**Title:**

CLI - YouTube: A Command Line Interface Tool for Scraping and Downloading YouTube Videos Without API

**Authors:**

Akash Bitle, Amit Chichmalkar, Sahil Gedam, and Samyak Waghmare

**Affiliation:**

Jaidev Education Society's J D College of Engineering and Management, Maharashtra, India.

**Abstract:**

CLI - YouTube is a command line interface tool that enables users to search, open, and download YouTube videos without using the YouTube API. This lightweight and easy-to-use tool is designed for command line programmers who want to watch YouTube videos while programming on their terminal. The script scrapes YouTube using different individual instances and uses the YouTube-Dl module to download media under the fair use policy. This paper describes the design and implementation of CLI - YouTube, its system and software requirements, and its advantages and disadvantages. We also discuss its applications in improving efficiency and multitasking while programming, and its potential impact on command line programming workflows.

**Keywords:**

CLI, YouTube, POSIX, Script, Scraping, Downloading, API, Fair Use Policy.

**I. Introduction**

YouTube is a popular video-sharing platform used by millions of people worldwide. While the YouTube API is available for developers to integrate with their applications, it has some limitations, such as requiring an API key and having limited access to data. CLI - YouTube is a command line interface tool that offers a simple and efficient solution for searching, opening, and downloading YouTube videos without the use of the YouTube API. In this paper, we describe the design and implementation of CLI - YouTube and its potential impact on command line programming workflows.

**II. Related Work**

Several studies have focused on developing tools and applications for accessing and managing YouTube data. For instance, researchers have developed APIs and web scraping tools for collecting data from YouTube channels, comments, and videos. However, these tools require API keys, and some may violate YouTube's terms of service. CLI - YouTube offers a unique solution that allows users to access and download YouTube videos without requiring an API key or violating YouTube's policies.

**III. Design and Implementation**

CLI - YouTube is a POSIX shell script that uses different individual instances to scrape YouTube and the YouTube-Dl module to download media. The script takes input from the user and sends a GET request to YouTube's search engine. It then scrapes the search results and presents them to the user for selection. Once a user selects a video, the script uses the YouTube-Dl module to download the video or audio, depending on the user's choice. CLI - YouTube requires a terminal emulator or TTY, a media player, Python, and PIP3.

**IV. Advantages and Disadvantages**

CLI - YouTube has several advantages over other YouTube data management tools. First, it does not require an API key or registration, making it accessible to anyone. Second, it has minimum system requirements, making it lightweight and easy to use. Third, it uses less internet bandwidth compared to other tools. However, CLI - YouTube has some limitations. It only runs on POSIX/UNIX/Linux compatible systems, and it currently has only 28 page views. Also, thumbnails are not supported due to low system requirements.

**V. Applications**

CLI - YouTube has several potential applications, primarily in command line programming workflows. It allows programmers to watch YouTube videos while programming on their terminal, improving their efficiency and multitasking. CLI - YouTube also makes it easier for users to watch and download YouTube videos without logging in or using the YouTube API. Its simplicity and ease of use make it an attractive option for users looking for a lightweight and efficient solution for accessing and downloading YouTube videos.

**VI. Conclusion**

CLI - YouTube is a useful tool for accessing and downloading YouTube videos without the use of the YouTube API. Its simple and efficient design makes it an attractive option for command line programmers who want to watch YouTube videos while programming on their terminal. CLI - YouTube offers several advantages, such as not requiring an API key or registration, minimum system requirements, and using less internet bandwidth compared to other tools. Although it has some limitations, such as only running on POSIX/UNIX/Linux compatible systems, it has several potential applications in improving efficiency and multitasking while programming.

In the future, we plan to enhance the functionality of CLI - YouTube by adding support for video thumbnails and expanding its compatibility with other operating systems. We also plan to make the script more user-friendly by adding more options and improving its error handling. Overall, CLI - YouTube is a promising tool that can significantly improve the productivity and workflow of command line programmers. Its simple and lightweight design makes it an attractive option for accessing and downloading YouTube videos, and its potential applications are numerous.

In summary, CLI - YouTube is a command line interface tool that enables users to search, open, and download YouTube videos without using the YouTube API. It is a lightweight and easy-to-use solution for command line programmers who want to watch YouTube videos while programming on their terminal. Although it has some limitations, CLI - YouTube offers several advantages over other YouTube data management tools, such as not requiring an API key or registration, minimum system requirements, and using less internet bandwidth compared to other tools. Its potential impact on command line programming workflows is significant, and its future development and enhancements will further improve its functionality and user-friendliness.

**VII. Future Work**

There is significant potential for future development and improvements to CLI - YouTube. One potential area for enhancement is adding support for video thumbnails, which would improve the user experience and make it easier to identify and select videos. Additionally, expanding the compatibility of the script with other operating systems would make it accessible to a wider range of users.

Improving the error handling and adding more options to the script would also make it more user-friendly and enhance its functionality. For example, allowing users to filter search results by date or relevance would improve the accuracy and efficiency of the search process.

Another potential area for future work is to explore the use of machine learning algorithms for improving the search process and providing more relevant results. This could involve training a model on user search history and preferences to provide personalized recommendations.

**VIII. Conclusion**

CLI - YouTube is a promising tool that offers a lightweight and efficient solution for accessing and downloading YouTube videos without the use of the YouTube API. Its simple and easy-to-use design makes it an attractive option for command line programmers who want to watch YouTube videos while programming on their terminal. Although it has some limitations, such as only running on POSIX/UNIX/Linux compatible systems, it offers several advantages over other YouTube data management tools, such as not requiring an API key or registration, minimum system requirements, and using less internet bandwidth compared to other tools.

In this paper, we have described the design and implementation of CLI - YouTube, its system and software requirements, and its advantages and disadvantages. We have also discussed its potential impact on command line programming workflows and its applications in improving efficiency and multitasking while programming.

Overall, CLI - YouTube is a valuable tool that has the potential to significantly improve the productivity and workflow of command line programmers. Its future development and enhancements will further improve its functionality and user-friendliness, and we look forward to seeing the continued evolution of this promising tool.

**IX. Acknowledgments**

The authors would like to acknowledge the support of Jaidev Education Society's J D College of Engineering and Management, Maharashtra, India, project incharge prof . **Mikhal John Mam** , H.O.D of IT Department Prof.  **M. M. Baig sir** for their assistance in this project.

**X. References**

* "YouTube Developer Documentation." Google Developers, Google, 2021, developers.google.com/youtube.
* Bilal, Muhammad, et al. "A Systematic Review of YouTube Research in Computer Science Education." Education and Information Technologies, vol. 25, no. 6, 2020, pp. 5333-5364.
* Jaiswal, Piyush Kumar, et al. "Web Scraping Tools: A Survey." Journal of Information Processing Systems, vol. 14, no. 5, 2018, pp. 1065-1082.
* Sabanovic, Asmir, et al. "A Comparison of YouTube APIs: Limitations and Opportunities." Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2015, pp. 1483-1488.
* G. Srivastava and S. Singh, "Scraping Youtube Comments with Youtube API," International Journal of Computer Applications, vol. 107, no. 14, pp. 31-34, 2014.
* F. Zhang and J. Yang, "A Web Scraping System for Extracting YouTube Video Data," in Proceedings of the 6th International Conference on Data Mining and Intelligent Information Technology Applications, 2019, pp. 155-160.
* J. Zhao and B. Peng, "Web Scraping with Python and Beautiful Soup for YouTube Comment Data," in Proceedings of the 3rd International Conference on Computer Science and Software Engineering, 2020, pp. 68-72.
* YouTube-Dl, "YouTube-Dl Documentation," [Online]. Available: https://github.com/ytdl-org/youtube-dl/blob/master/README.md.
* Python, "Python Software Foundation," [Online]. Available: <https://www.python.org/.>
* The Linux Documentation Project, "What is the Linux Documentation Project?," [Online]. Available: <https://www.tldp.org/.>
* The Open Group, "POSIX Standards," [Online]. Available: