Глубокая интеграция цифрового диалога в бизнес-процессы: практический гайд

Для примера мы разберем интеграцию веб-сайта с чат-ботом для Telegram, но все сказанное ниже справедливо для любого мессенджера.

Зачем это нужно

Представьте, что вам необходимо реализовать один или сразу все из представленных ниже сценариев:

- Привязать веб-сайт к чат-боту, чтобы отправлять уведомления зарегистрированным пользователям веб-сайта прямо в их Telegram, например, сообщать о смене статуса заказа;
- Привязать веб-сайт к чат-бот, чтобы зарегистрированные пользователи веб-сайта могли работать с вашим веб-сайтом через чат-бот в мессенджере, используя чат-бот как диалоговый пользовательский интерфейс;
- Авторизоваться или зарегистрироваться на веб-сайте через чат-бот, без ввода логина и пароля на самом веб-сайте;
- Открыть страницу веб-сайта в чат-боте с графическим интерфейсом, например, показать список товаров;
- Зарегистрировать пользователя на веб-сайте через чат-бот, чтобы упростить путь пользователя.

Эти и другие подобные сценарии применения можно реализовать на платформе Metabot с помощью глубокой интеграции мессенджеров и веб-сайта (deep messaging integration). Вам потребуются базовые навыки программирования, умение работать с REST API и немного терпения.

Что это даёт

С помощью глубокой интеграции вы получите следующие возможности:

- Сможете упростить онбординг пользователей в ваш бизнес/на сайт, тем самым повысить конверсию и снизить стоимость привлечения пользователя;
- Сможете захватывать пользователей, которые находятся на очень раннем этапе в воронке продаж, например, тех кто не готов к покупке и обычно не регистрируется при переходе по рекламе;
- Сможете автоматизировать задачи на любых участках пути покупателя;
- Отправлять уведомления в мессенджер в любой момент.

Как это работает

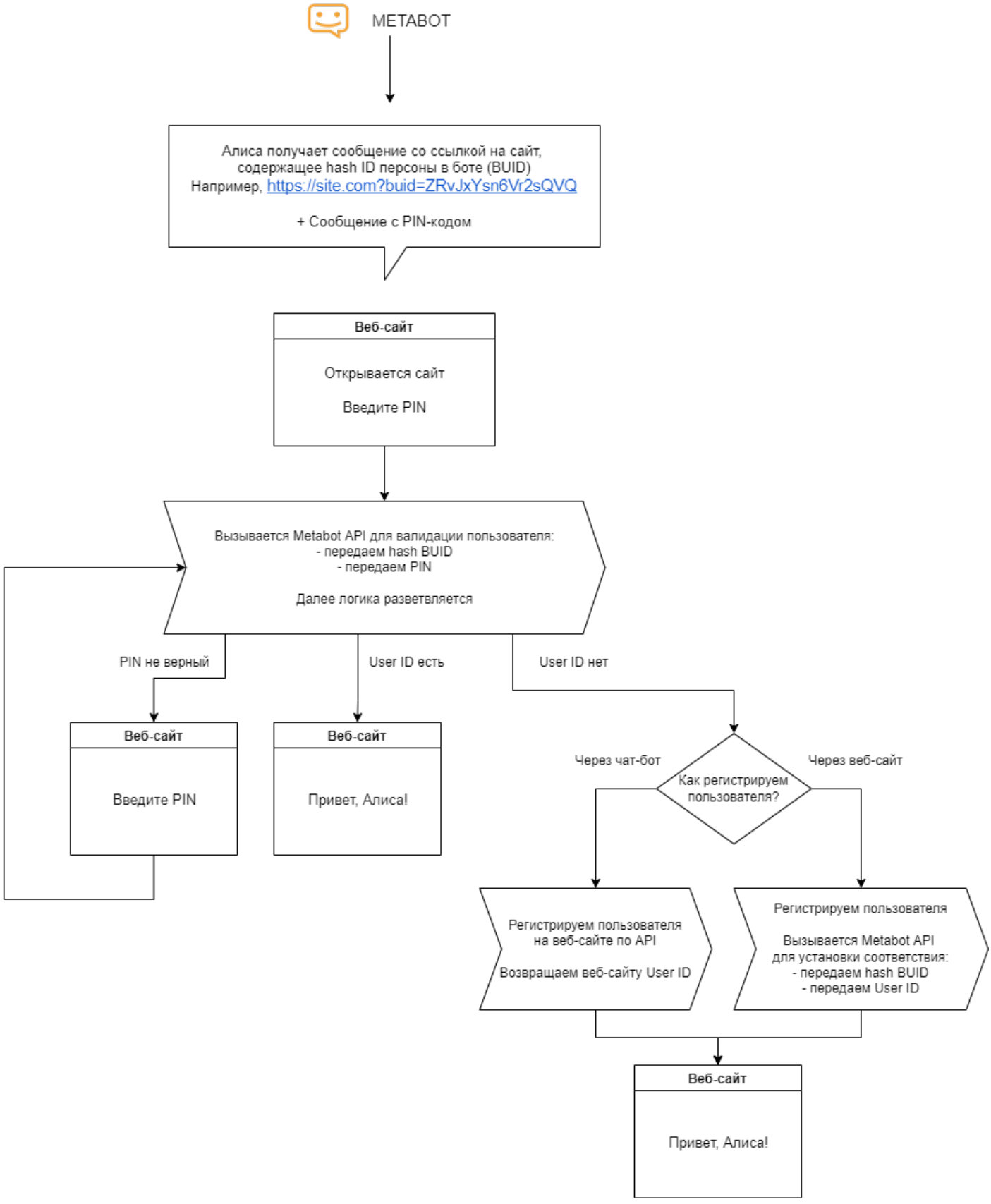
Все довольно просто! Хотя нам пришлось много потрудиться, чтобы сделать сложное простым.

Давайте пройдем по алгоритму используя вымышленную Алису, которая хочет одновременно пользоваться своим любимым сервисом как с помощью веб-сайта, так и с помощью чат-бота.

Основной момент, который важно понимать, заключается в том, что у пользователя (у Алисы) взаимодействие с веб-сайтом происходит в одном приложении т.е. в веб-браузере, а взаимодействие с чат-ботом в другом приложении т.е. в мессенджере.

Чтобы связать две разных системы вместе, мы будем на фронте последовательно переводить Алису из одного приложения в другое, а на бэке мы свяжем системы с помощью по REST API.

Схематично глубокую интеграцию можно изобразить следующим образом:



Теперь разберем логику интеграции подробнее.

В какой системе хранить ID

Есть три подхода связать чат-бот и веб-сайт вместе:

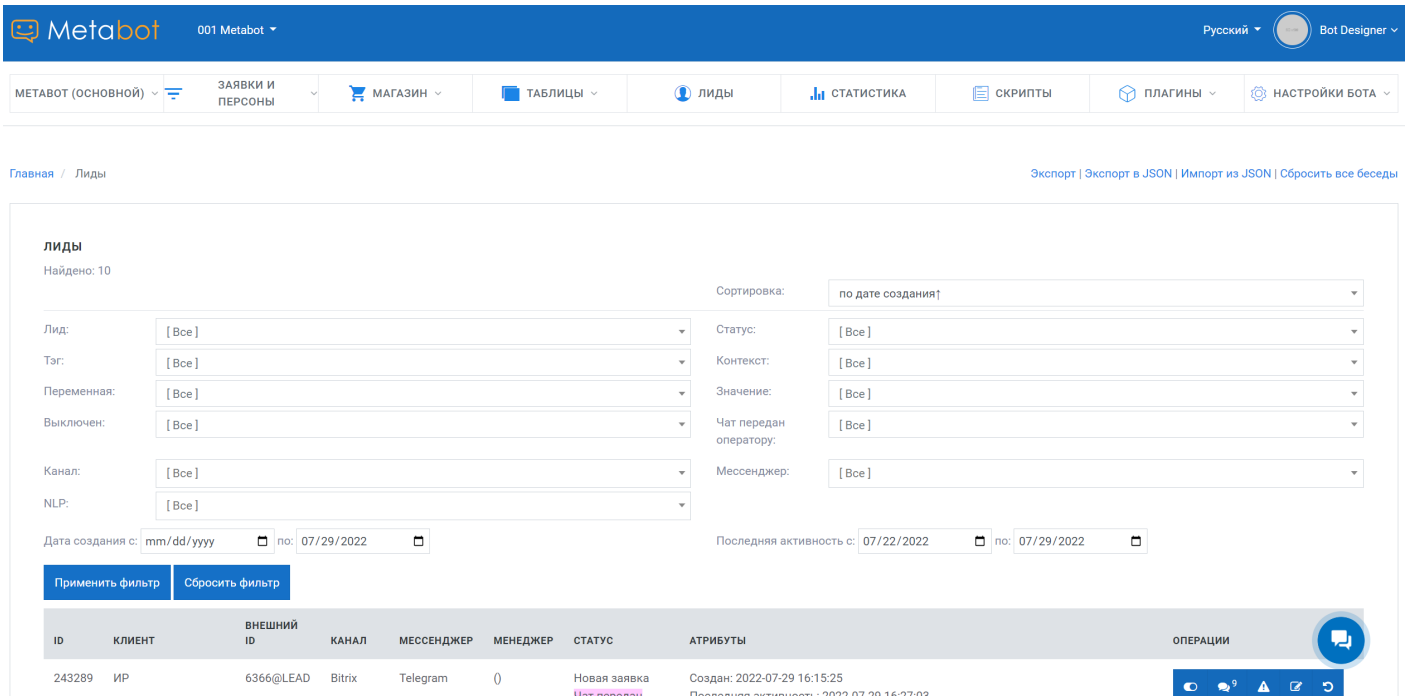
- Чат-бот хранит ID пользователя веб-сайта;
- Веб-сайт хранит ID персоны в чат-боте;
- Веб-сайт и чат-бот одновременно хранят ID пользователей другой системы.

Нет каких-либо жёстких правил для принятия решения о том, как лучше организовать хранение ID и что лучше использовать для идентификации пользователя при вызове API. Делайте как вам удобно, исходя из реалий конкретной ситуации.

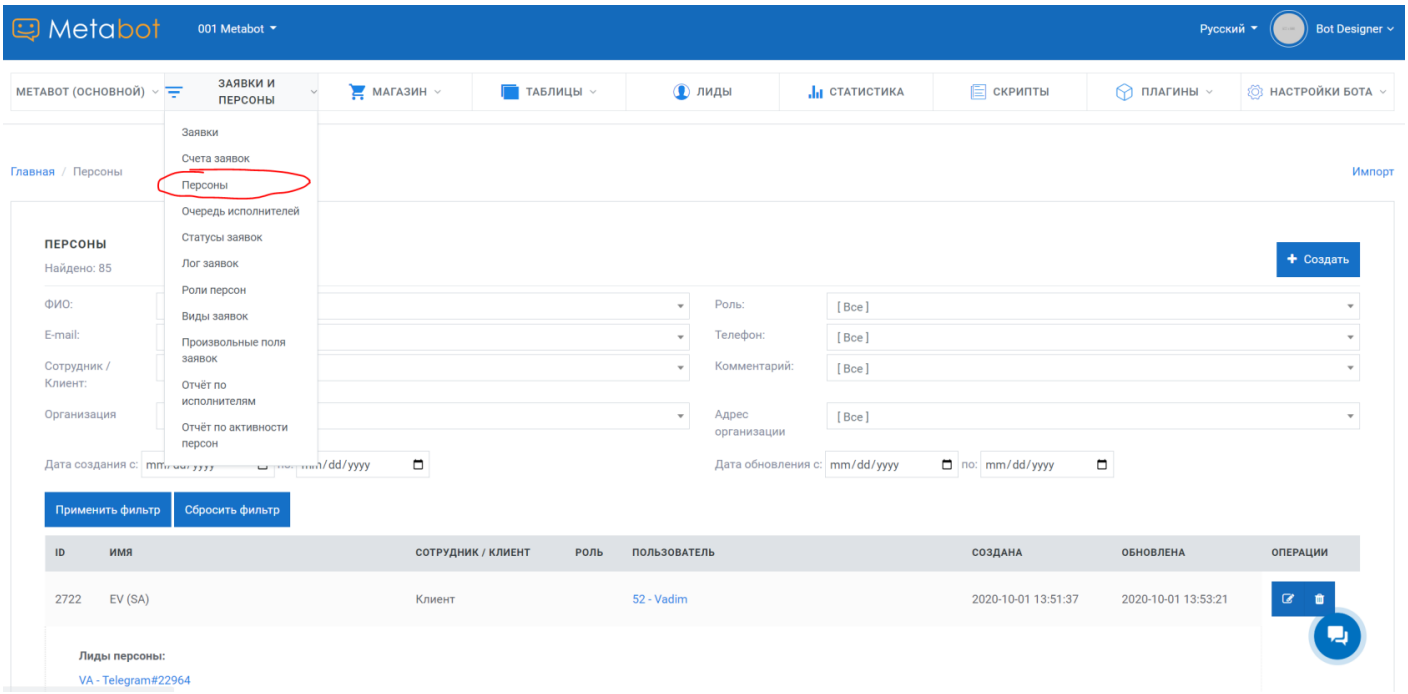
Например, если разработчикам веб-сайта затруднительно внести изменения в базу данных и код веб-сайта, вы можете принять решение хранить ID пользователя сайта у себя в боте, а разработчикам веб-сайта предложить API для отправки сообщений пользователям, используя User Id веб-сайта.

Разница между ID персоны и ID лида

В терминах Metabot **лид (lead)** — это подписчик на ваш чат-бот (можно сказать bot user), который подключен к конкретному мессенджеру. Лид создается на платформе автоматически в тот момент, когда человек поздоровался с вашим чат-ботом т.е. послал первое сообщение.



Персона (Person) — это запись о конкретном человеке, которая создается вручную разработчиком чат-бота во время регистрации или авторизации в чат-боте. У персоны теоретически может быть несколько лидов т.е. когда один и тот же человек написал в один и тот же чат-бот, подключенный к разным мессенджерам, но такие проекты не так часто встречается. Обычно у людей есть предпочтительный мессенджер, через который они общаются с компанией.

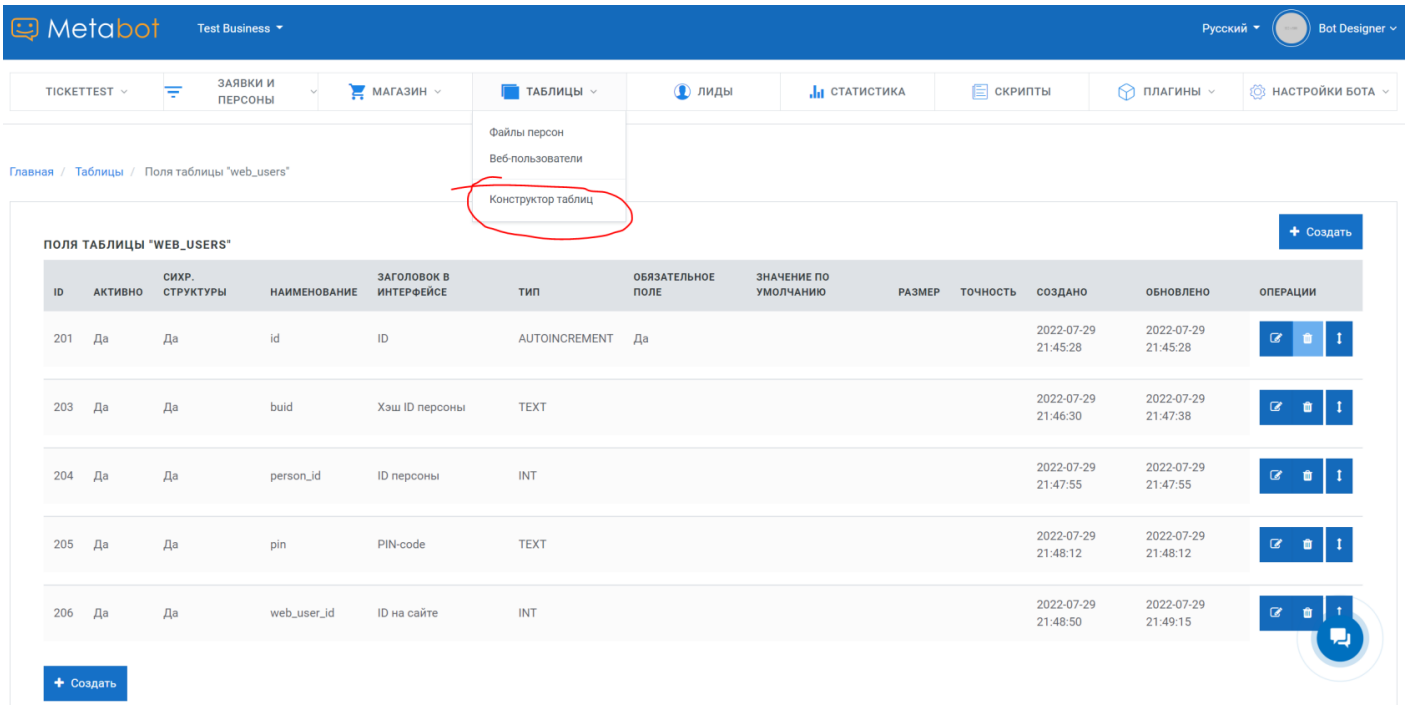


Мы обычно всегда рекомендуем во время регистрации и авторизации в чат-боте создавать в Metabot персону, привязывать к этой персоне лид и использовать ID персоны для связки с веб-сайтом, потому что это правильно. У вашего приложения должна быть реляционная структура данных, которая независима от того в каких мессенджерах кто предпочитает общается.

Но если проект простой и не надо сильно заморачиваться, то можете сделать очень просто: храните ID лида на веб-сайте и используйте ID лида для идентификации API запросов. В таком подходе есть свои плюсы. В Metabot вызов триггеров и API завязан на Lead ID, что означает код на стороне чат-бота существенно упрощается — вам не нужно будет писать прослойку, которая будет по ID персоны в боте или ID пользователя сайта будет искать ID лида, чтобы в этот лид отправить сообщение – вам достаточно вызвать API для отправки сообщения, указав конкретный лид.

Таблица соответствия

Для хранения данных соответствия ID в двух системах мы рекомендуем создать кастомную таблицу. Назовите эту таблицу например, **web_users** (веб-пользователи).



Ниже описан технический процесс организации совместной работы веб-сайта и мобильного приложения, которым является мессенджер, с помощью чат-бота:

1. В таблицу добавляем поле с хэшом ID персоны, чтобы не передавать ID персоны в явном виде в ссылках (для безопасности).
2. Также в таблицу добавляем поле PIN код, которое будем формировать случайным образом в чат-боте и затем просить ввести PIN на сайте для подтверждения идентификации.

Ссылка в боте для перехода на сайт

Для авторизации пользователя на сайте через чат-бот, формируйте временный PIN код и присылайте его в чат-бот с инструкциями для пользователя о том, что нужно перейти на сайт по ссылке и ввести PIN.

Сразу же присылайте ссылку на сайт, добавляя в нее GET параметр с хэшом персоны. Хэш персоны позволит сайту идентифицировать пришедшего пользователя из бота, без необходимости спрашивать логин и пароль, но нужно только проверить, что пользователь ввел именно тот PIN, который получил в чат-боте.

После формирования значений, сохраните данные в таблицу соответствия web_users.

Ссылка для перехода на сайт может выглядеть, например, следующим образом:

https://site.com?buid=ZRvJxYsn6Vr2sQVQ, где BUID — акроним от Bot User ID.

Подтверждение PIN-кода

В роутере разработчик сайта добавляет код, который при наличии параметра BUID загружает страницу для ввода PIN кода. После ввода, сайт на бэке по REST API проверяет валидность PIN, отправляя в бот три параметра:

- **BUID** — хэш ID персоны в чат-боте;
- **PIN** — введённый пользователем код;
- **ID** — ID пользователя сайта, если пользователь в текущий момент уже авторизован на сайте.

Чтобы обработать API запрос на валидацию PIN-кода со стороны чат-бота, в чат-боте создайте внутренний API Endpoint. Metabot поддерживает создание двух видов эндпоинтов: внутренних и внешних. Внутренние предназначены для обращения внешних систем к вашему боту, а внешние, наоборот, для обращения из вашего ботом к внешним системам.

Назовите эндпоинт, например, **authorize** (авторизация). Напишите код, который в таблице web_users будет искать запись с нужным нам BUID и проверит PIN на правильность. Если запись найдена, то сообщаем сайту об успехе, или в противном случае о провале.

Рассмотрим вероятные сценарии, представленные ранее на упрощенной схеме и дополненные новыми:

- **PIN не верный**:
  - Это означает перед нами либо пользователь, который ошибся в воде кода, либо злоумышленник, пытавшийся взломать чужой аккаунт, либо неосторожный пользователь, который по тем или иным причинам переслал страницу ввода PIN кода другому человеку;
  - В этом случае, на сайте нужно обработать ответ от бота, например, показав соответствующие сообщения об ошибке (например, что PIN истек) и предложить выслать новый PIN в мессенджер;
  - Или же можете создать новый PIN на лету и выслать его пользователю в чат-бот;
- **user_id есть в web_users, но нет в запросе**:
  - Это означает, что пользователь веб-сайта привязан к чат-боту. Возвращаем user_id в ответе, чтобы веб-сайт авторизовал пользователя;
- **user_id есть в web_users и есть в запросе**:
  - Если user_id совпадают, то это означает, пользователь уже авторизован на сайте и бот привязан к его аккаунте. Предлагаем вернуться на сайт;
  - Если user_id отличаются, то скорее всего пользователь пытается привязать свой новый аккаунт на сайте к боту, а тот уже привязан к другому аккаунте на сайте. Решите как хотите обработать эту исключительную ситуацию, например, предложите пользователю варианты на выбор: отвязать старый аккаунт и привязать новый, оставить привязку к другому аккаунту;
- **user_id нет в web_users**:
  - Если в запросе нет user_id, то это означает пользователь входит на сайте через бота и скорее всего аккаунта на сайте у него тоже нет, но это не точно;

- В идеале нужно поискать наличие аккаунта, например, запросив регистрационные данные и выполнив поиск по API;
- В самом простом сценарии, мы полагаем, что пользователь входит на сайт через чат-бот и одновременно регистрирует аккаунт на сайте.

Авторизация без ввода PIN

Для простых случаев, например, когда нам нужно показать форму ввода, принять значения, закрыть ее и вернуться в бота, мы можем не спрашивать PIN кода при переходе на сайт. Сайт может доверять хэш токену, который получили в GET, идентифицируя по нему пользователя без проверки PIN кода.

Это менее безопасный вариант, в котором есть риск, что Алиса свою ссылку со своим хэшом может переслать другому пользователю Боту, который откроет ее и будет идентифицирован на сайте под Алисой. Это допустимо в некоторых не критичных случаях, например, вы можете использовать ссылку для ввода данных, которая живет не более 1 часа.

Повышение безопасности авторизации

Опционально, вы можете улучшить безопасность, добавив дополнительные проверки, например, время жизни PIN-кода или сделав одноразовый ввод кода. Обработайте нужные вам ситуации и верните сайту ответ о результате проверки.

Что дальше?

А дальше все зависит от ваших конкретных задач и проблем. Теперь когда у вас в таблице соответствия есть взаимно-однозначная связь между ID персоны в боте и ID пользователя на сайте, вы можете как работать с сервисами веб-сайта через чат-бот от имени пользователя сайта, так и со стороны сайта инициировать в чат-боте те или иные действия для пользователя.

Также, мы рекомендуем разместить в боте функцию выхода из аккаунта, а на сайте опции для отвязки и привязки бота заново.

Версия #4
Юрий Гарашко создал 2023-06-14 08:43:18 UTC
Artem Garashko обновил 2026-03-24 07:09:51 UTC