



FM

FLOWMATRIX.DEV

Premium Workflow

A practical guide – step by step

Price Monitoring

Track competitor prices automatically: scrape product pages, detect changes and trigger alerts or repricing rules.

E-COMMERCE

ADVANCED

ABOUT 3 HOURS SETUP

OVERVIEW

Price Monitoring

Track competitor prices automatically: scrape product pages, detect changes and trigger alerts or repricing rules.

CATEGORY	DIFFICULTY	DURATION
E-commerce	Advanced	About 3 hours setup

What you use it for

E-commerce teams keeping prices competitive.

Tools in this workflow

n8n · Supabase · OpenAI API · Resend Email API

How to implement it

1

Define competitor product URLs

In this step, competitor product URLs are set up to be monitored. These URLs serve as a basis for price monitoring.

WHAT YOU NEED TO DO

1. Create a list of competitor product URLs.
2. Store the list in a Supabase database table named 'competitor_urls'.
3. Verify the accessibility of each URL by making a test call.

CHECK AT THE END

- [] All relevant product URLs are included in the list.
- [] The URLs are stored in the Supabase database.

RESULT

A list of competitor product URLs is stored in the Supabase database.

2

Scrape prices on a schedule

This creates a schedule for regularly scraping prices from defined URLs to ensure data stays up-to-date.

WHAT YOU NEED TO DO

1. Create an n8n workflow that retrieves URLs from the Supabase database.
2. Add an HTTP Request Node to scrape product pages.
3. Schedule the workflow to run at regular intervals (e.g. every 24 hours).

CHECK AT THE END

- [] The n8n workflow is correctly configured.
- [] The workflow runs according to the scheduled time.

RESULT

An n8n workflow scrapes prices regularly from defined URLs.

3

Detect and store changes

Collected price data is analyzed to recognize changes and store significant price changes.

WHAT YOU NEED TO DO

1. Process scraped data in an n8n workflow to recognize price changes.
2. Store recognized changes in a Supabase table named 'price_changes'.
3. Create a report on price changes for later analysis.

CHECK AT THE END

- [] Price changes are correctly recognized.
- [] Changes are stored in the Supabase database.

RESULT

Price changes are recognized and stored in the Supabase database.

4

Apply alert thresholds

Define threshold values that determine when a price change is considered significant, to avoid unnecessary notifications.

WHAT YOU NEED TO DO

1. Create a logic in your n8n workflow to compare price changes with the defined threshold values.
2. Define threshold values for significant price changes in a configuration file.
3. Update the workflow to trigger notifications only when the threshold values are exceeded.

CHECK AT THE END

- ☐ Threshold values are correctly defined.
- ☐ The workflow triggers notifications only for significant changes.

RESULT

The workflow considers only significant price changes based on the defined threshold values.

5

Notify the pricing team

Relevant team members of the pricing team will be informed about recognized changes to quickly respond to market changes.

WHAT YOU NEED TO DO

1. Integrate the Resend Email API into the n8n workflow to send notifications.
2. Configure the email template with relevant price information.
3. Ensure that emails are sent to the correct team members.

CHECK AT THE END

- ☐ Email notifications are correctly configured.
- ☐ Notifications are sent when significant changes occur.

RESULT

The pricing team will be notified via email about significant price changes.

RECOMMENDED TOOLS & RESOURCES

Recommended Tools & Resources

All tools in this workflow with their official website.

n8n

<https://n8n.io/?ref=flowmatrix>

Official URL

Supabase

<https://supabase.com/?ref=flowmatrix>

Official URL

OpenAI API

<https://platform.openai.com/?ref=flowmatrix>

Official URL

Resend Email API

<https://resend.com/?ref=flowmatrix>

Official URL

Full version as Premium PDF

All steps, tools and resources as a PDF – €4.99.

[Buy now – €4.99](#)