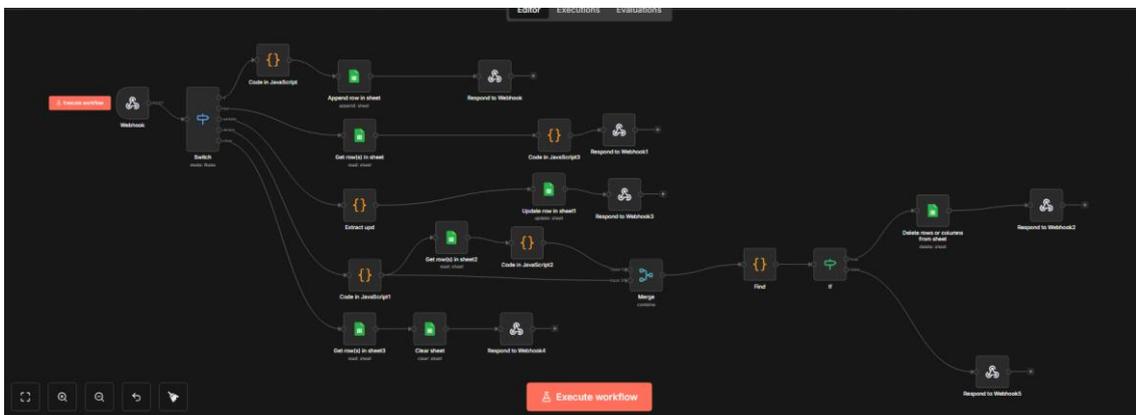


Практическая работа 10

Тема: Интеграция платформы с технологиями машинного обучения Подключение модели к существующей системе

Что в итоге должно получиться

- 1 Рабочий сайт с списком товаров и с возможностью изменения данных
- 2 Онлайн таблица с данными
- 3 Рабочий алгоритм для обработки данных и передачи между приложением и таблицей
- 4 Мини магазин



1 Создай workflow

1 Открой n8n

- Запусти CMD и пропиши команду: “n8n start”
- Дождись загрузки приложения, после чего нажми Английскую букву O

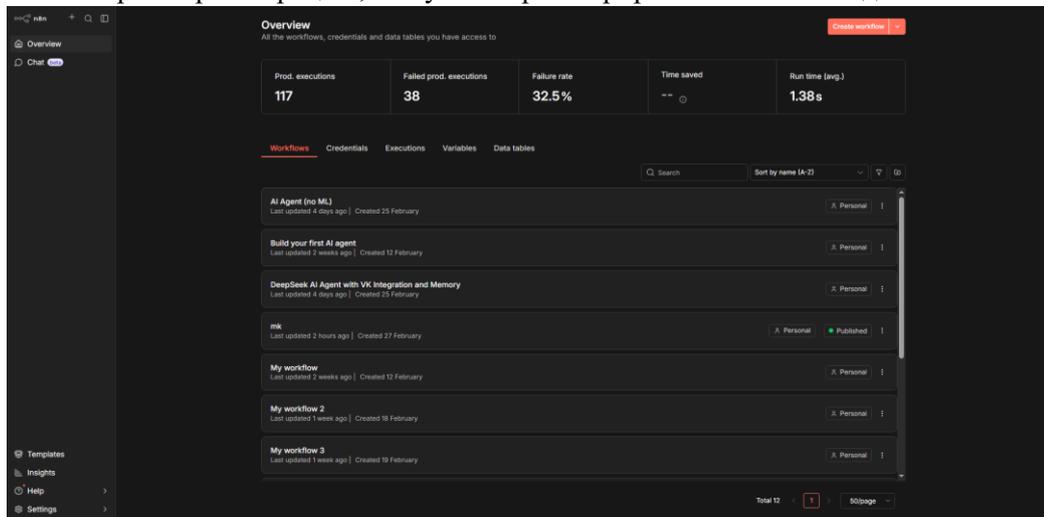
```
Windows PowerShell
Microsoft Windows [Version 10.0.17763.316]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\aaabelichenko>n8n
Initializing n8n process
ENOENT: no such file or directory, open 'C:\Users\aaabelichenko\.n8n\nodes\node_modules\n8n-nodes-llmlayer\package.json'
ENOENT: no such file or directory, open 'C:\Users\aaabelichenko\.n8n\nodes\node_modules\n8n-nodes-llmlayer\package.json'
n8n ready on ::, port 5678
n8n Task Broker ready on 127.0.0.1, port 5679
Failed to start Python task runner in internal mode, because Python 3 is missing from this system. Launching a Python ru
ner in internal mode is intended only for debugging and is not recommended for production. Users are encouraged to depl
y in external mode. See: https://docs.n8n.io/hosting/configuration/task-runners/#setting-up-external-mode
[license SDK] Skipping renewal on init: license cert is not initialized
Registered runner "JS Task Runner" (MLQdbI2phZAkPWk14qbR0)
Version: 2.8.3
Building workflow dependency index...
Start Active Workflows:
Finished building workflow dependency index. Processed 1 draft workflows, 0 published workflows.
Activated workflow "Vk" (ID: t2U5A0GAd4uUc7zz)

Editor is now accessible via:
http://localhost:5678

Press "o" to open in Browser.
(node:5676) [DEP0060] DeprecationWarning: The `util._extend` API is deprecated. Please use Object.assign() instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
```

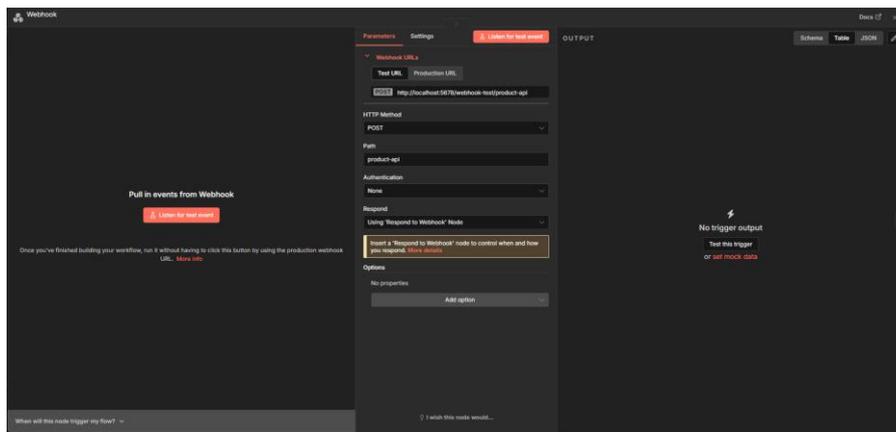
- Если запросит регистрацию, то нужно зарегистрироваться/войти под своей почтой



- На главной странице нажать на кнопку Create workflow/start from scratch
- Название: MiniMark_ФИО

2 Добавь узел webhook

- 1 Нажми на + по центру экрана
- 2 В правой части экрана появится список узлов, тебе необходимо написать в поиске слово «Webhook»
- 3 Нажми на него и у тебя откроется меню
- 4 Настрой меню как показано на картинке



6 После того как все заполнишь нажми на крестик в правом верхнем углу

3 Switch

1 Нажми на плюсики чуть правее webhook и Добавь узел switch

2 Настрой как показано на картинке

Parameters

Settings

Execute step

Mode

Rules

Routing Rules

`fx` `{{ $json.body.action }}`

T is equal to

add

Rename Output



`fx` `{{ $json.body.action }}`

T is equal to

list

Rename Output



Output Name

list

`fx` `{{ $json.body.action }}`

T is equal to

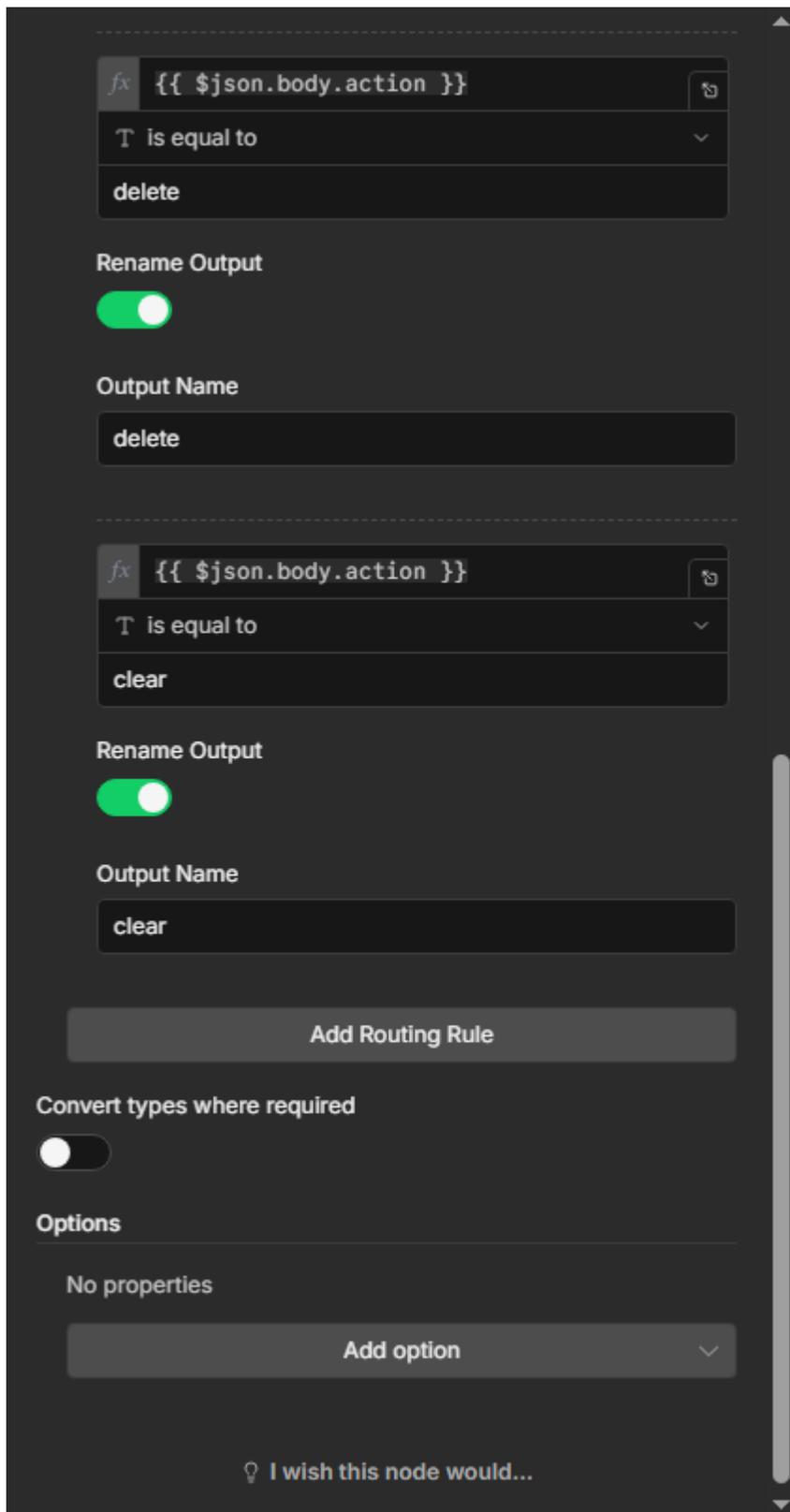
update

Rename Output



Output Name

update



3 После заполнения данных закрой это окно

4 Создание таблицы для нашего магазина(если ее не выдали)

4.1

1. Переходим на сайт <https://docs.google.com/spreadsheets/u/0/>, авторизуемся под своим акаунтом и создаем новую таблицу, называем ее «Product»
2. В таблице добавляем столбцы

id	name	category	price	qty	UpdateAt
----	------	----------	-------	-----	----------

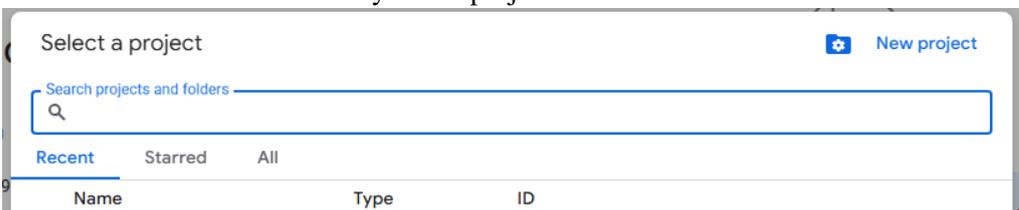
3. Сохраняем

4.2 Получение Api для работы с таблицами

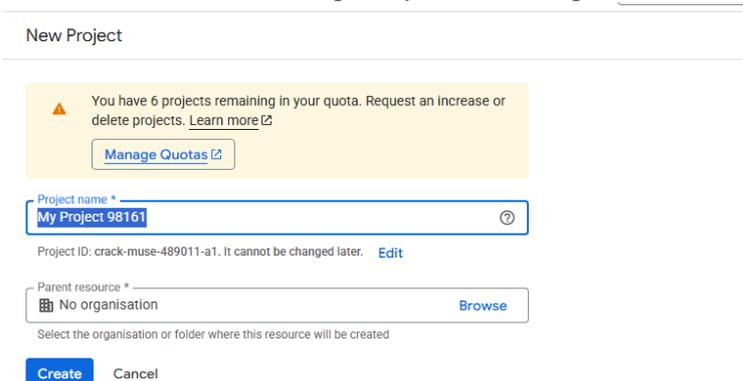
1. Заходим на сайт console.cloud.google.com/welcome?project=gen-lang-client-0927978487
2. Входим под своими акаунтами
3. нажимаем на кнопку справа от надписи Google Cloud



4. Нажимаем на кнопку “New project”



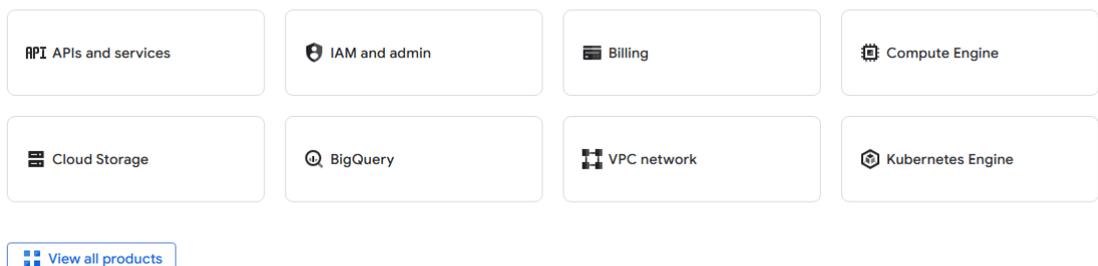
5. Даем название проекту и создаем проект

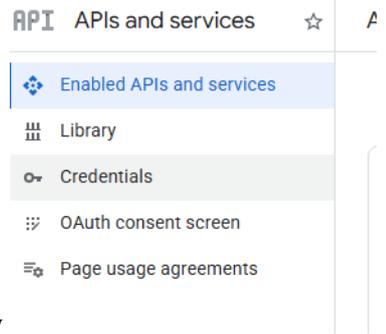


6. в меню из 3 пункта выбираем наш проект

7. В главном меню выбираем вкладку Api

Quick access





8. В открывшемся окне выбираем вкладку Library

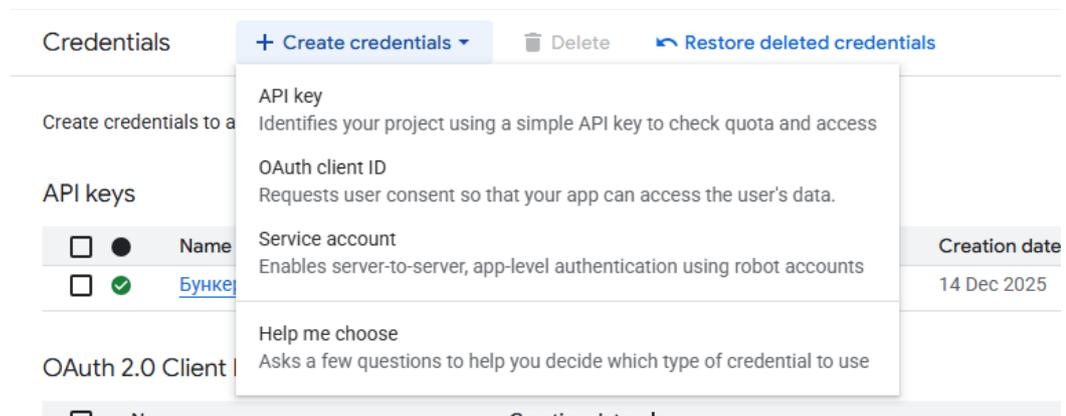
9. Ищем в поиске

«Google Sheets API» и «Google Drive API» и Подключаем их



10. Возвращаемся в окно из 7 пункта и выбираем Credentials

11. Нажимаем на Create credentials и выбираем OAuth client ID



12. Выбираем Web

← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *

Web application

Android

Chrome extension

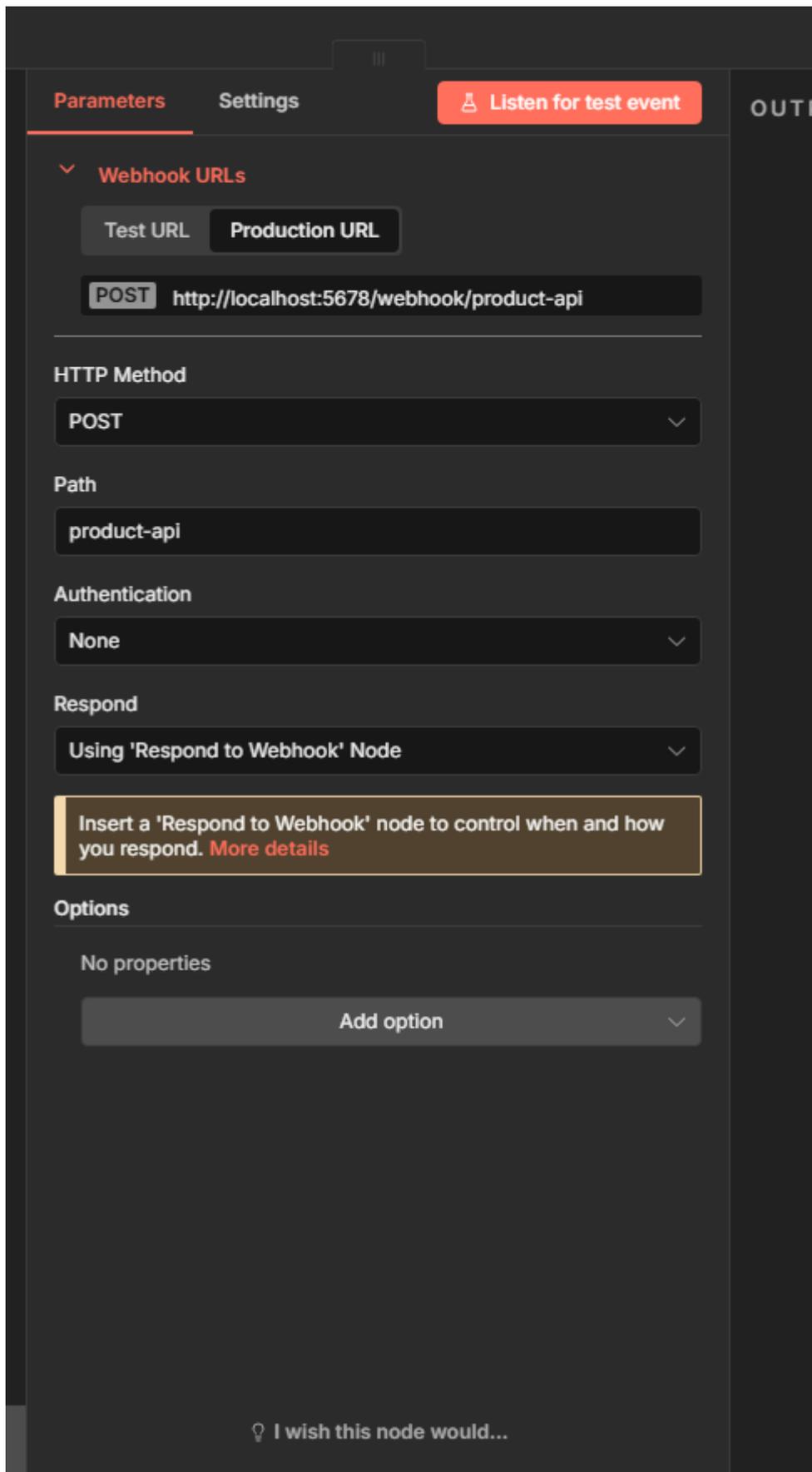
iOS

TVs and Limited Input devices

Desktop app

Universal Windows Platform (UWP)

13 Заходим в свой проект n8n и нажимаем 2 левой кнопкой мыши на webhook, выбираем Production URL и копируем ссылку



14 Пишем название

← Create OAuth client ID

[OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *
Web application

Name *
MiniMarket

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.



The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorised domains](#).

Authorised JavaScript origins ?

For use with requests from a browser

+ Add URI

Authorised redirect URIs ?

For use with requests from a web server

URIs 1 *
`http://localhost:5678/webhook/product-api`

+ Add URI

Note: It may take five minutes to a few hours for settings to take effect

Create

Cancel

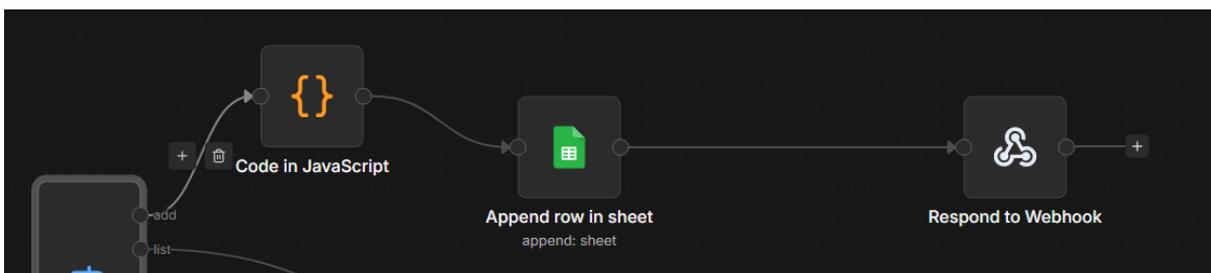
Now viewing project 'k

15 Нажимаем кнопку Create

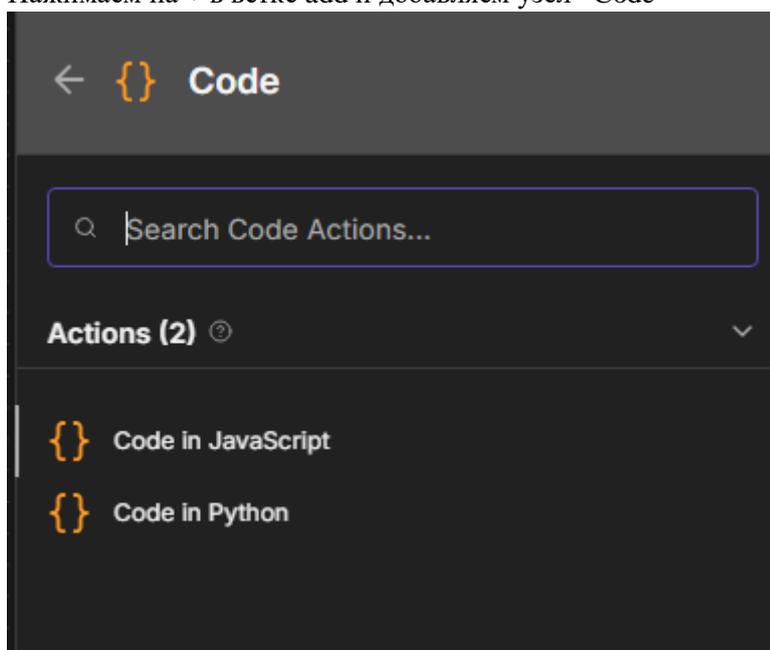
16 Копируем и сохраняем себе Client ID и Client secret, Они нам пригодятся для дальнейшей работы (можно снизу нажать Download JSON и скачать сразу файл с ними)

5 Case add – Добавление товара

На этой ветке мы будем реализовывать функцию добавления товара в таблицу.



1. Нажимаем на + в ветке add и добавляем узел “Code”



- 2.
3. Выбираем javascript
4. В Открывшемся окне вписываем код в javascript

```
const data = $input.first().json;
```

```
const b = data.body || {};
```

```
const name = String(b.name || "").trim();
```

```
const category = String(b.category || 'Без категории').trim();
```

```
const price = Number(b.price);
```

```
const qty = Number(b.qty);
```

```
if (name.length < 2) return [{ json: { ok: false, error: 'name too short' } }];
```

```
if (!Number.isFinite(price) || price <= 0) return [{ json: { ok: false, error: 'bad price' } }];
```

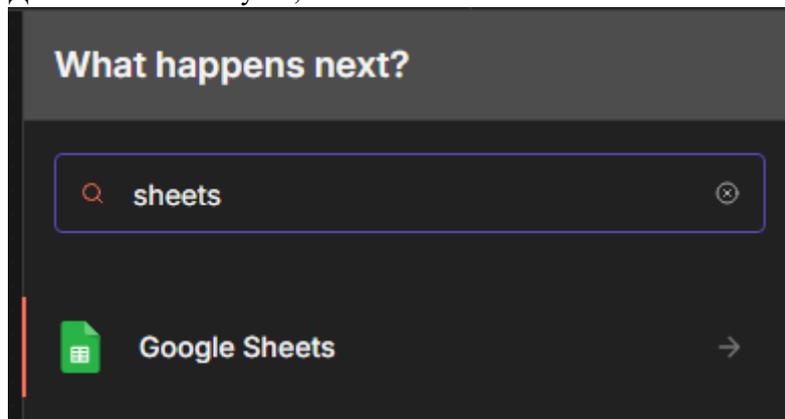
```
if (!Number.isFinite(qty) || qty < 0) return [{ json: { ok: false, error: 'bad qty' } }];
```

```
const id = Date.now().toString(36) + Math.random().toString(36).slice(2, 6);
```

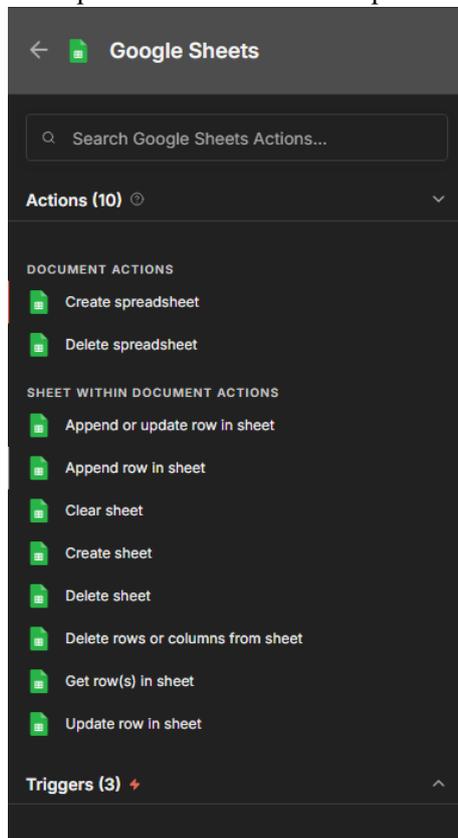
```
const UpdateAt = new Date().toISOString();
```

```
return [{ json: { id, name, category, price, qty, UpdateAt } }];
```

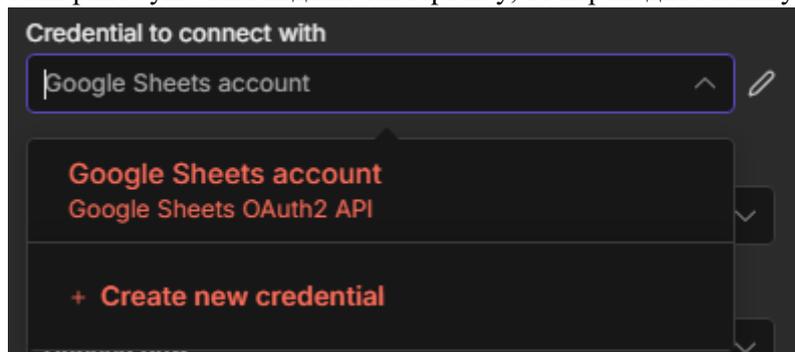
5. Закрываем окно
6. Добавляем новый узел, в поиск пишем



7. В открывшемся списке выбираем Append row in sheet

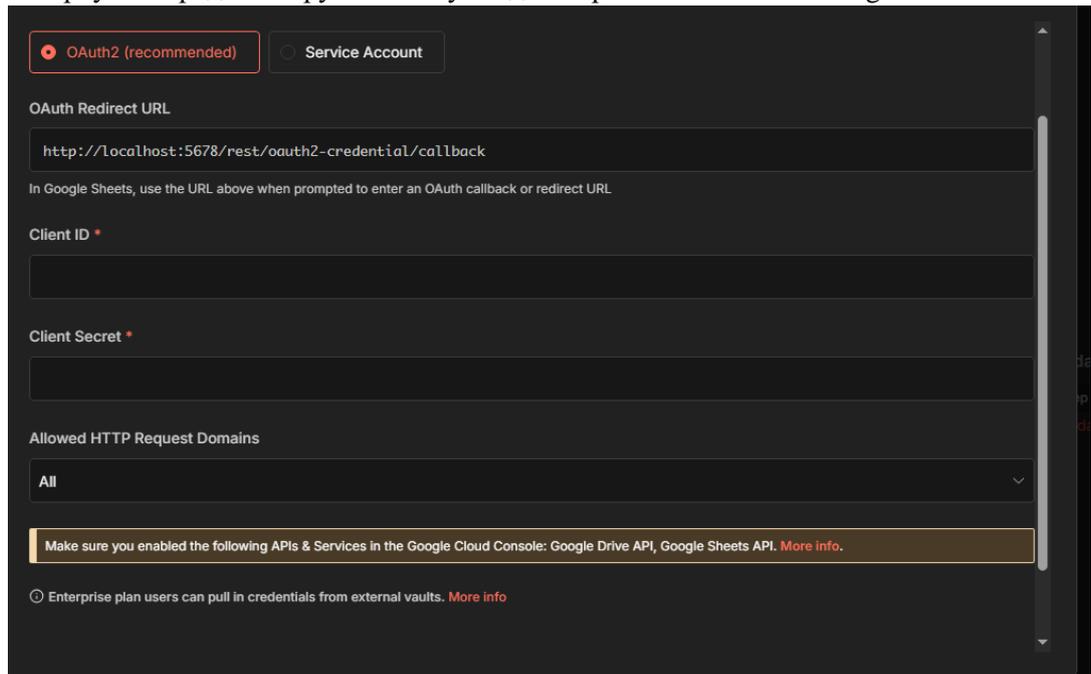


- 8.
9. В первой строке «Credential to connect with» нажимаем на «create new credential». Но если ранее уже был подключен Api Key, то переходите к шагу 24



- 10.

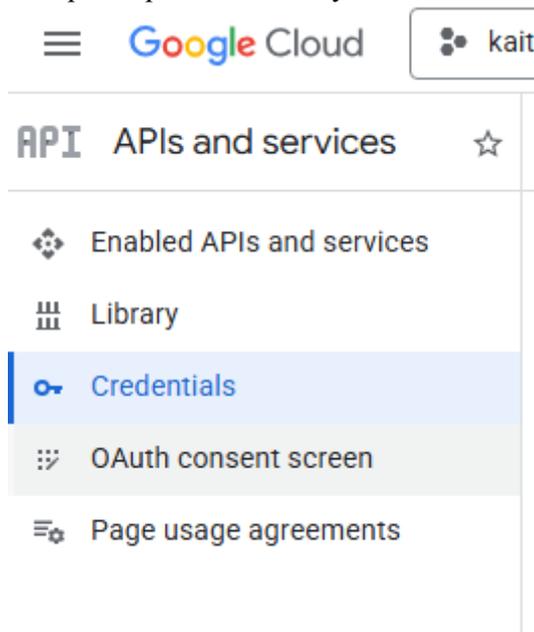
11. В первую очередь копируем ссылку и идем обратно на сайт Google cloud



The screenshot shows the Google Cloud OAuth2 configuration interface. At the top, there are two radio buttons: "OAuth2 (recommended)" which is selected, and "Service Account". Below this, the "OAuth Redirect URL" is set to "http://localhost:5678/rest/oauth2-credential/callback". A note below states: "In Google Sheets, use the URL above when prompted to enter an OAuth callback or redirect URL". There are empty input fields for "Client ID" and "Client Secret". The "Allowed HTTP Request Domains" dropdown is set to "All". A yellow warning box at the bottom says: "Make sure you enabled the following APIs & Services in the Google Cloud Console: Google Drive API, Google Sheets API. [More info.](#)". A small note at the very bottom indicates: "Enterprise plan users can pull in credentials from external vaults. [More info](#)".

12.

13. Теперь открываем вкладку "OAuth consent screen"

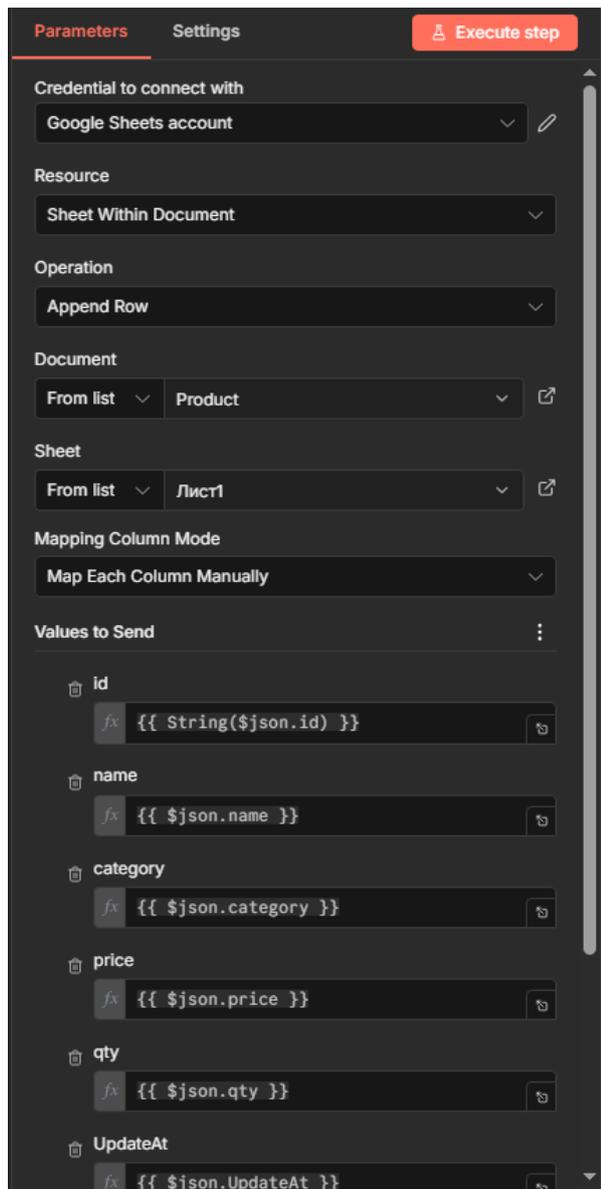


14.

15. Переходим во вкладку Clients

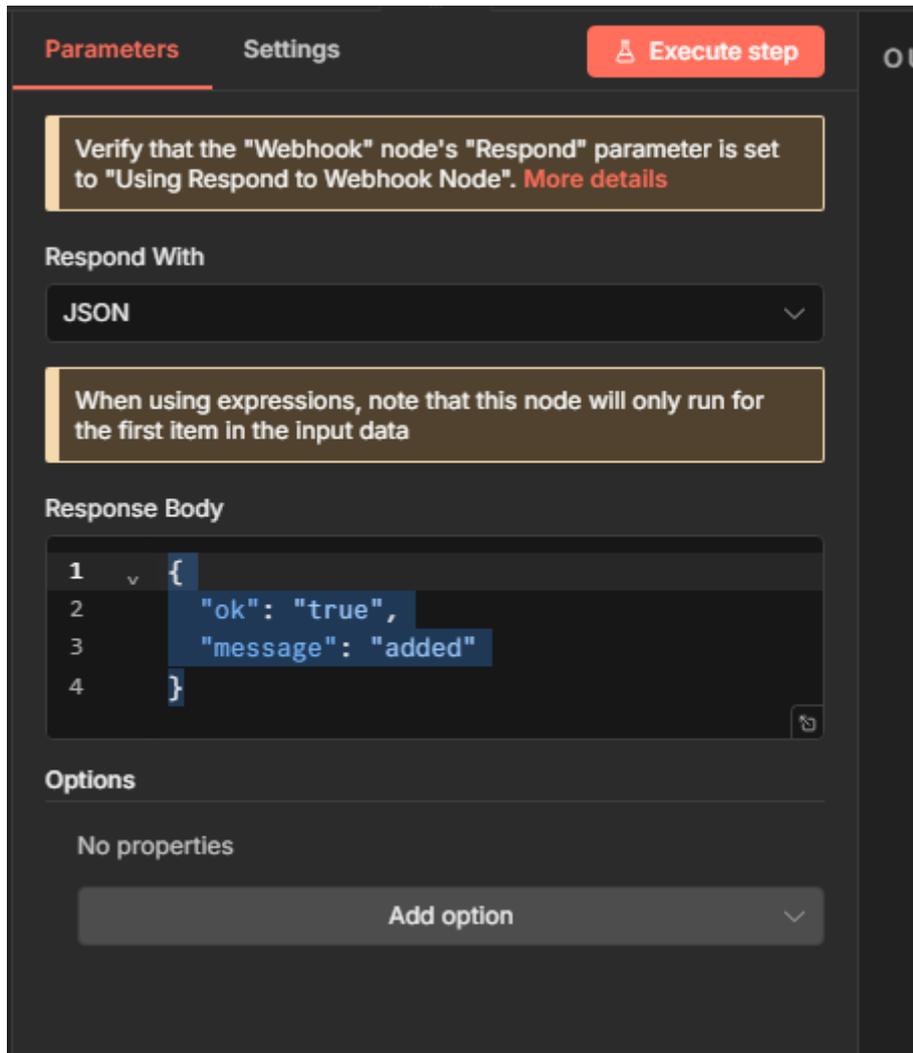
The screenshot shows the 'OAuth2 (recommended)' configuration page in the Google Cloud console. At the top, there are two radio buttons: 'OAuth2 (recommended)' (selected) and 'Service Account'. Below this, the 'OAuth Redirect URL' is set to 'http://localhost:5678/rest/oauth2-credential/callback'. A note states: 'In Google Sheets, use the URL above when prompted to enter an OAuth callback or redirect URL'. There are empty input fields for 'Client ID *' and 'Client Secret *'. The 'Allowed HTTP Request Domains' dropdown is set to 'All'. A yellow warning box contains the text: 'Make sure you enabled the following APIs & Services in the Google Cloud Console: Google Drive API, Google Sheets API. More info.' At the bottom, there is a note: 'Enterprise plan users can pull in credentials from external vaults. More info.'

- 22.
23. Сохраняем и закрываем
24. Выбираем только что созданное соединение и выбираем в полях Documents и sheets нашу таблицу и лист
25. Переписываем все настройки с скриншота



- 26.
27. Закрываем окно
28. Добавляем узел “Respond to Webhook”
29. Внутри меняем Response body на этот код:

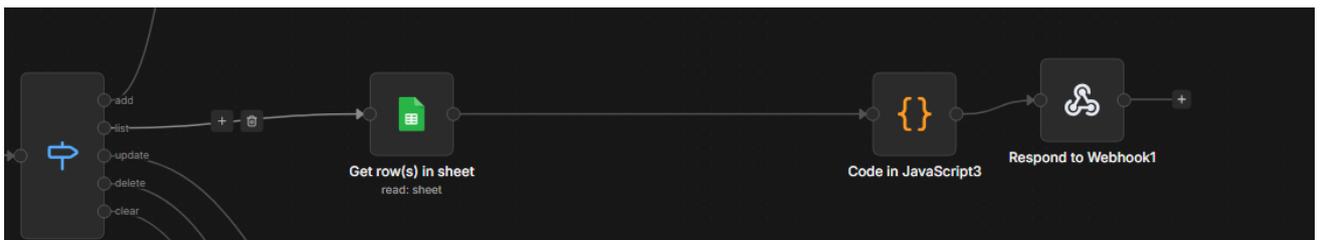
```
{  
  "ok": "true",  
  "message": "added"  
}
```



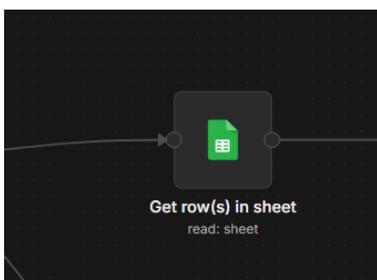
30.

31. Закрываем

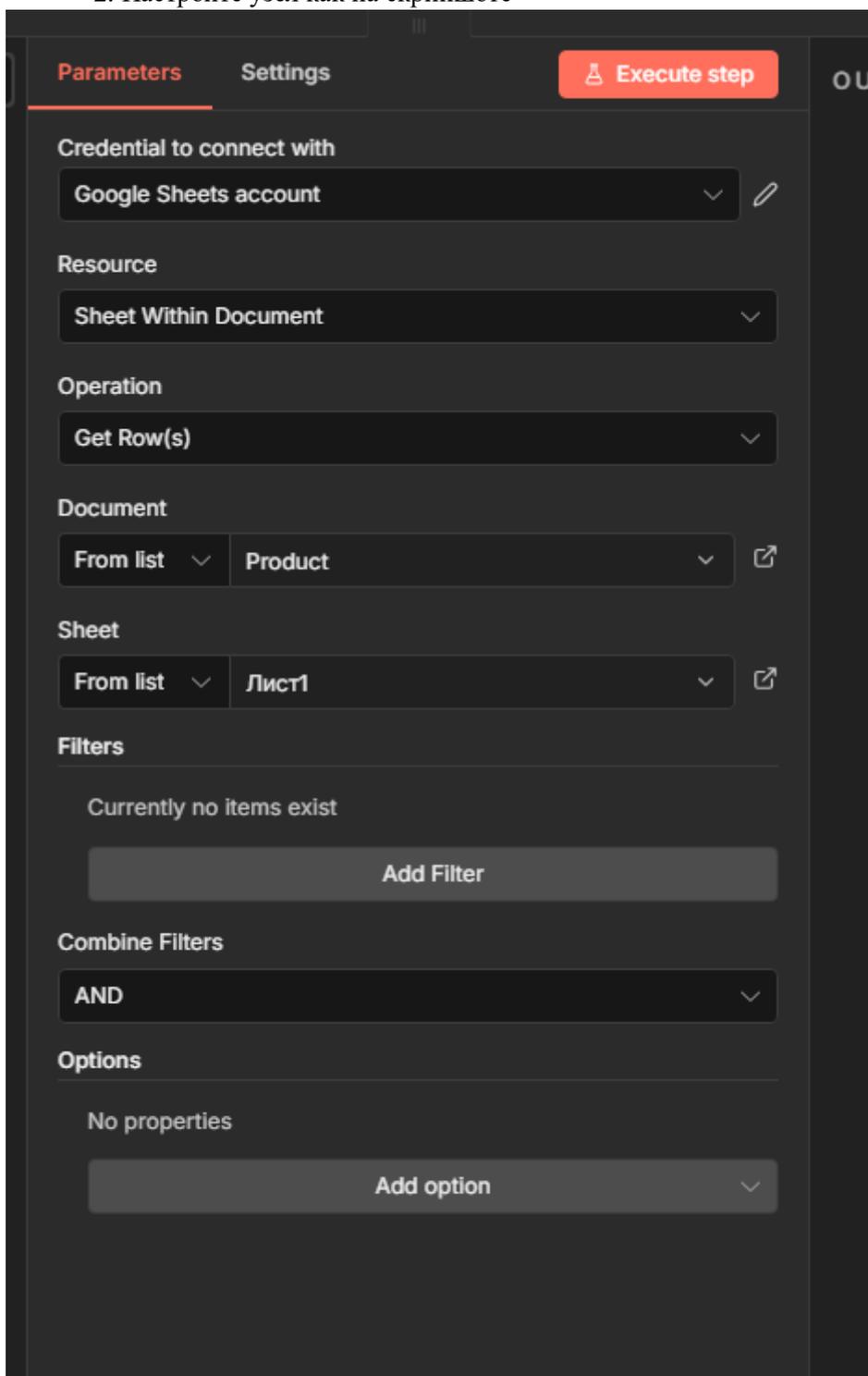
6. Case list – Просмотр списка товаров



1 Добавляем на ветку list узел get row in sheets



2. Настройте узел как на скриншоте

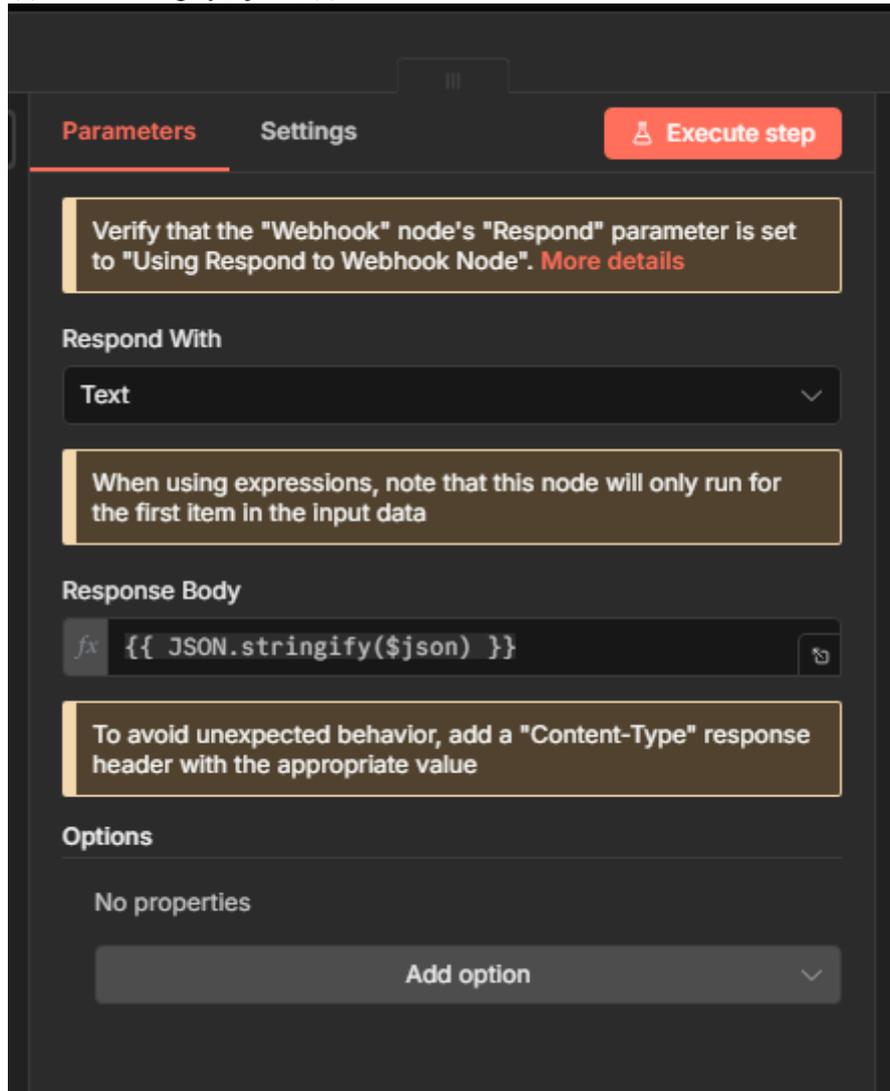


3. Добавьте узел Code JavaScript, и вставьте код

```
const rows = $input.all().map(i => i.json);  
return [{  
  json: { ok: true, items: rows }  
}];
```

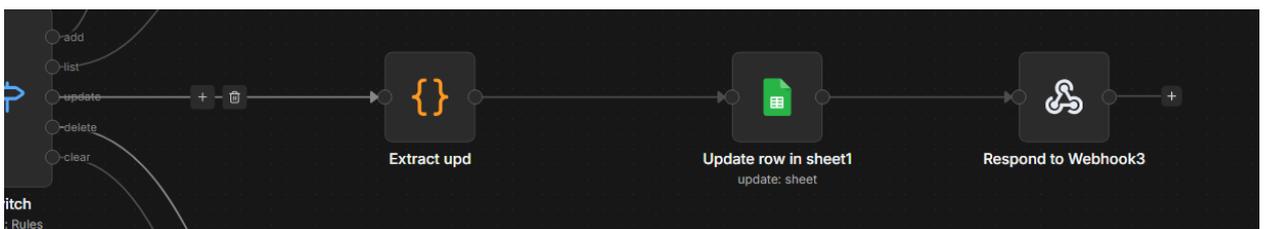
4. Добавьте узел Respond to webhook
5. Настройте как на скриншоте

6. `{{ JSON.stringify($json) }}`



7.

7 Ветка Update Обновления



1 Добавьте code на ветку update

2. В Открывшемся окне вписываем код в javascript

```
const data = $input.first().json;
```

```
const b = data.body || {};
```

```
const id = String(b.id || "").trim();
```

```
if (!id) return [{ json: { ok:false, error:'no id', debug: b } }];
```

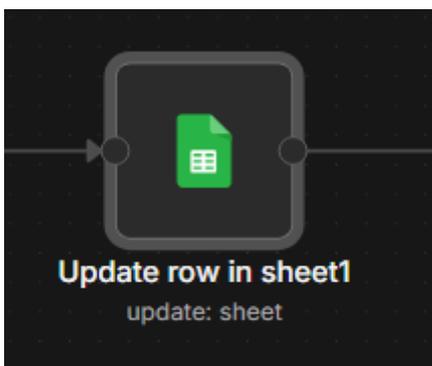
```
const name = String(b.name || "").trim();
const category = String(b.category || "").trim();
const price = Number(b.price);
const qty = Number(b.qty);

if (name.length < 2) return [{ json: { ok:false, error:'name too short' } }];
if (!Number.isFinite(price) || price <= 0) return [{ json: { ok:false, error:'bad price' } }];
if (!Number.isFinite(qty) || qty < 0) return [{ json: { ok:false, error:'bad qty' } }];

return [{
  json: {
    id,
    name,
    category,
    price,
    qty,
    UpdateAt: new Date().toISOString()
  }
}];
```

3. Закрываем окно

4. Добавляем узел Update Row



5. Заполняем в окне данные как на скриншоте

Parameters

Settings

Execute step

Credential to connect with

Google Sheets account

Resource

Sheet Within Document

Operation

Update Row

Document

From list

Product

Sheet

From list

Лист1

Mapping Column Mode

Map Each Column Manually

Column to match on

id

The column to use when matching rows in Google Sheets to the input items of this node. Usually an ID.

Values to Update

id (using to match)

`fx {{ $json.id }}`

name

`fx {{ $json.name }}`

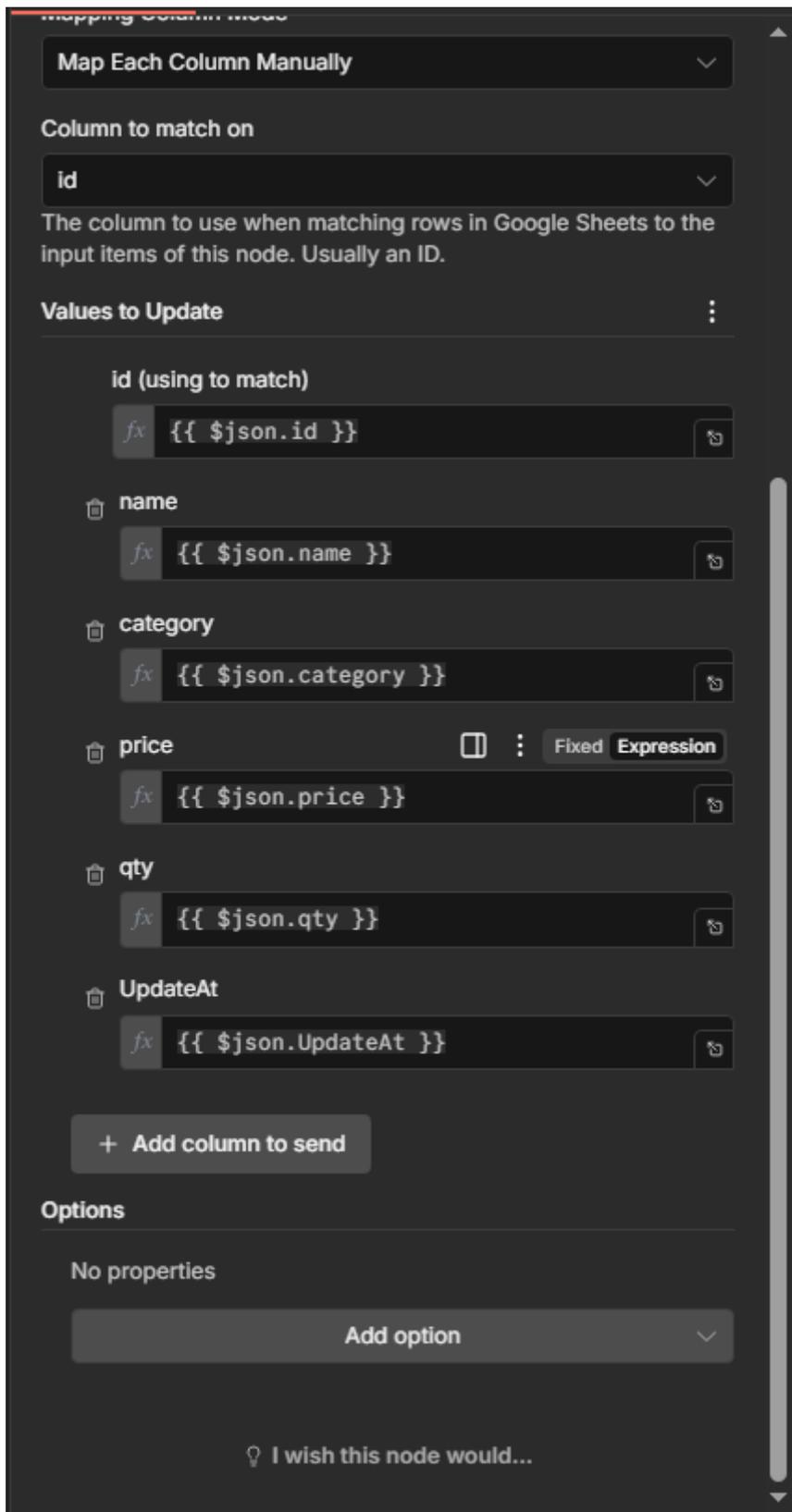
category

`fx {{ $json.category }}`

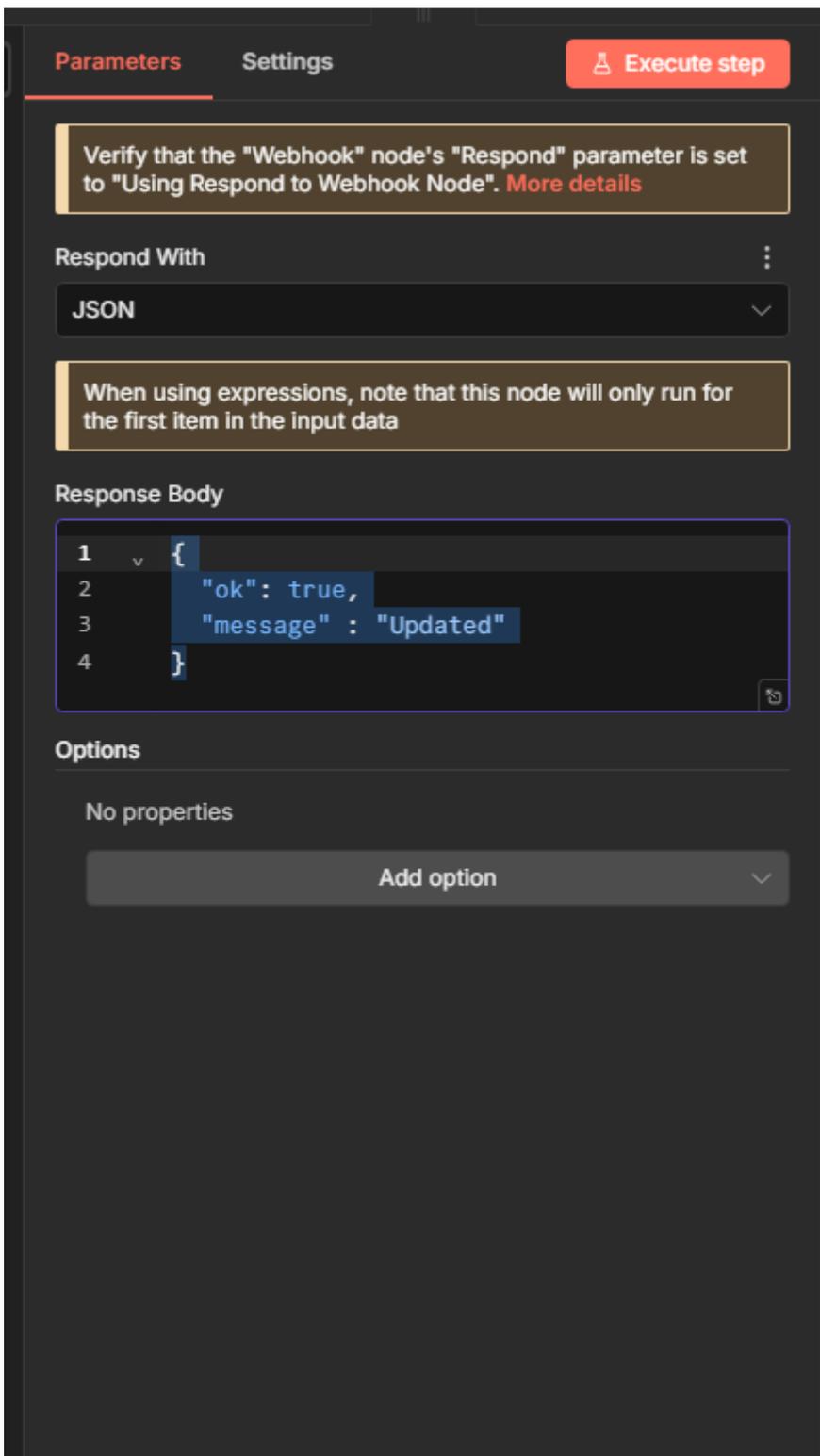
price

`fx {{ $json.price }}`

qty

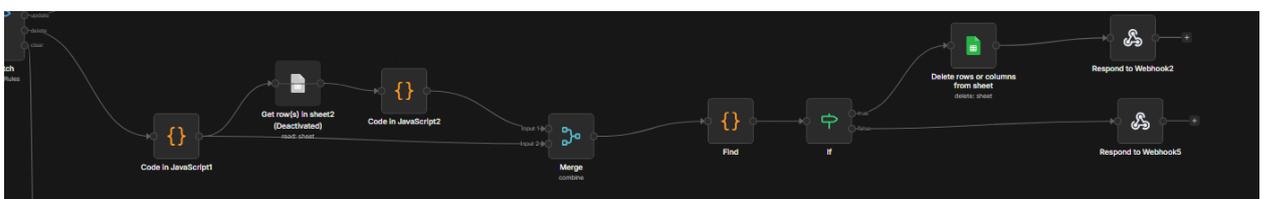


6. Закрываем
7. Создаем узел Respond to Webhook
8. вписываем данные как на скриншоте



8. Закрываем и переходим к следующей ветке

8. Ветка Delete – возможность удаления строки



1. На ветке Delete Добавляем Code java script

2. Вписываем код

```
const data = $input.first().json;
```

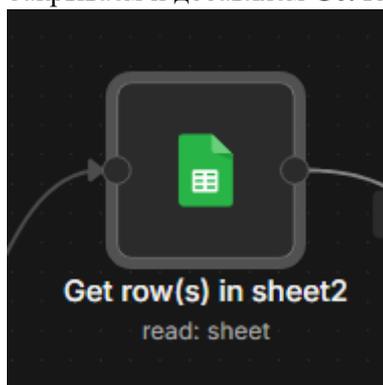
```
const body = data.body || {};
```

```
const targetId = String(body.id || "").trim();
```

```
if (!targetId) return [{ json: { ok: false, error: 'no id' } }];
```

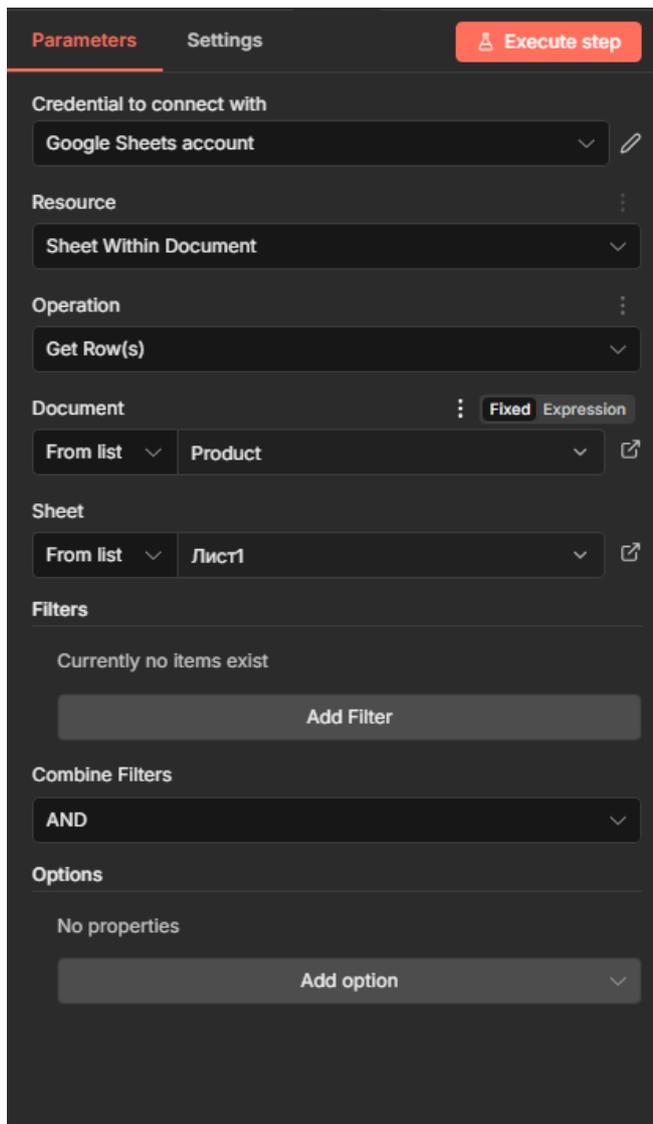
```
return [{ json: { __join: 1, targetId } }];
```

3. Закрываем и добавляем Get Row(поиск данной строки в таблице)



4.

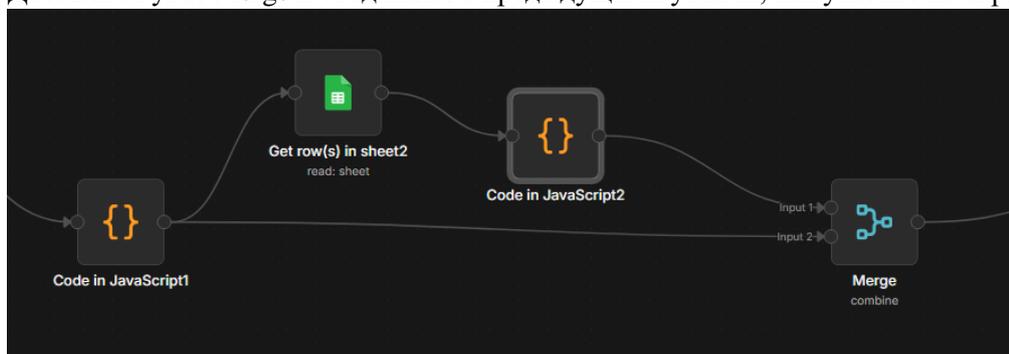
5. Делаем как на скриншоте



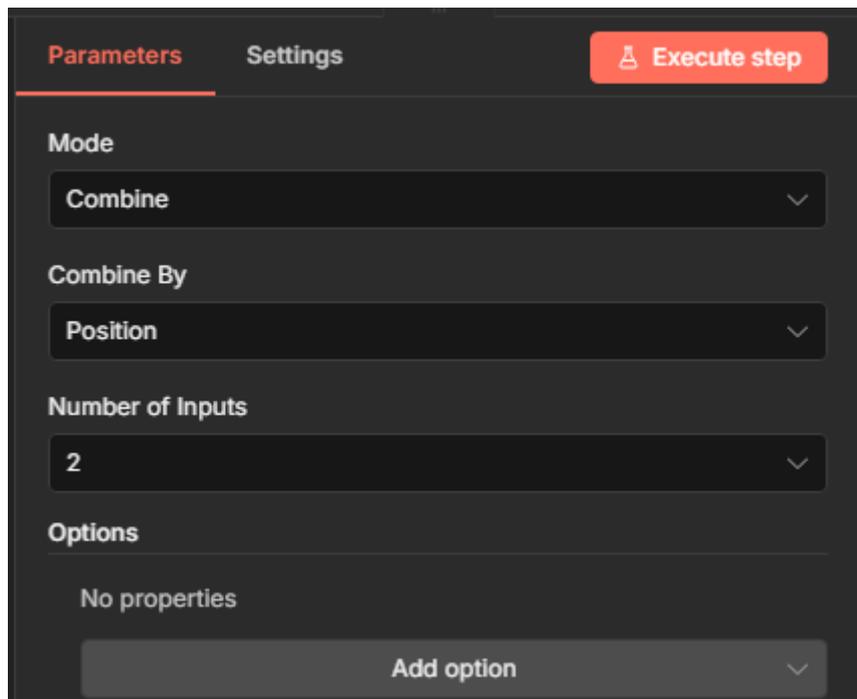
- 6.
7. Добавляем узел code javascript
8. Вписываем код

```
return $input.all().map(item => ({
  json: { ...item.json, __join: 1 }
}));
```

9. Добавляем узел Merge и соединяем с предыдущими узлами, как указано на скриншоте



- 10.
11. Настраиваем узел как на скриншоте



12.

13. Добавляем узел Code Javascript и вписываем следующий код:

```
const rows = $input.all().map(i => i.json);
```

```
// targetId теперь будет в каждой строке
```

```
const targetId = String(rows[0]?.targetId || "").trim();
```

```
if (!targetId) return [{ json: { ok: false, error: 'no id after merge' } }];
```

```
const row = rows.find(r => String(r.id || "").trim() === targetId);
```

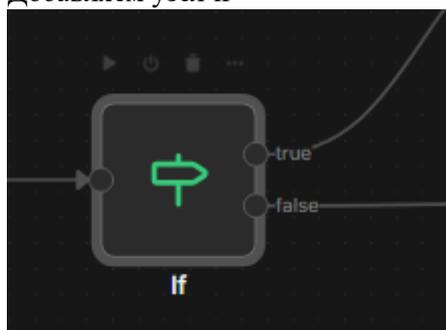
```
if (!row) return [{ json: { ok: false, error: 'not found' } }];
```

```
const row_number = Number(row.row_number);
```

```
if (!Number.isFinite(row_number)) return [{ json: { ok: false, error: 'row_number missing', row } }];
```

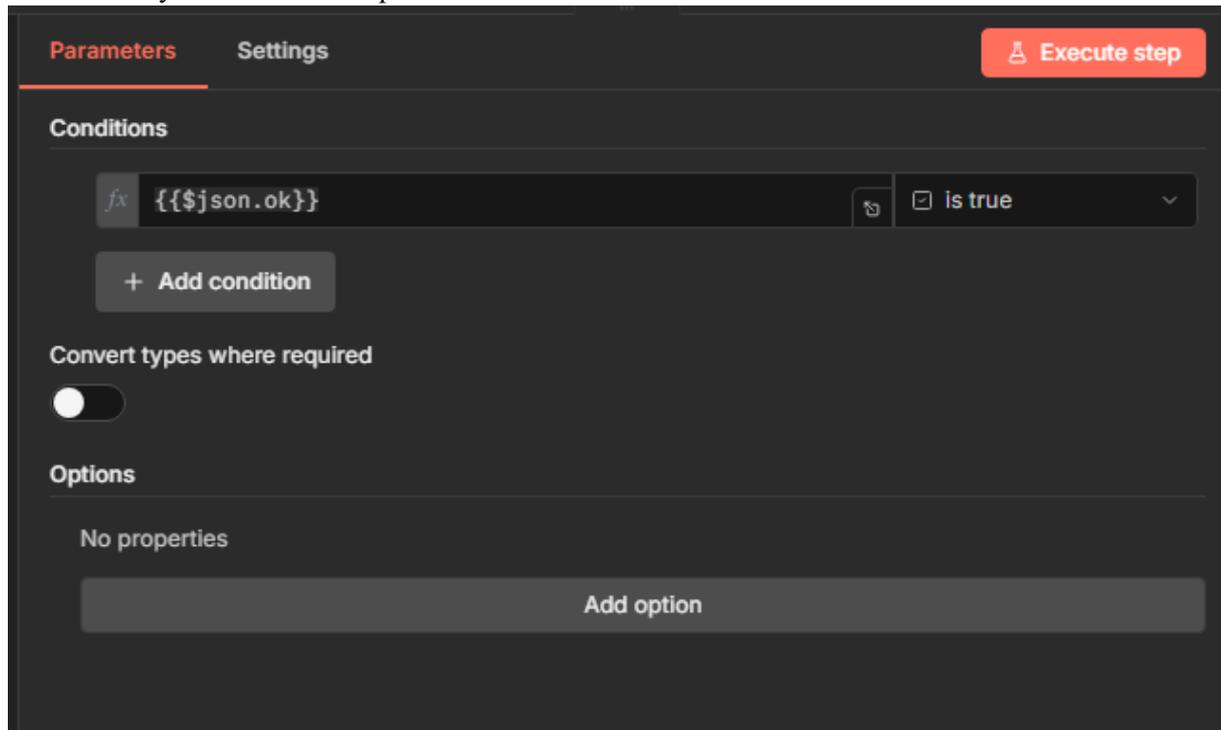
```
return [{ json: { ok: true, row_number } }];
```

14. Добавляем узел if



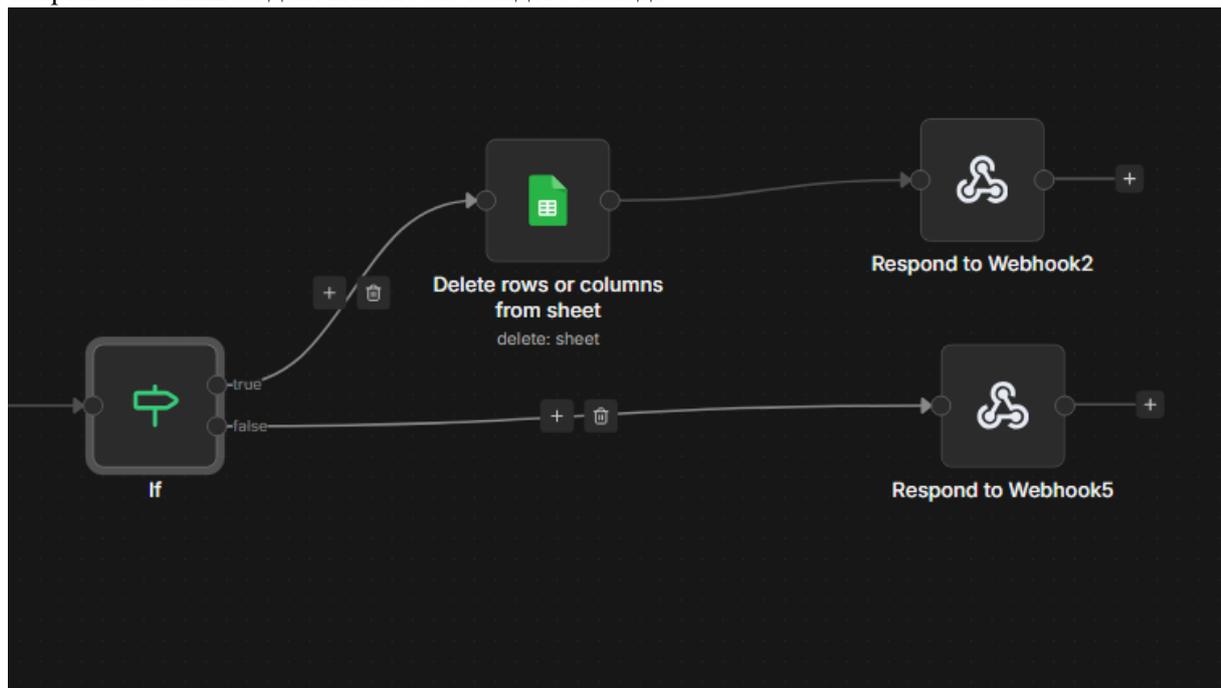
15.

16. Вписываем условие как на скриншоте



17.

18. Закрываем и пишем действия после каждого исхода

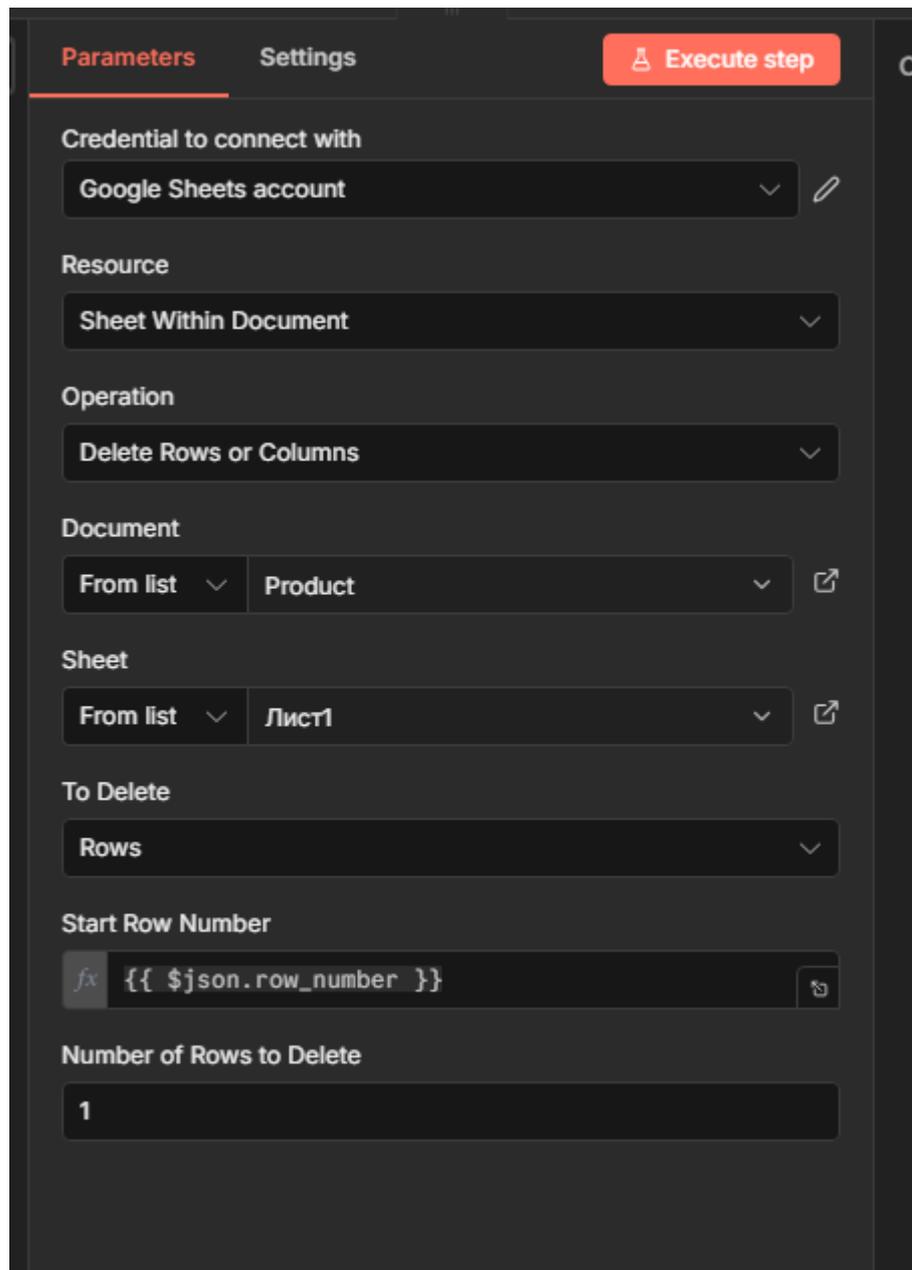


19.

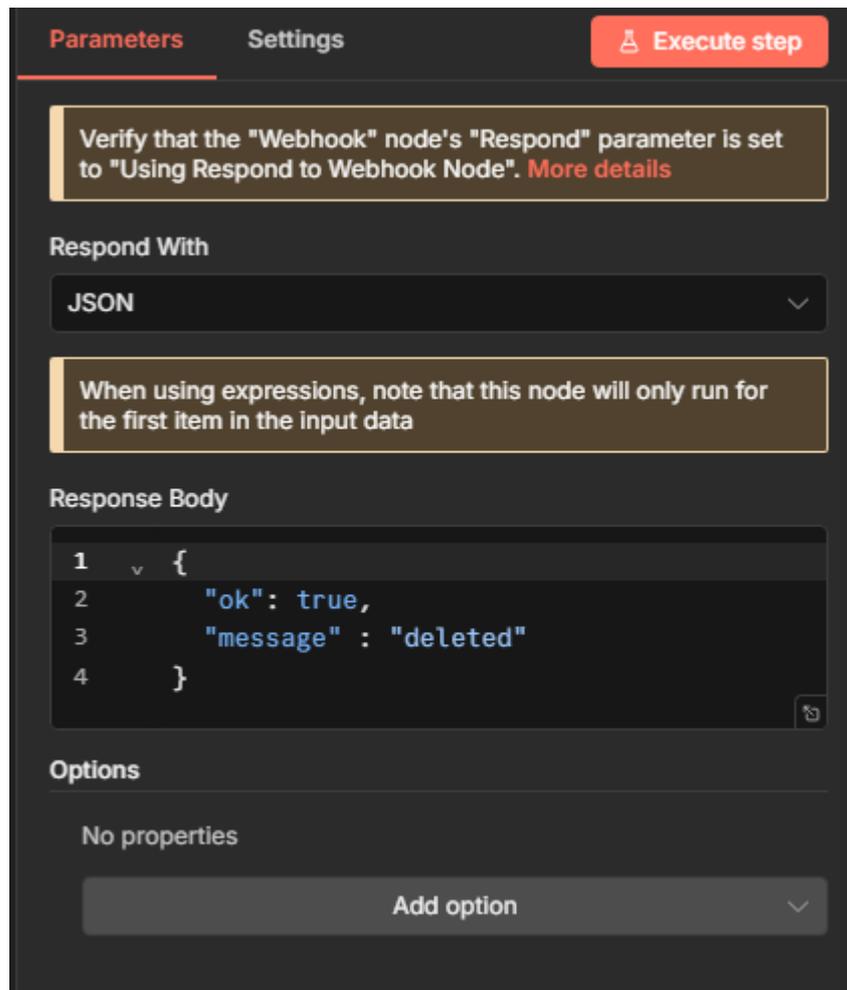
20. Если у нас нашелся id и товар, то мы идем по ветке true, если нет то по false

21. Ветка true

- a. Добавляем узел Delete rows or columns from sheet и настраиваем как на скриншоте

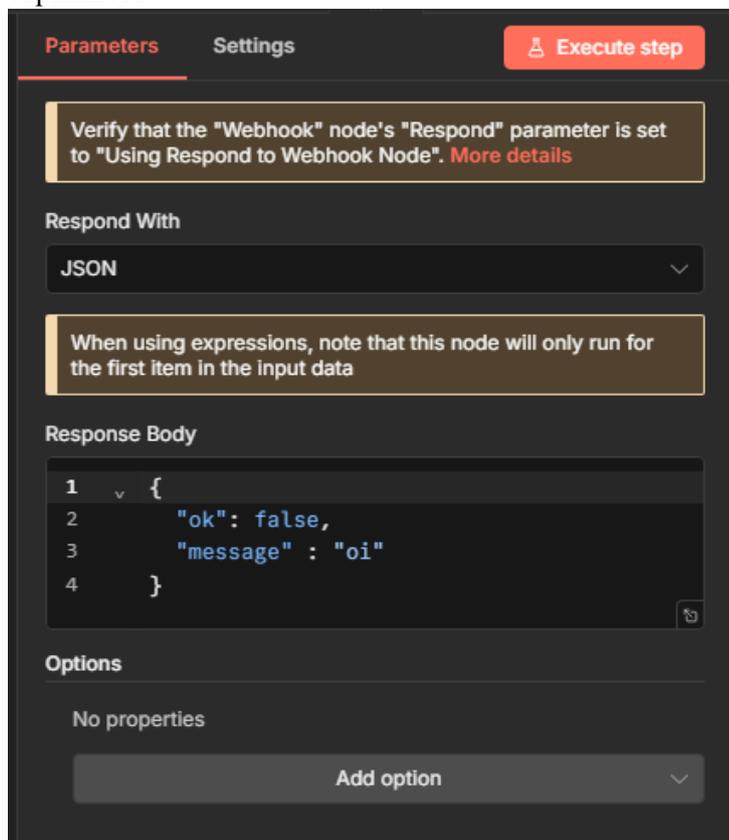


- b.
- c. Добавляем узел Respond to Webhook и настраиваем как на скриншоте



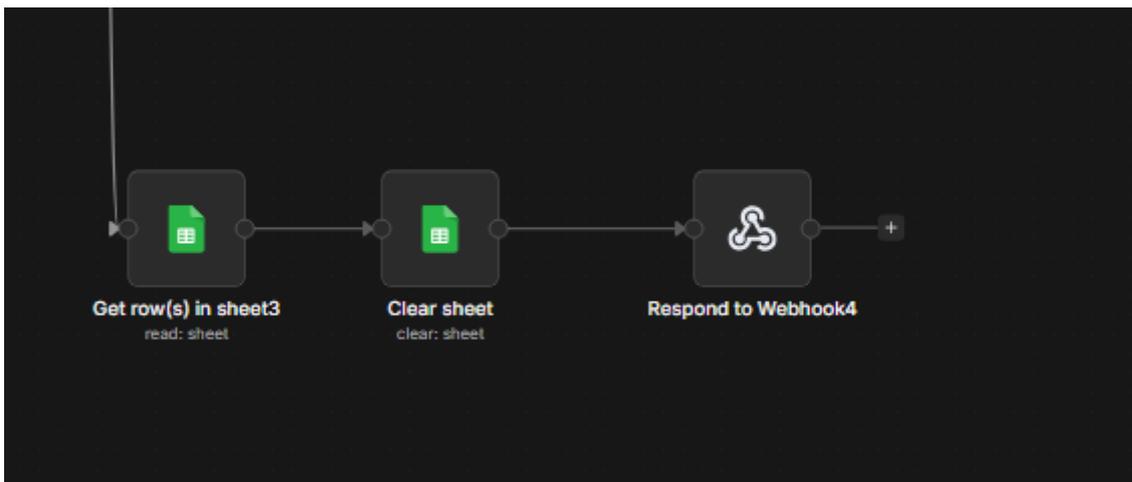
d.

22. На ветке False Также добавляем узел Respond to Webhook и настраиваем как на скриншоте



23.

9. Ветка Clear – очистка таблицы полностью

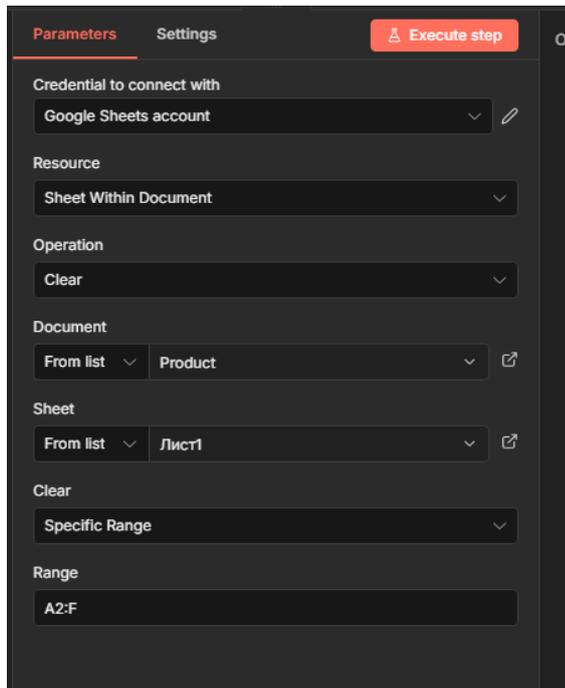


1. Добавляем узел Get row и настройте его

The screenshot shows the configuration panel for the 'Get row(s) in sheet3' step. It has two tabs: 'Parameters' (selected) and 'Settings'. An 'Execute step' button is in the top right corner. The configuration includes:

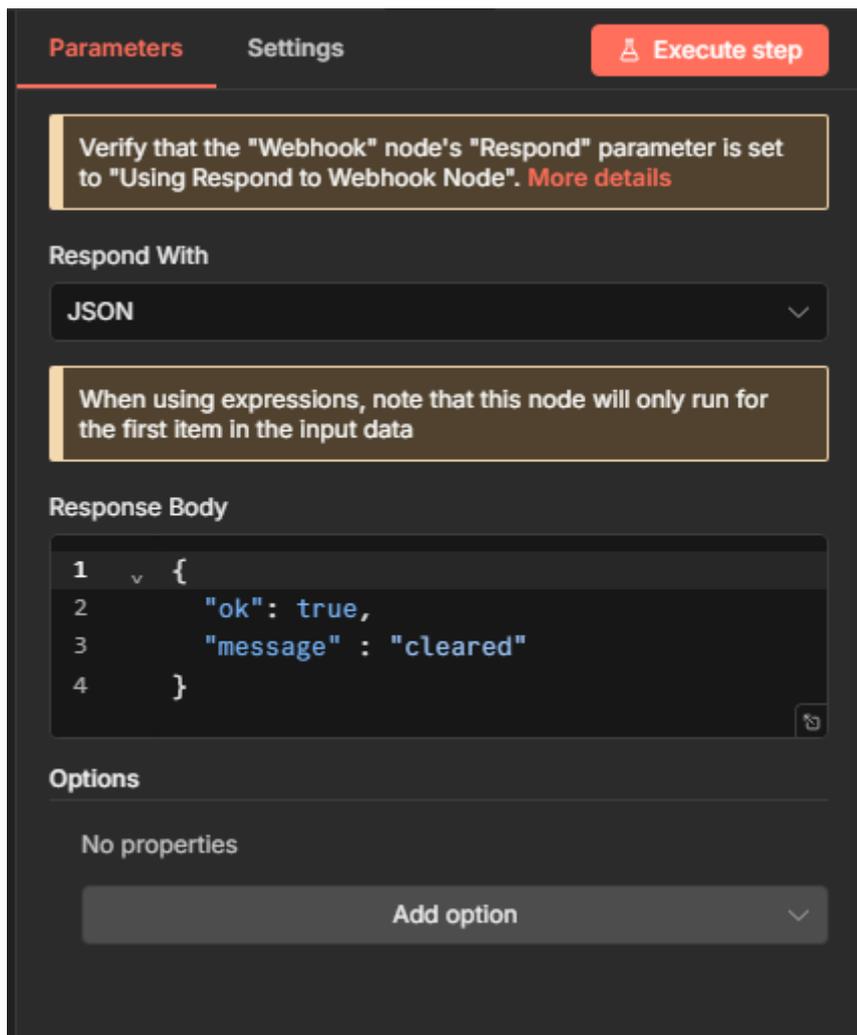
- Credential to connect with:** Google Sheets account
- Resource:** Sheet Within Document
- Operation:** Get Row(s)
- Document:** From list (Product)
- Sheet:** From list (Лист1)
- Filters:** Currently no items exist. Add Filter button.
- Combine Filters:** AND
- Options:** No properties. Add option button.

- 2.
3. Добавляем узел clear sheet и настраиваем



4.

5. Добавляем узел Respond to Webhook и настраиваем его



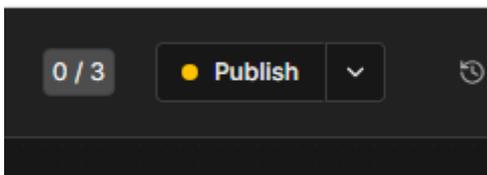
6.

7. Закрываем

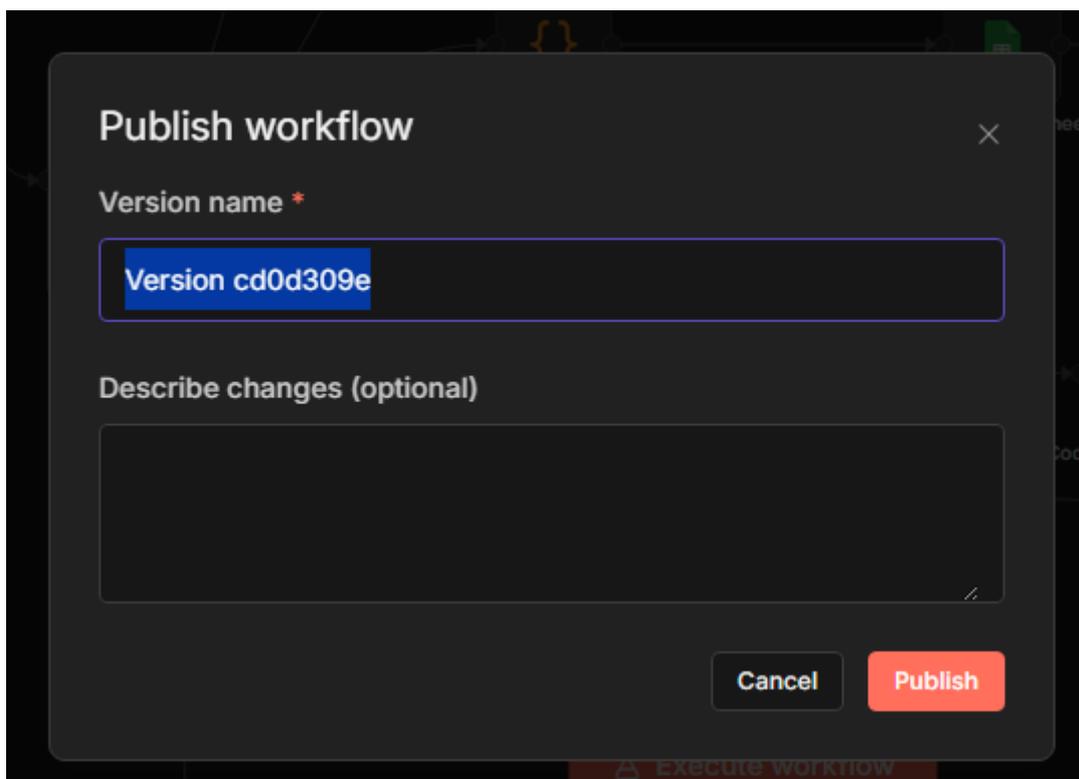
10 сайт

Теперь настал этап тестирования

В правом верхнем углу надо нажать на кнопку Publish



Нажимаем на кнопку Publish



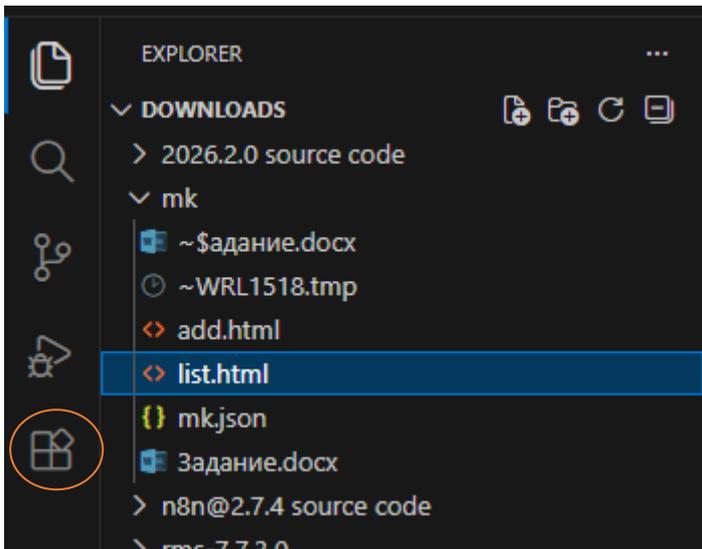
Если ранее не создали папку, то создаем папку с названием практики и вашим фио

В эту папку копируем файлы add.html и list.html

Запускаем приложение visual studio code и выбираем эту папку

Если нету расширений

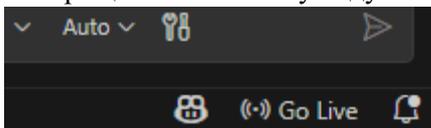
В этой папке нажимаем на значок расширений



В открывшееся вкладке ищем Live Server и устанавливаем его

Перезапускаем vscode

Возвращаемся к нашему коду и нажимаем в правом нижнем углу Go Live



11. Тестирование

A screenshot of the 'AI Анализ текста' web application. The interface is light purple and white. At the top, the title 'AI Анализ текста' is displayed, along with the API key 'API: 127.0.0.1:5678 • Ollama: 127.0.0.1:11434'. Below the title, there is a breadcrumb trail: 'Сайт → n8n Webhook → Ollama. Есть история запросов (localStorage)'. The main content area is divided into three columns. The left column is titled 'Текст для анализа' and contains a text input field with the example text: 'Например: Доставка быстрая, но упаковка была повреждена.' Below the input field are four buttons: 'Тональность' (purple), 'Резюме' (blue), 'Ключевые слова' (dark blue), and 'Сделать всё сразу' (green). The middle column contains three sections: 'Тональность' (with a minus sign), 'Резюме' (with a minus sign), and 'Ключевые слова' (with a minus sign). Each section currently shows 'Нет данных'. The right column is titled 'История запросов' and shows a list of requests. The first request is from '2026-03-03 10:58:00' with the label 'Тональность'. The text of the request is: 'Доставка пришла быстро, менеджер отвечал вежливо. Но коробка была слегка помята, и инструкция оказалась неполной. В целом покупкой доволен.' Below the text are buttons for 'Позитивный', 'Повторить', 'Вставить текст', and 'Удалить'. The second request is from '2026-03-03 10:26:37' with the label 'Ключевые'. The text of the request is: 'Быстрота доставки, вежливый менеджер, слегка помятая коробка, неполная инструкция.' Below the text are buttons for 'Повторить' and 'Удалить'. At the bottom of the right column, there is a note: 'Можно повторить любой запрос или удалить одну запись.'

Тестируем каждую кнопку и пишем разные предложения для теста.

Теперь у вас есть полностью рабочий веб-сервис с полноценным ии бекендом, и с базой данных в виде онлайн таблицы, которая обновляется в реальном времени с помощью p8n