NotebookLM Generation Tool

Complete Technical Documentation & User Guide

Version 1.0.0
December 2024

Table of Contents

1. Project Overview

1.1 What is NotebookLM Generation Tool?

The NotebookLM Generation Tool is a Python-based automation framework that interfaces with Google's NotebookLM service to automatically generate educational materials from PDF documents and other content sources.

1.2 Key Features

- **Batch Processing:** Process entire folders of PDFs automatically
- **Material Generation:** Create Audio Summaries, Videos, Mindmaps, Quizzes, Flashcards, and Infographics
- **Notebook Management:** Automatically create and name notebooks
- **Download Automation:** Bulk download all generated materials
- **Browser Automation:** Uses Selenium with persistent Chrome profiles

1.3 Why Browser Automation?

Google NotebookLM's API is **Enterprise-only** (paid). The public/consumer version has no API access. Browser automation is the only approach for free accounts.

2. Architecture

2.1 Project Structure

```
notebook-lm-generation/
├── src/
│   ├── main.py              # CLI entry point
│   ├── auth/google_auth.py     # Chrome profile & auth
│   ├── generators/
│   │   ├── notebooklm.py       # Core browser automation
│   │   └── studio_automator.py # Studio panel automation
│   └── processors/
├── install.sh                  # Installation script
└── requirements.txt            # Dependencies
```

2.2 Core Components

NotebookLMClient

The main browser automation client handling navigation, notebook creation, source management, and chat interactions.

StudioAutomator

Advanced Studio panel automation for source selection, material generation (all 6 types), language selection, and downloads.

GoogleAuthenticator

Chrome browser management with persistent profile at ~/.nlm_chrome_profile for session reuse.

3. Installation

3.1 Prerequisites

- Python 3.11 or higher
- Google Chrome browser
- Google account (logged into Chrome)

3.2 Quick Install

```
cd /path/to/notebook-lm-generation
./install.sh
```

The installer creates a virtual environment, installs dependencies, and makes the **nlmgen** command available globally.

4. Usage Guide

4.1 Single File Processing

```
# Process PDF with name prompt
nlmgen "/path/to/document.pdf"

# Process with custom notebook name
nlmgen "/path/to/document.pdf" --name "My Study Notes"

# Auto-name from filename
nlmgen "/path/to/document.pdf" --auto-name
```

4.2 Batch Processing

```
# Process all PDFs in folder (prompts for each name)
nlmgen --batch "/path/to/folder/"

# Auto-name all notebooks
nlmgen --batch "/path/to/folder/" --auto-name

# Generate only specific materials
nlmgen --batch "/path/to/folder/" --materials audio video --auto-name
```

4.3 Material Types

Material	Flag	Time	Language	Download
Audio Summary	`audio`	3-10 min	English	.mp3
Video Overview	`video`	5-15 min	English	.mp4
Mindmap	`mindmap`	2-5 min	Auto	.png
Quiz	`quiz`	1-3 min	Auto	No
Flashcards	`flashcards`	1-3 min	Auto	No
Infographic	`infographic`	3-8 min	English	No

4.4 Download Materials

```
# Download all completed materials from a notebook
nlmgen --notebook-url "https://notebooklm.google.com/notebook/..." --action
download
```

5. Implementation Details

5.1 Browser Automation Approach

The tool uses Selenium WebDriver with multiple fallback strategies for each UI element to handle NotebookLM's dynamic interface.

Element Finding Strategy

Each UI element has multiple XPath and CSS selectors as fallbacks:

- Primary: Specific aria-label patterns
- Secondary: Text content matching
- Tertiary: CSS class patterns
- Fallback: Generic element search

German UI Support

The tool fully supports the German NotebookLM interface:

- "Quellen" → Sources
- "Karteikarten" → Flashcards
- "Erstellen" → Create
- "Herunterladen" → Download

5.2 Studio Panel Automation Flow

For each source in the notebook:

8. Deselect all sources
9. Select single target source
10. Open Studio panel
11. Generate each material type
12. For Audio/Video/Infographic: Select English language
13. Click Create/Erstellen button
14. Move to next source

6. API Reference

6.1 CLI Arguments

Argument	Short	Description
`--batch FOLDER`	`-b`	Process all PDFs in folder
`--notebook-url URL`	`-n`	Use existing notebook
`--materials`	`-m`	Materials to generate
`--auto-name`	`-a`	Auto-name from filename
`--name NAME`		Custom notebook name
`--list-sources`	`-l`	List sources in notebook
`--download`	`-d`	Download all materials
`--action`		studio, download, chat
`--headless`		Run browser headless

6.2 Python API

```
from src.generators.studio_automator import StudioAutomator, MaterialType

# Initialize
studio = StudioAutomator(driver)

# Generate materials
studio.process_all_sources(materials=[MaterialType.AUDIO])

# Download all
results = studio.download_all_materials()
```

7. Troubleshooting

7.1 Common Issues

"Element not found"

- The UI may have changed - update selectors
- Wait longer for page load
- Check if logged into Google

"Download fails"

- Item may still be generating (wait longer)
- Item type may not support download (Quiz, Flashcards)
- Check Downloads folder permissions

7.2 Debug Mode

```
# Run with verbose logging
nlmgen document.pdf -v

# Check error reports
ls ~/nlm_error_*.png
```

7.3 Reset Chrome Profile

```
# Remove cached profile
rm -rf ~/.nlm_chrome_profile
```

Next run will create a fresh profile and require Google login.

End of Documentation

For updates and support, visit the project repository.