Web Scraper & API - Documentation

Overview

This project consists of a Node.js web scraper using Puppeteer and a Python Flask server to host the scraped data. The application runs as a Docker container using a multi-stage build to keep the image lean.

How to Build the Docker Image

Run the following command in the project directory:

docker build -t web-scraper .

How to Run the Container

Run the container and specify the URL to scrape:

docker run -p 5000:5000 -e SCRAPE_URL=https://example.com web-scraper

How to Access the Hosted Scraped Data

Once the container is running, visit:

http://localhost:5000

This will return the extracted page title and first heading in JSON format.

scrape.js

```
const puppeteer = require('puppeteer');
const fs = require('fs');

(async () => {
    const url = process.env.SCRAPE_URL || 'https://example.com';
    const browser = await puppeteer.launch({
        headless: 'new',
        args: ['--no-sandbox', '--disable-setuid-sandbox']
    });
    const page = await browser.newPage();
    await page.goto(url, { waitUntil: 'domcontentloaded' });

    const data = await page.evaluate(() => {
        return {
            title: document.title,
            heading: document.querySelector('h1')?.innerText || 'No heading found'
```

```
        };
    });

    fs.writeFileSync('/app/scraped_data.json', JSON.stringify(data, null, 2));
    console.log('Scraping complete:', data);

    await browser.close();
})();
```

server.py

```
from flask import Flask, jsonify
import json

app = Flask(__name__)

@app.route('/')
def serve_data():
    try:
        with open('/app/scraped_data.json', 'r') as file:
            data = json.load(file)
        return jsonify(data)
    except Exception as e:
        return jsonify({"error": str(e)})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Dockerfile

```
# Stage 1: Scraper
FROM node:18-slim AS scraper
WORKDIR /app
COPY package.json package-lock.json ./
RUN npm install
COPY scrape.js ./
RUN apt-get update && apt-get install -y chromium
ENV PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=true
ENV SCRAPE_URL=https://example.com
CMD ["node", "scrape.js"]

# Stage 2: Server
FROM python:3.10-slim AS server
WORKDIR /app
COPY --from=scraper /app/scraped_data.json ./
COPY server.py ./
```

```
RUN pip install flask
EXPOSE 5000
CMD ["python", "server.py"]
```

package.json

```
{
  "name": "scraper",
  "version": "1.0.0",
  "dependencies": {
    "puppeteer": "latest"
  }
}
```

requirements.txt

```
flask
```