

Blog Content to LinkedIn Post Automation Tool

Project Description

This Python utility automates the transformation of long-form blog content into highly engaging, professional LinkedIn posts. It uses robust web scraping techniques to extract clean article text and then leverages the **Gemini API** with an expert copywriter prompt to generate structured, ready-to-publish social media updates.

A key feature is the automatic organization of all generated posts into a designated output folder, ensuring a clean and manageable content pipeline.

Features

- **Robust Web Scraping:** Uses requests and BeautifulSoup with a specific User-Agent to successfully fetch and clean the main body text from most blog URLs.
- **Expert AI Copywriting:** Utilizes a detailed, optimized prompt in the Gemini model to ensure the output is tailored specifically for LinkedIn (strong hook, key takeaways, clear CTA, and relevant hashtags).
- **Secure Authentication:** Loads the API key securely from a local .env file using python-dotenv.
- **Automatic File Saving:** Saves all generated posts to a dedicated output_summaries folder with unique, timestamped filenames derived from the blog URL.
- **Interactive CLI:** Simple command-line interface for specifying the target URL.

Prerequisites

Before running the script, ensure you have:

1. **Python 3.8+**
2. A **Gemini API Key** (required for the AI generation step).

Setup and Installation

1. File Structure

Ensure your project directory contains the following two files (you already have them):

```
/  
└── main.py      # The core scraping, generation, and saving script.  
└── .env        # File containing your secret API key.
```

2. Install Required Python Libraries

This project relies on four external libraries:

```
pip install google-genai requests beautifulsoup4 python-dotenv
```

3. Configure Your API Key (Mandatory)

The script is set up for secure key handling. You **must** create a file named `.env` in the same directory as `main.py` and add your API key in the following format:

```
GEMINI_API_KEY="YOUR_API_KEY_HERE"
```

(Replace `YOUR_API_KEY_HERE` with your actual key.)

Note on Authentication: The `python-dotenv` library is used in `main.py` to automatically load this variable into your system environment when the script starts, resolving common "API key missing" errors.

Usage

1. Execute the main script from your terminal:
`python main.py`

2. When prompted, enter the full URL of the blog post you wish to summarize:
Please enter the blog link (URL) you want to summarize:
<https://www.onsaas.me/blog/b2b-growth-strategies>

Output & File Organization

Upon successful execution, the full LinkedIn post will be displayed in your console, and a new folder named `output_summaries` will be created (if it doesn't exist).

The full output, including the URL and generation time, will be saved to a uniquely named file, such as:

```
/output_summaries/20251024_193045_b2bgrowthstrategies.txt
```

Output Example

The script provides the ready-to-use output directly to the console:

```
=====  
🚀 GENERATED LINKEDIN POST 🚀  
=====
```

Tired of generic "growth hacks"? 🙌 This is the playbook for 2025 B2B success.

Our latest research breaks down the 12 non-negotiable strategies that market leaders are using right now:

- ✓ **ICP Precision:** Stop selling to everyone. Hyper-focus your ideal customer profile.
- 💡 **The Handoff:** Optimize the moment Marketing hands a lead to Sales—where most revenue is lost.
- 🚀 **CS as Sales:** Turn customer success stories and retention into your most powerful acquisition engine.

Ready to leave the guesswork behind? Read the full analysis now: [BLOG_URL_HERE]

#B2BGrowth #SaaS #MarketingStrategy #LinkedInTips

```
=====
```

✓ Full output saved to:
/path/to/project/output_summaries/20251024_193045_b2bgrowthstrategies.txt
Folder created/used: /path/to/project/output_summaries

```
=====
```