Module `copytube`

CopyTube's main module.

This module contains all the functionality of CopyTube allowing a user to copy a public playlist of up to 195 videos (due to API quota restrictions) to a private playlist on their own channel after configuring their own API key and client_secret file through YouTube's Data V3 API.

Global variables

`var scopes : List[str]`

> List[str]: Scope of requests to be made by CopyTube.
>
> The scope for requests was determined from this source, https://developers.google.com /youtube/v3/docs/playlistItems/insert, (https://developers.google.com/youtube/v3/docs/playlistItems /insert,) under the section, Authorization.

Functions

`def add_videos_to_playlist(client: Any, playlist_id: str, songs: List[str])`

> Inserts videos from list to new playlist.
>
> Loops through the list of song IDs obtained and uses the YouTube API client to insert the songs into the newly created playlist.
>
> ### Parameters
>
> `client : Any`
> > YouTube API client to make requests through
>
> `playlist : str`
> > String of the new playlist ID
>
> `songs : List[str]`
> > List of song IDs to be added to the new playlist

```
def create_new_playlist(client: Any, title: str) -> str
```

Creates a new playlist on the user's YouTube account.

Inserts a playlist into the user's YouTube account with the name following the format: -Copy

Parameters

```
client : Any
```
    Instance of YouTube API client to make requests through

```
title : str
```
    Title of the playlist to be copied

Returns

```
str
```
    String of the new playlist's ID


```
def get_credentials(secret: str) -> Any
```

Obtains credentials necessary to interact with YouTube's API

Parameters

```
secret : str
```
    String containing the file name of the client secret JSON file

Returns

```
Instance
```
    Credentials instance for use with the YouTube API client


```
def get_playlist_id(playlist_url: str) -> str
```

Retrieves ID of playlist from URL provided using a URL parser.

Parameters

```
playlist_url : str
```
    String of URL that points to a YouTube playlist

Returns

```
str
```

String of the playlist ID

```python
def get_song_ids(client: Any, playlist_id: str) -> Dict[str, Any]
```

Retrieves a list of the song IDs in a YouTube playlist

Parameters

```
client : Any
```
   YouTube API client to make requests through

```
playlist_id : str
```
   Playlist ID of the YouTube playlist to retrieve song IDs from

Returns

```
Dict[str: Any]
```
   Returns a dictionary containing the following information:

```
{
    'songs': List of song IDs,
    'title': Title of playlist to be copied
}
```

```python
def main()
```

```python
def printProgressBar(iteration, total, prefix='', suffix='', decimals=1, length=100,
                     fill='█', printEnd='\r')
```

Progress bar that prints progress of songs added to new playlist.

Source of progress bar can be found here: https://stackoverflow.com/questions/3173320/text-progress-bar-in-the-console (https://stackoverflow.com/questions/3173320/text-progress-bar-in-the-console)

Parameters

```
iteration : int
```
   current iteration

```
total : int
```
   total iterations

`prefix : str,` optional
   prefix string

`suffix : str,` optional
   suffix string

`decimals : int,` optional
   positive number of decimals in percent complete

`length : int,` optional
   character length of bar

`fill : str,` optional
   bar fill character

`printEnd : str,` optional
   end character (e.g. "\r", "\r\n")

```
def valid_input(url_to_check: str) -> bool
```

Checks for valid URL input from user.

Checks for a quit signal or a YouTube URL with a 'list' query parameter.

Parameters

`url_to_check : str`
   String containing the URL input by the user.

Returns

`bool`
   True, if the URL is YouTube URL with a 'list' parameter. False, otherwise. If the user inputs the quit signal 'q', the program exits.

Index

Global variables

scopes

Functions

add_videos_to_playlist
create_new_playlist
get_credentials
get_playlist_id
get_song_ids
main
printProgressBar
valid_input

Generated by *pdoc* 0.9.2 (https://pdoc3.github.io/pdoc).