

تعدادی از ویدیوهای یوتیوب، خصوصا مواردی که از Google videos به یوتیوب منتقل شده‌اند، دارای زیر نویس هستند. این زیر نویس هم فرمت جالبی دارد:

```
<?xml version="1.0" encoding="UTF-8"?>
<transcript>
<text start="23.49" dur="5.97">
    &gt;&gt;&gt;Commentator: We have here today, Steve with
    us whom I met not too long ago at something
</text>
...
</transcript>
```

و از هر لحاظ بهتر است از فرمت srt متداول (خصوصا از لحاظ قابلیت بهتر parse آن) ولی ... این فرمت xml ای، متداول نیست و بهتر است جهت استفاده از آن در برنامه‌های پخش کننده فایل‌های ویدیویی، به srt تبدیل شود. ابتدا شاید بپرسید که این‌ها را چگونه می‌توان دریافت کرد. روش کار به صورت زیر است:

<http://video.google.com/timedtext?hl=en&v=Id&lang=en>

در قسمت‌های hl و lang ، زبان مورد نظر قرار می‌گیرد و قسمت v همان id معروف ویدیوی یوتیوب مورد نظر است. برای مثال اگر لینک ویدیوی ما <http://www.youtube.com/watch?v=-PA-buwI3q4> باشد، لینک زیرنویس انگلیسی هماهنگ با آن <http://video.google.com/timedtext?hl=en&v=-PA-buwI3q4&lang=en> خواهد بود.

البته زمان‌های این فایل xml یک نکته‌ی خاص گوگلی دارد!

اگر عدد 3.4 را مشاهده کردید یعنی سه ثانیه و 400 میلی ثانیه

اگر عدد 3.04 را مشاهده کردید یعنی سه ثانیه و 40 میلی ثانیه

به عبارتی با قسمت اعشاری آن باید به صورت رشته برخورد کنید. اگر طول آن یک بود، دو صفر و اگر طول آن 2 بود، 1 صفر در جلوی آن باید قرار گیرد تا قابل استفاده شود.

کلاس تهیه شده زیر کار تبدیل Xml گوگل را به Srt انجام می‌دهد:

```
using System;
using System.Linq;
using System.Text;
using System.Web;
using System.Xml.Linq;

namespace YtSubs
{
    public class GoogleXmlToSrt
    {
        public string ConvertXmlToSrt(string url)
        {
            var transcript = loadTranscript(url);

            var srtBuf = new StringBuilder();
            var lineNumber = 0;

            foreach (var item in transcript.Root.Elements("text"))
            {
                lineNumber++;
                srtBuf.Append(lineNumber.ToString());

                var timeLine = getTimeLine(item);
```

```

        srtBuf.AppendFormat("{0}{1} --> {2}{3}", Environment.NewLine, timeline.Item1,
        timeline.Item2, Environment.NewLine);

        var msg = decodeText(item);
        srtBuf.AppendLine(msg + Environment.NewLine);
    }
    return srtBuf.ToString();
}

private static XDocument loadTranscript(string url)
{
    var transcript = XDocument.Load(url);

    if (transcript.Root == null)
        throw new InvalidOperationException("couldn't load received data.");
    return transcript;
}

private static string decodeText(XElement text)
{
    var textValue = text.Value.Split('\n');
    var msg = textValue.Aggregate(
        string.Empty, (current, line) =>
            current + (HttpUtility.HtmlDecode(line.Trim()) + Environment.NewLine)
    );
    return msg.Trim();
}

private static int fractionalPart(string data)
{
    var idx = data.IndexOf(".");
    if (idx == -1) return 0;
    var fractionalPart = data.Substring(idx + 1);
    if (fractionalPart.Length == 1) //3.4 --> 3.400
        return int.Parse(fractionalPart + "00");

    if (fractionalPart.Length == 2) //3.04 --> 3.040
        return int.Parse(fractionalPart + "0");

    return int.Parse(fractionalPart.Substring(0, 3));
}

private Tuple<string, string> getTimeLine(XElement text)
{
    var startTs = getStartTs(text);
    TimeSpan endTs = getEndTs(text, startTs);
    return new Tuple<string, string>(timeSpanToString(startTs), timeSpanToString(endTs));
}

private static TimeSpan getStartTs(XElement text)
{
    var startData = text.Attribute("start");
    if (startData == null)
        throw new InvalidOperationException("This is not a valid subtitle file.");

    var start = startData.Value;
    var startTs = new TimeSpan(0, 0, 0, (int)Math.Truncate(double.Parse(start)),
fractionalPart(start));
    return startTs;
}

private static TimeSpan getEndTs(XElement text, TimeSpan startTs)
{
    TimeSpan endTs;
    var durData = text.Attribute("dur");
    if (durData == null)
    {
        endTs = startTs + new TimeSpan(0, 0, 0, 3, 0);
    }
    else
    {
        var dur = durData.Value;
        var durTs = new TimeSpan(0, 0, 0, (int)Math.Truncate(double.Parse(dur)),
fractionalPart(dur));
        endTs = startTs + durTs;
    }
    return endTs;
}

private static string timeSpanToString(TimeSpan lineTs)
{

```

```
        return string.Format("{0}:{1}:{2},{3}",  
                                lineTs.Hours.ToString("D2"),  
                                lineTs.Minutes.ToString("D2"),  
                                lineTs.Seconds.ToString("D2"),  
                                lineTs.Milliseconds.ToString("D3"));  
    }  
}
```

و یک نمونه از استفاده آن می‌تواند به شکل زیر باشد:

```
using System.IO;  
  
namespace YtSubs  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var url = "http://video.google.com/timedtext?hl=en&v=-PA-buwI3q4&lang=en";  
            //var url = "subtitle.xml";  
            var srtBuf = new GoogleXmlToSrt().ConvertXmlToSrt(url);  
            File.WriteAllText("-PA-buwI3q4.srt", srtBuf);  
        }  
    }  
}
```

مطالب تکمیلی:

[همه چیز درباره زیرنویس ویدئوهای گوگل](#)

## نظرات خوانندگان

نویسنده: Sniper\_528  
تاریخ: ۰۲:۳۶:۲۴ ۱۳۹۰/۰۶/۰۱

اکثر زیرنویسهای فارسی که تو اینترنت هست واسه دانلود، UTF8 نیستن و کاراکترها ناخوانا هستن. شما برای این مشکل چه راه حلی دارید؟

نویسنده: وحید نصیری  
تاریخ: ۰۸:۱۳:۵۵ ۱۳۹۰/۰۶/۰۱

<http://www.dotnettips.info/2010/12/subtitle-tools.html>

نویسنده: نیما  
تاریخ: ۱۱:۵ ۱۳۹۲/۰۹/۲۵

با سلام؛ ممنون از مطلب مفیدتون. میخوام عرض کنم فرمتی که شما فرمودید فرق کرده و دیگر attribute به نام dur نداره و به همین دلیل تبدیل کننده ای که زحمتش رو کشیدید با ارور مواجه میشه. موفق باشید.

نویسنده: وحید نصیری  
تاریخ: ۱۱:۳۴ ۱۳۹۲/۰۹/۲۵

چیزی عوض نشده. الان سه مورد را بررسی کردم مانند قبل بود: ^ و ^ و ^