «Привет!»
Б: «Привет! Я бот по поиску отелей. Смогу подобрать для вас подходящие отели, где бы вы ни были. В каком городе вы хотите остановиться? Чтобы увидеть полный список моих функций, напишите /help».

Подсказки для пользователя

Во время диалога с пользователем его нужно постоянно направлять, потому что он не знает, как работает ваш бот. Для этого в конце каждого сообщения стоит оставлять рекомендации или подсказки в виде вопросов. Они выступают в роли указателей, которые помогают пользователю сориентироваться.

Пример плохого завершения диалога

- Бот: «Результаты по вашему запросу».

Это неудачный вариант завершения диалога, так как пользователю непонятно, что нужно делать дальше: снова ввести запрос или выбрать команду.

Пример хорошего завершения диалога

- Бот: «Результаты по вашему запросу. Чтобы продолжить, введите новый запрос или команду из списка».

Обработка ошибок

Бывают ситуации, когда пользователь пишет нестандартное сообщение или в программе что-то ломается, например работа с API. Важно обработать этот момент: написать, что ввод был некорректным, и направить пользователя в нужную сторону. Если что-то пошло не так, нужно предупредить об этом клиента и подсказать, что можно сделать.

Примеры плохого диалога

- Б: «Выберите количество товаров».
П: «Сколько времени?»
Б: «Извините, что-то пошло не так».
- Б: «Выберите количество товаров».
П: «750».Б: молчание.

Пользователь не понимает, что он сделал не так и что нужно делать дальше.

Пример хорошего диалога

- Б: «Выберите количество товаров».
П: «Сколько времени?»
Б: «Извините, я вас не понимаю. Введите количество товаров или напишите /help, чтобы получить справку».
- Б: «Выберите количество товаров».
П: «750».
Б: «Введите число от 1 до 100».

Вариативность ответов бота

Здорово, когда бот реагирует как на «Привет», так и на «Добрый день», а также отвечает по-разному. Это создаёт ощущение, что общаешься не с роботом, а с реальным человеком. С точки зрения разработки это достигается путём лемматизации сообщения пользователя, то есть разбивки текста на слова. Например, если в сообщении есть слово «Привет», скорее всего, это приветствие.

Пример плохой вариативности диалога

- П: «Привет, бот!»
Б: «Извините, я вас не понимаю».

Бот оказался не очень дружелюбным. Хорошо, когда бот умеет обрабатывать базовые фразы типа «Привет», «Пока», «Спасибо» и так далее.

Пример хорошей вариативности диалога

- П: «Привет, бот!»
Б: «Рад вас снова видеть, Александр!»

В этом материале вы получили общее представление о ботах. Пришло время его создать!