



TERM PROJECT

Instructor: Le Thanh Son

Project Task: Music Crawler and Music Player

Trương Công Trung ITITIU19059

Contents

1	Introduction	3
1.1	Background.....	3
1.2	Objectives.....	3
2	Music Crawler.....	3
2.1	Collect list of songs.....	3
2.1.1	Request.....	6
2.1.2	Request URL.....	7
2.1.3	SIG	8
2.1.4	Cookie.....	11
2.1.5	List of songs	11
2.2	Getting songs' information	12
2.3	Getting songs' lyric.....	14
2.4	Getting songs' audio.....	16
2.5	Main function	17
3	Music Player	18
3.1	webserver.py.....	19
3.2	music.html.....	20
3.3	song_info.html	21
3.4	song.html.....	22
4	Usage.....	23
5	Conclusion	23
6	Referrer	23

1 Introduction

1.1 Background

In order to gain practical experience in the network programming course, I pick up the term project of music crapper, which plays on detecting and understanding the API from an online music site.

1.2 Objectives

In this project, I will dig on zingmp3.vn, understand how data work in there and try to collect as much as valuable information from them. Then build my own system to deploy the simplest website which shows all local data.

2 Music Crawler

2.1 Collect list of songs

The first step is to determine what data to get and where it can be collected.

I choose a target that is the most unstable page but also the most valuable page, which will always have new songs and have a real-time update.

<https://zingmp3.vn/moi-phat-hanh>

Open the “Network” tab in “Develop tools” of the browser, I could observe the activities of the website, and find out the way that data was requested.

Following the “Request URL”, the json data shows up and our job is just to collect “encodedID”, which is the representative of song.

Zing mp3

Nhập tên bài hát, nghệ sĩ hoặc MV...

- Cá Nhân
- Khám Phá
- #zingchart
- Radio **LIVE**
- Theo Dõi
- Nhạc Mới**
- Thể Loại
- Top 100
- MV

Đang nhập để khám phá playlist dành riêng cho bạn

ĐĂNG NHẬP

+ Tạo playlist mới

Mới Phát Hành

Số	Tên	Ca sĩ / Nhóm	Mô tả	Thời lượng
1	Không Thể Yêu, Không Thể Quên	Khang Việt	Không Thể Yêu, Không Thể Quên (Single)	05:33
2	hai mươi hai (22)	Hứa Kim Tuyền, AMEE	hai mươi hai (22) (Single)	03:59
3	Chỉ Còn Một Đêm	Quang Hùng MasterD, F.HERO	Chỉ Còn Một Đêm (Single)	04:25
4	Động Phòng Hoa Chúc	Lâm Chấn Khang, Jombie	Động Phòng Hoa Chúc (Single)	03:56
5	Người Anh Yêu	Anh Quân Idol	Người Anh Yêu (Single)	04:41
6	Tình Đầu	Tăng Phúc	Tình Đầu (Single)	03:54
7	Biển Cát Nắng Xanh Vàng Trắng	Nguyễn Vũ	Biển Cát Nắng Xanh Vàng Trắng (Single)	04:06

Network tab in browser developer tools showing network requests for the "Mới Phát Hành" chart page.

Request URL: <https://zingmp3.vn/api/v2/page/get/newrelease-chart?ctime=1653766417&version=1.6.27&sig=eac6f1d96983a3b7d2b18ba4c09f4231319e2cefd6078a2e28f7ca4deab13379531116df234c17ae3de1d3a910dcf373b5ebd1e82504c818ede0d1f4bbcd8&apiKey=88265e23d4284f25963e6eedac8fbfa3>

Request Method: GET

Status Code: 200 OK

Remote Address: 120.138.68.154:443

Referrer Policy: strict-origin-when-cross-origin

Response Headers:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Headers: Accept, Origin, Content-Type, Session-Key , Api-Key
- Access-Control-Allow-Origin: https://zingmp3.vn
- Access-Control-Request-Method: GET, POST , OPTIONS ,PUT, DELETE, PATCH
- Cache-Control: private
- Connection: keep-alive
- Content-Encoding: gzip
- Content-Type: application/json;charset=utf-8

Network timeline showing request timing and resource details.

{"err":0,"msg":"Success","data":{"banner":"https://zmp3-static.zmdcdn.me/skins/zmp3-v5.2/images/chart-new-release-vn.png","type":"song","link":"/moi-phat-hanh","sectionType":"newReleaseChart","sectionId":"hNewRelease","viewType":"slider","items":[{"encodeId":"ZZBAWCZ","title":"Không Thé Yêu, Không Thé Quén","alias":"Khong-The-Yeu-Khong-The-Quen-Khang-Viet","isOfficial":true,"username":"","artistsNames":"Khang Việt","artists":[{"id":"IWZ92B7D","name":"Khang Việt","link":"/Khang-Viet","spotlight":false,"alias":"Khang-Viet","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/7/7/a/f/77a7df3537177703cab72e92350aca6.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/7/7/a/f/77a7df3537177703cab72e92350aca6.jpg","isOA":true,"isOABrand":false,"playlistId":"Z6BUIAFU"}, {"isWorldWide":true,"thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/7/7/a/f/77a7df3537177703cab72e92350aca6.jpg","link":"/bai-hat/Khong-The-Yeu-Khong-The-Quen-Khang-Viet/ZZBAWCZ.html","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w94_r1x1.jpeg/cover/7/f/3/5/f7f353f61642fa8b40c807578a7485ab4.jpg","duration":333,"zingChoice":false,"isPrivate":false,"preRelease":false,"releaseDate":1653393600,"genreIds":["IWZ92F08I","IWZ97FCD"],"album":{"encodeId":"6B0U06BU","title":"Không Thé Yêu, Không Thé Quén (Single)","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w165_r1x1.jpeg/cover/6/5/0/7/6507c467a9e8d80d6538b29bb321143.jpg","isOfficial":true,"link":"/album/Khong-The-Yeu-Khong-The-Quen-Single-Khang-Viet/6B0U06BU.html","isIndie":false,"releaseDate":"24/05/2022","sortDescription":"","PR":false,"artists":[{"id":"IWZ92B7D","name":"Khang Việt","link":"/Khang-Viet","spotlight":false,"alias":"Khang-Viet","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/7/7/a/f/77a7df3537177703cab72e92350aca6.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/7/7/a/f/77a7df3537177703cab72e92350aca6.jpg","isOA":true,"isOABrand":false,"playlistId":"Z6BUIAFU","totalFollow":50934}], "artistNames":"Khang Việt","indicators":[],"isIndie":false,"streamingStatus":1,"allowAudioAds":false,"hasLyric":true,"rankingStatus":0,"releasedAt":1653393600}, {"encodeId":"ZZBFZ0F8","title":"hai muoi hai (22)","alias":"hai-muoi-hai-22-Hua-Kim-Tuyen-AMEE","isOfficial":true,"username":"","artistsNames":"Hứa Kim Tuyên, AMEE","artists":[{"id":"IWZCOZ0I","name":"Hứa Kim Tuyên","link":"/nghe-si/Hua-Kim-Tuyen","spotlight":false,"alias":"Hứa-Kim-Tuyen","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/e/9/0/c/e90c61137614fdfeff31100e93447bf4d.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/e/9/0/c/e90c61137614fdfeff31100e93447bf4d.jpg","isOA":false,"isOABrand":false,"playlistId":"Z6B0F78A"}, {"id":"IW6WBZwO","name":"AMEE","link":"/AMEE","spotlight":false,"alias":"AMEE","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/9/8/4/d/984de8b3596498462d5f954eadbb1f47.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/9/8/4/d/984de8b3596498462d5f954eadbb1f47.jpg","isOA":true,"isOABrand":false}], "isWorldWide":true,"thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/cover/b/6/d/b6de6c0057b19e1c921f2d379817d491.jpg","duration":239,"zingChoice":false,"isPrivate":false,"preRelease":false,"releaseDate":1653448000,"genreIds":["IWZ92F08I","IWZ97FCD"],"album":{"encodeId":"6B06AU68","title":"hai muoi hai (22) (Single)","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w165_r1x1.jpeg/cover/f/a/0/d/faa0dd90fce6538b52f1f3b87999f5ed1.jpg","isOfficial":true,"link":"/album/hai-muoi-hai-22-Single-Hua-Kim-Tuyen-AMEE/6B06AU68.html","isIndie":false,"releaseDate":"25/05/2022","sortDescription":"","PR":false,"artists":[{"id":"IWZCOZ0I","name":"Hứa Kim Tuyên","link":"/nghe-si/Hua-Kim-Tuyen","spotlight":false,"alias":"Hứa-Kim-Tuyen","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/e/9/0/c/e90c61137614fdfeff31100e93447bf4d.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/e/9/0/c/e90c61137614fdfeff31100e93447bf4d.jpg","isOA":false,"isOABrand":false,"playlistId":"Z6B0F78A","totalFollow":11136}, {"id":"IW6WBZwO","name":"AMEE","link":"/AMEE","spotlight":false,"alias": "AMEE","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/9/8/4/d/984de8b3596498462d5f954eadbb1f47.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/9/8/4/d/984de8b3596498462d5f954eadbb1f47.jpg","isOA":true,"isOABrand":false,"totalFollow":373853}], "artistsNames":"Hứa Kim Tuyên, AMEE","indicators":[],"isIndie":false,"mvLink":"video-clip/hai-muoi-hai-22-Hua-Kim-Tuyen-AMEE/ZZBFZ0F8.html","streamingStatus":1,"allowAudioAds":true,"hasLyric":true,"rankingStatus":2,"releasedAt":16534480000}, {"encodeId":"ZZBEU099","title":"Chi Còn Một Dêm","alias":"Chi-Con-Mot-Dem-Quang-Hung-MasterD-F-HERO","isOfficial":true,"username":"","artistsNames":"Quang Hùng MasterD, F.HERO","artists":[{"id":"IWZAZFC0","name":"Quang Hùng MasterD","link":"/Quang-Hung-MasterD","spotlight":false,"alias":"Quang-Hung-MasterD","thumbnail":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/avatars/b/f/b/bfb0aab82efdf18aebefdee6b82f3d9f3.jpg","thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w360_r1x1.jpeg/avatars/b/f/b/bfb0aab82efdf18aebefdee6b82f3d9f3.jpg","isOA":true,"isOABrand":false,"playlistId":"Z6B0F78A"}, {"isWorldWide":true,"thumbnailM":"https://photo-resize-zmp3.zmdcdn.me/w240_r1x1.jpeg/cover/b/6/d/b6de6c0057b19e1c921f2d379817d491.jpg","duration":239,"zingChoice":false,"isPrivate":false,"preRelease":false,"releaseDate":1653448000}, {"isOA":true,"isOABrand":false,"playlistId":"Z6B0F78A"}]}]}]

2.1.1 Request

The next step is to build a valid request in order to legally get data from the website. I use <https://reqbin.com/> to make a test base-on “Request URL” and “Request Headers”.

Request URL: `https://zingmp3.vn/api/v2/page/get/newrelease-chart?ctime=1653766417&version=1.6.27&sig=eac6f1d96983a3b7d2b18ba4c09f4231319e2ceef06078a2e28f7ca4deab13379531116df234c17ae3de1d3a910dcef373b5ebd1e82504c818ede0d1f4bbcccd8&apiKey=88265e23d4284f25963e6eedac8fbfa3`

Request Header:

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
Gecko/20100101 Firefox/91.0

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate, br

Upgrade-Insecure-Requests: 1

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: none

Sec-Fetch-User: ?1

Connection: keep-alive

Cookie:

zmp3_rqid=MHwyNy43Ni4xMTAdUngMjA5fG51WeBGx8MTY1MzmUsIC5NjE4MjU0Mg

Cache-Control: max-age=0

https://zingmp3.vn/

GET US Send

Status: 200 (OK) Time: 1549 ms Size: 208.65 kb

Authorization Content Headers (12) Raw (14)

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:9.0.1) Gecko/20100101 Firefox/9.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Connection: keep-alive
Cookie: zmp3_rqid=MHwyNy43Ni4xMTAdUngMjA5fG51WeB; Cache-Control: max-age=0
```

Content (6633) Headers (13) Raw (15) JSON

Timings

```
{
  "type": "Song",
  "link": "/moi-phat-hanh",
  "sectionType": "newReleaseChart",
  "sectionId": "hNewrelease",
  "viewType": "slider",
  "items": [
    {
      "encodeId": "ZZBAWCCZ",
      "title": "Không Thể Yêu, Không Thể",
      "alias": "Khong-The-Yeu-Khong-The",
      "isOfficial": true,
      "username": "",
      "artistsNames": "Khang Việt",
      "...": ...
    }
  ]
}
```

2.1.2 Request URL

In “Request URL”, there are several arguments:

- Host: <https://zingmp3.vn/>
- API type: /api/v2/page/get/newrelease-chart
- Time: ?ctime=1653766417
- Version: &version=1.6.27
- Signature:
`&sig=eac6f1d96983a3b7d2b18ba4c09f4231319e2cefd6078a2e
 28f7ca4deab13379531116df234c17ae3de1d3a910dcef373b5ebd
 1e82504c818ede0d1f4bbcccd8`
- API key: &apiKey=88265e23d4284f25963e6eedac8fbfa3

Host simply is the website url.

API type could be discovered like “top 100 of genre”, “lyric”, “song”, “stream”.

```
/api/v2/page/get/newrelease-chart  
/api/v2/page/get/song  
/api/v2/lyric/get/lyric  
/api/v2/song/get/streaming
```

Time is to avoid an unknowledge crawl bot. Each time the user makes request, it will get current time and mix with other values in order to calculate valid “sig”. But it will be a big flaw when I discovered their encrypt algorithms, “ctime” could be a constant value and still be a valid request.

```
_time = str(int(datetime.datetime.now().timestamp()))
```

Version is a constant value.

2.1.3 SIG

Request being handled by “main.min.js”

Searching for “apikey”, I could find “apiKey” argument and also point out and make a breakpoint for debugging, and discover their encrypt algorithms.

```

main.min.js      main.min.js:formatted
6679 }
6680     }
6681     var g = function(t) {
6682         var e = {};
6683         return Object.keys(t).sort().forEach((function(n) {
6684             e[n] = t[n]
6685         })),
6686         e
6687     }
6688     , v = function(t) {
6689         var e = arguments.length > 1 && void 0 !== arguments[1] ? arguments[1] : "&"
6690         , n = encodeURIComponent;
6691         return Object.keys(t).map((function(e) {
6692             return n(t[e]).length > 5e3 ? "" : "".concat(n(e), "=").concat(n(t[e]))
6693         })
6694         ).filter((function(t) {
6695             return "" !== t
6696         })
6697         ).join(e)
6698     }
6699     , b = n(44);
6700     function y(t) {
6701         var e = arguments.length > 1 && void 0 !== arguments[1] ? arguments[1] : {};
6702         return e.ctime = Object(u.currentTimeMillis)(),
6703         e.version = o.d,
6704         e.sig = m(t, e),
6705         e.apiKey = h,
6706         e
6707     }
6708     function w(t, e, n, u, c) {
6709         var l = a({
6710             credentials: o.j ? "include" : "same-origin"
6711         }, n);
6712         o.k && (e ? e.userid = 1055798857 : e = {
6713             userid: 1055798857
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
6999

```

In summary, it will take arguments from request url, concat and calculate their SHA256. The result will concat with “apitype” string and hash by SHA1 with secret key SEC=”2aa2d1c561e809b267f3638c4a307aab”

```

6662 , s = n(123)
6663 , u = n(28)
6664 , c = n(598)
6665 , l = n.n(c)
6666 , f = n(599)
6667 , d = n.n(f)
6668 , h = "88265e23d4284f25963e6eedac8fbfa3"
6669 , p = ["ctime", "id", "type", "page", "count", "version"];
6670 function m(t, e) {
6671   var n = function(t) {
6672     var e = g(t), t = "/api/v2/page/get/newrelease-chart"
6673     , n = [];
6674     for (var r in e) e = {ctime: 1653769070, version: '1.6.27'}
6675     -1 !== p.indexOf(r) && null != t[r] && void 0 !== t[r] && "" !== t[r] && (n[r] = e[r]), r = in
6676     return v(n, "")
6677   }(e = {ctime: 1653769070, version: '1.6.27'}
6678   , r = l()(":" + n); n = "ctime=1653769070version=1.6.27"
6679   return d()(t + r) + "2aa2d1c561e809b267f3638c4a307aab")
6680 }
6681 var g = function(t) {
6682   var e = {};
6683   return Object.keys(t).sort().forEach((function(n) {
6684     e[n] = t[n]
6685   )), e
6686   )
6687   , v = function(t) {
6688     var e = arguments.length > 1 && void 0 !== arguments[1] ? arguments[1] : "&"
6689     , n = encodeURIComponent;
6690     return Object.keys(t).map((function(e) {
6691       return n(t[e]).length > 5e3 ? ":" + ".concat(n(e), "=").concat(n(t[e]))"
6692     })).filter((function(t) {
6693       return "" !== t
6694     })
6695     , return ""
6696   )
6697 }
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328
7329
7330
7331
7332
7333
7334
7335
7336
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7399
7400
7401
7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423
7424
7425
7426
7427
7428
7429
7430
7431
7432
7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7778
7779
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7798
7799
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7898
7899
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7998
7999
7999
8000
8001
8002
8003
8004
8005
8006
8007
8008
8009
8009
8010
8011
8012
8013
8014
8015
8016
8017
8018
8019
8019
8020
8021
8022
8023
8024
8025
8026
8027
8028
8029
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8039
8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8078
8079
8079
8080
8081
8082
8083
8084
8085
8086
8087
8088
8089
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8098
8099
8099
8100
8101
8102
8103
8104
8105
8106
8107
8108
8109
8109
8110
8111
8112
8113
8114
8115
8116
8117
8118
8119
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8149
8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178
8178
8179
8179
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8198
8199
8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8219
8220
8221
8222
8223
8224
8225
8226
8227
8228
8229
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8259
8260
8261
8262
8263
8264
8265
8266
8267
8268
8269
8269
8270
8271
8272
8273
8274
8275
8276
8277
8278
8278
8279
8279
8280
8281
8282
8283
8284
8285
8286
8287
8288
8289
8289
8290
8291
8292
8293
8294
8295
8296
8297
8298
8298
8299
8299
8300
8301
8302
8303
8304
8305
8306
8307
8308
8309
8309
8310
8311
8312
8313
8314
8315
8316
8317
8318
8319
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8378
8379
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8398
8399
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8439
8440
8441
8442
8443
8444
8445
8446
8447
8448
8449
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8478
8479
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8498
8499
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8549
8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8578
8579
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8598
8599
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8619
8620
8621
8622
8623
8624
8625
8626
8627
8628
8629
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638
8639
8639
8640
8641
8642
8643
8644
8645
8646
8647
8648
8649
8649
8650
8651
8652
8653
8654
8655
8656
8657
8658
8659
8659
8660
8661
8662
8663
8664
8665
8666
8667
8668
8669
8669
8670
8671
8672
8673
8674
8675
8676
8677
```

2.1.4 Cookie

One more problem is a cookie. It has expired time and I had to identify after several times.

Luckily, website automatically resets a cookie when it expired.

I just make an invalid request containing an expired cookie and update my cookie by using “Set-Cookie” value in “Response Headers” of response data..

The screenshot shows a browser developer tools interface with the Network tab selected. At the top, there are dropdown menus for URL (https://zingmp3.vn/), Method (GET), and Location (US). To the right, the status is shown as "Status: 200 (OK)" with a time of "993 ms" and a size of "0.30 kb". Below the status, there are tabs for Authorization, Content, Headers (14), Raw (16), Content (6), Headers (15), Raw (17), and JSON. The Headers (14) tab is active, showing the client's request headers. The Raw (16) tab is also visible. On the right, there is a "Timings" section and another set of tabs for Content (6), Headers (15), Raw (17), and JSON. The Headers (15) tab is active, showing the server's response headers. A specific header, "Set-Cookie", is highlighted with a red box, indicating its modification. The response body is also visible below the headers.

```
Accept: /
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,vi;q=0.8
Connection: keep-alive
Cookie: zmp3_rqid=HwyNy43Ni4xMTAdUngMjA5fHYXLjYdI
DNT: 1
Referer: https://zingmp3.vn/moi-phat-hanh
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
```

```
Server: nginx
Date: Sat, 28 May 2022 20:40:01 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 308
Connection: keep-alive
Set-Cookie: zmp3_rqid=MHwyMDYdUngMTg5LjIwNS4yNTF; expires=Thu, 01 Jan 1970 00:00:00 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: private
Content-Encoding: gzip
Access-Control-Allow-Origin: https://zingmp3.vn
Access-Control-Allow-Credentials: true
```

```
22 def checking(response: str) -> None:
23     if b'err':-201' in response.content:
24         print("\n[*] Error: Cookie expired! Trying update the new one!")
25         updateCookie()
26         main()
27         exit(0)
28     elif b'err':0' not in response.content:
29         print(response.headers)
30         print(response.content)
31         exit(0)
32
33 def updateCookie() -> None:
34     with eventlet.Timeout(10):
35         url = "https://zingmp3.vn/api/v2/page/get/newrelease-chart?ctime=1652468472&vers"
36         response = requests.get(url)
37         HEADERS["Cookie"] = response.headers["Set-Cookie"].split(';')[0]
38         with open("HEADERS.json", 'w') as f:
39             f.write(json.dumps(HEADERS))
```

2.1.5 List of songs

Request is good now, just catch the response data and find the “encodedID”. This is the representation code for each song.

```

62 def getTopList(name_api:str = '/api/v2/page/get/newrelease-chart', param:dict = None) -> list:
63     print('\n[*] Getting list of song')
64     with eventlet.Timeout(10):
65         if param:
66             url = urlGeneration(name_api, {'ctime':_time,'id':param["id"],'version':'1.6.25'})
67         else:
68             url = urlGeneration(name_api, {'ctime':_time,'version':'1.6.25'})
69         response = requests.get(url,headers=HEADERS)
70         checking(response)
71
72         data = re.findall(r'\\"items\"\s*:\s*\[.*\]', response.content.decode('utf-8'))
73         data = json.loads(data[0].split("\\"items\" : ")[1])
74
75         res = []
76         for items in data:
77             res.append(items["encodeId"])
78     return res

```

```

[*] Getting list of song
[*] Number of song: 100
['ZZBAWCCZ', 'ZZBFZ0F8', 'ZZBF8CBW', 'ZZBEU099', 'ZZB8DABW', 'ZZC00UI6', 'ZZC0WOCZ', 'ZZBAFB9A', 'ZZBUE7ZA', 'ZZC0W06C', 'ZZBDE0WC',
'ZZBDZ789', 'ZZBAWE0F', 'ZZBB0ZEC', 'ZZBFAIFC', 'ZZBBZ887', 'ZZBAUIBI', 'ZZC00FB', 'ZZBZA099', 'ZZBAFF60', 'ZZB6D1W', 'ZZC00U00',
'ZZB6BZU', 'ZZB6BB8C', 'ZZBEAZWA', 'ZZB80IW0', 'ZZB8UW0E', 'ZZC0968D', 'ZZBCCWUD', 'ZZBUFCWD', 'ZZBZAU6C', 'ZZBF68UA', 'ZZC000CE', 'ZZAU7CA0', 'ZZB7C60I',
'ZZBAUADC', 'ZZBBUZEW', 'ZZBZ016F', 'ZZBIIIAE', 'ZZBAU97B', 'ZZB97ED6', 'ZZC0ZIWM', 'ZZBBZ0ZF', 'ZZBAUOWB', 'ZZB0IC89', 'ZZB8E00B', 'ZZB6WDW',
'ZZBEZ0B0', 'ZZBZB788', 'ZZAEU67', 'ZZBAFCE0', 'ZZB6D176', 'ZZBAU8DI', 'ZZBAWEAU', 'ZZBZA6A', 'ZZB8UW6Z', 'ZZBE1B0E', 'ZZBBZAU8', 'ZZB8BUU8', 'ZZB8UDA8',
'ZZB97FOZ', 'ZZB8F9B6', 'ZZBAUÁZC', 'ZZC000B0', 'ZZB86ADB', 'ZZB8BWIE', 'ZZB8A8C0', 'ZZAFW8Z', 'ZZB8B18F', 'ZZBAF8AC', 'ZZBWAUW6', 'ZZB97E6A',
'ZZB6B8AI', 'ZZB6BA66', 'ZZCI98W', 'ZZB6Z688', 'ZZBACUF', 'ZZAF000E', 'ZW70W609', 'ZZB6UZ0', 'ZZB92C97', 'ZZBZB09I', 'ZZBCZ0W0', 'ZZC0UBAI', 'ZZB6CBAI',
'ZZB8C9BF', 'ZZBWE9AZ', 'ZZACB90D', 'ZZB6A9E8', 'ZZB8WEI', 'ZZB6CB6I', 'ZZB8008D', 'ZZC00UII', 'ZZD9WB8B', 'ZZB8W0C0', 'ZZA8A70B', 'ZZB800UA']

```

2.2 Getting songs' information

Everything gonna easy right now, based-on API link founded, I can use “encodeld” to request the json data.

In this task, parameter contain 3 elements, “ctime”, “id”, and “version” followed by an order.

```

https://zingmp3.vn/api/v2/page/get/song?id=ZZBUE7ZA&ctime=1652
534700&version=1.6.25&sig=fa906944628776791050d19321ce2ade1227
667cb1d032a77801e52461c62687fbcea9509e04c7f741f1c656ac0f562d2f
8d7fc5554767c3051332e27f92234a&apiKey=88265e23d4284f25963e6eed
ac8fbfa3

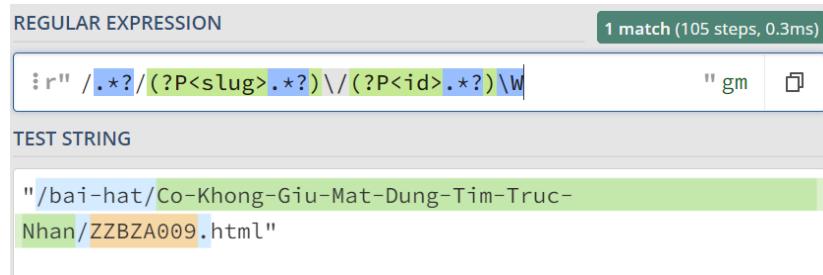
```

```

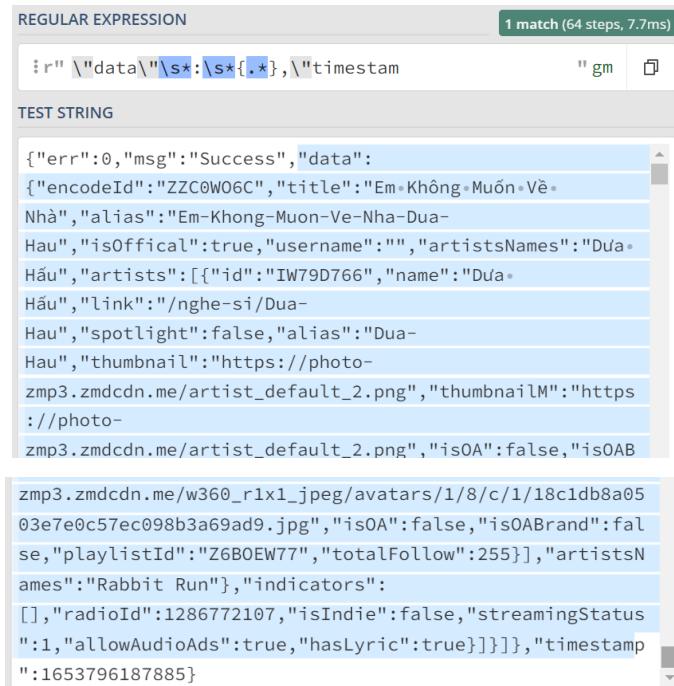
155 def getMediaFile(obj__id: str) -> None:
156     with eventlet.Timeout(10):
157         url = urlGeneration('/api/v2/page/get/song', {'ctime':_time,'id':obj__id,'version':'1.6.25'})
158         response = requests.get(url,headers=HEADERS)
159         checking(response)
160
161         #print(response.content.decode('utf-8'))
162         data = re.findall(r'\\"data\"\s*:\s*\{.*\},\\"timestamp", response.content.decode('utf-8'))[0][-10]
163         data = json.loads(data.split("\\"data\" : ")[1])
164
165         title = data["title"]

```

One more way, I can use regex to extract “encodeId” from song’s link.



Receive response and get worth data into json list.



There is an argument named “hasLyric”, if the value is true, I can collect lyric’s song.

```
    "artistsNames": "Trúc Nhân",
},
"indicators": [],
"isIndie": false,
"streamingStatus": 1,
"allowAudioAds": false,
"hasLyric": true,
"userid": 456,
"genres": [
{
  "id": "IWZ97FCD",
  "name": "V-Pop",
  "title": "V-Pop",
  "alias": "v-pop",
  "link": "/the-loai-album/V-Pop/IWZ97FCD.html"
},
```

Then, download the audio file by linux commandline.

```

155 def getMediaFile(obj__id: str) -> None:
156     with eventlet.Timeout(10):
157         url = urlGeneration('/api/v2/page/get/song', {'ctime': _time, 'id': obj__id, 'version': '1.6.25'})
158         response = requests.get(url, headers=HEADERS)
159         checking(response)
160
161         #print(response.content.decode('utf-8'))
162         data = re.findall(r'\"data\"\s*:\s*{.*},\\"timestamp\", response.content.decode('utf-8'))[0][-10:]
163         data = json.loads(data.split("\\"data\"":")[1])
164
165         title = data["title"]
166
167         lyric = ""
168         if "hasLyric" not in data:
169             print(f"\n[*] \\"{title}\\" does not has lyric!")
170         elif data["hasLyric"]:
171             print(f"\n[*] Download \\"{title}\\" lyric...")
172             lyric = downloadLyric(obj__id)
173         else:
174             print(f"\n[*] \\"{title}\\" does not has lyric!")
175
176         if "composers" in data: del data["composers"]
177         if "sections" in data: del data["sections"]
178         data.update({"lyric": lyric})
179
180         print(f'* Download \\"{title}\\" audio...')
181         streams = getStreamInfo(obj__id)
182         if streams == {}:
183             return
184         else:
185             data.update(streams)
186
187         for i in streams["streams"]:
188             for quality, link in i.items():
189                 downloadMedia(link, title, quality)
190
191         #print(json.dumps(data))
192         json_local.append(data)

```

2.3 Getting songs' lyric

API link:

<https://zingmp3.vn/api/v2/lyric/get/lyric?id=ZW6I979F&ctime=1652539551&version=1.6.25&sig=0c781dd12f369c3f062af7ee0fee8298c6c17edbb38ea87fac73cb989c239ccca94f4eb7fce6e80468b014380ea08a74c94b3f2170deb4756d54ffb5a81f1bc&apiKey=88265e23d4284f25963e6eedac8fbfa3>

Filter the response data:

```

[*] Download "Không Thể Yêu, Không Thể Quên" lyric...
b'{"err":0,"msg":"Success","data":{"sentences":[{"words":[{"startTime":34580,"endTime": "\xc4\x91\xc3\xaa3"}, {"startTime":35330,"endTime":35700,"data":"n\xc3\xb3i"}, {"startTime":36460,"data":"nhau"}, {"startTime":36460,"endTime":37210,"data":"r\xe1\xba\xaa5t"}, {"startTime":38330,"endTime":38720,"data":"Nh\xc6\xb0ng"}, {"startTime":38720,"endTime":39110,"data":"ai"}, {"startTime":39110,"endTime":39480,"data":"mu\xe1\xbb\x91n"}, {"startTime":40990,"endTime":42870,"data":"ngh"}, {"startTime":43260,"endTime":43260,"data":"l\xe1\xba\xbd"}, {"startTime":43260, "data": "cho"}, {"startTime":44380,"endTime":44750,"data": "ri\xc3\xaaang"}, {"startTime":46260,"endTime":46640,"data": "kh\xc3\xb4ng"}, {"startTime":47010,"endTime":47770,"data": "ph\xe1\xba\xaa3i"}, {"startTime":51130,"data": "ngh\xc4\xaa9"}]}, {"words": [{"startTime":51160,"endTime":51520,"data": "7n"}, {"startTime":51910,"endTime":52280,"data": "t\xe1\xbb\x8f"}, {"startTime":52280, "data": "nh"}, {"startTime":53030,"endTime":53400,"data": "kh\xc3\xb4ng"}, {"startTime":53400, "endTime":54910,"data": "Kh\xc3\xb4ng"}, {"startTime":54910,"endTime":55280,"data": "qu"}, {"startTime":56040,"endTime":56420,"data": "lo"}, {"startTime":56420,"endTime":59420,"data": "l\xe1\xbb\x83"}, {"startTime":59810,"endTime":60190,"data": "em"}, {"startTime":60190, "data": "v\xc6\xaa1"}, {"startTime":60950,"endTime":61320,"data": "ri\xc3\xaaang"}, {"startTime":62830,"endTime":63580,"data": "m\xc3\xacnh"}]}, {"words": [{"startTime":62830,"endTime":63580,"data": "\xc4\x91\xc3\xaa2u"}, {"startTime":64340, "data": "\xc4\x91\xc3\xaa"}, {"startTime":68100,"endTime":68890,"data": "Th\xc3\xb4i"}], {"words": [{"startTime":68100,"endTime":68890,"data": "Th\xc3\xb4i"}]}

```

REGULAR EXPRESSION

1 match (7 434 steps, 8.4ms)

```
:r" {\\"sentences\\":\\s*:\\s*[.*\\]}\\"]" gm
```

TEST STRING

```
{"err":0,"msg":"Success","data": {"sentences": [{"words": [{"startTIme":34580,"endTIme":34950,"data":"M\xc3\xacnh"}]}
```

```
{"startTIme":294150,"endTIme":296040,"data":"b\xc3\xb3ng"}, {"startTIme":296040,"endTIme":297040,"data":"h\xc3\xacnh"}]}],"file":"https://static-zmp3.zmdcdn.me/lyrics/f/4/0/3/f403a231e1ff745f0e2980ef2f6121ea.lrc","enabledVideoBG":true,"defaultIBGUrls": ["https://photo-resize-zmp3.zmdcdn.me/w1920_r3x2_jpeg/cover/f/2/8/f/f28fcfd62e10
```

There are 3 types of saved lyric:

- #### - First:

- #### - Second:

- ### - Third:

Get lyric function:

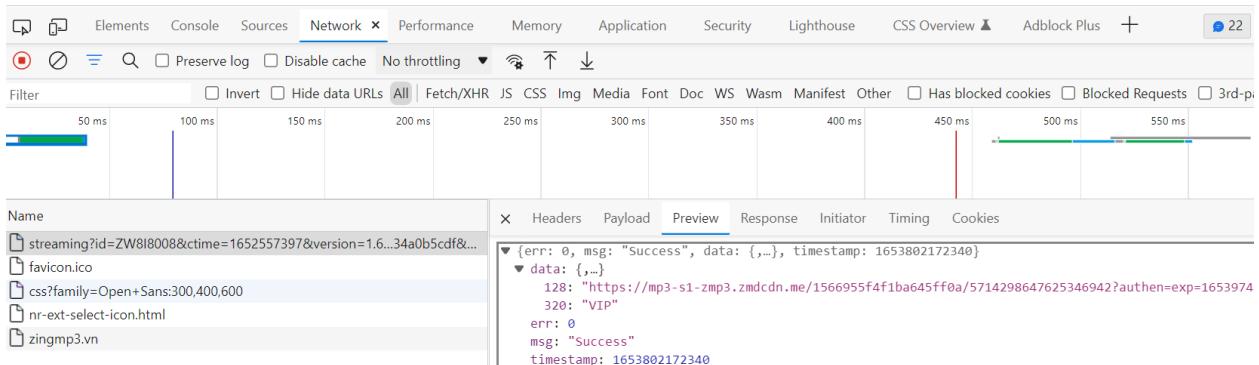
```
98  def downloadLyric(target_id: str) -> str:
99      with eventlet.Timeout(10):
100          url = urlGeneration('/api/v2/lyric/get/lyric', {'ctime':_time,'id':target_id,'version':'1.6.25'})
101          response = requests.get(url,headers=HEADERS)
102          checking(response)
103          #print(response.content)
104
105          lyric = ""
106
107          data = re.findall(r'\"sentences\"\s*:\s*\[.*\]\}\]", response.content.decode('utf-8'))
108          if data == []:
109              data = re.findall(r'\"lyric\"\s*:\s*\".*?\"", response.content.decode('utf-8'))
110              if data == []:
111                  data = re.findall(r'\"file\"\s*:\s*\".*?\"", response.content.decode('utf-8'))
112                  data = data[0].split("file\":\"")[-2]
113                  os.system(f"curl -s \"{data}\" -o temp")
114                  with open('temp', 'r', encoding='utf-8') as f:
115                      data = f.read()
116                      data = data.split('\n')
117                      for i in range(len(data)):
118                          if '][' in data[i]:
119                              data[i] = data[i].split(']')
120                              if data[i] != []:
121                                  for j in data[i]:
122                                      if j == '' or j[0] != '[': tmp = j; break
123                                      data[i] = tmp
124                                  else:
125                                      data[i] = ''
126
127                                  else:
128                                      data[i] = data[i].split(']')[1] if data[i].split(']') != [] else ''
129
130                      os.system(f"rm temp")
131                      lyric = '\n'.join(data)
132
133
134
135
136
137
138
return lyric
```

2.4 Getting songs' audio

This function, “getStreamInfo” will get all streams of one song, choose the one available to download.

```
80  def getStreamInfo(target_id: str) -> dict:
81      with eventlet.Timeout(10):
82          url = urlGeneration('/api/v2/song/get/streaming', {'ctime':_time,'id':target_id,'version':'1.6.25'})
83          response = requests.get(url,headers=HEADERS)
84          if b'"err":1150' in response.content:
85              print('*] Error : VIP requirement!')
86              return {}
87          checking(response)
88
89          data = re.findall(r'\"data\"\s*:\s*{.*}', response.content.decode('utf-8'))[0][-1]
90          data = (data.split('\"data\":')[1][1:-1]).split(',')
91
92          streams = {"streams" : []}
93          for i in data:
94              i = '{' + i + '}'
95              streams["streams"].append(json.loads(i))
96
return streams
```

```
{"err":0,"msg":"Success","data":{"128":"https://mp3-s1-zmp3.zmdcdn.me/1566955f4f1ba645ff0a/5714298647625346942?authen=exp=1653974811~acl=/1566955f4f1ba645ff0a/*~hmac=ae32974eaf14caa317eaa247fa52bf42&fs=MTY1MzgwmjAxMTmUsIC50Hx3ZWJWnwxMDYzNjmUsICyNTg5fDExNS43Ni4zOS4","timestamp":1653802172340}}
```



“downloadMedia” use linux commandline to download and save the audio file as {title}_[quality] in DOWNLOAD folder.

```

140  def downloadMedia(link, title, quality) -> None:
141      if link == "VIP":
142          return
143
144      DirDownload = os.path.join(os.getcwd(), "DOWNLOAD")
145      if not os.path.exists(DirDownload):
146          os.mkdir(DirDownload)
147
148      filename = f"{title}_{quality}"
149      if os.path.exists(f"{DirDownload}/{filename}"):
150          return
151
152      link = link.replace('https://', 'https://vnno-vn-5-tf-')
153      os.system(f"curl -s \"{link}\" -o \"{DirDownload}/{filename}\"")

```

2.5 Main function

Finally, combine all to one json element and add them to json object.
Elsewhere, check if “encodId” for existing songs.

```

194 def main() -> None:
195     # Default, 100 new songs
196     listOfSong = getTopList()
197     # Custom, get 100 songs of one playlist
198     #listOfSong = getTopList('/api/v2/page/get/playlist', {"id": "ZWZB96A9"})
199     #listOfSong = getTopList('/api/v2/page/get/playlist', {"id": "ZWZB96A0"})
200     print(f"[*] Number of song: {len(listOfSong)}\n{listOfSong}")
201
202     for x in listOfSong:
203         flag = 0
204         for i in json_local:
205             if x == i["encodeId"]:
206                 flag = 1
207                 break
208         if flag:
209             continue
210
211         getMediaFile(x)
212         #print(json_local)
213         #input()
214         with open("music_info.json", 'w', encoding='utf-8') as f:
215             f.write(json.dumps(json_local))
216
217         print('\n[*] Crawling done!')
218
219 main()

```

Otherwise, download the list of top 100 songs by getting a playlist id.

3 Music Player

Simply make html and js webpages, list information of song and response audio file.

3.1 webserver.py

Processing for each webpage.

3.2 music.html

Listing all the songs.

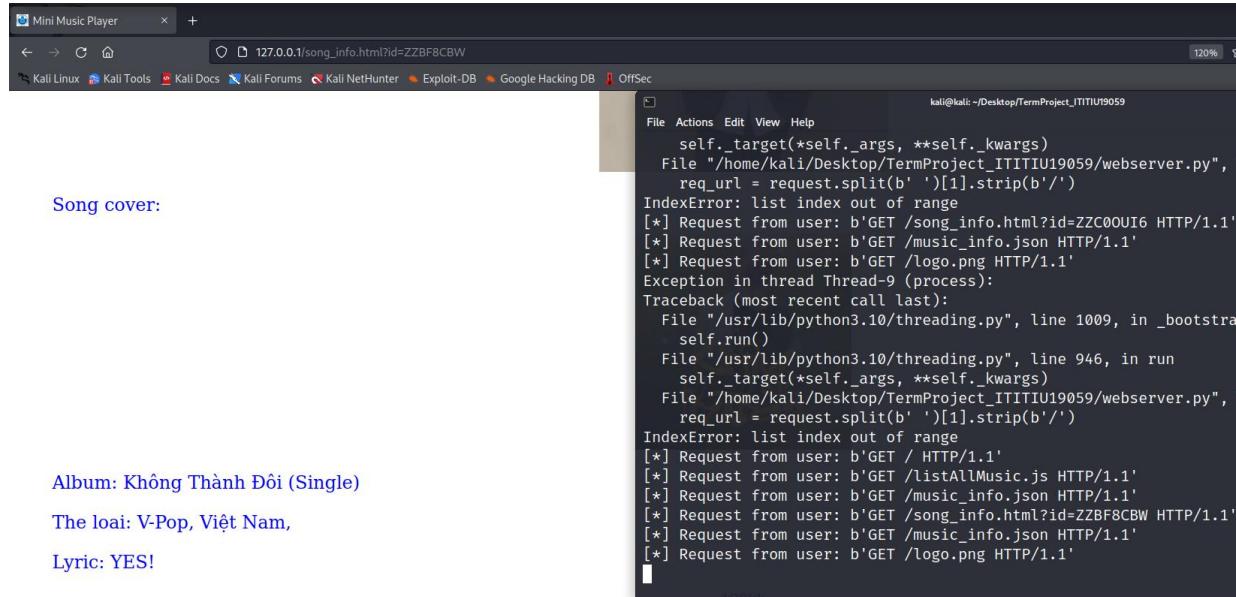
The terminal window shows the directory structure and a log of requests from a browser:

```
kali㉿kali:~/Desktop/TermProject_ITITIU19059
File Actions Edit View Help
└─(kali㉿kali)-[~/Desktop]
  $ cd TermProject_ITITIU19059
  └─(kali㉿kali)-[~/Desktop/TermProject_ITITIU19059]
    $ python webserver.py 127.0.0.1 80
    [*] Successfully bind on 127.0.0.1:80!
    [*] Successfully listen on 127.0.0.1:80!
    [*] Web successfully start on http://127.0.0.1:80
    [*] Request from user: b'GET / HTTP/1.1'
    [*] Request from user: b'GET /listAllMusic.js HTTP/1.1'
    [*] Request from user: b'GET /favicon.ico HTTP/1.1'
    [*] Request from user: b'GET /music_info.json HTTP/1.1'
```

3.3 song_info.html

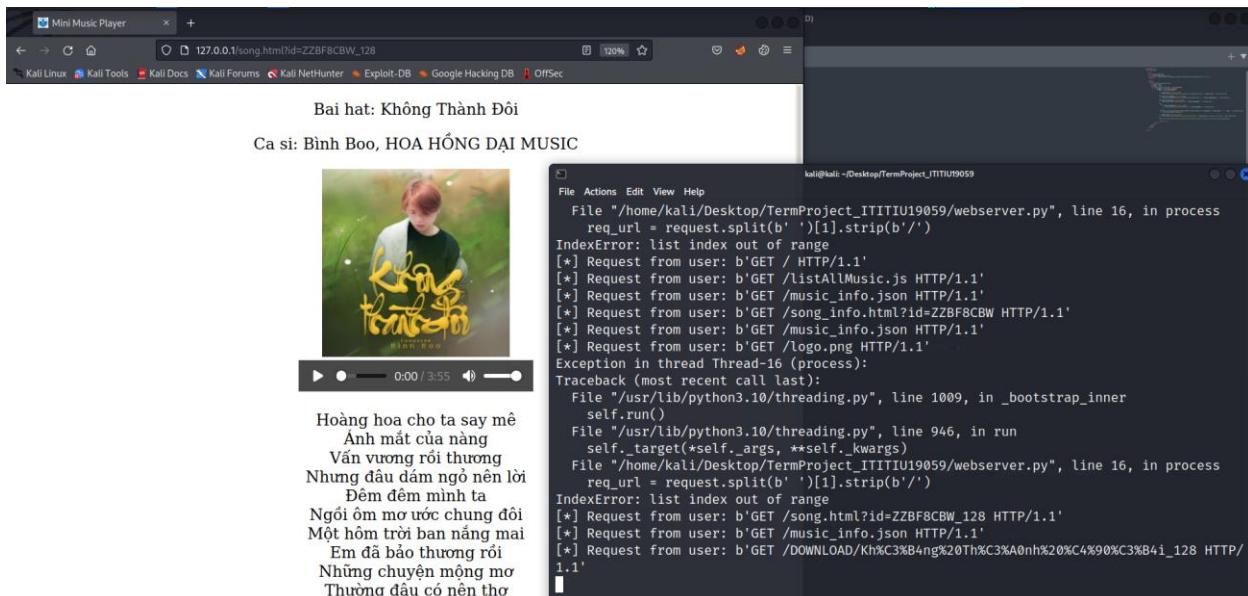
Getting main information of song, click spotify button for play.

```
song_info.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4     <head>
5         <meta charset="utf-8">
6         <title>Mini Music Player</title>
7         <script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
8     </head>
9
10    <body>
11        <ul></ul>
12        <script type="text/javascript">
13            let mID = "%b";
14            $.getJSON( "music_info.json", function(obj) {
15                $.each(obj, function(key, value) {
16                    if (mID == value.encodeId) {
17
18                        if (value.title != undefined) {
19                            $("ul").append('<p style="color:blue;font-size:20px">Bai hat: ' + value.title + '</p>');
20                        }
21                        if (value.artistsNames != undefined) {
22                            $("ul").append('<p style="color:blue;font-size:20px">Ca si: ' + value.artistsNames + '</p>');
23                        }
24                        if (Object(value.artists[0]).thumbnailM != undefined) {
25                            $("ul").append('<center></center>');
26                        }
27                        else
28                            if (Object(value.artists[0]).thumbnail != undefined) {
29                                $("ul").append('<center></center>');
30                            }
31                        if (value.thumbnailM != undefined) {
32                            $("ul").append('<p style="color:blue;font-size:20px">Song cover: <center></center>');
33                        }
34                        else
35                            if (value.thumbnail != undefined) {
36                                $("ul").append('<p style="color:blue;font-size:20px">Song cover: <center></center></p>');
37                            }
38                        if (value.album != undefined) {
39                            $("ul").append('<p style="color:blue;font-size:20px">Album: ' + Object(value.album).title + '</p>');
40                        }
41                    }
42                }
43            }
44        </script>
45    </body>
46</html>
```



3.4 song.html

Show play place and lyric.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Mini Music Player</title>
    <script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
</head>
<body>
<ul></ul>
<script type="text/javascript">
let mID = "%b";
let sKB = "%b";
$.getJSON("music_info.json", function(obj) {
    $.each(obj, function(key, value) {
        if (mID == value.encodeId) {
            if (value.title != undefined) {
                $("ul").append('<center><p style="font-size:20px">Bai hat: ' + value.title + '</p></center>');
            }
            if (value.artistsNames != undefined) {
                $("ul").append('<center><p style="font-size:20px">Ca si: ' + value.artistsNames + '</p></center>');
            }
            if (value.thumbnailM != undefined) {
                $("ul").append('<center></center>');
            }
            else {
                if (value.thumbnail != undefined) {
                    $("ul").append('<center></center>');
                }
            }
            $("ul").append('<center><audio width="500px" height="500px" src="DOWNLOAD/' + value.title + '_' + sKB + '" type="audio/mp3"></audio></center>');
            if (value.lyric != undefined) {
                $("ul").append('<center><p style="font-size:20px">' + value.lyric.replace(/\n/g, '<br>') + '</p></center>');
            }
            else $("ul").append('<center><p style="font-size:20px">Lyric: Missing! Do you have it?</p></center>');
        }
        return false;
    });
});
</script>
</body>
</html>
```

Bai hat: Không Thành Đôi
Ca si: Binh Boo, HOA HỒNG DẠI MUSIC



Hoàng hoa cho ta say mê
Ánh mắt của nàng
Vẫn vướng rối thương
Nhưng đâu dám ngó nên lời
Đêm đêm mình ta
Ngồi ôm mợ ước chung đôi
Một hôm trời ban nắng mai
Em đã bảo thương rồi
Những chuyện mong mơ
Thường đâu có nên thơ

```
File "/home/kali/Desktop/TermProject_ITITIU19059/webserver.py", line 16, in process
    req_url = request.split(b' ')[1].strip(b'/')
IndexError: list index out of range
[*] Request from user: b'GET / HTTP/1.1'
[*] Request from user: b'GET /listAllMusic.js HTTP/1.1'
[*] Request from user: b'GET /music_info.json HTTP/1.1'
[*] Request from user: b'GET /song_info.html?id=ZZBF8CBW_128 HTTP/1.1'
[*] Request from user: b'GET /music_info.json HTTP/1.1'
[*] Request from user: b'GET /logo.png HTTP/1.1'
Exception in thread Thread-16 (process):
Traceback (most recent call last):
  File "/usr/lib/python3.10/threading.py", line 1009, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.10/threading.py", line 946, in run
    self._target(*self._args, **self._kwargs)
  File "/home/kali/Desktop/TermProject_ITITIU19059/webserver.py", line 16, in process
    req_url = request.split(b' ')[1].strip(b'/')
IndexError: list index out of range
[*] Request from user: b'GET /song.html?id=ZZBF8CBW_128 HTTP/1.1'
[*] Request from user: b'GET /music_info.json HTTP/1.1'
[*] Request from user: b'GET /DOWNLOAD/Kh%C3%B4ng%20Th%C3%A0nh%20D%C3%A1i%20H%C3%A1o%20Boo%20H%C3%A0o%20H%C3%A0ng%20D%C3%A1i%20Music HTTP/1.1'
```

4 Usage

- musiccrawler.py
 - o Simply run it with command

```
python3 musiccrawler.py
```
 - o If you want to collect a playlist, following part 2.5
- webserver.py
 - o This script receive 2 arguments, <ip> and <port>.

```
python3 webserver.py 127.0.0.1 80
```

5 Conclusion

This project gives me more experience and a sense of working with web applications, changing my mind that web is the simplest part of programming.

Handling json data, know how to work and choose between multiple types of custom data or information.

6 Referrer

zingmp3 crawler on previous zingmp3's version:

<https://github.com/hatienl0i261299/Zingmp3>