✅ Top 3 APIs for n8n + Scraping/Search

1. Apify
    - Integration with n8n: n8n has a built-in “Apify” node. You can run Apify Actors (scrapers), fetch datasets, or scrape a single URL.
    - What it’s good at:
        - Full browser-based scraping (Playwright / Puppeteer) or lightweight HTTP client → handle complex, JS-heavy e-commerce sites.
        - Use “Actors” (pre-built scrapers) or build custom ones to extract product details, reviews, listings, etc.
        - Automate workflows: schedule scrapes, chain scrapers, trigger other n8n nodes when data is ready.
    - Why choose it: Very flexible, scalable, and well-integrated into n8n for more advanced / large-scale scraping tasks.
2. ScraperAPI
    - Integration with n8n: There is an official / community node for ScraperAPI in n8n.
    - What it’s good at:
        - Simple “send URL, get back HTML (or parsed response)” model.
        - Handles proxy rotation, CAPTCHAs, rendering of JS automatically.
        - You can configure device type (desktop/mobile), geo-targeting, and level of proxy / rendering.
        - Supports integration with n8n’s MCP client for real-time AI agent use.
    - Why choose it: Lightweight and reliable for “on-demand scraping” from workflows. If your agent just needs to fetch product pages and extract info, this is easy to plug in.
3. ScrapingBee

- Integration with n8n: There is a verified ScrapingBee integration in n8n.
- What it's good at:
  - Web scraping API that supports JavaScript rendering, rotating proxies, and CAPTCHAs.
  - Good for extracting structured data (e.g., product details from Amazon, Alibaba) because you can reliably fetch fully rendered HTML.
  - Their Amazon-specific endpoint helps with product APIs (ScrapingBee has Amazon Search / Amazon Product API).
- Why choose it: Balanced between power and simplicity: more robust than plain HTTP + proxies, but lighter than full-blown browser automation.

⚠️ Bonus Option to Consider

- Scrapfly: They have an official n8n integration. Their Web Scraping API supports browser rendering, anti-bot, and they also have an Extraction API for parsing product data directly.
- Bright Data: Also supported in n8n. You can use their HTTP API or MCP integration to do large-scale scraping.
- ScrapeNinja: Has an n8n community node for browser + JS scraping, extracting content via custom JS.

💡 My Recommendation (for AI Agents that Search + Scrape)

- Use Apify for your core scraping: run actors to get product pages, crawl, extract structured product data — fits well in an "agent picks a URL → scrape → return JSON → agent reasons."
- Use ScraperAPI for lighter "on-the-fly" scraping from agents when they need a quick look at a page or search result.

- Use ScrapingBee when you specifically want product data from e-commerce sites and need JS rendering + stable proxy but don’t need full actor control.

📄 Workflow Report: "Content Idea Generator"

This report details the structure, purpose, and function of the automated workflow named **"Content Idea Generator"**. This workflow is a sophisticated content pipeline designed to harvest news from an automation-focused RSS feed, transform the articles into structured YouTube video concepts using a large language model (LLM), and record the results for production planning.

---

1. Workflow Summary

The **"Content Idea Generator"** workflow is an end-to-end automation process comprising ten connected nodes. Its primary objective is to streamline the content brainstorming process for a YouTube channel specializing in AI and automation.

- **Input Source:** The RSS feed for the `/r/Automation` subreddit.
- **Core Action:** An **AI Agent** powered by a Groq Chat Model generates three structured video ideas per article.
- **Output Destination:** A Google Sheet named **"youtube idea generator"** for tracking and production.

2. Node-by-Node Analysis

The workflow executes in a linear fashion, with distinct phases for data ingestion, cleaning, processing, and output.

Phase 1: Data Ingestion and Normalization (Nodes 1-4)

Node Name	Node Type	Description
When clicking ‘Execute workflow’	`manualTrigger`	Serves as the starting point, manually initiating the process.
HTTP Request	`httpRequest`	Fetches the raw XML data (RSS feed) from `https://www.reddit.com/r/Automation/.rss`.

XML	xml	Converts the fetched XML content into a structured, machine-readable JSON format.
Code	code	Executes a robust custom script to parse the varied structures of RSS/Atom JSON output, extracting and standardizing the **title**, **link**, and **pubDate** for all article entries.

Phase 2: AI Processing and Content Generation (Nodes 5-9)

Node Name	Node Type	Description
Code1	code	Filters the data to a minimalist payload, passing only the **title** and **link** for each article to the AI Agent, reducing token usage and focusing the input.
AI Agent	agent	The central component. It uses the input article details and a detailed **System Message** (acting as an experienced YouTube content strategist) to generate **three unique video concepts** per item.
Groq Chat Model	lmChatGroq	The powerful language model (openai/gpt-oss-20b) that executes the creative generation logic defined by the AI Agent's prompt.
Simple Memory	memoryBufferWindow	Provides short-term memory (30 messages) for the AI Agent, ensuring context is maintained during potentially complex or multi-step interactions.

Code2	code	A critical cleaning step that **parses the raw text output** from the AI Agent. It specifically removes JSON delimiters (\``json`) and converts the resulting string into a structured array of JSON objects.

Phase 3: Final Output (Node 10)

Node Name	Node Type	Description
Google Sheets	googleSheets	Takes the cleaned and structured JSON output and performs an **append** operation to save the video ideas—specifically **videoTitle**, **description**, **hook**, and **format**—into the "**Sheet1**" of the "**youtube idea generator**" spreadsheet.

3. Conclusion

The **"Content Idea Generator"** workflow successfully automates a complex content strategy process. It reliably converts disparate web data into a clean, actionable dataset ready for immediate use by a content team, demonstrating effective use of data transformation and advanced language model capabilities within a unified automation platform.