Instagram APK Scraper: Technical Documentation

Overview

Instagram APK Scraper is a web application designed to scrape and analyze Instagram APK versions from APKMirror. The application provides an interface for scraping data, viewing APK details, and assessing compatibility of the APK versions with different Android devices based on user agent strings.

Features

1. Scraping Instagram APKs

- **Functionality**: Scrapes APK versions of Instagram from APKMirror.
- **Implementation**: Utilizes Axios and Cheerio for HTTP requests and DOM parsing.
- **Endpoint**: `/start-scrape` (POST)

2. Displaying Scraped Data

- **Functionality**: Displays a list of scraped APK versions with release dates.
- **UI**: Table format in the frontend.
- **Endpoint**: `/api/versions` (GET)

3. Detailed APK Information

- **Functionality**: Shows detailed information about each APK version, including variants.
- **UI**: Implemented as a modal in the frontend.
- **Endpoint**: `/api/versions/:id` (GET)

4. Deleting APK Versions

- **Functionality**: Allows deletion of APK versions from the database.
- **UI**: 'Delete' button next to each APK version.
- **Endpoint**: `/api/versions/:id` (DELETE)

5. Compatibility Check

- **Functionality**: Checks if an APK version is compatible with a user's device based on agent string.
- **UI**: Form input for agent string and alert messages for results.
- **Endpoint**: `/check-compatibility` (POST)
- **Agent Parsing**: Extracts version ID, variant ID, Android version, and DPI from the agent string.

6. Dockerization

- **Functionality**: Application is containerized using Docker for easy deployment.
- **Dockerfile**: Includes instructions for building the Docker image.
- **Usage**: Image can be built and run using standard Docker commands.

Technologies Used

- **Backend**: Node.js, Express
- **Frontend**: HTML, Bootstrap, JavaScript
- **Database**: MongoDB
- **Web Scraping**: Axios, Cheerio
- **Containerization**: Docker

Setup and Installation

1. **Prerequisites**: Node.js, MongoDB, Docker (optional).
2. **Running the Application**:
    - Without Docker: `npm install` and `node app.js`.
    - With Docker: Build the Docker image and run the container as per the Docker documentation.

Endpoints

1. **Start Scraping**:
    - `POST /start-scrape`
    - Initiates the scraping process.
2. **Get Versions**:
    - `GET /api/versions`
    - Retrieves all scraped APK versions.
3. **Get Specific Version**:
    - `GET /api/versions/:id`
    - Retrieves details of a specific APK version.

4. **Delete Version**:
    - `DELETE /api/versions/:id`
    - Deletes a specific APK version.
5. **Check Compatibility**:
    - `POST /check-compatibility`
    - Checks the compatibility of a version with a user's device.

Error Handling

- The application includes basic error handling for database operations and scraping processes.
- Responses include appropriate HTTP status codes and descriptive messages.

Future Enhancements

- **Automatic Scraping**: Implement a scheduler for automatic scraping at regular intervals.
- **Enhanced UI/UX**: Improve frontend design for a more intuitive user experience.
- **Advanced Compatibility Checking**: Incorporate more parameters for a thorough compatibility check.

Conclusion

Instagram APK Scraper serves as a comprehensive tool for analyzing and managing Instagram APK versions, with a focus on compatibility assessment. The application's Dockerization simplifies deployment, making it suitable for various environments.