

Tiny YouTube Downloader

免費開源不限速完整畫質媒體下載器

介紹

用途顧名思義就是用來下載 YouTube 媒體，雖然下載器在網路上的種類繁多，但總會有些限制及短處亦或者是安全性問題，例如：「付費解鎖完整畫質、實際檔案被壓縮過 或 轉檔錯誤或是速率限制」。我使用下載器的目的是為了在不額外付費的前提下，能夠在通勤時或沒有網路的時候可以觀看 YouTube，在這些前提下我開始開發能夠安全穩定且沒有次數限制的 YouTube 下載工具。

影片資訊擷取

點擊影片網址後，YouTube 會提供當前影片的資訊，像是標題、影像、音訊、介紹、觀看次數與發布日期等等...，為了不讓使用者輕易的取得影像、音訊，所以這些資訊都會需要 signature 才能取得，而且 signature 會被特定的方法加密，需要解開才能取得影像資訊。另外音訊是獨立於影像外的，這就是為什麼有時候網路品質差時，會有影像與音訊不同步的情況發生。

YouTube 在 2021 年二度修改了影片資訊的取得方式，目前資訊的取得方式已經變得非常簡單，可以直接從 HTML 中取得，但如需下載影片，還要「解密」才能達成。

```
1332 <body dir="ltr" no-y-overflow>
1333 <script nonce="8MZW8qu/sVf5IF8D0FQS1w">
1334   var ytInitialPlayerResponse = {
1335     "responseContext": {
1336       "playabilityStatus": {
1337         "streamingData": {
1338           "expiresInSeconds": "21540",
1339           "formats": [{
1340             "itag": 18,
1341             "mimeType": "video/mp4; codecs=\"avc1.42001E, mp4a.40.2\"",
1342             "bitrate": 630994,
1343             "width": 640,
1344             "height": 360,
1345             "lastModified": "1624987724234745",
1346             "contentLength": "16724503",
1347             "quality": "medium",
1348             "fps": 25,
1349             "qualityLabel": "360p",
1350             "projectionType": "RECTANGULAR",
1351             "averageBitrate": 630842,
1352             "audioQuality": "AUDIO_QUALITY_LOW",
1353             "approxDurationMs": "212091",
1354             "audioSampleRate": "44100",
1355             "audioChannels": 2,
1356             "signatureCipher": "s=V%3DQK%3DQADdZLMszIMX5YkNDIvJWboLg64KrcU95lHgotycMgI
```

▲ 影片資訊，紅框是需解密的簽名

原始媒體網址解密

```
"itag": 135,  
"url": "https://r1---sn-8pcv-u2x6.go  
"mimeType": "video/mp4; codecs=\"avc  
"bitrate": 635273,  
"width": 480,  
"height": 480,  
"initRange": {  
"indexRange": {  
"lastModified": "1623497373266928",  
"contentLength": "3439823",  
"quality": "large",  
"fps": 30,  
"qualityLabel": "480p",  
"projectionType": "RECTANGULAR",  
"averageBitrate": 604365,  
"approxDurationMs": "45533"
```

▲ 未被加密，可直接取得 url

```
"itag": 137,  
"mimeType": "video/mp4; codecs=\"avc  
"bitrate": 4690450,  
"width": 1920,  
"height": 1080,  
"initRange": {  
"indexRange": {  
"lastModified": "1628130818054956",  
"contentLength": "93990963",  
"quality": "hd1080",  
"fps": 25,  
"qualityLabel": "1080p",  
"projectionType": "RECTANGULAR",  
"averageBitrate": 3546159,  
"approxDurationMs": "212040",  
"signatureCipher": "s=%3D%3DY2%3DY2q
```

▲ 需透過 signatureCipher 解密

1. 步驟 1：取得簽名與網址

將 signatureCipher 依照&拆開，可以得到三筆資料

s: ==AG3C_xAw...

sp: sig

url: https://r4---sn-ab5sxn7e.googlevideo.com/videoplayback...

得到的資料，分別是 s(signature 簽名)、sp(signature parameter 簽名參數)、url(還沒加入簽名的網址)。

2. 步驟 2：尋找解密簽名的步驟

因為簽名的步驟跟代號都是隨機的，所以需要取得 YouTube 網頁中將影音資料解密的程式，它叫 base.js，其中包含解密的步驟以及步驟代號。

Youtube 解密的代號三種，可將 signature 簽名解密：

1. 交換：將第 0 個字元跟第 n 個字元交換
2. 反轉：將所有字元反轉(倒過來)
3. 切除：去掉最前面的 n 個字元

```
6717 yb.prototype.getVariableAudioTracks=function(){return this.audioTracks};  
6718 var NB={  
6719   N2:function(a,b){var c=a[0];a[0]=a[b%a.length];a[b%a.length]=c},  
6720   Vn:function(a){a.reverse()},  
6721   nl:function(a,b){a.splice(0,b)}  
6722 };  
6723
```

這三個步驟會對應不同的代號，以上圖舉例，N2 是交換；Vn 是反轉；nl 是切除。

```
1757 OB=function(a){a=a.split("");NB.nl(a,3);NB.Vn(a,54);NB.nl(a,3);return a.join("")};
```

接著找到解密步驟就可以依照步驟代號將簽名解密。

3. 步驟 3：將簽名解密

```
1757 OB=function(a){a=a.split("");NB.nl(a,3);NB.Vn(a,54);NB.nl(a,3);return a.join("")};
```

解密步驟如下

1. `split("")`先將字串轉成一個一個的字元
2. 依步驟 `nl`→`Vn`→`nl`，**切除**前三個字元→**反轉**→**切除**前三個字元
3. `join("")`把字元組回字串，這樣就解密完成了

4. 步驟 4：將簽名與網址組合

將 url 跟簽名組合起來，需要將網址加上簽名

url(網址)：`https://r4---sn-ab5szn7e.googlevideo.com/videoplayback...`

sp(簽名參數)：`sig`

解密好的簽名字串：`AG3C_xAwRQIhAKRd2QqpcA...`

首先在網址後面加上`&`代表新增參數，接著加上 `sig`(就是簽名參數的值)代表參數的名字，再加上`=`以及解密好的簽名字串就完成了，最終網址如下圖：

```
https://r4---sn-8pcv-u2xz.googlevideo.com/videoplayback?expire=1629455729&ei=ETBfyd78BYfi4wKkpaX4Dg
ip=180.217.242.207&id=o-AK-EXSQeoHRGY2ebOK2wflSJ0xdHdKUqGrKboNwzpgjg&itag=137&
aitags=133,134,135,136,137,160,242,243,244,247,278,394,395,396,397,398,399&source=youtube&
requiressl=yes&mh=7c&mm=31,29&mn=sn-8pcv-u2xz,sn-un57en7e&ms=au,rdu&mv=m&mvi=4&pl=21&
initcwndbps=477500&vprv=1&mime=video/mp4&ns=BxYChGu9zMzN2-WtQ__bYjkG&gir=yes&clen=93990963&
dur=212.040&lmt=1628130818054956&mt=1629433757&fvip=4&keepalive=yes&fexp=24001373,24007246&
c=WEB&txp=5535434&n=FUBGmlMjLEcjib-1t&
sparams=expire,ei,ip,id,aitags,source,requiressl,vprv,mime,ns,gir,clen,dur,lmt&
lsparams=mh,mm,mn,ms,mv,mvi,pl,initcwndbps&
lsig=AG3C_xAwRQIqDqErFTkQEapQt3LvSoUTKKuRJPo2RAU0QUCe5a8og0MCIQc8GaRnolGZV0lv7E8ILvKWRYzXKvKUj1I5tmJU13HUgw==&
sig=AOq0QJ8wRQIgENw13RKZCOMW3zGr9pNzkFy4dRBw8Snnz5g-Aq3-Ka4CIQCfU4Crqwj04g_pKt7uw9Rn7Ug_KWRlm8s2fBa79u8NA==
```

最底下那行可以看到解密的簽名

使用正則表達式解密

正則表達式能依規則尋找相符的字串，雖然內容可能不太一樣，但符合規則就可以被找到。

若要尋找解密步驟，先尋找下圖範例中標識的關鍵字：

```
1757 OB=function(a){a=a.split("");NB.nl(a,3);NB.Vn(a,54);NB.nl(a,3);return a.join("")};
```

找到這些關鍵字後就能撰寫正則表達式，正則表達式雖然看起來複雜，但其實沒有很難，常用的參數如下：

- **\w** 代表任意英文字元
- **.** 代表除了換行符號的任意字元，在[]中沒有作用，視為一般字元
- **[]** 代表中括號內的任意字元
- **+** 代表前面的字元需一個以上
- ***** 代表前面的字元需要零個以上
- **()** 代表要選取的範圍
- **** 跳脫字元，如果要查詢的字元是特殊字元，像是我們要查找 "(" 前面必須加上 **\"** 變成 **\"** "("，代表它是括號而不是要選取的範圍。

```
function\(\w+\)\{[\w=.,(\")]+\;([\w=.,(\")];)*\};return \w+\.join[(\")]*\};
```

這個正則表達式是用於搜尋解密步驟的，把剛剛找到的固定特徵寫進去，像

"function("、"){"、"return"、".join"還有"};"，其他不確定的用 **\w** 或是 **[]** 來代替，

就可以找到字串並取得在括號裡的解密步驟，也就是 **"NB.nl(a,3);NB.Vm(a,54)**

NB.nl(a,3);"，接下來要找的是代號對應的方法就可以完成解密，在未整理的情況下三種方法的原始碼長這樣：

```
AB.prototype.getAvailableAudioTracks=function(){return this.audioTracks};var OB={QQ:function(a,b){a.splice(0,b)},z6:function(a){a.reverse()},yU:function(a,b){var c=a[0];a[0]=a[b%a.length];a[b%a.length]=c}};g.PB.prototype.set=function(a,b){this.i[a]!==b&&
```

是有換行的，搜尋的方法是：

```
\w+ NB=\{(.*\n*.*\n*.*)\};
```

接著把換行去掉後就可以得到代號對應的解密方法：

```
6717 yb.prototype.getAvailableAudioTracks=function(){return this.audioTracks};
6718 var NB={
6719     N2:function(a,b){var c=a[0];a[0]=a[b%a.length];a[b%a.length]=c},
6720     Vn:function(a){a.reverse()},
6721     nl:function(a,b){a.splice(0,b)}
6722 };
6723 PB.prototype.set=function(a,b){this.i[a]!==b&&
```

N2 是交換；Vn 是反轉；nl 是切除。

下載器

下載器使 Java 撰寫，依照上述原理來下載影片。

程式碼連結 <https://github.com/WavJaby/TinyYoutubeDownloader>