

Telegram-бот для бізнесу: **практичний посібник із запуску сценарію**

Як перевірити придатність процесу, спроектувати маршрут,
підготувати інтеграції та прийняти результат без меню-тупиків

У ЦЬОМУ ГАЙДІ

Telegram-бот працює тоді, коли допомагає людині завершити конкретну повторювану дію в чаті: залишити заявку, записатися, передати дані, дізнатися статус або погодити внутрішнє рішення. Робота над ботом починається не з кнопок, а з процесу, ролей, даних і меж автоматизації. Нижче — послідовність, за якою команда може підготувати сценарій до реалізації та перевірити його перед запуском.

- 01 **1. Відіберіть процес, який справді варто переносити в чат**
- 02 **2. Намалюйте маршрут користувача до конкретного результату**
- 03 **3. Підготуйте дані, ролі та правила доступу**
- 04 **4. Опишіть обмін із CRM, базою та іншими системами**
- 05 **5. Протестуйте маршрут і передайте його в підтримку**
- Q **Часті питання**

Не кожна бізнес-задача стає зручнішою в Telegram. Спочатку потрібно визначити одну дію, яку користувач має завершити без складної навігації, великого обсягу інформації чи повноцінного особистого кабінету. Це захищає команду від розробки бота, який дублює сайт, але не доводить користувача до результату.

Почніть з опису не функції, а завершеної дії. Наприклад: користувач передає заявку, отримує статус за ідентифікатором, обирає параметри для запису або запускає просте внутрішнє погодження. У кожній дії має бути зрозумілий старт, послідовність кроків і фінал, який бачить користувач або відповідальний працівник.

Чат доречний для короткого керованого маршруту: повідомлення, кнопка, уточнення, передавання даних, відповідь. Якщо людині потрібно порівнювати багато товарів, читати довгі матеріали, шукати по структурі сайту, працювати з документами, історією операцій, ролями або складними налаштуваннями, основну дію краще залишити сайту чи вебкабінету.

Бот і сайт можуть працювати разом. Бот може бути точкою входу, каналом статусів, нагадувань і коротких запитів, а сайт — місцем для каталогу, пошуку, оформлення або складної роботи з профілем.

01 Сформулюйте одну цільову дію

Запишіть її у формі результату: «користувач залишив заявку», «користувач отримав статус», «відповідальний працівник погодив запит». Не починайте з формулювання на кшталт «потрібен бот для клієнтів».

02 Перевірте довжину маршруту

Розкладіть дію на послідовність повідомлень, виборів і даних. Якщо маршрут вимагає великої навігації, паралельного порівняння або багатьох винятків, винесіть складну частину у вебінтерфейс.

03 Визначте точку входу

Зафіксуйте, чому користувач починає дію саме в Telegram: він уже пише в месенджер, отримує посилання з іншого каналу або має повернутися за статусом чи нагадуванням.

04 Позначте межу автоматизації

Окремо перелічіть випадки, які бот виконує за правилами, і випадки, що мають перейти людині: нестандартні звернення, неповні дані, конфліктні ситуації або запити, де потрібне рішення відповідального працівника.

ДЕ ВТРАЧАЮТЬ

Найбільше часу втрачається, коли бот намагаються перетворити на каталог, сайт або особистий кабінет у повідомленнях. У такому сценарії команда ускладнює чат, а користувач усе одно не може зручно завершити задачу.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Сформулюйте одну завершену дію для першого сценарію.
Без конкретного результату бот перетворюється на набір загальних кнопок.
- ☐ Перевірте, чи вистачає для дії повідомлень, кнопок і коротких форм.
Якщо потрібен складний інтерфейс, чат стане перешкодою для користувача.
- ☐ Зафіксуйте, які звернення бот передає людині.
Без цієї межі нестандартні випадки зависатимуть між автоматизацією та ручною роботою.

Сценарій потрібен, щоб кожна дія в чаті мала передбачуване продовження. Він показує не лише основний шлях, а й стани діалогу, необхідні дані, повідомлення про помилки та правила передачі процесу далі. Саме карта маршруту дає команді спільне розуміння того, що потрібно реалізувати.

Опишіть сценарій як послідовність станів. Користувач може лише почати взаємодію, обрати категорію, передати контакти, очікувати відповідь від зовнішньої системи, отримати підтвердження або повідомлення про помилку. Стан пояснює, яке повідомлення бот очікує зараз і яку відповідь має сформувати.

Кожна кнопка повинна запускати дію, переводити людину в новий стан або пояснювати, чому наступний крок недоступний. Кнопка без маршруту не є функцією: вона створює тупик, після якого користувач повертається до ручного листування.

Паралельно з переходами підготуйте тексти. Користувач має розуміти, що від нього потрібно, які дані прийнято, що сталося після натискання кнопки і що робити, якщо відповідь не отримано.

01 Зафіксуйте стартові умови

Для кожного сценарію визначте, що його запускає: команда, кнопка, текстове повідомлення, дані форми або дія в групі.

02 Опишіть основний маршрут

Побудуйте ланцюжок «дія користувача — перевірка — наступне повідомлення». Додайте всі точки, де бот збирає дані, показує вибір або звертається до іншої системи.

03 Додайте стани та переходи

Назвіть кожен етап так, щоб команда розуміла контекст: початок, вибір категорії, очікування контакту, перевірка статусу, підтвердження або передача оператору.

04 Пропишіть винятки

Для кожного кроку дайте відповідь, що робити, якщо дані не знайдено, доступ заборонено, користувач надіслав не те значення або зовнішня система не відповідає.

05 Підготуйте тексти інтерфейсу

Напишіть повідомлення, назви кнопок, уточнення й помилки до початку реалізації. Текст має не просто інформувати, а підказувати наступну дію.

ДЕ ВТРАЧАЮТЬ

Усна домовленість «бот прийматиме заявки» не є сценарієм. Вона не визначає поля заявки, правила неповних даних, отримувача заявки та момент, коли користувач бачить підтвердження.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ **Намалюйте основний маршрут від старту до фінального результату.**
Без маршруту розробка почнеться з окремих екранів і не складе цілісного процесу.
- ☐ **Для кожної кнопки вкажіть дію, новий стан або зрозуміле повідомлення.**
Інакше в чаті з'являться тупики, які доведеться обробляти вручну.
- ☐ **Пропишіть реакцію на помилкові та неповні дані.**
Без цього користувач не знатиме, як повернутися до правильного кроку.
- ☐ **Погодьте тексти повідомлень із відповідальним за процес.**
Пізня зміна текстів часто змінює саму логіку сценарію.

03 · 3. ПІДГОТУЙТЕ ДАНІ, РОЛІ ТА ПРАВИЛА ДОСТУПУ

Бот не працює з абстрактними зверненнями: він отримує конкретні значення, перевіряє умови та передає результат далі. До початку реалізації потрібно визначити, які дані потрібні на кожному кроці, звідки вони беруться, хто має право їх бачити та хто відповідає за актуальність правил.

Для кожного сценарію складіть перелік вхідних і вихідних даних. Вхідними можуть бути контакт, ідентифікатор, дата або інший параметр, який надсилає користувач. Вихідними — підтвердження, статус, повідомлення для групи або запис, переданий у CRM чи іншу систему.

Права доступу є частиною сценарію, а не загальним налаштуванням «для всіх». Обмеження можуть залежати від ролі користувача, конкретної команди або даних, які він намагається отримати. Правило доступу потрібно описати поруч із відповідним переходом, а не залишати його на розсуд реалізації.

Визначте власника процесу після запуску. Це людина або роль, яка оновлює тексти, правила, сценарії та зв'язки з іншими системами. Без відповідального бот швидко починає повертати застарілі повідомлення або вести користувача до неактуальної дії.

01 Створіть таблицю даних сценарію

Для кожного поля зазначте: назву, хто його надає, на якому кроці бот його запитує, куди передає та що робить, якщо значення відсутнє або не підходить.

02 Розподіліть ролі користувачів

Опишіть, які маршрути доступні клієнту, працівнику, відповідальному за погодження або іншій ролі. Не показуйте однакові дії всім, якщо результат залежить від доступу.

03 Встановіть правила перевірки

Для кожної дії визначте умови: які дані приймаються, коли доступ заборонено, що потрібно уточнити та коли сценарій можна продовжити.

04 Призначте відповідальних у команді

Власник процесу відповідає за правила й тексти; технічна сторона реалізує сценарій та інтеграції; відповідальний працівник приймає звернення, які бот передає людині.

05 Погодьте критерії приймання

Для кожного маршруту опишіть перевірний результат: користувач може пройти шлях, дані передаються в потрібну систему, обмеження доступу спрацьовують, а помилка має зрозумілий текст.

ДЕ ВТРАЧАЮТЬ

Якщо права доступу та власника даних не визначити до запуску, бот може або показувати інформацію не тій ролі, або зупиняти корисні сценарії без зрозумілої причини.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Зберіть усі дані, які бот читає, запитує та передає.
Без переліку полів команда не зможе перевірити повноту сценарію.
- ☐ Опишіть доступи для кожної ролі та дії.
Без цього правила доступу з'являтимуться випадково під час розробки.
- ☐ Призначте відповідального за актуальність текстів і сценаріїв.
Без власника бот залишиться без підтримки після запуску.
- ☐ Зафіксуйте критерії, за якими команда приймає кожен маршрут.
Без критеріїв перевірка зведеться до суб'єктивного «ніби працює».

Telegram показує повідомлення, кнопки й форми, але відповідь часто залежить від зовнішніх даних. Щоб бот передав заявку, отримав статус або записав інформацію, команда має заздалегідь описати, яка система є джерелом даних, що саме бот читає або записує та як поводить ся у разі збою.

У робочому маршруті подія з Telegram надходить до серверної логіки. Логіка визначає тип події та стан діалогу, перевіряє права, за потреби виконує запит до інтеграції, обирає наступний крок і повертає повідомлення або кнопки в чат.

Інтеграція може читати або записувати дані. Наприклад, бот передає заявку в CRM, отримує статус із системи обліку або звертається до бази даних. Для кожного такого обміну важливо знати, яке саме значення передається, що система має повернути та що побачить користувач, якщо відповідь не надійшла.

Не залишайте ручні дії між кроками неописаними. Якщо після натискання кнопки працівник має сам здогадатися, що робити з інформацією, процес не автоматизовано: плутанина просто переміщується з листування в чат.

01 Визначте джерело істини для кожного значення

Вкажіть, де бот бере статус, куди передає заявку, у якій системі зберігаються дані та яка сторона відповідає за їхню актуальність.

02 Опишіть запит і відповідь інтеграції

Для кожної взаємодії зафіксуйте, що запускає запит, які дані передаються, яку відповідь очікує бот і як ця відповідь впливає на наступний стан діалогу.

03 Передбачте відсутність відповіді

Сформулюйте окремий маршрут для ситуації, коли зовнішня система не відповідає або не повертає даних. Користувач має отримати зрозуміле повідомлення, а не залишитися без реакції.

04 Відокремте читання від запису

Позначте, де бот лише отримує дані для відповіді, а де створює або змінює запис в іншій системі. Це допомагає перевірити наслідки кожної дії.

05 Перевірте маршрут на реальних сценаріях

Пройдіть успішний варіант, помилкове значення, відмову в доступі та недоступність інтеграції. Результат кожної перевірки внесіть у журнал.

ДЕ ВТРАЧАЮТЬ

Найчастіший провал інтеграції — описати лише успішний запит. Коли CRM, база або інший сервіс не повертає дані, бот без окремого маршруту не пояснює користувачу, що сталося і що робити далі.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Для кожної інтеграції вкажіть, що бот читає та що записує.
Без цього неможливо перевірити межі обміну даними.
- ☐ Опишіть окрему відповідь для недоступної зовнішньої системи.
Без неї користувач отримає мовчання замість зрозумілого наступного кроку.
- ☐ Перевірте, чи не залишилися ручні дії між кнопкою та результатом.
Неописана ручна робота руйнує передбачуваність автоматизованого маршруту.

Запуск не завершується перевіркою команди /start або кількох кнопок. Потрібно прийняти кожен сценарій як завершений процес: від стартової події до результату, включно з помилками, доступами, інтеграціями та передаванням звернення людині. Після цього команда має мати журнал перевірки й зрозумілий порядок оновлень.

Тестуйте не елементи інтерфейсу окремо, а маршрути. Кнопка може відправляти значення коректно, але сценарій буде неповним, якщо після неї не створюється заявка, не повертається статус або користувач не розуміє наступного кроку.

Для приймання зручно використовувати журнал перевірки. У ньому фіксують назву сценарію, роль користувача, стартову дію, очікуваний результат, фактичний результат, стан інтеграції, помилку та відповідального за виправлення. Такий журнал перетворює тестування на керований процес, а не на список повідомлень у чаті.

Після запуску будь-яка зміна тексту, правила або інтеграції може вплинути на маршрут. Тому оновлення має проходити через власника процесу: він перевіряє, які стани, дані та повідомлення змінюються, і передає завдання технічній стороні.

01 Складіть перелік сценаріїв для приймання

Внесіть основні маршрути, усі ролі, помилкові дані, заборонений доступ, відсутні дані та недоступність зовнішньої системи.

02 Заповніть журнал перевірки

Використайте шаблон: «сценарій — роль — стартова дія — очікуваний результат — фактичний результат — статус — відповідальний». Фіксуйте навіть дрібні неясності у текстах чи переходах.

03 Перевірте критерії приймання

Підтвердіть, що кожна кнопка веде до дії, нового стану або зрозумілого повідомлення; дані передаються за призначенням; доступи працюють; винятки не залишають користувача без відповіді.

04 Передайте правила підтримки

Зафіксуйте, хто оновлює тексти, хто погоджує зміни сценарію, хто відповідає за інтеграції та хто приймає звернення, що бот передає людині.

05 Визначте порядок змін

Будь-яку нову кнопку, поле або інтеграцію спочатку додайте до карти сценарію, даних і винятків. Лише після цього передавайте її в реалізацію та повторне тестування.

ДЕ ВТРАЧАЮТЬ

Запуск без журналу перевірки створює приховані дефекти: команда пам'ятає, що перевіряла бот, але не може підтвердити, який саме маршрут, за якою роллю та з яким результатом був прийнятий.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Пройдіть кожен сценарій від старту до фінального результату.
Перевірка окремих кнопок не гарантує, що процес завершується.
- ☐ Протестуйте помилки, відмову в доступі та недоступність інтеграції.
Саме ці ситуації найчастіше залишають користувача без маршруту.
- ☐ Заповніть журнал перевірки та призначте відповідальних за виправлення.
Без фіксації проблеми повторюються або губляться між учасниками команди.
- ☐ Погодьте порядок оновлення сценаріїв після запуску.
Без процесу змін бот поступово розходиться з реальним бізнес-процесом.

Чи потрібен Telegram-боту сайт?

Ні, якщо вся потрібна дія вміщується у короткий чат-сценарій. Сайт або вебкабінет потрібен, коли користувачеві необхідні пошук, каталог, довге читання, складне порівняння чи робота з великим обсягом даних.

Чим бот відрізняється від Telegram-каналу?

Канал призначений для односторонніх дописів підписникам. Бот реагує на події користувача: повідомлення, команди, кнопки або дані форми, і веде його через визначений сценарій.

Що має бути готово до початку розробки?

Потрібно підготувати цільову дію, карту сценарію, ролі, дані, тексти повідомлень, правила доступу, опис інтеграцій і винятків. Окремо варто призначити відповідального за підтримку логіки після запуску.

Чи може бот передавати заявки в CRM?

Так, серверна логіка може передавати дані до CRM через інтеграцію, якщо система має придатний спосіб обміну. У сценарії потрібно визначити, які поля передаються, хто бачить результат і що відбувається, якщо CRM не відповідає.

Коли звернення потрібно передавати людині?

Людині варто передавати нестандартні випадки, запити без потрібних даних, конфліктні ситуації та дії, де потрібне рішення відповідального працівника. Цей перехід має бути окремим кроком сценарію, а не неформальним припущенням.

ЩО ДАЛІ

Обговорити з Apricode



apri-code.com/kontakty/