Home Popular Modules

Python youtube_dl.YoutubeDL() Examples

The following are code examples for showing how to use *youtube_dl.YoutubeDL()*. They are extracted from open source Python projects. You can vote up the examples you like or vote down the ones you don't like. You can also save this page to your account.

+ Save to library

Example 1

Project: *MusicTools* Author: *kalbhor* File: *musictools.py* (MIT License) *View Source Project* 10 votes

```
def download_song(song_url, song_title):
    """
    Download a song using youtube url and song title
    """

    outtmpl = song_title + '.%(ext)s'
    ydl_opts = {
        'format': 'bestaudio/best',
        'outtmpl': outtmpl,
        'postprocessors': [
            {'key': 'FFmpegExtractAudio','preferredcodec': 'mp3',
             'preferredquality': '192',
            },
            {'key': 'FFmpegMetadata'},
        ],
    }

    with youtube_dl.YoutubeDL(ydl_opts) as ydl:
        info_dict = ydl.extract_info(song_url, download=True)
```

Example 2

Project: *apex-sigma-core* Author: *lu-ci* File: *music.py* (license) *View Source Project* 9 votes

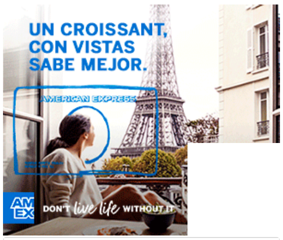
```
def __init__(self, requester, item_info):
        self.requester = requester
        self.item_info = item_info
        self.url = self.item_info.get('webpage_url')
        self.video_id = self.item_info.get('id') or self.url
        self.uploader = self.item_info.get('uploader') or 'Unknown'
        self.title = self.item_info['title']
        if 'thumbnail' in self.item_info:
            thumb = self.item_info.get('thumbnail')
            if thumb:
                self.thumbnail = thumb
            else:
                self.thumbnail = 'https://i.imgur.com/CGPNJDT.png'
        else:
            self.thumbnail = 'https://i.imgur.com/CGPNJDT.png'
        self.duration = int(self.item_info.get('duration') or 0)
        self.downloaded = False
        self.loop = asyncio.get_event_loop()
        self.threads = ThreadPoolExecutor()
        self.ytdl_params = ytdl_params
        self.ytdl = youtube_dl.YoutubeDL(self.ytdl_params)
        self.token = self.tokenize()
        self.location = None
        print(self.thumbnail)
```

Example 3

Project: *playlistfromsong* Author: *schollz* File: *playlistfromsong.py* (license) *View Source Project* 8 votes

```
def downloadURL(url, preferredCodec=None, preferredQuality=None):
    """ Downloads song using youtube_dl and the song's youtube
    url.
    """
    codec, quality = getCodecAndQuality()

    ydl_opts = {
        'format': 'bestaudio/best',
        'quiet': True,
        'no_warnings': True,
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': codec,
            'preferredquality': quality,
        },
```


UN CROISSANT, CON VISTAS SABE MEJOR.
AMERICAN EXPRESS
AM EX
DON'T live life WITHOUT IT

Related Functions

- sys.exit()
- sys.argv()
- re.compile()
- time.time()
- os.listdir()
- re.sub()
- time.sleep()
- re.search()
- os.remove()
- os.makedirs()
- logging.getLogger()
- random.randint()
- datetime.timedelta()
- json.loads()
- json.dumps()
- random.choice()
- json.load()
- functools.partial()
- requests.get()
- bs4.BeautifulSoup()

Related Modules

- os
- sys
- re
- time
- logging
- datetime
- random
- subprocess
- shutil
- traceback
- json
- collections
- functools
- requests
- datetime.datetime

Others in *youtube_dl*

- .DownloadError()
- .main()
- .YoutubeDL()

```
            {'key': 'FFmpegMetadata'},
        ],
    }

    try:
        with YoutubeDL(ydl_opts) as ydl:
            return ydl.extract_info(url, download=True)
    except:
        print("Problem downloading " + url)
    return None
```

Example 4

Project: *MusicNow* Author: *kalbhor* File: *command_line.py* (license) *View Source Project* 8 votes 👍 👎

```
def download_song(song_url, song_title):
    '''
    Downloads song from youtube-dl
    '''
    outtmpl = song_title + '.%(ext)s'
    ydl_opts = {
        'format': 'bestaudio/best',
        'outtmpl': outtmpl,
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192',
        },
            {'key': 'FFmpegMetadata'},
        ],

    }

    with youtube_dl.YoutubeDL(ydl_opts) as ydl:
        info_dict = ydl.extract_info(song_url, download=True)
```

Example 5

Project: *lynda-video-downloader* Author: *22nds* File: *download.py* (license) *View Source Project* 7 votes 👍 👎

```
def get_videos(url):
    """Download all videos in the course.
    """

    # Lynda.com login and video filename options
    options = {
        'username': USERNAME,
        'password': PASSWORD,
        'outtmpl': u'%(playlist_index)s-%(title)s.%(ext)s',
        'writesubtitles': SUBTITLES,
        'allsubtitles': SUBTITLES,
        'download_archive': ARCHIVE,
        'external_downloader': EXTERNAL_DL
    }

    try:
        with youtube_dl.YoutubeDL(options) as ydl:
            ydl.download([url])
    except:
        print('Could not download the video in course: {}'.format(url))
```

Example 6

Project: *tubeup* Author: *bibanon* File: *test_tubeup.py* (license) *View Source Project* 7 votes 👍 👎

```
def test_create_basenames_from_ydl_info_dict_playlist(self):
        ydl = YoutubeDL()
        result = self.tu.create_basenames_from_ydl_info_dict(
            ydl, info_dict_playlist)

        expected_result = set([
            'Live Streaming Rafid Aslam-7gjgkH5iPaE',
            'Live Streaming Rafid Aslam-q92kxPm-pqM',
            'Cara Membuat Laptop Menjadi Hotspot WiFi Dengan CMD-YjFwMSDNphM',
            '[CSO] Defeat Boss in Dead End With Thanatos 7-EEm6MwXLse0',
            'Cara Bermain Minecraft Multiplayer Dengan LAN-g2vTZ2ka-tM',
            'Live Streaming Rafid Aslam-AXhuSS5_9YU',
            'Cara Membuat Disk Baru di Komputer-KDOygJnK7Sw',
            'Cara Mendownload Lewat Torrent-cC-9RghkvXs']
        )

        self.assertEqual(result, expected_result)
```

Example 7

Project: *search_within_videos* Author: *iRapha* File: *caption.py* (license) *View Source Project* 7 votes 👍 👎

```
def getCaptions(url, progress_cb, so_far, task_weight):
    ydl = youtube_dl.YoutubeDL({'writesubtitles': True, 'allsubtitles': True, 'writeautomaticsub': True})
    with ydl:
        res = ydl.extract_info(url, download=False)
        if res['requested_subtitles'] and res['requested_subtitles']['en']:
            print ('Grabbing vtt file from ' + res['requested_subtitles']['en']['url'])
            response = requests.get(res['requested_subtitles']['en']['url'], stream=True)
            b = BytesIO()
            for block in response.iter_content(1024):
                b.write(block)
            b.seek(0)
            arr = WebVTTReader().read(b.read().decode('ascii'))
            progress_cb(so_far + task_weight, so_far + task_weight)
            return arr.get_captions('en-US')
        else:
            return []
            print ('Youtube Video does not have any english captions')
            return None
```

Example 8

Project: *musicbot* Author: *ArrowsX* File: *bot.py* (license) *View Source Project* 6 votes 👍 👎

```
def download(title, video_url):
    ydl_opts = {
        'outtmpl': '{}.%(ext)s'.format(title),
        'format': 'bestaudio/best',
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192',
        }],
    }
    with youtube_dl.YoutubeDL(ydl_opts) as ydl:
        ydl.download([video_url])

    return {
        'audio': open('{}.mp3'.format(title), 'rb'),
        'title': title,
    }
```

Example 9

Project: *youtube-closed-captions* Author: *mkly* File: *download.py* (license) *View Source Project* 6 votes 👍 👎

```
def get_result(self, video_id: str) -> int:
        with youtube_dl.YoutubeDL(self.opts) as ydl:
            try:
                return ydl.download([self.get_url_from_video_id(video_id)])
            except youtube_dl.utils.DownloadError as err:
                raise DownloadException(
                    "Unable to download captions: {0}".format(str(err)))
            except youtube_dl.utils.ExtractorError as err:
                raise DownloadException(
                    "Unable to extract captions: {0}".format(str(err)))
            except Exception as err:
                raise DownloadException(
                    "Unknown exception downloading and extracting captions: {0}".format(
                        str(err)))
```

Example 10

Project: *optimalvibes* Author: *littlemika* File: *helper.py* (license) *View Source Project* 6 votes 👍 👎

```
def expect_info_dict(self, got_dict, expected_dict):
    expect_dict(self, got_dict, expected_dict)
    # Check for the presence of mandatory fields
    if got_dict.get('_type') not in ('playlist', 'multi_video'):
        for key in ('id', 'url', 'title', 'ext'):
            self.assertTrue(got_dict.get(key), 'Missing mandatory field %s' % key)
    # Check for mandatory fields that are automatically set by YoutubeDL
    for key in ['webpage_url', 'extractor', 'extractor_key']:
        self.assertTrue(got_dict.get(key), 'Missing field: %s' % key)

    # Are checkable fields missing from the test case definition?
    test_info_dict = dict((key, value if not isinstance(value, compat_str) or len(value) < 250 else 'md5:' + md5(value))
                          for key, value in got_dict.items()
                          if value and key in ('id', 'title', 'description', 'uploader', 'upload_date', 'timestamp', 'uploader_id', 'location',
    missing_keys = set(test_info_dict.keys()) - set(expected_dict.keys())
    if missing_keys:
        def _repr(v):
            if isinstance(v, compat_str):
                return "'%s'" % v.replace('\\', '\\\\').replace("'", "\\'").replace('\n', '\\n')
            else:
                return repr(v)
        info_dict_str = ''
        if len(missing_keys) != len(expected_dict):
            info_dict_str += ''.join(
                '    %s: %s,\n' % (_repr(k), _repr(v))
                for k, v in test_info_dict.items() if k not in missing_keys)
            if info_dict_str:
                info_dict_str += '\n'
        info_dict_str += ''.join(
            '    %s: %s,\n' % (_repr(k), _repr(test_info_dict[k]))
            for k in missing_keys)
        write_string(
            '\n\'info_dict\': {\n' + info_dict_str + '},\n', out=sys.stderr)
        self.assertFalse(
            missing_keys,
            'Missing keys in test definition: %s' % (
                ', '.join(sorted(missing_keys))))
```

Example 11

Project: *optimalvibes* Author: *littlemika* File: *test_YoutubeDL.py* (license) *View Source Project* 6 votes

```
def test_postprocessors(self):
        filename = 'post-processor-testfile.mp4'
        audiofile = filename + '.mp3'

        class SimplePP(PostProcessor):
            def run(self, info):
                with open(audiofile, 'wt') as f:
                    f.write('EXAMPLE')
                return [info['filepath']], info

        def run_pp(params, PP):
            with open(filename, 'wt') as f:
                f.write('EXAMPLE')
            ydl = YoutubeDL(params)
            ydl.add_post_processor(PP())
            ydl.post_process(filename, {'filepath': filename})

        run_pp({'keepvideo': True}, SimplePP)
        self.assertTrue(os.path.exists(filename), '%s doesn\'t exist' % filename)
        self.assertTrue(os.path.exists(audiofile), '%s doesn\'t exist' % audiofile)
        os.unlink(filename)
        os.unlink(audiofile)

        run_pp({'keepvideo': False}, SimplePP)
        self.assertFalse(os.path.exists(filename), '%s exists' % filename)
        self.assertTrue(os.path.exists(audiofile), '%s doesn\'t exist' % audiofile)
        os.unlink(audiofile)

        class ModifierPP(PostProcessor):
            def run(self, info):
                with open(info['filepath'], 'wt') as f:
                    f.write('MODIFIED')
                return [], info

        run_pp({'keepvideo': False}, ModifierPP)
        self.assertTrue(os.path.exists(filename), '%s doesn\'t exist' % filename)
        os.unlink(filename)
```

Example 12

Project: *optimalvibes* Author: *littlemika* File: *test_age_restriction.py* (license) *View Source Project* 6 votes

```
def _download_restricted(url, filename, age):
    """ Returns true if the file has been downloaded """

    params = {
        'age_limit': age,
        'skip_download': True,
        'writeinfojson': True,
        'outtmpl': '%(id)s.%(ext)s',
    }
    ydl = YoutubeDL(params)
    ydl.add_default_info_extractors()
    json_filename = os.path.splitext(filename)[0] + '.info.json'
    try_rm(json_filename)
    ydl.download([url])
    res = os.path.exists(json_filename)
    try_rm(json_filename)
    return res
```

Example 13

Project: *Python-Scripts* Author: *Logan1x* File: *vid.py* (license) *View Source Project* 6 votes

```
def download(link,data):
    try:
        with youtube_dl.YoutubeDL(data) as ydl:
            ydl.download([link])
    except youtube_dl.utils.DownloadError as err:
        print(err)
```

Example 14

Project: *yt2audiobot* Author: *gise88* File: *ythelper.py* (license) *View Source Project* 6 votes

```
def __init__(self, url, progress_hook):
        self._url = url
        self._progress_hook = progress_hook
        self._INFO_OPTS = {
            'logger': YoutubeDLLogger(),
            'socket_timeout': 10
        }

        ydl = ytdl.YoutubeDL(self._INFO_OPTS)
        self._progress_hook({
            'status': 'getting_information'
        })
        self._info = ydl.extract_info(self._url, download=False)
        self._progress_hook({
            'status': 'information_downloaded',
            'type': 'playlist' if self.is_playlist() else 'video'
        })
```

Example 15

Project: *GUIYoutube* Author: *coltking* File: *GUIYoutube.py* (license) *View Source Project* 6 votes 👍 👎

```
def descargarPorLink(self):
        ydl_opts = {
                    'format': self.format,
                    'postprocessors': [{
                        'key': 'FFmpegVideoConvertor',
                        'preferedformat': self.preferedformat
                        }]
                    }
        with YT(ydl_opts) as yt:
            homedir = os.getenv("HOME")
            exdir = os.getcwd() #exdir es el directorio actual, se guarda para saber donde volver una vez completada la descarga
            #print(exdir)
            if not os.path.exists(homedir+ "/Descargas/GUIYoutube"): os.makedirs(homedir+"/Descargas/GUIYoutube")
            os.chdir(homedir+"/Descargas/GUIYoutube")
            #print(os.getcwd())
            yt.download([self.link])
            os.chdir(exdir)
            #self.popup = QtGui.QMessageBox.information(self, "Informacion", """Descarga finalizada (revise la carpeta Descargas)""",QtGui.QMess
            subprocess.Popen(["notify-send", "-t","4000", "Descarga finalizada (revise su carpeta de descargas)"])
```

Example 16

Project: *GUIYoutube* Author: *coltking* File: *VideoWidget.py* (license) *View Source Project* 6 votes 👍 👎

```
def descargar(self):
        self.parent().parent().parent().parent().parent().seleccionDeCalidad()
        self.parent().parent().parent().parent().parent().link = self.descarga[0]
        #Dialogo que consulta la calidad y formato
        #self.dialogo = self.formato()
        """
        ydl_opts = {
                    'format': 'bestaudio/best',
                    'postprocessors': [{
                        'key': 'FFmpegVideoConvertor',
                        'preferedformat': 'mp4'
                        }]
                    }

        with YT(ydl_opts) as yt:
            homedir = os.getenv("HOME")
            exdir = os.getcwd() #exdir es el directorio actual, se guarda para saber donde volver una vez completada la descarga
            #print(exdir)
            if not os.path.exists(homedir+ "/Descargas/GUIYoutube"): os.makedirs(homedir+"/Descargas/GUIYoutube")
            os.chdir(homedir+"/Descargas/GUIYoutube")
            #print(os.getcwd())
            yt.download([self.descarga[0]])
            os.chdir(exdir)"""
            #self.popup = QtGui.QMessageBox.information(self, "Informacion", """Descarga finalizada (revise la carpeta Descargas)""",QtGui.QMess
```

Example 17

Project: *Shallus-Bot* Author: *cgropp* File: *audio.py* (license) *View Source Project* 6 votes 👍 👎

```
def get_info(self):
        if self._yt is None:
            self._yt = youtube_dl.YoutubeDL(youtube_dl_options)
        if "[SEARCH:]" not in self.url:
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)
        else:
            self.url = self.url[9:]
            yt_id = self._yt.extract_info(
                self.url, download=False)["entries"][0]["id"]
            # Should handle errors here ^
            self.url = "https://youtube.com/watch?v={}".format(yt_id)
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)

        self.song = Song(**video)
```

Example 18

Project: *pyjam* Author: *10se1ucgo* File: *downloader.py* (license) *View Source Project* 6 votes 👍 👎

```
def run(self):
        song_urls = iter(self.song_urls)
        errors = []
        with youtube_dl.YoutubeDL(self.opts) as yt:
            while not self.is_aborted():
                try:
                    song_url = next(song_urls)
                    logger.info("Downloading audio/video from {url}".format(url=song_url))
                    try:
                        yt.download([song_url])
                    except youtube_dl.DownloadError:
                        errors.append(song_url)
                    self.downloaded += 100
                    wx.CallAfter(self.parent.download_update, message=self.downloaded)
                except StopIteration:
                    wx.CallAfter(self.parent.download_complete, errors=errors)
                    break
```

Example 19

Project: *pyjam* Author: *10se1ucgo* File: *downloader.py* (license) *View Source Project* 6 votes

```
def yt_extract(links):
    """Extract individual URLs from a playlist.

    Args:
        links (list[str]): A list of the URLs.

    Returns:
        list[str]: The list of extracted URLs.
    """
    songs = []
    with youtube_dl.YoutubeDL() as yt_dl:
        for url in links:
            try:
                result = yt_dl.extract_info(url, download=False, process=False)
            except youtube_dl.DownloadError:
                return songs

            if 'entries' in result:
                songs.extend(vid['url'] for vid in list(result['entries']))
            else:
                songs.append(url)
        return songs
```

Example 20

Project: *leetcode* Author: *WUJISHANXIA* File: *Youtube.py* (license) *View Source Project* 6 votes

```
def download(self,command,id):
        #print command
        download=subprocess.Popen(command,shell=True)
        #download = subprocess.Popen('echo "hello"',shell=True)
        download.wait()
        #with youtube_dl.YoutubeDL() as ydl:
        #    ydl.download(['http://www.bilibili.com/video/av2331280?from=search&seid=8314675707341925231'])
        print 'download video successfully'
        gif_command='ffmpeg -ss 20 -t 20 -i'+' '+str(id)+'.mp4  -s 320x240 -f gif'+' '+str(id)+'.gif'
        gif=subprocess.Popen(gif_command,shell=True)
        gif.wait()
        print 'generate gif successfully'
        cp_command='cp'+' '+str(id)+'.gif'+' /home/www/public/media'
        cp = subprocess.Popen(cp_command, shell=True)
        cp.wait()
        print 'copy gif to media successfully'
```

Example 21

Project: *leetcode* Author: *WUJISHANXIA* File: *Youtube.py* (license) *View Source Project* 6 votes

```
def youtube_download(self,url,id):
        #download_command=['youtube-dl -f bestvideo[ext=mp4]+bestaudio[ext=m4a]/bestvideo+bestaudio\
        #                  --merge-output-format mp4 -o /Users/wujishanxia/Downloads/test.mp4',url]
        command='youtube-dl'+' -f bestvideo[ext=mp4]+bestaudio[ext=m4a]/best[ext=mp4]/best'+' -o'+' '+str(id)+'.mp4'+' '+url
        #thread=threading.Thread(target=self.download,args=(command,id))
        #thread.setDaemon(True)
        #thread.start()
        download=subprocess.Popen(command,shell=True)
        #download = subprocess.Popen('echo "hello"',shell=True)
        download.wait()
        #with youtube_dl.YoutubeDL() as ydl:
        #    ydl.download(['http://www.bilibili.com/video/av2331280?from=search&seid=8314675707341925231'])
        print 'download video successfully'
        gif_command='ffmpeg -ss 20 -t 20 -i'+' '+str(id)+'.mp4  -s 320x240 -f gif'+' '+str(id)+'.gif'
        gif=subprocess.Popen(gif_command,shell=True)
        gif.wait()
        print 'generate gif successfully'
        cp_command='cp'+' '+str(id)+'.gif'+' /home/www/public/media'
        cp = subprocess.Popen(cp_command, shell=True)
        cp.wait()
        print 'copy gif to media successfully'
        rm_command='rm'+' '+str(id)+'.mp4'
        rm = subprocess.Popen(rm_command,shell=True)
        rm.wait()
        print 'rm mp4 successfully'
```

Example 22

Project: *TheaterTrailers* Author: *Electronickss* File: *theaterTrailers.py* (license) *View Source Project* 6 votes

```
def videoDownloader(string, passedTitle, yearVar):
  # Options for the video downloader
  ydl1_opts = {
    'outtmpl': os.path.join(TheaterTrailersHome, 'Trailers', '{0} ({1})'.format(passedTitle, yearVar), '{0} ({1}).mp4'.format(passedTitle, yearV
    'ignoreerrors': True,
    'format': 'mp4',
  }
  with youtube_dl.YoutubeDL(ydl1_opts) as ydl:
    logger.info("downloading {0} from {1}".format(passedTitle, string))
    ydl.download([string])
    shutil.copy2(
        os.path.join(trailerLocation, '{0} ({1})'.format(passedTitle, yearVar), '{0} ({1}).mp4'.format(passedTitle, yearVar)),
        os.path.join(trailerLocation, '{0} ({1})'.format(passedTitle, yearVar), '{0} ({1})-trailer.mp4'.format(passedTitle, yearVar))
      )
    shutil.copy2(
        os.path.join(TheaterTrailersHome, 'res', 'poster.jpg'),
        os.path.join(trailerLocation, '{0} ({1})'.format(passedTitle, yearVar))
      )
    updatePlex()


# Downloads info for the videos from the playlist
```

Example 23

Project: *Addon* Author: *GitAcrown* File: *music.py* (license) *View Source Project* 6 votes

```
def get_info(self):
        if self._yt is None:
            self._yt = youtube_dl.YoutubeDL(youtube_dl_options)
        if "[SEARCH:]" not in self.url:
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)
        else:
            self.url = self.url[9:]
            yt_id = self._yt.extract_info(
                self.url, download=False)["entries"][0]["id"]
            # Should handle errors here ^
            self.url = "https://youtube.com/watch?v={}".format(yt_id)
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)

        self.song = Song(**video)
```

Example 24

Project: *get-youtube-subtitle-url-node* Author: *joegesualdo* File: *test_age_restriction.py* (license) *View Source Project* 6 votes

```
def _download_restricted(url, filename, age):
    """ Returns true if the file has been downloaded """

    params = {
        'age_limit': age,
        'skip_download': True,
        'writeinfojson': True,
        'outtmpl': '%(id)s.%(ext)s',
    }
    ydl = YoutubeDL(params)
    ydl.add_default_info_extractors()
    json_filename = os.path.splitext(filename)[0] + '.info.json'
    try_rm(json_filename)
    ydl.download([url])
    res = os.path.exists(json_filename)
    try_rm(json_filename)
    return res
```

Example 25

Project: *KeekoBot* Author: *DavidNeon* File: *audio.py* (license) *View Source Project* 6 votes

```
def get_info(self):
        if self._yt is None:
            self._yt = youtube_dl.YoutubeDL(youtube_dl_options)
        if "[SEARCH:]" not in self.url:
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)
        else:
            self.url = self.url[9:]
            yt_id = self._yt.extract_info(
                self.url, download=False)["entries"][0]["id"]
            # Should handle errors here ^
            self.url = "https://youtube.com/watch?v={}".format(yt_id)
            video = self._yt.extract_info(self.url, download=False,
                                          process=False)

        self.song = Song(**video)
```

Example 26

Project: *enigma2-plugins* Author: *opendreambox* File: *Videos.py* (license) *View Source Project* 6 votes

```
def _request(self):
                ie_key = "YoutubeLive" if "live" in self._baseurl.lower() else "Youtube"
                try:
                        self._setupFormatMap()
                        with YoutubeDL(self._params) as ytdl:
                                result = ytdl.extract_info(self._baseurl, ie_key=ie_key, download=False, process=True)
                                if self.KEY_ENTRIES in result: # Can be a playlist or a list of videos
                                        entry = result[self.KEY_ENTRIES][0] #TODO handle properly
                                else:# Just a video
                                        entry = result
                                        fmt = entry.get(self.KEY_FORMAT_ID)
                                        url = ""
                                        suburi = ""
                                        for f in entry.get(self.KEY_REQUESTED_FORMATS, []):
                                                if not url and f.get(self.KEY_VCODEC, u"none") != u"none":
                                                        url = str(f.get(self.KEY_URL, ""))
                                                elif not suburi and f.get(self.KEY_ACODEC, u"none") != u"none":
                                                        suburi = str(f.get(self.KEY_URL, ""))
                                        if not url:
                                                url = str(entry.get(self.KEY_URL, ""))
                                        self._onResult(True, url, fmt, suburi)
                except Exception as e:
                        Log.w(e)
                        self._onResult(False, None, -1)
```

Example 27

Project: *yt-browser* Author: *juanfgs* File: *youtube.py* (GNU General Public License v3.0) *View Source Project* 5 votes

```
def download(self):
        config = self.configuration.get_config()
        self.options['format'] = config['quality'] + "[ext="+config['format']+"]+bestaudio/"+config['quality']+ "+bestaudio"
        with youtube_dl.YoutubeDL(self.options) as ydl:
            ydl.download(["http://www.youtube.com/watch?v=" + self.video])
```

Example 28

Project: *Harmonbot* Author: *Harmon758* File: *search.py* (license) *View Source Project* 5 votes

```
def youtube(self, *, search : str):
                '''Find a Youtube video'''
                ydl = youtube_dl.YoutubeDL({"default_search": "auto", "noplaylist": True, "quiet": True})
                func = functools.partial(ydl.extract_info, search, download = False)
                info = await self.bot.loop.run_in_executor(None, func)
                if "entries" in info:
                        info = info["entries"][0]
                await self.bot.reply(info.get("webpage_url"))
```

Example 29

Project: *Harmonbot* Author: *Harmon758* File: *audio_player.py* (license) *View Source Project* 5 votes

```
def _get_song_info(self, song):
                ydl = youtube_dl.YoutubeDL(self.ytdl_options)
                func = functools.partial(ydl.extract_info, song, download = False)
                info = await self.bot.loop.run_in_executor(None, func)
                if "entries" in info:
                        info = info["entries"][0]
                logging.getLogger("discord").info("playing URL {}".format(song))
                return info
```

Example 30

Project: *Harmonbot* Author: *Harmon758* File: *audio_player.py* (license) *View Source Project* 5 votes

```
def _download_song(self, song):
                ydl = youtube_dl.YoutubeDL(self.ytdl_download_options)
                func = functools.partial(ydl.extract_info, song, download = True)
                info = await self.bot.loop.run_in_executor(None, func)
                return ydl.prepare_filename(info)
```

Example 31

Project: *CorpBot.py* Author: *corpnewt* File: *downloader.py* (license) *View Source Project* 5 votes

```
def __init__(self, download_folder=None):
        self.thread_pool = ThreadPoolExecutor(max_workers=2)
        self.unsafe_ytdl = youtube_dl.YoutubeDL(ytdl_format_options)
        self.safe_ytdl = youtube_dl.YoutubeDL(ytdl_format_options)
        self.safe_ytdl.params['ignoreerrors'] = True
        self.download_folder = download_folder

        if download_folder:
            otmpl = self.unsafe_ytdl.params['outtmpl']
            self.unsafe_ytdl.params['outtmpl'] = os.path.join(download_folder, otmpl)
            # print("setting template to " + os.path.join(download_folder, otmpl))

            otmpl = self.safe_ytdl.params['outtmpl']
            self.safe_ytdl.params['outtmpl'] = os.path.join(download_folder, otmpl)
```

Example 32

Project: *modis* Author: *Infraxion* File: *_musicplayer.py* (license) *View Source Project* 5 votes 👍 👎

```
def download_next_song(self, song):
        """Downloads the next song and starts playing it"""

        dl_ydl_opts = dict(ydl_opts)
        dl_ydl_opts["progress_hooks"] = [self.ytdl_progress_hook]
        dl_ydl_opts["outtmpl"] = self.output_format

        # Move the songs from the next cache to the current cache
        self.move_next_cache()

        self.state = 'ready'
        self.play_empty()
        # Download the file and create the stream
        with youtube_dl.YoutubeDL(dl_ydl_opts) as ydl:
            try:
                ydl.download([song])
            except DownloadStreamException:
                # This is a livestream, use the appropriate player
                future = asyncio.run_coroutine_threadsafe(self.create_stream_player(song, dl_ydl_opts), client.loop)
                try:
                    future.result()
                except Exception as e:
                    logger.exception(e)
                    self.vafter_ts()
                    return
            except PermissionError:
                # File is still in use, it'll get cleared next time
                pass
            except youtube_dl.utils.DownloadError as e:
                self.logger.exception(e)
                self.statuslog.error(e)
                self.vafter_ts()
                return
            except Exception as e:
                self.logger.exception(e)
                self.vafter_ts()
                return
```

Example 33

Project: *modis* Author: *Infraxion* File: *_musicplayer.py* (license) *View Source Project* 5 votes 👍 👎

```
def download_next_song_cache(self):
        """Downloads the next song in the queue to the cache"""
        if len(self.queue) == 0:
            return

        cache_ydl_opts = dict(ydl_opts)
        cache_ydl_opts["outtmpl"] = self.output_format_next

        with youtube_dl.YoutubeDL(cache_ydl_opts) as ydl:
            try:
                url = self.queue[0][0]
                ydl.download([url])
            except:
                pass
```

Example 34

Project: *tumdlr* Author: *FujiMakoto* File: *containers.py* (license) *View Source Project* 5 votes 👍 👎

```
def _parse_post(self):
        """
        Parse all available photos using the best image sizes available
        """
        super()._parse_post()

        video_info = YoutubeDL().extract_info(self.url.as_string(), False)

        self.title = video_info.get('title')

        self.description    = video_info.get('description')
        self.duration       = int(video_info.get('duration', 0))
        self.format         = video_info.get('format', 'Unknown')

        self.files.append(TumblrVideo(video_info, self))
```

Example 35

Project: *kvve* Author: *diegodukao* File: *kvveapp.py* (license) *View Source Project* 5 votes 👍 👎

```
def download(self, url):
        options = {}
        options['ignoreerrors'] = True
        options['outtmpl'] = 'videos/%(title)s-%(id)s.%(ext)s'

        with youtube_dl.YoutubeDL(options) as ydl:
            dl_thread = threading.Thread(target=ydl.download, args=([url],))
            dl_thread.start()
```

Example 36

Project: *yt_dl_music* Author: *jtara1* File: *yt_dl_music.py* (license) *View Source Project* 5 votes

```
def my_hook(d):
    '''
    DESCRIPTION:
        This function gets called by youtube_dl; check their docs for more info (YoutubeDL.py)
        Upon completion of download of a video, print string
    '''
#    print (d) # debug - can be used to extract percentage or bytes of file downloaded or time spent downloading or check if download finished
#    with open('my_hook.txt', 'w') as f:
#        f.write(d)
    if d['status'] == 'finished':
        print('Done downloading, now converting ...')
```

Example 37

Project: *yt_dl_music* Author: *jtara1* File: *yt_dl_music.py* (license) *View Source Project* 5 votes

```
def get_ydl_options(dir_downloads_playlist, last_dl_index, end_dl_index,
                    extract_audio=True, preferred_audio_codec='m4a',
                    embed_metadata=True, embed_thumbnail=True):
    """Get youtube-dl options"""
    # check YoutubeDL.py in the youtube_dl library for more info
    # m4a seems to be consistantly available
    postprocessors = []
    if embed_thumbnail:
        postprocessors.append(
            {
                'key': 'EmbedThumbnail', # embed thumbnail in file
            }
        )
    if embed_metadata:
        postprocessors.append(
            {
                'key': 'FFmpegMetadata', # embed metadata in file (uploader & upload date, I think)
            }
        )
    if extract_audio: # only downloads audio
        postprocessors.append(
            {
                'key': 'FFmpegExtractAudio',
                'preferredcodec': preferred_audio_codec,
                'preferredquality': '192',
            }
        )
    ydl_opts = {
        'outtmpl': os.path.join(dir_downloads_playlist, '%(title)s.%(ext)s'), # location & template for title of output file
        'usetitle': True,
        'writethumbnail': True,             # needed if using postprocessing EmbedThumbnail
        'playliststart': last_dl_index + 1,
        'playlistend': end_dl_index,        # stop downloading at this index
        'format': 'bestaudio/best',
        'ignoreerrors': True,               # allows continuation after errors such as Download Failed from deleted or removed videos
        'postprocessors': postprocessors,
        'logger': MyLogger(),
        'progress_hooks': [my_hook],
    }

    return ydl_opts
```

Example 38

Project: *eitbapi* Author: *erral* File: *tv.py* (license) *View Source Project* 5 votes

```
def episode(request):
    """ Get all the information and the video links from a given episode.
        How: use youtube-dl to get the information
    """
    episode_url = request.matchdict['episode_url']

    url = EITB_VIDEO_BASE_URL + episode_url
    try:
        playlist_title, playlist_id, video_title, video_id = episode_url.split('/')
    except ValueError:
        return {}

    result = {
        '@context': 'http://www.w3.org/ns/hydra/context.jsonld',
        '@id': request.route_url('episode', episode_url=episode_url),
        '@type': 'Episode',
        'parent': request.route_url('playlist', playlist_id=playlist_id),
    }

    try:
        ydl = youtube_dl.YoutubeDL({'outtmpl': '%(id)s%(ext)s'})
        video_data = ydl.extract_info(url, download=False)
    except youtube_dl.DownloadError:
        return result

    result.update(video_data)
    return result
```

Example 39

Project: *globibot* Author: *best-coloc-ever* File: *player.py* (license) *View Source Project* 5 votes 👍 👎

```
def extract_info(self):
        opts = {
            'simulate': True
        }
        opts.update(YTDL_PLAYER_OPTS)
        ydl = YoutubeDL(opts)
        return ydl.extract_info(self.resource)
```

Example 40

Project: *youtube-updater* Author: *Korons* File: *youtube-updater.py* (license) *View Source Project* 5 votes 👍 👎

```
def download(video_title, opts, url):
    if args.l:
        if video_title not in open(args.l).read():
            with open(args.l, mode='a') as f:
                f.write(video_title + '\n')
    with youtube_dl.YoutubeDL(opts) as ydl:
        ydl.download([url])
    with open(downloaded, mode='a') as f:
        f.write(url + '\n')
```

Example 41

Project: *youtube_downloader* Author: *aksinghdce* File: *backend_youtube_dl.py* (license) *View Source Project* 5 votes 👍 👎

```
def _fetch_basic(self):
        """ Fetch basic data and streams. """
        if self._have_basic:
            return

        with youtube_dl.YoutubeDL(self._ydl_opts) as ydl:
            try:
                self._ydl_info = ydl.extract_info(self.videoid, download=False)
            # Turn into an IOError since that is what pafy previously raised
            except youtube_dl.utils.DownloadError as e:
                raise IOError(str(e).replace('YouTube said', 'Youtube says'))

        self.callback("Fetched video info")

        self._title = self._ydl_info['title']
        self._author = self._ydl_info['uploader']
        self._rating = self._ydl_info['average_rating']
        self._length = self._ydl_info['duration']
        self._viewcount = self._ydl_info['view_count']
        self._likes = self._ydl_info['like_count']
        self._dislikes = self._ydl_info['dislike_count']
        self._username = self._ydl_info['uploader_id']
        self._category = self._ydl_info['categories'][0] if self._ydl_info['categories'] else ''
        self._bigthumb = g.urls['bigthumb'] % self.videoid
        self._bigthumbhd = g.urls['bigthumbhd'] % self.videoid
        self.expiry = time.time() + g.lifespan

        self._have_basic = True
```

Example 42

Project: *optimalvibes* Author: *littlemika* File: *test_http.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_unicode_path_redirection(self):
        # XXX: Python 3 http server does not allow non-ASCII header values
        if sys.version_info[0] == 3:
            return

        ydl = YoutubeDL({'logger': FakeLogger()})
        r = ydl.extract_info('http://localhost:%d/302' % self.port)
        self.assertEqual(r['entries'][0]['url'], 'http://localhost:%d/vid.mp4' % self.port)
```

Example 43

Project: *optimalvibes* Author: *littlemika* File: *test_http.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_nocheckcertificate(self):
        if sys.version_info >= (2, 7, 9):  # No certificate checking anyways
            ydl = YoutubeDL({'logger': FakeLogger()})
            self.assertRaises(
                Exception,
                ydl.extract_info, 'https://localhost:%d/video.html' % self.port)

        ydl = YoutubeDL({'logger': FakeLogger(), 'nocheckcertificate': True})
        r = ydl.extract_info('https://localhost:%d/video.html' % self.port)
        self.assertEqual(r['entries'][0]['url'], 'https://localhost:%d/vid.mp4' % self.port)
```

Example 44

Project: *optimalvibes* Author: *littlemika* File: *test_http.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_proxy(self):
        geo_proxy = 'localhost:{0}'.format(self.geo_port)
        ydl = YoutubeDL({
            'proxy': 'localhost:{0}'.format(self.port),
            'geo_verification_proxy': geo_proxy,
        })
        url = 'http://foo.com/bar'
        response = ydl.urlopen(url).read().decode('utf-8')
        self.assertEqual(response, 'normal: {0}'.format(url))

        req = compat_urllib_request.Request(url)
        req.add_header('Ytdl-request-proxy', geo_proxy)
        response = ydl.urlopen(req).read().decode('utf-8')
        self.assertEqual(response, 'geo: {0}'.format(url))
```

Example 45

Project: *optimalvibes* Author: *littlemika* File: *test_http.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_proxy_with_idn(self):
        ydl = YoutubeDL({
            'proxy': 'localhost:{0}'.format(self.port),
        })
        url = 'http://??.tw/'
        response = ydl.urlopen(url).read().decode('utf-8')
        # b'xn--fiq228c' is '??'.encode('idna')
        self.assertEqual(response, 'normal: http://xn--fiq228c.tw/')
```

Example 46

Project: *optimalvibes* Author: *littlemika* File: *test_write_annotations.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_info_json(self):
        expected = list(EXPECTED_ANNOTATIONS)  # Two annotations could have the same text.
        ie = youtube_dl.extractor.YoutubeIE()
        ydl = YoutubeDL(params)
        ydl.add_info_extractor(ie)
        ydl.download([TEST_ID])
        self.assertTrue(os.path.exists(ANNOTATIONS_FILE))
        annoxml = None
        with io.open(ANNOTATIONS_FILE, 'r', encoding='utf-8') as annof:
            annoxml = xml.etree.ElementTree.parse(annof)
        self.assertTrue(annoxml is not None, 'Failed to parse annotations XML')
        root = annoxml.getroot()
        self.assertEqual(root.tag, 'document')
        annotationsTag = root.find('annotations')
        self.assertEqual(annotationsTag.tag, 'annotations')
        annotations = annotationsTag.findall('annotation')

        # Not all the annotations have TEXT children and the annotations are returned unsorted.
        for a in annotations:
            self.assertEqual(a.tag, 'annotation')
            if a.get('type') == 'text':
                textTag = a.find('TEXT')
                text = textTag.text
                self.assertTrue(text in expected)  # assertIn only added in python 2.7
                # remove the first occurrence, there could be more than one annotation with the same text
                expected.remove(text)
        # We should have seen (and removed) all the expected annotation texts.
        self.assertEqual(len(expected), 0, 'Not all expected annotations were found.')
```

Example 47

Project: *optimalvibes* Author: *littlemika* File: *test_YoutubeDL.py* (license) *View Source Project* 5 votes 👍 👎

```
def test_prepare_filename(self):
        info = {
            'id': '1234',
            'ext': 'mp4',
            'width': None,
            'height': 1080,
        }

        def fname(templ):
            ydl = YoutubeDL({'outtmpl': templ})
            return ydl.prepare_filename(info)
        self.assertEqual(fname('%(id)s.%(ext)s'), '1234.mp4')
        self.assertEqual(fname('%(id)s-%(width)s.%(ext)s'), '1234-NA.mp4')
        # Replace missing fields with 'NA'
        self.assertEqual(fname('%(uploader_date)s-%(id)s.%(ext)s'), 'NA-1234.mp4')
        self.assertEqual(fname('%(height)d.%(ext)s'), '1080.mp4')
        self.assertEqual(fname('%(height)6d.%(ext)s'), '  1080.mp4')
        self.assertEqual(fname('%(height)-6d.%(ext)s'), '1080  .mp4')
        self.assertEqual(fname('%(height)06d.%(ext)s'), '001080.mp4')
        self.assertEqual(fname('%(height) 06d.%(ext)s'), ' 01080.mp4')
        self.assertEqual(fname('%(height)   06d.%(ext)s'), ' 01080.mp4')
        self.assertEqual(fname('%(height)0 6d.%(ext)s'), ' 01080.mp4')
        self.assertEqual(fname('%(height)0   6d.%(ext)s'), ' 01080.mp4')
        self.assertEqual(fname('%(height)   0   6d.%(ext)s'), ' 01080.mp4')
        self.assertEqual(fname('%%(height)06d.%(ext)s'), '%(height)06d.mp4')
        self.assertEqual(fname('%(width)06d.%(ext)s'), 'NA.mp4')
        self.assertEqual(fname('%(width)06d.%%(ext)s'), 'NA.%(ext)s')
        self.assertEqual(fname('%%(width)06d.%(ext)s'), '%(width)06d.mp4')
```

Example 48

Project: *optimalvibes* Author: *littlemika* File: *test_YoutubeDL.py* (license) *View Source Project* 5 votes

```
def test_format_note(self):
        ydl = YoutubeDL()
        self.assertEqual(ydl._format_note({}), '')
        assertRegexpMatches(self, ydl._format_note({
            'vbr': 10,
        }), r'^\s*10k$')
        assertRegexpMatches(self, ydl._format_note({
            'fps': 30,
        }), r'^30fps$')
```

Example 49

Project: *optimalvibes* Author: *littlemika* File: *test_download.py* (license) *View Source Project* 5 votes

```
def __init__(self, *args, **kwargs):
        self.to_stderr = self.to_screen
        self.processed_info_dicts = []
        super(YoutubeDL, self).__init__(*args, **kwargs)
```

Example 50

Project: *Personal_AI_Assistant* Author: *PratylenClub* File: *Youtube_Player.py* (license) *View Source Project* 5 votes

```
def download_song(self,textToSearch):
        url = query_youtube_video(textToSearch)[0]
        ydl_opts = {'format': 'bestaudio/best',
                    'postprocessors': [{'key': 'FFmpegExtractAudio',
                                        'preferredcodec': 'mp3',
                                        'preferredquality': '192',
                                        }],
                    }
        with youtube_dl.YoutubeDL(ydl_opts) as ydl:
            ydl.download([url])
```