

Інтеграція Telegram бота з Google Sheets через Azure Logic Apps

Мета: Налаштувати автоматизований процес (Flow), де Azure Logic App приймає повідомлення з Telegram, логує їх у Google Таблицю, а при отриманні спеціальної команди `/db` — зчитує останній запис з таблиці та відправляє його назад користувачу.

Необхідні ресурси:

1. Створений бот у **Telegram** (через @BotFather, потрібен API Token).
 2. Аккаунт **Google** (Google Drive/Sheets).
 3. Аккаунт **Microsoft Azure** (Logic App типу Consumption).
-

Крок 1: Підготовка бази даних (Google Sheets)

1. Зайдіть на [Google Sheets](#).
2. Створіть нову таблицю з назвою, наприклад, `TelegramLog`.
3. Перейменуйте аркуш (знизу) на `Log`.
4. **Обов'язково** створіть заголовки у першому рядку (A1, B1, C1):
 - **Date**
 - **User**
 - **Message**

The screenshot shows a Google Sheet interface. At the top, the title bar displays 'TelegramLog' in a red box, along with icons for file operations (Файл, Змінити, Вигляд, Вставити ...). Below the title bar is a toolbar with search, undo, redo, print, and other icons, followed by a zoom level of 100% and a currency symbol (грн. % .0). The main area of the sheet shows a table with columns labeled A, B, and C. The first row contains the headers 'Date', 'User', and 'Message', which are enclosed in a red box. The second row is highlighted in blue. The bottom right corner of the sheet features a 'Log' button, also enclosed in a red box. Below the sheet, a caption reads: 'Місце для скріншота: Готова Google Таблиця із заголовками'.

| | A | B | C |
|---|------|------|---------|
| 1 | Date | User | Message |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |

Крок 2: Створення Logic App та налаштування Тригера

Детальна інструкція: Створення Logic App (Consumption)

Цей етап критично важливий для успішного виконання роботи без зайвих витрат.

1. Пошук сервісу:

- Увійдіть на portal.azure.com.
- У верхньому рядку пошуку введіть **Logic Apps**.


- Оберіть сервіс **Logic Apps** (іконка з ракетою/схемою).

2. Початок створення:

- На сторінці сервісу натисніть кнопку **+ Create** (або "Створити") у верхньому лівому куті.

3. Налаштування вкладки "Basics" (Основні):

Заповніть поля наступним чином:

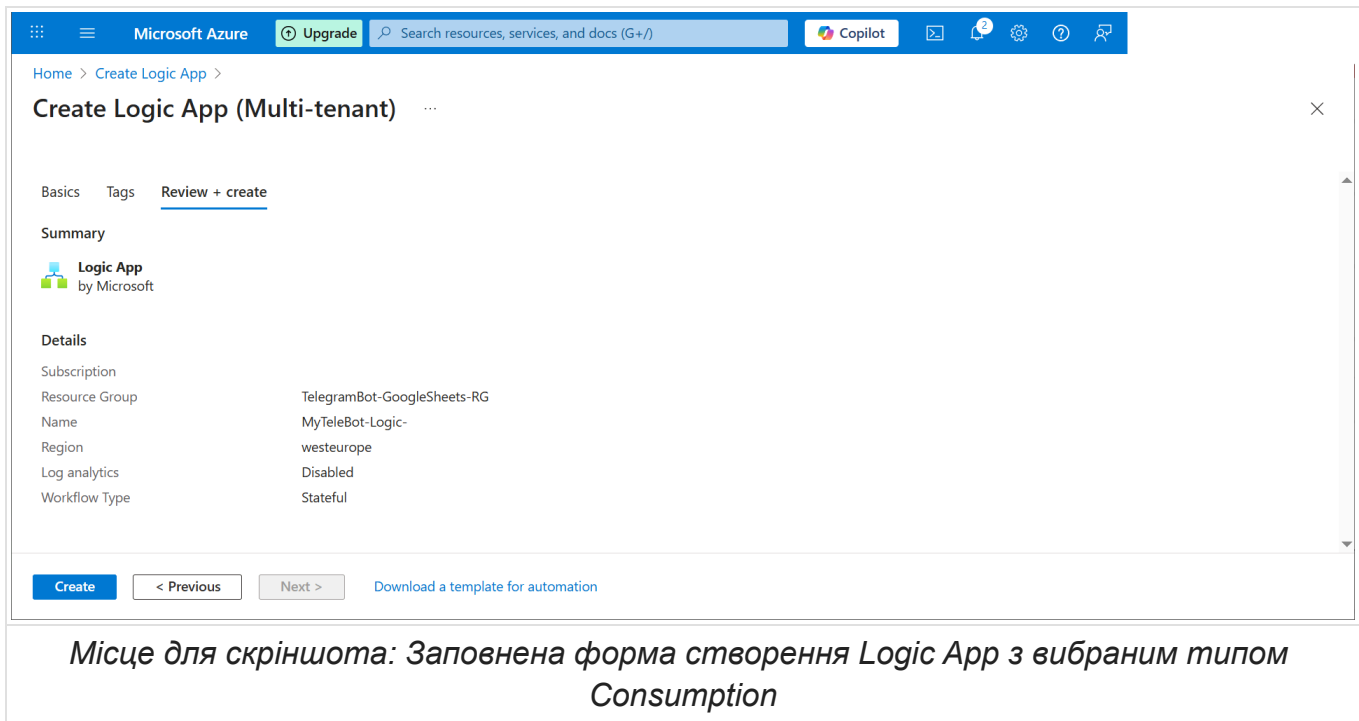
- **Plan type (Тип плану):**  **УВАГА! Це найважливіше.**
 - Оберіть **Consumption** (Споживання).
 - *Чому:* Цей план працює за моделлю "serverless" (платите лише за виконання). Якщо виберете *Standard*, вам доведеться платити за оренду сервера щомісяця.*
- **Subscription (Підписка):** Оберіть вашу активну підписку (наприклад, "Azure for Students").
- **Resource Group (Група ресурсів):**
 - Натисніть "Create new" і назвіть її, наприклад, TelegramBot-GoogleSheets-RG .
- **Logic App name:** Придумайте унікальне ім'я (наприклад, MyTeleBot-Logic-[ВашеПрізвище]).
- **Region (Періон):** Оберіть (Europe) West Europe або Germany West Central .

4. Завершення:

- Натисніть кнопку **Review + create** (Перегляд і створення) внизу.
- Дочекайтеся перевірки (має з'явитися напис **Validation Passed**).
- Натисніть кнопку **Create** (Створити).

5. Перехід до ресурсу:

- Зачекайте розгортання (30-60 секунд).
- Коли з'явиться напис "Your deployment is complete", натисніть **Go to resource**.
- У меню зліва знайдіть розділ **"Development Tools"** і натисніть **Logic App Designer**, щоб почати роботу.



1. Перейдіть у **Logic App Designer**.
2. Оберіть шаблон **"Blank Logic App"**.
3. Натисніть на об'єкт та в пошуку знайдіть і додайте тригер: **Request -> When a HTTP request is received**.
4. У полі **Request Body JSON Schema** вставте наступну схему (вона описує структуру повідомлення від Telegram (при розширенні функціонала бота, цю схему можна буде розширити для отримання більшої кількості даних)):

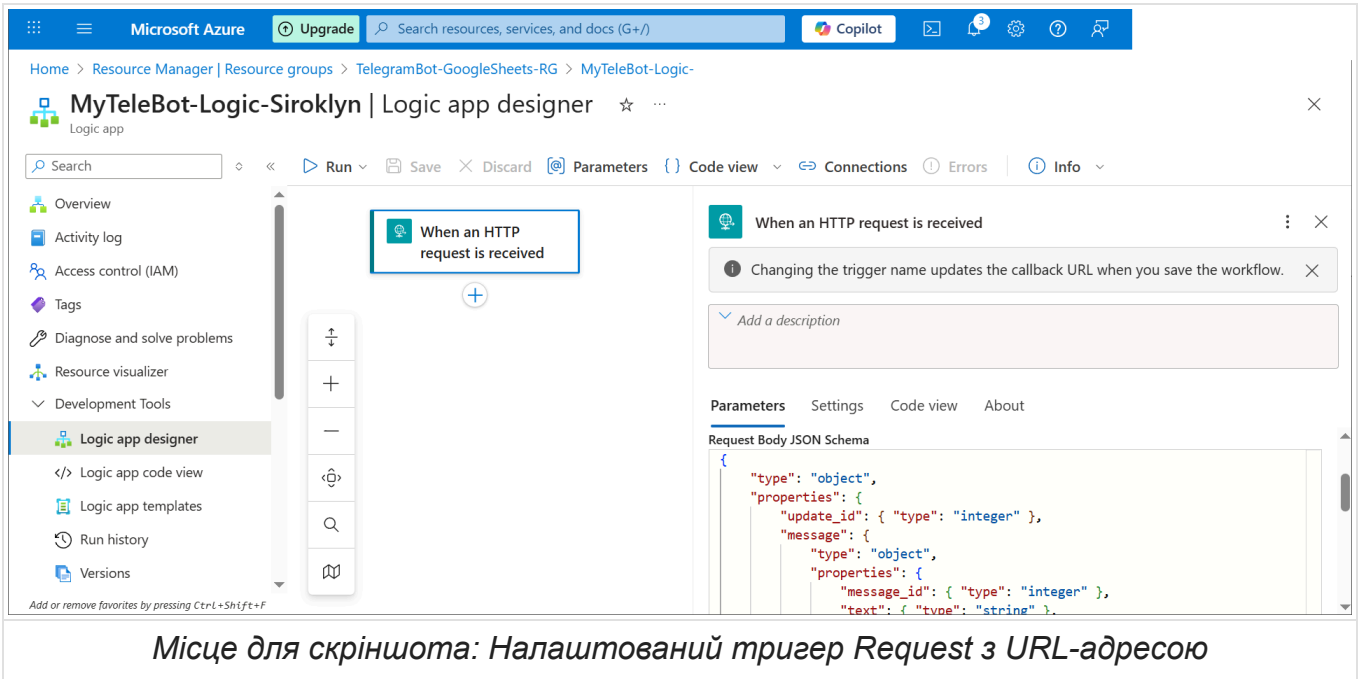
```
{
  "type": "object",
  "properties": {
    "update_id": { "type": "integer" },
    "message": {
      "type": "object",
      "properties": {
        "message_id": { "type": "integer" },
        "text": { "type": "string" },
        "date": { "type": "integer" },
        "from": {
          "type": "object",
          "properties": {
            "id": { "type": "integer" },
            "first_name": { "type": "string" },
            "username": { "type": "string" }
          }
        }
      }
    }
  }
}
```

```

    },
    "chat": {
      "type": "object",
      "properties": {
        "id": { "type": "integer" },
        "type": { "type": "string" }
      }
    }
  }
}

```

6. Натисніть **Save**. Після збереження у полі **HTTP POST URL** (всередині блока **When an HTTP request is received**) з'явиться посилання. Скопіюйте його.



Крок 3: Створення бота в Telegram та отримання токена

Для виконання лабораторної роботи вам знадобиться власний Telegram-бот. Усіма ботами керує головний бот — **BotFather**.

1. Пошук BotFather:

- Відкрийте Telegram.
- У полі пошуку введіть: @BotFather.

- Оберіть бота з синьою галочкою верифікації.
- Натисніть кнопку **Start** (або введіть `/start`).

2. Створення нового бота:

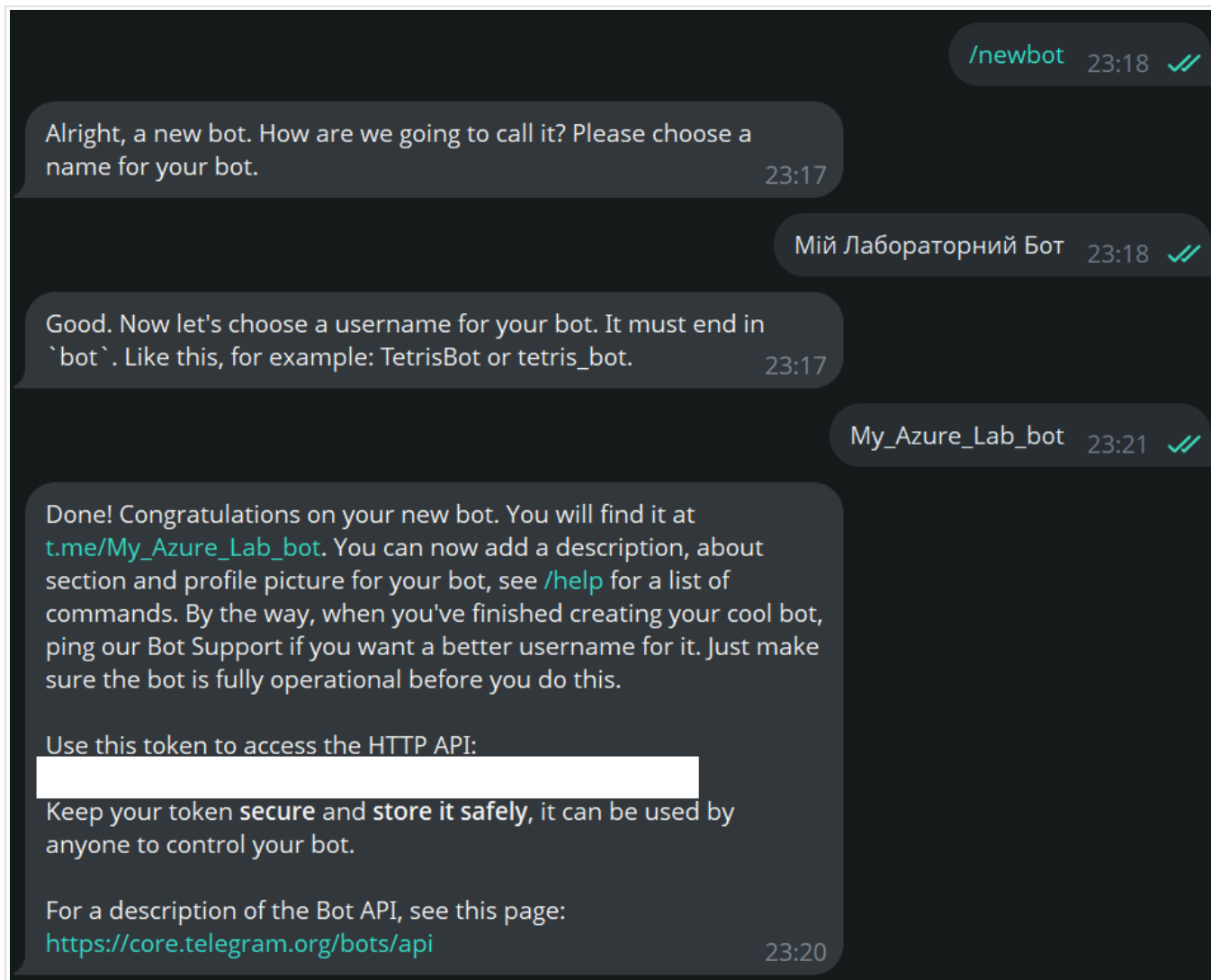
- Надішліть команду:

```
/newbot
```

- **Крок 1 (Ім'я):** BotFather попросить ввести ім'я для вашого бота. Це те, що користувачі бачитимуть у списку контактів.
 - *Приклад:* Мій Лабораторний Бот
- **Крок 2 (Юзернейм):** Тепер потрібно придумати унікальний юзернейм. Він має писатися латиницею і обов'язково закінчуватися на `bot` .
 - *Приклад:* `My_Azure_Lab_[ВашеПрізвище]_bot` або `[ВашеПрізвище]_Lab_bot` (обов'язкові унікальна назва та закінчення `_bot`).
 - *Примітка:* Якщо юзернейм зайнятий, BotFather попросить ввести інший.

3. Отримання токена:

- Після успішного створення BotFather надішле вам повідомлення з текстом: *"Done! Congratulations on your new bot..."*.
- У цьому повідомленні знайдіть рядок, що починається з `Use this token to access the HTTP API:` .
- Нижче буде довгий набір символів (наприклад, `123456789:ABCdefGHIjklMNOpqrs...`).
- **Це і є ваш API Token.** Скопіюйте його та збережіть. Він знадобиться для налаштування Logic App.



Місце для скріншота: Діалог з BotFather та повідомлення з успішно створеним ботом і токеном

Додавання команди /db у меню бота (опціонально)

Щоб користувачам не доводилося писати команду вручну, додамо її у меню швидкого доступу (те, що з'являється при введенні / або натисканні кнопки Menu).

1. Відкрийте BotFather:

- Поверніться до чату з **@BotFather**.

2. Розпочніть налаштування:

- Надішліть команду:

```
/setcommands
```

- Якщо у вас декілька ботів, оберіть потрібного зі списку (того, якого ви створили для цієї лабораторної).

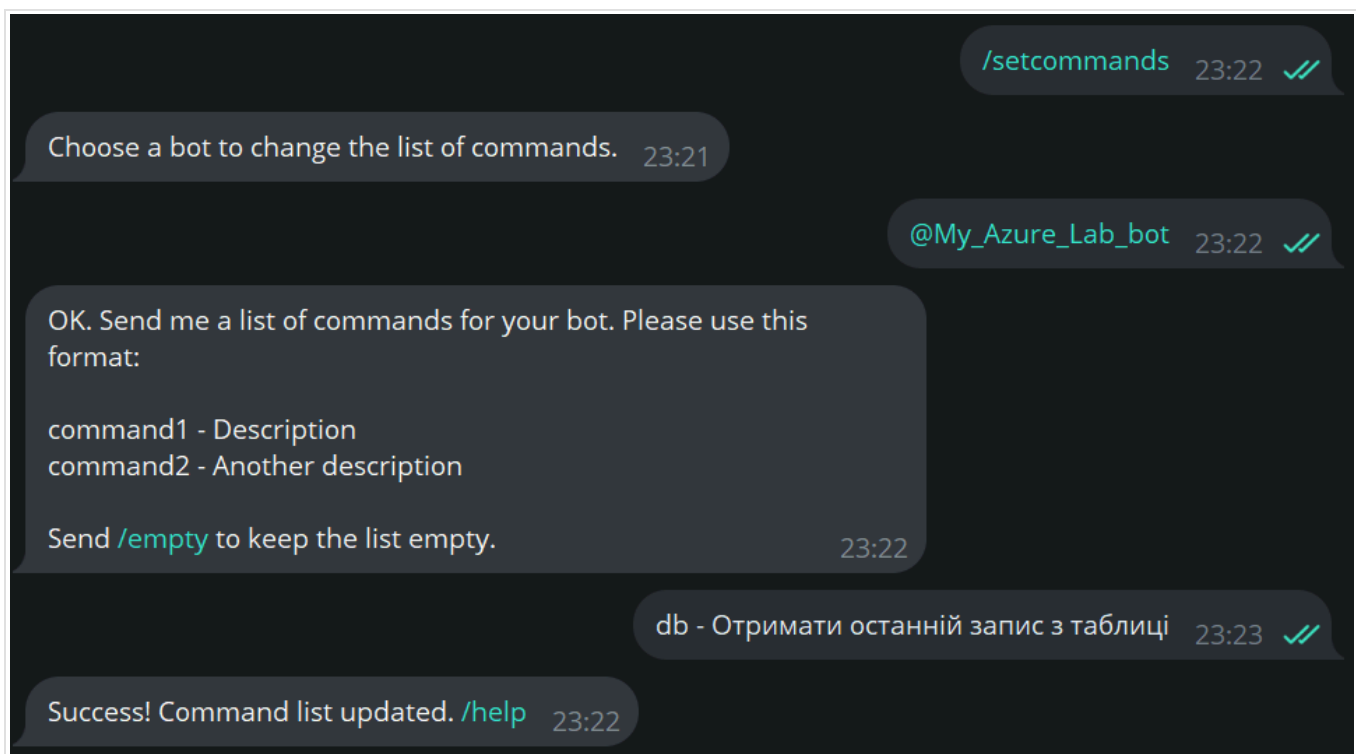
3. Введіть опис команди:

- BotFather попросить надіслати список команд у форматі команда – опис .
- Скопіюйте та надішліть наступний рядок:

```
db – Отримати останній запис з таблиці
```

4. Завершення:

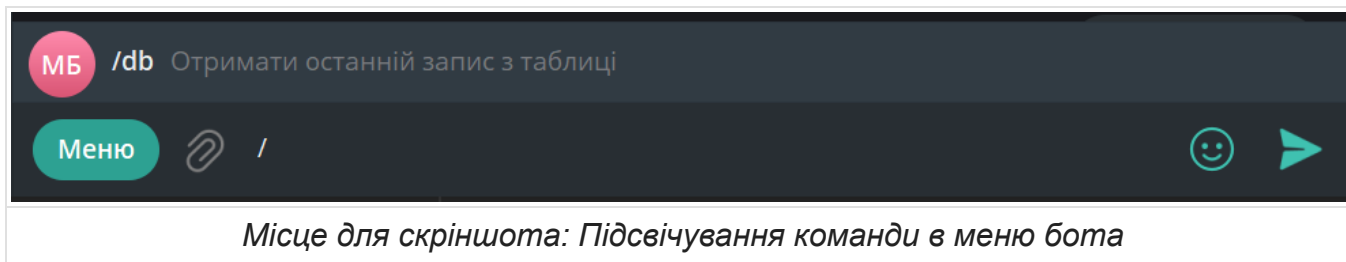
- BotFather має відповісти: *Success! Command list updated.*



Місце для скріншота: Процес додавання команди через BotFather

5. Перевірка:

- Перейдіть у чат зі своїм ботом.
- Натисніть кнопку **Menu** (зліва від поля вводу) або просто введіть / . Ви маєте побачити підказку з вашою новою командою.



Крок 4: Реєстрація Webhook (Зв'язок Telegram та Azure)

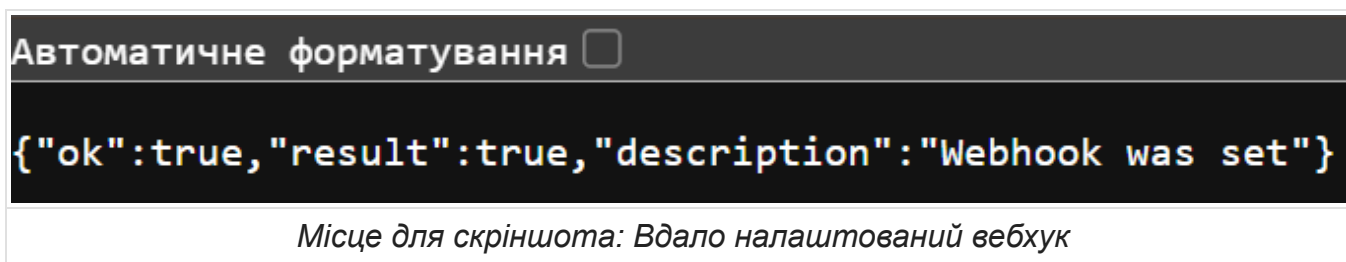
Щоб Telegram почав надсилати повідомлення у ваш Logic App, потрібно зареєструвати URL, отриманий на попередньому кроці.

1. Відкрийте нову вкладку в браузері.
2. Сформууйте та перейдіть за наступним посиланням (підставте свої дані):

```
https://api.telegram.org/bot<ВАШ_БОТ_ТОКЕН>/setWebhook?url=<URL_3_ЛОДЖИК_АПП>
```

- <ВАШ_БОТ_ТОКЕН> — токен, який дав BotFather (напр. 123456:ABC-DEF...).
- <URL_3_ЛОДЖИК_АПП> — довге посилання з кроку 2 (воно вже закодоване, просто вставте його).

3. Ви повинні побачити відповідь: `{"ok":true, "result":true, "description":"Webhook was set"}`.



Крок 5: Розгалуження логіки (Condition)

Ми маємо розділити потік: якщо це команда, робимо одне; якщо текст — інше.

1. У Logic App Designer натисніть **+ Add an action**.
2. Оберіть **Control -> Condition**.

3. Налаштуйте умову перевірки тексту повідомлення:

- Зліва оберіть динамічний контент: `text` (з розділу "When a HTTP request is received").
- Оператор: **starts with** (починається з) (можна використати "=", але в групах це може викликати проблеми).
- Значення: `/db`.

Альтернативно, у режимі виразу (Expression):

`startsWith(triggerBody()?['message'])?['text'], '/db')` is equal to `true`.

Місце для скріншота: Блок Condition з налаштованою умовою

Крок 6: Гілка "True" (Читання з БД)

Ця гілка виконується, якщо користувач написав `/db`.

1. У блоці **True** натисніть **Add an action**.

2. Знайдіть **Google Sheets -> Get rows**.

- Увійдіть у свій Google акаунт (ВАЖЛИВО надати доступ до дій в акаунті, галочки на яких, за замовчуванням не виставлені).
- File: Оберіть файл `TelegramLog`.
- Worksheet: Оберіть аркуш `Log`.

3. Додайте наступну дію: **HTTP**.

- **URI:** `https://api.telegram.org/bot<BAШ_БОТ_ТОКЕН>/sendMessage`
- **Method:** `POST`

- **Headers:**
 - Key: Content-Type
 - Value: application/json
- **Body:** (Ми використовуємо функцію `last()`, щоб взяти останній елемент з масиву отриманих рядків):

```
{
  "chat_id": "@{triggerBody()?['message']?['chat']?['id']}",
  "text": "📄 Останній запис у базі:\nКористувач: @{last(body('Get_rows')?['value'])?['User']}\nПовідомлення: @{last(body('Get_rows')?['value'])?['Message']}",
  "parse_mode": "HTML"
}
```

Місце для скріншота: Налаштовані дії у гілці True (Get Rows + HTTP)

Крок 7: Гілка "False" (Запис у БД)

Ця гілка виконується для будь-яких інших повідомлень.

1. У блоці **False** натисніть **Add an action**.

2. Знайдіть **Google Sheets** -> **Insert row**.

- **File:** TelegramLog
- **Worksheet:** Log
- У **Advanced parameters** виберіть усі три доступні стовпця (вони з'являються з затримкою, тож якщо ви щось міняли по ходу роботи, може бути необхідним запустити бота). Це ті самі назви стовпців, що ви давали в Кроці 1.
 - **Date:** У **Insert expression** (нижня кнопка **fx**, що виникає зліва, коли ви натискаєте на текстове поле) оберіть **Expression** і напишіть **utcNow()**.
 - **User:** Оберіть **first_name** з тригера (верхня кнопка "блискавка", що виникає зліва, коли ви натискаєте на текстове поле).
 - **Message:** Оберіть **text** з тригера.

3. Додайте дію **HTTP** (щоб сповістити користувача про успіх). Можна скопіювати через натискання **ПКМ** по блоку з гілки **True** з подальшою заміною **Body** (ПКМ (блок) -> **Copy action** -> ПКМ (+) - **Paste action**).

- **URI:** `https://api.telegram.org/bot<ВАШ_БОТ_ТОКЕН>/sendMessage`
- **Method:** **POST**
- **Headers:** `Content-Type: application/json`
- **Body:**

```
{
  "chat_id": "@{{triggerBody()?['message']?['chat']?['id']}}",
  "text": "✅ Повідомлення збережено в базу!",
  "reply_to_message_id": "@{{triggerBody()?['message']?['message_id']}}"
}
```

Місце для скріншота: Налаштовані дії у гілці False (Insert Row + HTTP)

Крок 8: Тестування та демонстрація

1. Збережіть Logic App (**Save**).
2. Відкрийте свій бот у Telegram.
3. **Тест 1 (Запис):** Напишіть Привіт, лабораторна! .
 - *Очікуваний результат:* Бот відповідає реплаєм "✅ Повідомлення збережено в базу!". У Google Таблиці з'являється новий рядок.
4. **Тест 2 (Читання):** Напишіть /db .
 - *Очікуваний результат:* Бот відповідає повідомленням, де вказано ім'я та текст "Привіт, лабораторна!" (або інший останній запис).

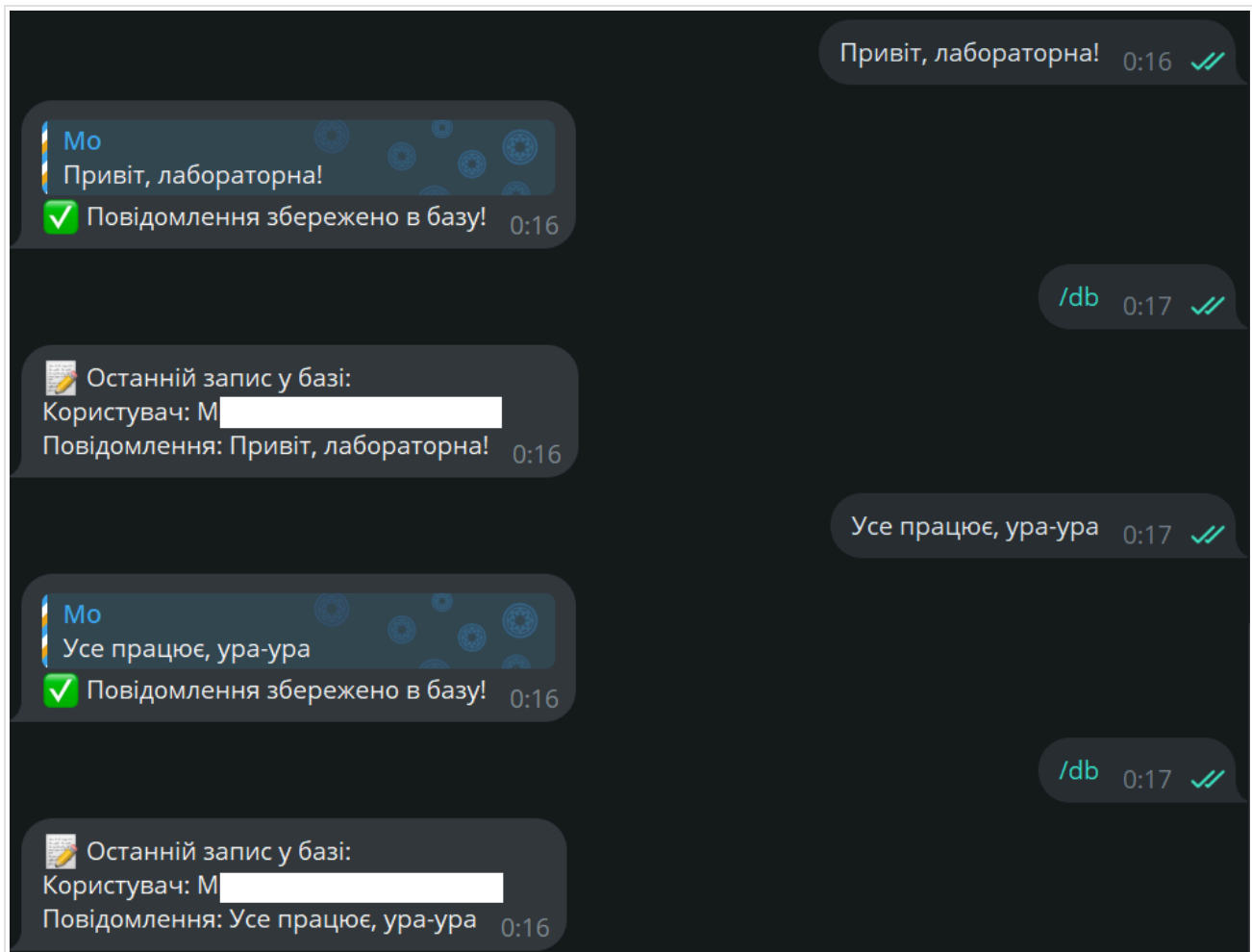
Важливо: Кодування URL-адреси (URL Encoding)

Посилання на Azure Logic App містить спеціальні символи (& , : , /), які можуть "ламати" запит до Telegram API, якщо їх не закодувати. Якщо вебхук не працює (у Development Tools -> Run history немає жодного запису), виконайте наступне:

1. Скопіюйте **HTTP POST URL** з тригера Logic App.

2. Відкрийте сайт urldecoder.org.
3. Натисніть кнопку **ENCODE** (Закодувати), згори.
4. Вставте посилання у поле.
5. Натисніть кнопку **ENCODE**, після текстового поля.
 - Результат має містити символи на зразок `%3A`, `%2F`, `%26` замість звичайних знаків пунктуації.
6. Скопіюйте цей **закодований** текст.
7. Використайте його для реєстрації вебхука:

```
https://api.telegram.org/bot<ВАШ_TOKEN>/setWebhook?url=
<ВСТАВТЕ_ЗАКОДОВАНИЙ_URL_ТУТ>
```



Місце для скріншота: Результат переписки в Telegram

| | A | B | C | D |
|---|-----------------|------|-----------------|-----------------|
| 1 | Date | User | Message | __PowerAppsid__ |
| 2 | 2025-11-27T22:1 | M | єдлждлж | 1bea9fbb22b34ac |
| 3 | 2025-11-27T22:1 | M | Привіт, лаборат | d81eb6a221794e |
| 4 | 2025-11-27T22:1 | M | фвіфв | e9c8c5d8b7bd4e |
| 5 | 2025-11-27T22:1 | M | віфвфів | 81a37bf6a3ce4dc |
| 6 | 2025-11-27T22:1 | M | Привіт, лаборат | b84a8196508749 |
| 7 | 2025-11-27T22:1 | M | Усе працює, ур | 6a55e2383b4749 |
| Місце для скріншота: Заповнена Google Таблиця | | | | |

Зверніть увагу, що з'явився стовбець **PowerAppsid**. Він позначає унікальні ідентифікатори, які позначають Row id та використовуються всередині блоків **Google Sheets** для посилання на відповідні рядки. Цікавий факт: ви можете замінити їх значення на свої, які є більш "корисними". Наприклад, помістити в них ID користувача.

Крок 9: Опціонально

Даний крок не є обов'язковим до виконання та тут показані просто цікаві речі, які можна використати при створенні бота.

Connector: Telegram Bot (Independent Publisher)

В **Logic app** існує спеціальний тип блоків Telegram Bot (Independent Publisher), але їх функціонал досить обмежений, тож при побудові власних сценаріїв, доцільно використовувати описаний метод запитів до **Telegram**, або створювати Custom Connector.

Logic app code view

Дана вкладка (у меню зліва Development Tools -> Logic app code view) є ширшою для модифікації, на відміну від Logic app designer, проте якщо ви туди напишете щось зайве, це поле зітре, після збереження в Logic app designer.

- Наприклад, якщо в блоці Telegram Bot (Independent Publisher) -> Send Message спробувати додати рядок "reply_to_message_id": "@{triggerBody()?['message']}?['message_id']}", , його зітре і він не буде працювати.

Parameters

Для створення більш універсальних сценаріїв, доцільно використовувати параметри. Доступ до них, можна отримати на вкладці `Logic app designer`, у рядку нагорі, неподалік від поля `Save`.

- Серед них існують `Secure` типи... Але вони зберігаються відкрито та навіть сам **Azure** каже нам `It is not recommended to set a default value for type 'SecureString' because it will be stored as plain text.` Для паролів та ключів API рекомендується використання **Key Vault**.
- У них можна зберегти текстові поля, які у вас часто використовуються та які ви плануєте змінювати іноді (наприклад, ім'я листа в таблиці).

Whitelist / Blacklist

Якщо у вас є бажання обмежити доступ до бота, то можна є кілька схожих варіантів:

- Створити параметр типу `String` з ID користувачів, записаними через певний роздільник (`-`, `&`, `\n` і т.д.). Наприклад, `(123456789-234567891-345678912)`. А після блоку `When an HTTP request is received`, поставити `Condition` типу `contains`, що перевіряє параметр `@{parameters('MembersIDs')}` на наявність у собі `string(triggerBody()?['message']?['from']?['id'])`.
- Створити параметр типу `Array`. Наприклад, `[123456789, 234567891, 345678912]`. Та обробляти його, по аналогії, циклом у блоці `For each`, який всередині мав би `Condition` типу `=`.

А для оповіщення адміністратора, про несанкціонований доступ, поставити блок з вмістом:

```
{
  "chat_id": "@{items('For_each')}",
  "text": "👤 Користувач \"<a href='\"tg://user?id=@{triggerBody()?['message']?['from']?['id']}\">@{triggerBody()?['message']?['from']?['first_name']}</a>\" - @concat('@', triggerBody()?['message']?['from']?['username']), з ID: <code>@{triggerBody()?['message']?['from']?['id']}</code>, написав мені: \n<blockquote>@{coalesce(triggerBody()?['message']?['text'], triggerBody()?['message']?['caption'], '')}</blockquote>",
  "parse_mode": "HTML"
}
```

Google Sheets: формули

Якщо спробувати вивести ячейку з формулою в таблицях, ви отримаєте результат її обчислення. За допомогою цього, в нашому варіанті, можна зробити підрахунок кількості повідомлень (`=COUNT(A2:A)`) та їх подальший вивід.

Google Sheets: вивід чисел та формул

При вставленні чисел та формул через деякі методи, перед ними, може додаватися апостроф ' , після чого, всередині вони починають сприйматися, як об'єкти типу `string` . Виходом з цього, може послугувати сусідня ячейка з формулою `=VALUE(A1)` , або подібні підходи.

Або ж, можна створити наступний скрипт: у Розширення -> App Script :

```
function formatRaidsColumn() {
    var sheetName = "Лист 1"; // Назва вашого листа
    var colIndex = 11; // Номер колонки K (A=1, B=2... K=11). Змініть, на той,
    що треба.
    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var sheet = ss.getSheetByName(sheetName);
    if (sheet) {
        // Беремо діапазон від K2 (бо в K1 знаходиться наша назва колонки) до
        кінця
        var range = sheet.getRange(2, colIndex, sheet.getLastRow() - 1, 1);

        // 1. Встановлюємо числовий формат (або будь-який, що вам треба)
        range.setNumberFormat("#,###;-#,###;0");

        // 2. (Опціонально) Якщо там є апострофи, цей хак змушує Google
        перерахувати значення:
        // Ми просто беремо значення і перезаписуємо їх назад.
        // Це "зриває" текстові мітки.
        /*
        var values = range.getValues();
        range.setValues(values);
        */
    }
}
```

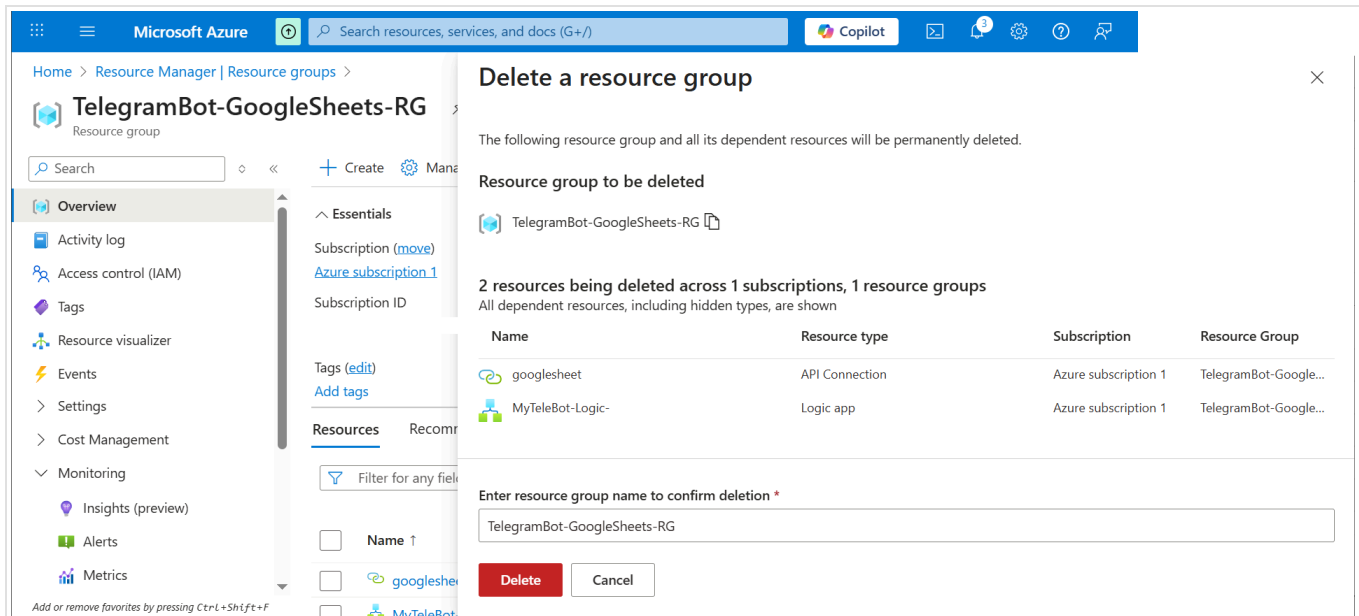
Крок 10: Видалення ресурсів

ВАЖЛИВО: Після завершення роботи та демонстрації результатів, обов'язково видаліть створені ресурси. Це не тільки хороша практика (очищення робочого простору), але й гарантія того, що з вашого рахунку не будуть списуватися кошти за активні з'єднання або збережені дані, а ваш бот не буде використаний зловмисниками.

10.1. Видалення ресурсів в Azure

Найпростіший спосіб очистити все в Azure — видалити групу ресурсів. Це автоматично видалить Logic App, історію запусків та API-з'єднання з Google Sheets.

1. Зайдіть на portal.azure.com.
2. Перейдіть у розділ **"Resource groups"** (Групи ресурсів).
3. Знайдіть групу, яку ви створили на початку (наприклад, TelegramBot-RG).
4. Натисніть кнопку **"Delete resource group"** (Видалити групу ресурсів) на верхній панелі.
5. Введіть назву групи для підтвердження та натисніть **"Delete"**.



Місце для скріншота: Підтвердження видалення групи ресурсів в Azure

10.2. Видалення Telegram-бота

Щоб "вимкнути" бота та видалити його зі списку ваших ботів:

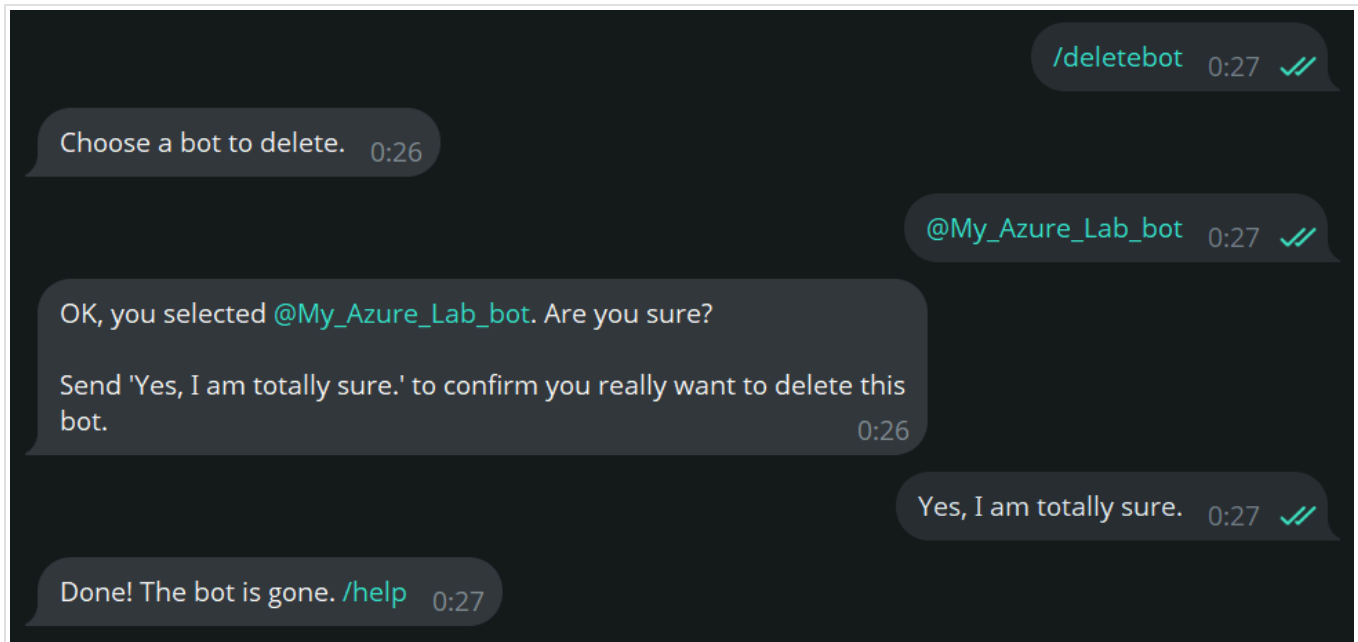
1. Відкрийте чат з **@BotFather** у Telegram.
2. Надішліть команду:

```
/deletebot
```

3. Оберіть вашого бота зі списку (того, якого ви створили для цієї лаби).
4. BotFather попросить підтвердження. Напишіть у відповідь:

```
Yes, I am totally sure.
```

5. Після цього бот буде безповоротно видалений, а його токен перестане працювати.



Місце для скріншота: Процес видалення бота в чаті з BotFather