URL to MP3

Abstract

My project takes a soundcloud and or youtube link and performs automation to reach a site, download the video and output it on the user's desktop as a MP3 file in which they can later listen to while offline. The user will only have to input a valid link, wait for the program to run and then when prompted name the file, and identify their user directory so that the program can output the file.

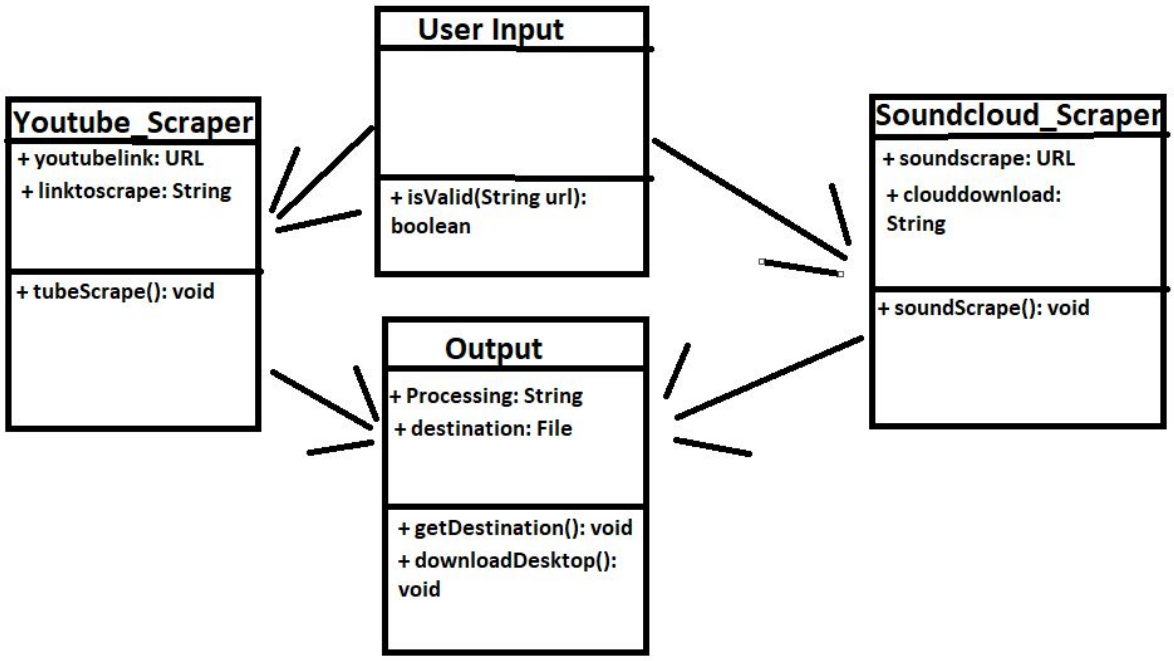
Introduction

This project was inspired by the many sites I use to access in order to download specific songs off youtube for offline listening purposes, I always wondered how it was possible to do something like this so I decided to try my best to recreate it using newly learnt java skills and prior coding knowledge. At first when I attempted to create this project I ran into a problem, the api in which I was pulling from had suddenly changed servers and could no longer be accessed forcing me to completely change my approach. Before I had a system that took input, encoded it and appended it to a URL then sent an http GET request to reach the site and while doing so use file input and output stream to save any data to a mp3 file, but due to the sudden server switch I had to instead use automation with Selenium web driver which seemed to have done the trick.

Detailed System Description

To get into more detail, let's break this system down by its class and explain how each one interacted with each other and on its own. The first class is "UserInput" this class contains no attributes as it is mainly a class to pass other values to another. It does however have many methods one of them being "isValid" which checks to see if the first user input is a valid link. Once this is done the link is sent through a series of if statements to check if the link is a youtube or soundcloud link, and if not it will exit out of the program instructing the user to run it again with a valid youtube or soundcloud url. Next it is sent to the respective class depending on the type of link, both classes do the same thing however they access different sites. Both the "Youtube_scrapper" and "Soundcloud_scrapper" class both have two attributes one to pass the user input link to the current class, and another to save the download url from the website reached to send to the next class. Both classes begin by creating a WebDriver class specifically a chromedriver which is imported from the Selenium dependency. This allows me to access any website on chrome so long as the user also has the latest version installed as well. This is of course headless meaning the user won't see the chrome window pop up and go through the download process. When it is reached both classes then input the original user link into the websites converter bar and do a series of clicks to retrieve the download link and save it to an attribute to be passed down to the last file. The output class is last and it contains two attributes, one to accept the download link from the soundcloud or youtube class in order to reach the site, and another to set the file's destination. It then also has two methods one called fileSelection that allows the user to identify the path so that the program can install on the user's desktop it also lets the user name the file. The last method named getDestionation creates an Http object in order to access the website the youtube and soundcloud classes returned if the

connection returns with the response code 200 meaning it was reached it then downloads the content into the file path set by the user in a while loop that basically states while there is information to be downloaded put it in the newly named and created user file. When it is complete it will then show a “Download complete” text in the program and the user will have their desired mp3 file on the desktop. Look below for a visual UMI representation.



Requirements

This program specifically solves the problem of downloading a song from youtube or soundcloud for offline purposes with ease. It is mainly addressing the problem of listening to music offline, as sometimes you want to listen to a artist who only produces music on youtube or soundcloud and have to be connected to the internet in order to listen to their songs, this becomes inconvenient when there's a power outage or your internet is down so this program fixes that by allowing the user to access and listen to the song at any time even with no internet connection.

Literature Survey

Other works have achieved the same result, mostly websites that allow you to download the said song so long as you have the link to pass into the converter. Others have also made whole programs to accomplish the same thing. They mainly use the API method I attempted earlier where they reach a specific site that has the audio content for the specific link to easily connect and download information reliably.

User Manual

The user manual for this program is short and sweet, it is designed so that the user only has to interact with the program 3 times to get the desired results. The user also needs to have the latest version of google chrome browser installed. 1st the user must have a valid youtube or soundcloud link and they must let the song load for at least 2 seconds so that the audio information can be generated. Afterward the user must input that link into the first input and then

wait for the next prompt. The user will see red text but don't worry this is not an error this is just the headless browser accessing the download from the site. Afterword the user will be prompted to enter a name for the file the user then must input a name and not add an extension, the name has no requirements it can be anything. Lastly the user must input their directories name for example mine would be Users\\Jelani\\, all the user has to do is input the name their computer is under in this case i would input “Jelani” and this ends the instructions for the user, everything should run normally if following these steps.

Conclusion

In conclusion this program will allow users to download any youtube or soundcloud link for offline listening purposes, it will be able to access and save any valid links to a new file on the user computer without any complications.

Bibliography

For my project I mainly used stack overflow it would be hard to show every single answer but here are the links i mainly pulled my ideas and methods from

https://bridgei2i.com/extracting-data-from-webpages-in-java-with-help-of-htmlunit/

https://devqa.io/webdriver-headless-mode-chrome-driver/

https://www.browserstack.com/guide/selenium-with-java-for-automated-test

These three links among mass google searching and a few scattered stack overflow answers lead to the creation of my project.