Javinator9889 – GitHub

YouTube MD Bot Downloader – SRS

A Python 3 bot (design for working with Telegram) for downloading YouTube videos (preferably music ones) in multimedia formats such as MP3, MP4, OGG and more

Javinator9889

June 16th, 2018

1. Introduction

1.1. Purpose

This SRS (*Software Requirement Specification*) aims to show the developer and the standard user how is this product going to be created and how it works, detailing its functionalities and restricting what is it able to achieve and what not.

Therefore, any developer that wants to create a Python application must read this guide and every user that has curiosity about the capabilities of the application itself.

1.2. Scope

This SRS describes the process of creation, innovation and development of *YouTube MD Bot Downloader* (henceforth, YTMB), which is a *YouTube Bot* created for *Downloading Media* in *Multiple Formats* such as MP3, MP4, OGG and more.

As its name means, this app will allow Telegram users to download almost every YouTube video in a multimedia format, aiming principally music ones. When executing, you will be able to search for a song just by typing its name and for a better results adding the artist. In addition, you can share a YouTube link with the bot and it will start downloading it automatically.

Multiple user options and preferences will be able for customization such as the possibility for choosing the song quality, media format and the capacity for creating custom metadata for a video in which that information were not found. In addition, a user history is saved so he can see what he has downloaded and share it as fast as possible with his contacts.

In contrast to the previous, this app will only download media in any reproducible format such as MP3, MP4, OGG, etc. but will not download just videos and send them to the user, as it is not its purpose.

As there are many applications of the same style as the one we are describing here, this aims to be very user friendly and developer friendly, supporting the OpenSource[1] project by making its code free for use, download and distribution, helping other developers by applying CleanCode principles[2]. In addition, the user must find this application easy to use and understand all its capabilities, which will improve the quantity of users that will use this bot.

Finally, here will be also described some system specifications needed in order to a better and correct working at the *Software system attributes* section.

1: see the appendix at section 1 for more information.

2: see the appendix at section 2 for more information.

1.3. Definitions, acronyms and abbreviations

- Telegram: "*Telegram is a cloud-based instant messaging and voice over IP service developed by Telegram Messenger LLP, a privately held company registered in London, United Kingdom, founded by the Russian entrepreneur Pavel Durov.*" [1]
- Python: "*Python is a programming language that lets you work more quickly and integrate your systems more effectively.*" [2]
- Telegram bot: Telegram bots "*are simply Telegram accounts operated by software – not people – and they'll often have AI features. They can do anything – teach, play, search, broadcast, remind, connect, integrate with other services, or even pass commands to the Internet of Things.*" [3]
- API: "*an Application Programming Interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.*" [4]
- YouTube: "*YouTube is an American video-sharing website headquartered in San Bruno, California. […] YouTube allows users to upload, view, rate, share, add to favorites, report, comment on videos, and subscribe to other users. It*

offers a wide variety of user-generated and corporate media videos. Available content includes video clips, TV show clips, music videos, short and documentary films, audio recordings, movie trailers, live streams, and other content such as video blogging, short original videos, and educational videos. Most of the content on YouTube is uploaded by individuals, but media corporations including CBS, the BBC, Vevo, and Hulu offer some of their material via YouTube as part of the YouTube partnership program." [5]

- YouTube-dl: "*youtube-dl is a command-line program to download videos from YouTube.com and a few more sites. It requires the Python interpreter (2.6, 2.7, or 3.2+), and it is not platform specific."* [6]
- FFmpeg: "*FFmpeg is a free software project, the product of which is a vast software suite of libraries and programs for handling video, audio, and other multimedia files and streams. At its core is the FFmpeg program itself, designed for command-line-based processing of video and audio files*" [7]
- Docker: "*Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud.*" [8]
- Python Telegram Bot: python-telegram-bot is an API developed by the community that provides an interface for creating bots that communicate with the Telegram API. [9]
- TG: *Telegram.*
- YTMB: *YouTube MD Bot Downloader.*
- Py: *Python*.
- YT: *YouTube.*
- API: *Application Programming Interface.*
- YT-DL: *YouTube-dl.*
- MM: *MultiMedia*.
- HDD: *Hard Drive Disk*.
- OS: *Operative System*.
- DB: *DataBase*.
- PTB: *Python-Telegram-Bot*.

1.4. References

[1] Community, "Wikipedia - Telegram (service)," Wikipedia Org, [Online]. Available: https://en.wikipedia.org/wiki/Telegram_(service). [Accessed June 2018].

[2] Community, "Python," Python Software Fundation, [Online]. Available: https://www.python.org/. [Accessed June 2018].

[3] Telegram, "Telegram Bot Platform," Telegram Org, [Online]. Available: https://telegram.org/blog/bot-revolution. [Accessed June 2018].

[4] Community, "Wikipedia - API," Wikipedia Org, [Online]. Available: https://en.wikipedia.org/wiki/Application_programming_interface. [Accessed June 2018].

[5] Community, "Wikipedia - YouTube," Wikipedia Org, [Online]. Available: https://en.wikipedia.org/wiki/YouTube. [Accessed June 2018].

[6] R. G. González, "youtube-dl," GitHub, [Online]. Available: https://rg3.github.io/youtube-dl/. [Accessed June 2018].

[7] Community, "Wikipedia - FFmpeg," Wikipedia Org, [Online]. Available: https://en.wikipedia.org/wiki/FFmpeg. [Accessed June 2018].

[8] Docker, "What is Docker?," Docker, [Online]. Available: https://www.docker.com/what-docker. [Accessed June 2018].

[9] Community, "python-telegram-bot," python-telegram-bot, [Online]. Available: https://python-telegram-bot.org/. [Accessed June 2018].

[10] O. Initiative, "About the OpenSource Initiative," OpenSource, [Online]. Available: https://opensource.org/about. [Accessed June 2018].

[11] O. Initiative, "The Open Source Definition," OpenSource Initiative, [Online]. Available: https://opensource.org/docs/osd. [Accessed June 2018].

[12] R. C. M. Series, Clean Code, a Handbook of Agile Software Craftsmanship, Boston: Pearson Education, Inc., 2009.

1.5. Global vision

The following content will describe you at first the general factors that directly affect the product, its general requirements and some explanations in order to a better understanding of the specific requirements.

2. General description

2.1. Product perspective

This product is an evolution of its previous version *@dwnmp3bot* but with a redefined algorithm and logic, for a better performance and less errors. There are some similar bots, being the most significantly similar *@YTAudioBot*.

As it is a Py application, this bot depends on the system that is executing it. Independently if it is Windows, Linux or MacOS, they need to have Py version 3 installed with the required dependencies. Those dependencies are automatically included if not present when installing the bot for its execution. In addition, in a future, a docker version will be available for downloading and installing, speeding-up all this process.

The product interface with the system is simple: a Py container/installation with a minimum hardware available (2 GB of RAM, at least dual-core processor higher than 1 GHz), and the user interface is provided by the TG application itself, so there is no need of developing a custom view for the user.

The required APIs and packages are provided at the GitHub's project web page with all the important information available on each ones sites (as it can change, here is not specified because today can be version 1.1.5. and in three days it is updated to 1.5.3., so it is better to include each package site at the *README.md* file of the project).

In addition, as it is properly a web service listening to an URL, YTMB needs completely Internet access for listening to petitions, but it does not need access to any specific port at the running machine.

Finally, there is no song stored on the local machine because YTMB uses TG servers, which saves every file sent. But, as specified previously, at least 2 GB of RAM is required and enough space at the primary HDD in order to store and save the database data (with 10 GB or less may be enough for this purpose, but recommended having at least 20 GB).

2.2. Product functions

Primarily, this product will have the capability for:

a. Downloading YT videos at the highest quality available, in order to get the best results while converting to MM format.
b. Converting downloaded videos to user specified MM format in a list of available ones.
c. Applying user chosen quality options, allowing the user to choose the bitrate of the MM file.
d. Obtain song metadata if available. Else, requesting it to the user so he can customize MM file.

2.3. User characteristics

YTMB is designed for every user: there is not previously required experience or acknowledgment, just having a smartphone with TG app installed on it. Also, a little tutorial is added in order to help people that may have some troubles using the application.

2.4. Restrictions

As this application is designed to work on each system, there may be some restrictions that will make the user experience worse. For example, if your system does not have enough memory or space in your HDD, very probably YTMB will crash or stop working.

Also, if the HDD speed is so low, there will be some timeout problems reading and sending files and maybe some users will not get the requested file.

2.5. Supposals and dependencies

Continuing with what said at *Product perspective*, the OS is independent as it can run a Py application or a docker container. In addition, the required dependencies are installed when running setup of YTMB else, they are specified at the GitHub project page.

2.6. Proposed requirements

There is only one proposed requirement that must be studied if it is possible to be included in future versions of the software: creating an inline mode for the bot with all user history for sending songs fastest from there.

3. Specific requirements

3.1. External interface requirements

At this section, all the points described at *2.1* are detailed by specifying for which inputs/outputs this product is working, with which values and the expected result.

For that reason is why the following interfaces are described, in order to generate a context where the application should work.

3.1.1. User interface

As said at *Product perspective* point, the user interface is provided by the TG application, so there is no need to create a GUI. When possible, friendly texts and *emoji* will be used for making the text easier to read.

In addition, TG buttons will be used and the bot will try to send less messages as possible, by using the editing method for sent messages.

3.1.2. Hardware interface

As this application is running under Py version 3, there is no specific hardware interface in order to achieve a correct working and performance: it is only necessary to have Py version 3 (or higher) installed.

If running under *dockerized* system, it is probably limited to Linux-based systems, as Docker runs at most on Linux-based ones. However, if a Windows version found and working, it can perfectly work on it.

3.1.3. Software interface

This bot will work with some different APIs such as:

- TG API, for communicating with the servers in order to receive and send messages interacting with the user.
- YT API, for looking for videos/songs with the given keywords (such as title and/or artist)
- Last.FM API, for searching for song metadata with the given title, artist, etc.

At this first approach, no other API is required but maybe, in a future, another one is included for adding new features that need that API.

3.1.4. Communications interface

In order to keep user privacy safe, this application must work with a cypher protocol for communication, such as HTTPS or similar one that grants full security and privacy for the user.

3.2. Functional requirements

At this section, all the requirements the application must satisfy are specified and detailed, by adding: description, inputs, processing, outputs and, if necessary, errors.

3.2.1. User preferences

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
The user must be able to save its data and preferences, in order to not to setup the bot every time he wants to use it.	The user can modify/setup his language, audio quality (320Kbps, 256 Kbps, 128 Kbps), audio format (MP3, MP4, OGG, AAC) and whether or not is asked for metadata if not found.	The new data will be stored at the user entry in the database, whenever he changes it, and immediately available for using it when requesting songs to the bot.	A pop-up will appear when the changes are saved so the user can know that he has changed his configuration.	No errors are expected here, as all the interaction will be done by using custom keyboards with a predefined output, so the user cannot choose a

				non-valid option.

3.2.2. User history

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
When a user downloads a video, its data and information is stored in a DB only available for the user if requested. In addition, there is a possibility where the user can delete all his history.	By using commands, the user can request all his history, navigating through it by using a custom keyboard. A button for deleting the history is also included.	Accessing to the DB, all history will be requested and saved in an object for manipulating it. If delete option is chosen, all entries in DB for this user are removed.	The user will see a message with each history and the possibility to request again the song and see more information about it. Only three entries are displayed for each page. A delete button is included.	As said before, no errors are expected, as it is a custom keyboard with a predefined behavior.

3.2.3. Video searching

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
YTMB must be capable for searching every YT video at the highest quality available.	An URL or video ID is sent to the bot. Also, a title followed by an artist can be provided.	The video is looked for at YT and obtained detailed information about it, whether if exists or not and its title, description, author and rating.	A message for the found result is shown to the user with a confirm button for starting download.	Some errors may occur when looking for a video and downloading it: • Video is longer than an hour and a half, so the bot will discard it for downloading. • Video is private, so the bot will discard it for downloading.

3.2.4. Video downloading

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
With a given valid URL, YTMB must be able to download the requested video at the highest quality available.	A valid YT URL.	Using an external lib (YT-DL) the bot will try to receive the YT video.	A message showing the progress of the download.	Some errors may occur: • The video duration is higher than an hour and a half, so the bot rejects the download. • The copyright laws protect the video, so it is not possible to download it. A message is shown to the user.

3.2.5. Video converting

Description	***Inputs***	***Processing***	***Outputs***	***Errors***

With a given video, YTMB must be able to convert it to the user specified format and with the user specified quality.	A downloaded video with a MM format (webm, MP4) which will be converted.	Using FFmpeg, the video will be saved at the specified format with the metadata.	An MM file with the user specified format and quality, ready to be sent to the user.	No errors are expected at this point, as the FFmpeg lib will use the required HDD space for its operation by checking it before.

3.2.6. Music sending

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
Once the video is converted, the bot must send it to the user.	A MM file.	By using native bot functions, the file will be sent.	A message that contains the audio file.	If HDD is being used, maybe a timeout error may occur. Also, if the file is bigger than 50 MB, the bot is not able to send it via TG so it will use an external service for sending it.

3.2.7. Metadata searching

Description	***Inputs***	***Processing***	***Outputs***	***Errors***

With a given title, YTMB must search for metadata with an external API (Last.FM).	Unicode title for searching.	The bot will do an API request to the external site for searching the metadata.	A title, artist, album and cover of the song.	• The song is not found, so no metadata is recovered. If in user settings the request metadata to the user is specified, the bot will ask the user for custom metadata. Else, default options will be used.

3.2.8. Developer contacting

Description	***Inputs***	***Processing***	***Outputs***	***Errors***

At the help section, the bot will have an option for contacting the developer for communication about errors or whatever the user wants.	Help command.	Receive and send messages.	A message with a direction for contacting the developer.	No errors expected.

3.2.9. Inline mode

Description	***Inputs***	***Processing***	***Outputs***	***Errors***
[THIS IS AN OPTIONAL FUTURE REQUIREMENT] The user can use the bot inline for looking for songs at his history.	Name of the bot in every chat.	List songs available and send them as a Py list.	A list with all the history.	• Empty history. • List too big.

3.3. Performance requirements

The YTMB must support at least 100 simultaneous users. This can be achieved my making DB access as optimum as possible

3.4. Design restrictions

As YTMB uses PTB API, there are some limitations imposed by TG for bots (in order to avoid problems such as spam or similar). For this reason, the bot cannot send files bigger than 50 MB or cannot start a conversation with a user if that user has not spoken the bot yet.

3.5. Software system attributes

YTMB must be:

- ✓ Secure: keeping user data protected and the user concerned about what is stored about.
- ✓ Private: a user can only access its own downloads, but not other users ones.
- ✓ Available: this application must be designed for working 24/7 every day.
- ✓ Maintainable & evolutionary: this application must be accessible for maintenance and being able to admit new features and characteristics.
- ✓ Scalability: the application must be designed for admitting many users without compromising the performance.

3.6. Other attributes

As mentioned before, the DB must be optimized when possible for granting the fastest possible access to the data. In addition, the application must apply al software engineering attributes and design techniques for developing a more optimized application.

Appendix

1. OpenSource Project

"The Open Source Initiative (OSI) is a non-profit corporation with global scope formed to educate about and advocate for the benefits of open source and to build bridges among different constituencies in the open source community.

Open source enables a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is higher quality, better reliability, greater flexibility, lower cost, and an end to predatory vendor lock-in.

One of our most important activities is as a standards body, maintaining the Open Source Definition for the good of the community. The Open Source Initiative Approved License trademark and program creates a nexus of trust around which developers, users, corporations and governments can organize open source cooperation." [10]

List of the Open Source Definition [11]:

1. Free distribution: "*The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.*"
2. Source code: "*The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.*"
3. Derived works: "*The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.*"
4. Integrity of the author's source code: "*The license may restrict source-code from being distributed in modified form* only *if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of*

software built from modified source code. The license may require derived works to carry a different name or version number from the original software.”

5. No discrimination against persons or groups: “*The license must not discriminate against any person or group of persons.”*
6. No discrimination against fields or endeavor: “*The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.”*
7. Distribution of license: “*The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.”*
8. License must not be specific to a product: “*The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.”*
9. License must not restrict other software: “*The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.”*
10. License must be technology-neutral: “*No provision of the license may be predicated on any individual technology or style of interface.”*

2. *CleanCode Principles*

Applying when possible all CleanCode principles written by Robert C. Martin Series on his book *Clean Code, a handbook of agile software craftsmanship* [12].

Index

18/06/2018

X

Javinator9889
Head developer
Firmado por: Javinator9889