

URL Resolver CLI Setup

Requirements

Create a `requirements.txt` file with these dependencies:

```
flask==2.3.3
requests==2.31.0
beautifulsoup4==4.12.2
lxml==4.9.3
pyngrok==7.0.0
```

Installation

1. Install Python dependencies:

```
bash
pip install -r requirements.txt
```

2. Install ngrok (for public exposure):

- Go to <https://ngrok.com/download>
- Download and install ngrok for your OS
- Sign up for a free account and get your auth token

Usage Examples

1. CLI Mode - Single URL

```
bash
python url_resolver.py --mode cli --url "https://news.google.com/articles/example"
```

2. CLI Mode - Batch Processing

Create a JSON file with URLs:

```
json
```

```
{  
  "urls": [  
    "https://news.google.com/articles/example1",  
    "https://example.com/redirect",  
    "https://short.ly/abc123"  
  ]  
}
```

Then run:

```
bash  
  
python url_resolver.py --mode cli --batch urls.json --output results.json
```

3. Server Mode - Local Only

```
bash  
  
python url_resolver.py --mode server --port 5000
```

Then test with:

```
bash  
  
curl "http://localhost:5000/resolve?url=https://news.google.com/articles/example"
```

4. Server Mode - With ngrok (Public Access)

```
bash  
  
python url_resolver.py --mode server --port 5000 --ngrok --ngrok-token YOUR_NGROK_TOKEN
```

This will give you a public URL like: `https://abc123.ngrok.io`

5. Batch API Request

```
bash  
  
curl -X POST http://localhost:5000/resolve-batch \  
-H "Content-Type: application/json" \  
-d '{"urls": ["https://example.com", "https://google.com"]}'
```

API Endpoints

When running in server mode, you get these endpoints:

- **GET /resolve?url= <url>** - Resolve a single URL
- **POST /resolve-batch** - Resolve multiple URLs (JSON body with `urls` array)
- **GET /health** - Health check endpoint

Command Line Options

- `--mode {cli,server}` - Run in CLI or server mode (default: server)
- `--url URL` - Single URL to resolve (CLI mode only)
- `--batch FILE` - JSON file with URLs to resolve (CLI mode only)
- `--port PORT` - Port for local server (default: 5000)
- `--ngrok` - Expose server via ngrok tunnel
- `--ngrok-token TOKEN` - Your ngrok authentication token
- `--output FILE` - Output file for results (CLI mode only)

Features

✓ **Google News URL Decoding** - Handles complex Google News URLs using multiple methods ✓

Standard URL Resolution - Follows redirects and finds canonical URLs ✓ **Batch Processing** - Handle multiple URLs at once ✓ **CLI & Web API** - Use as command-line tool or web service ✓ **ngrok**

Integration - Easily expose to public internet ✓ **Multiple Detection Methods** - Canonical tags, Open Graph, JSON-LD, etc. ✓ **Error Handling** - Robust error handling and timeouts

Example Response

```
json
{
  "original": "https://news.google.com/articles/example",
  "final_url": "https://example.com/actual-article",
  "resolved_article_url": "https://example.com/actual-article",
  "source": "google_news_batchexecute",
  "status_code": 200
}
```

Notes

- The application limits batch processing to 10 URLs to prevent abuse
- Google News URL resolution uses multiple fallback methods
- All requests include realistic User-Agent headers
- Timeouts are set to prevent hanging requests
- The server runs on all interfaces (0.0.0.0) for ngrok compatibility