Documentation for NPM Package Analyzer

Overview

The NPM Package Analyzer is a Python-based tool designed to compare README files of different versions of NPM packages. It aims to identify changes, especially breaking changes, between versions by leveraging AI-based comparison methods

Prerequisites

Before running the program, ensure you have Python installed on your system. The program has been developed and tested with Python 3.8. Additionally, you will need to install the following dependencies:

1. requests
2. openai
3. google-generativeai

These dependencies can be installed using the following command:

pip install -r requirements.txt

The `requirements.txt` file is in the root directory of the project.

Running the Program

1. Open a terminal or command prompt.

2. Navigate to the directory containing the program files.

3. Run the program using the following command:

python CLI\_Client.py

Follow the on-screen prompts to enter the package name, the number of versions to compare, and the comparison method.

Setting API Keys

The program requires API keys for certain comparison methods (e.g., Google Generative AI). If an API key is needed, the program will prompt you to enter it. Ensure that the API key is set in your environment variables or be prepared to input it when prompted.

Program Explanation and Design Decisions

The program is divided into two main files:

1. CLI\_Client.py
2. and npm\_package\_analyzer.py

The decision to separate concerns into two files enhances maintainability and allows for easier extension of the program's capabilities in the future.

**CLI\_Client.py**

Handles user input and output, providing an interactive command-line interface for the user to specify the package name, the number of versions to compare, and the comparison method.

**npm\_package\_analyzer.py**

Contains the core functionality for fetching README files from NPM or GitHub, comparing README files using AI-based methods, and identifying changes between versions.

The program uses the NPM registry (<https://registry.npmjs.org/>) in order to get information about the package. Information like: package versions, GitHub repo, NPM package url README file name or number of files in package.

The program attempts to download the README file from GitHub before NPM because downloading only the README is more efficient than downloading the entire package from NPM, extracting the README and then deleting the package.

I added the option to use Google Generative AI because my free tokens in openai were expired and openai did not let me renew them even for a new user

Because I didn't want to buy tokens with money I was looking for another good model that I could use for free.

Assumptions

There is an assumption that the environment where the script runs has the necessary permissions to download, extract, and manage files and directories.

If npm\_package\_analyzer.py is used directly there should be a correct API key set in your environment variables. In here there is an assumption that the user knows how to set environment variables in their operating system.

If your user is (for example) out of tokens and you still uses its API key, there will be error msg printed.

Other than that, The program doesn’t really assume anything, but an error msg will be printed if information is missing like:   
the NPM package doesn’t have its information in the NPM registry.   
Or in the registry, there isn’t the GitHub repo URL under repository and URL, or there isn’t a link to the NPM package under dist and tarbal.

Limitations

It relies on external AI services (OpenAI, Google Generative AI) for comparison, which incur costs and have usage limits.

The accuracy of the comparison and the identification of breaking changes depend on the quality and detail of the README files and the AI model.