

# Quote Fetcher

Fetch a personalized set of quote IDs from a dynamic page, retrieve the corresponding quotes from a public API, save them to `quotes.json`, and generate shareable PNG images for each quote using a color scheme also derived from the dynamic page.

The script prompts for your student ID (A.M.) at runtime and uses it to fetch both the list of quote IDs and a color scheme.

---

## Table of Contents

- [Features](#)
  - [Architecture](#)
  - [Prerequisites](#)
  - [Installation](#)
  - [Usage](#)
  - [Outputs](#)
  - [Configuration & Customization](#)
  - [Error Handling & Troubleshooting](#)
  - [Privacy & Notes](#)
  - [Project Structure](#)
  - [API Endpoints Used](#)
  - [FAQ](#)
  - [License](#)
  - [Ελληνικά — Σύντομες οδηγίες](#)
- 

## Features

- **ID extraction:** Scrapes numbers from a dynamic page tied to your student ID and selects up to the first 50 unique IDs.
- **Quote retrieval:** Calls a public API ( `dummyjson.com` ) to fetch quote text and author per ID.
- **JSON export:** Stores all fetched quotes in `quotes.json` with UTF-8 encoding.
- **Image generation:** Produces PNG images ( `1200×200` ) per quote with background/text colors read from the dynamic page.
- **Graceful failure:** Network issues and missing IDs are handled without crashing the whole run.

## Architecture

The script consists of the following functions:

- `fetch_quote_ids(student_id)`: Requests a Netlify function URL derived from `student_id`, parses the HTML, and extracts numeric tokens from common tags (`p`, `span`, `div`, `li`, `td`). Returns up to 50 unique sorted IDs.
- `fetch_quotes(quote_ids)`: For each ID, fetches quote data from `https://dummyjson.com/quotes/{id}` and collects `{id, quote, author}` objects.
- `save_quotes_to_json(quotes, filename="quotes.json")`: Serializes quotes to a JSON file using `ensure_ascii=False`.
- `fetch_color_scheme(student_id)`: Re-requests the dynamic page and reads `<div id="colors" data-bg="#..." data-color="#...">` attributes; defaults to black on white if absent.
- `generate_quote_images(quotes, bg_color, text_color)`: Calls `https://dummyjson.com/image/1200x200/{bg}/{fg}?text=...&fontSize=18` per quote and writes PNGs to `quotes/`.
- `main()`: Prompts for A.M., orchestrates the steps, and prints progress in Greek.

## Prerequisites

- **Python**: 3.8+ (tested on 3.10/3.11)
- **Pip**: for installing dependencies

## Python packages

```
requests
beautifulsoup4
```

Optionally install `lxml` for faster/stricter HTML parsing, though the built-in `html.parser` works.

## Installation

```
# 1) Create and activate a virtual environment (recommended)
python -m venv .venv
# Windows
.venv\Scripts\activate
# macOS/Linux
source .venv/bin/activate

# 2) Install dependencies
pip install -U pip
pip install requests beautifulsoup4
```

## Usage

Assuming the script is saved as `main.py` in your project root:

```
python main.py
```

You will be prompted:

Εισάγετε τον Α.Μ. σας:

Enter your student ID (A.M.) and press **Enter**.

### Example session (abridged)

Ανακτήθηκαν τα ακόλουθα Quote IDs: [3, 7, 12, ...]  
Τα αποφθέγματα αποθηκεύτηκαν επιτυχώς στο quotes.json  
Εικόνα 3.png δημιουργήθηκε.  
Εικόνα 7.png δημιουργήθηκε.  
...  
Η διαδικασία ολοκληρώθηκε! Οι εικόνες αποθηκεύτηκαν στο φάκελο quotes/.

## Outputs

- `quotes.json`: Array of quote objects:

```
[
  {
    "id": 3,
    "quote": "Your time is limited...",
    "author": "Steve Jobs"
  },
  { "id": 7, "quote": "...", "author": "..." }
]
```

- `quotes/`: Directory containing one PNG per quote, named `{id}.png` (e.g., `quotes/3.png`).

## Configuration & Customization

- **Limit of IDs:** By default, the first **50** unique numeric tokens are used. Change the slice in `fetch_quote_ids` to adjust:

```
return sorted(numbers)[:50]
```

- **Image size & font:** The image service URL includes size and `fontSize`. Adjust in `generate_quote_images`:

```
f"https://dummyjson.com/image/1200x200/{bg}/{fg}?text=...&fontSize=18"
```

- **Output directory:** Images are written to `quotes/`. Change the path in `generate_quote_images`.
- **HTML tags scanned:** Modify the list in `soup.find_all(["p", "span", "div", "li", "td"])` to broaden or restrict extraction.
- **Default colors:** Fallbacks are black background (`000000`) and white text (`FFFFFF`). Update in `fetch_color_scheme`.

## Error Handling & Troubleshooting

- **Network errors:** Requests are wrapped with `response.raise_for_status()` and `try/except`. Failures for individual quote IDs are logged and skipped.
- **No IDs found:** You'll see `Δεν βρέθηκαν έγκυρα Quote IDs`. Confirm your A.M. and that the Netlify function is reachable.
- **Missing color scheme:** If the `<div id="colors">` is absent, defaults are used.
- **API limits/404s:** If `dummyjson.com` lacks a quote for a given ID, that entry is skipped; the rest proceed.

## Verbose Debugging (optional)

You can add temporary `print()` statements or logging to inspect responses, e.g., `print(response.text[:500])` after fetching the dynamic page.

## Privacy & Notes

- Your student ID (A.M.) is sent as a query parameter to the Netlify endpoint to generate a personalized page. No additional personal data is collected by this script.
- Generated files are stored locally in your project folder.

## Project Structure

```
.
├─ main.py           # The script (name may vary)
├─ quotes.json       # Generated after a successful run
└─ quotes/           # PNG images (auto-created)
```

## API Endpoints Used

- **Dynamic page (IDs & colors):** `https://tma111.netlify.app/.netlify/functions/generate?id={student_id}`
- **Quote by ID:** `https://dummyjson.com/quotes/{id}`
- **Image generator:** `https://dummyjson.com/image/1200x200/{bg}/{fg}?text={encoded_quote}&fontSize=18`

## FAQ

### Q: Do I need a browser or JavaScript engine?

A: No. The dynamic page is fetched server-side; we parse the returned HTML for numbers and color metadata.

### Q: Can I run this non-interactively?

A: Yes—wrap `main()` to accept a CLI flag (e.g., `--id`) and pass it into the functions instead of `input()`.

### Q: Why are some image files missing?

A: If a quote fetch fails for that ID, no image is generated.

## License

This project is licensed under the **MIT License**. See `LICENSE` (add one if needed).

---

## Ελληνικά — Σύντομες οδηγίες

1. Εγκατάσταση βιβλιοθηκών:

```
pip install requests beautifulsoup4
```

2. Εκτέλεση:

```
python main.py
```

3. Όταν ζητηθεί, πληκτρολογήστε τον **A.M.** σας.

4. Τα αποτελέσματα αποθηκεύονται σε `quotes.json` και στον φάκελο `quotes/` (εικόνες PNG).

**Παραμετροποίηση:** - Αλλάξτε το όριο των IDs (`[:50]`), το μέγεθος εικόνας (`1200x200`), το `fontSize=18`, ή τα προεπιλεγμένα χρώματα (`000000`, `FFFFFF`) μέσα στις αντίστοιχες συναρτήσεις.