制作一个在线音乐播放器。

首先身为一个在线音乐播放器，需要前端和数据库的搭配使用。

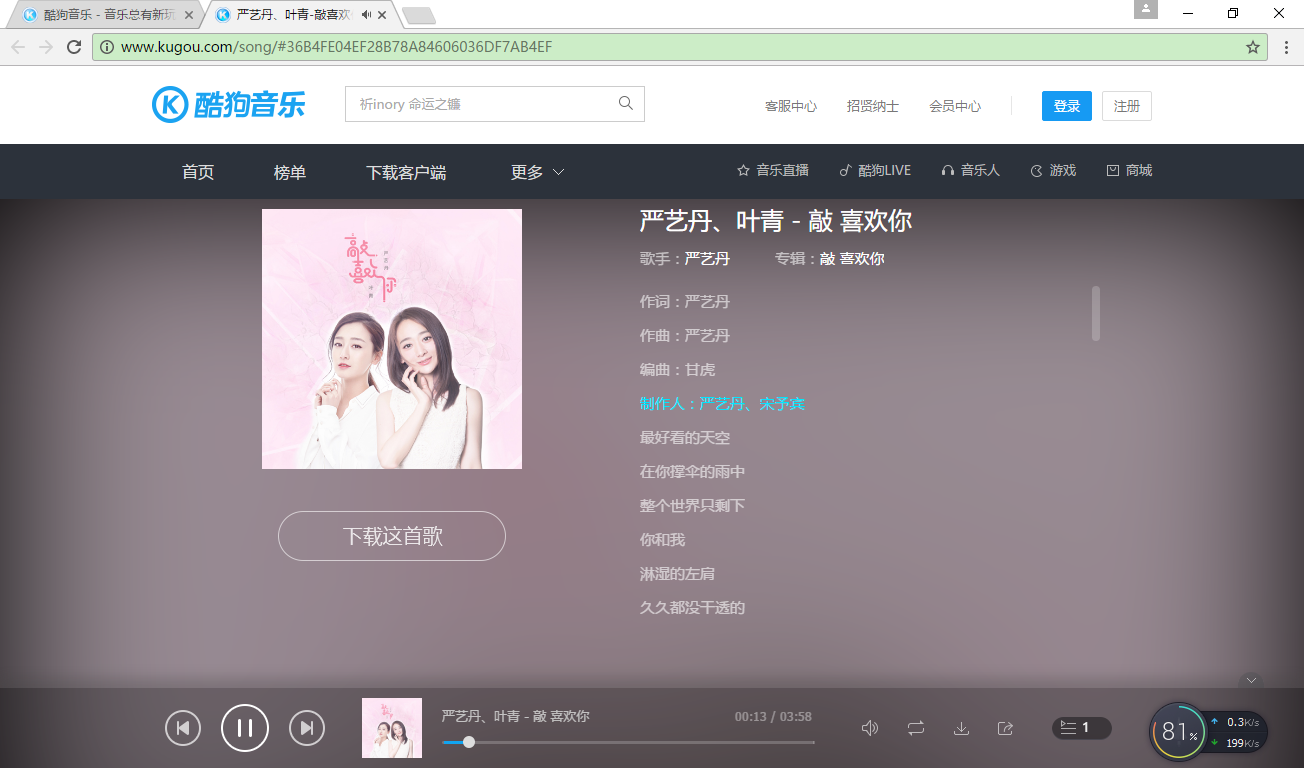
在数据库方面，我们没有办法制作，首先是版权问题，再加上数据量。所以我们需要借用其他网络播放器的数据库。

但是这些在线播放器，如百度，酷狗，酷我等在线音乐播放器，是不会提供他们的数据库接口的，所以这个我们需要，在线抓取这些在线音乐播放器的接口，首先，这个事情，是属于违规的。是属于侵犯这些公司的版权的。所以，不能用于商业用途

我们以酷狗音乐为例子，抓取他的连接数据库的API接口

我是用的是谷歌，进入私密模式，因为在这种模式下，不会缓存歌曲

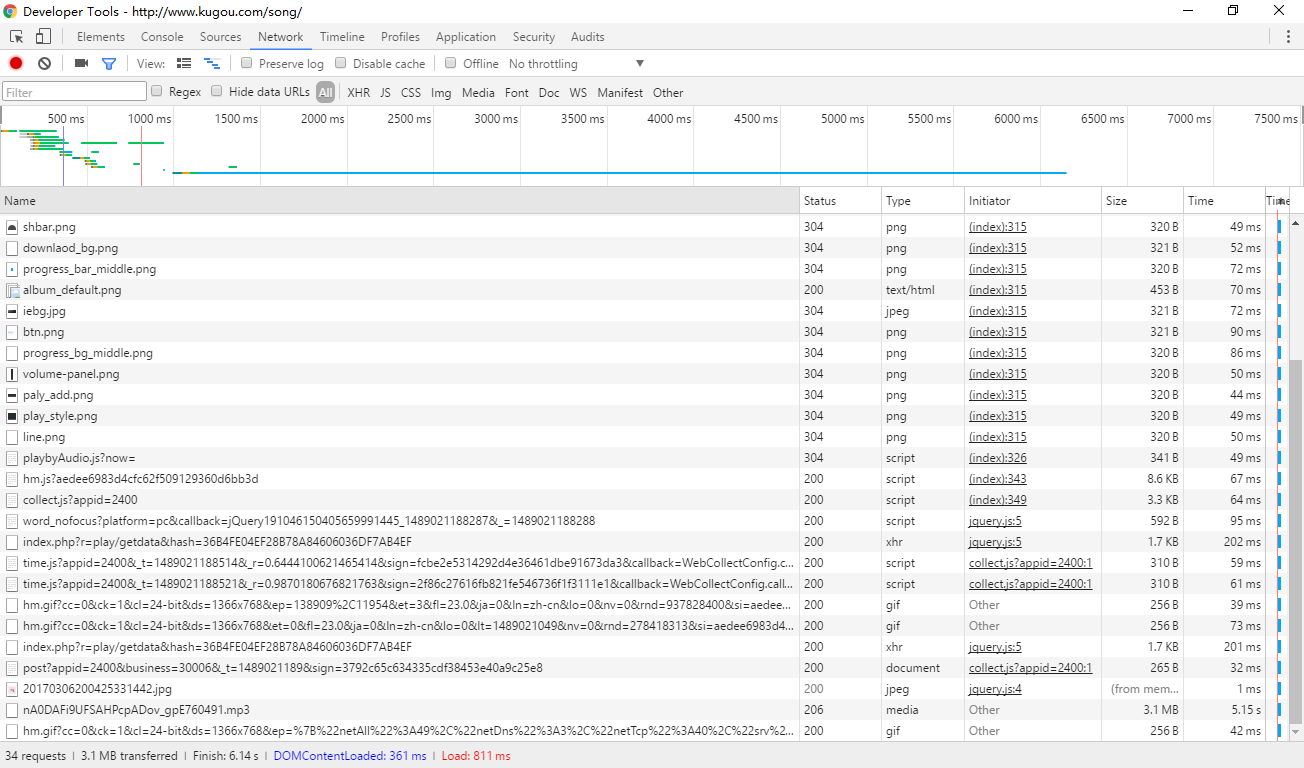
1. 进入酷狗的官网 [www.kugou.com](http://www.kugou.com) 然后随便打开一首歌曲



会出现上面的窗口

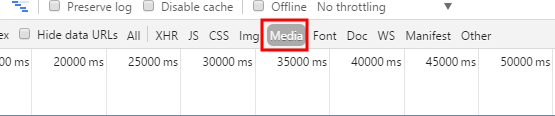
F12打开调试者工具

刷新页面，然后记录network中的页面传输信息



然后我们会发现，在记录中，有一个数据是加载时间最长的。这个数据，就是页面中数据量最大的，也就是歌曲

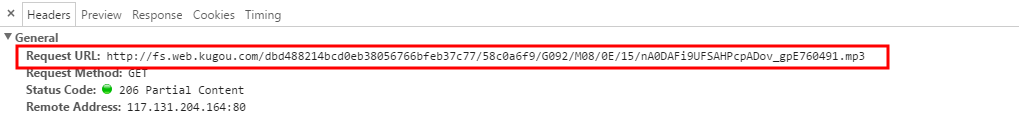
筛选数据类型 多媒体



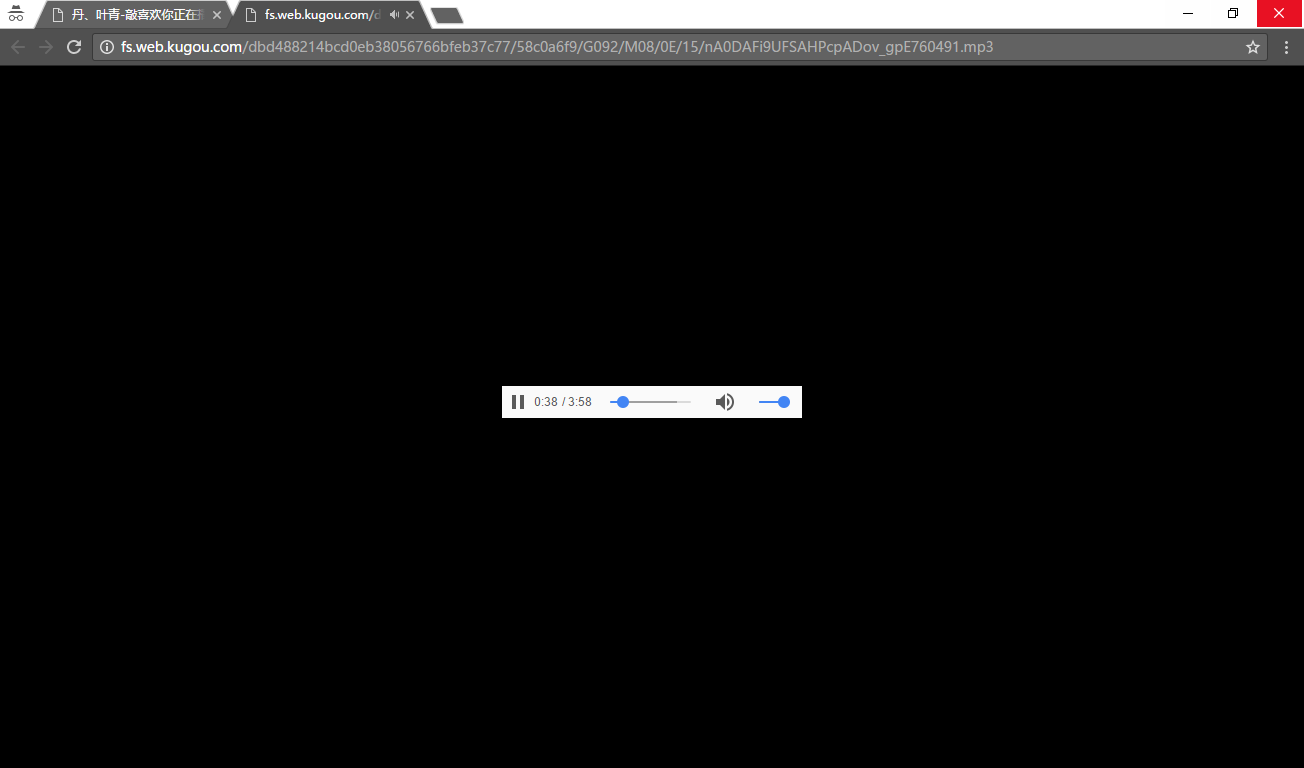
然后就发现列表中的音乐



查看音乐头信息，就会发现歌曲的地址，

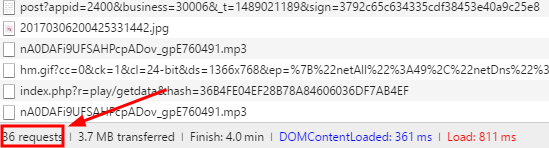


我们在新窗口中打开这个地址

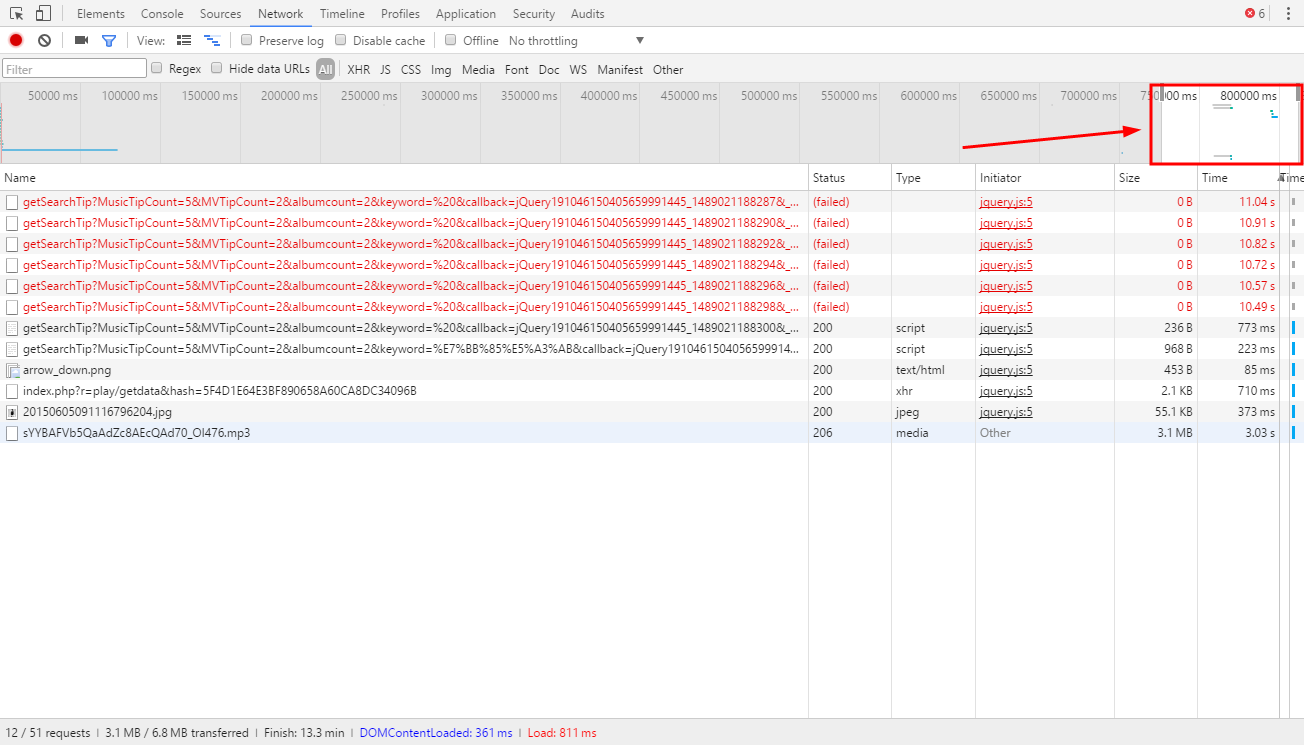


这就会是一首歌曲

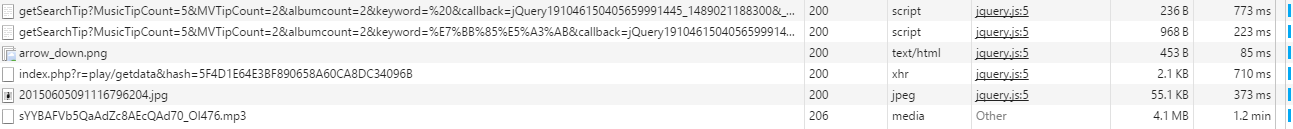
查看全部信息可以发现共有36个信息被记录



这时候我们，可以在查询一首歌，并添加到列表中

这时候我们查看开发者工具中，会最新加载的数据

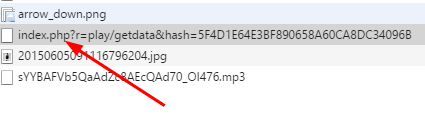
查看这些数据会发现除了歌曲外，还会加载一些文本



查看这些文本的头部，发现，里面会传递一个参数callback。这是回调函数，用来进行json传递的，是传递数据的常用方式。但是这种方式通常是不安全的，所以说，歌曲下载地址一般不是在这里的



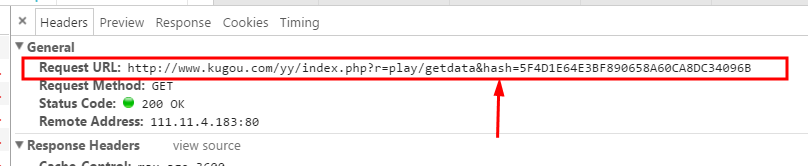
然后我们会找到一个传递到php页面的

、

查看内容



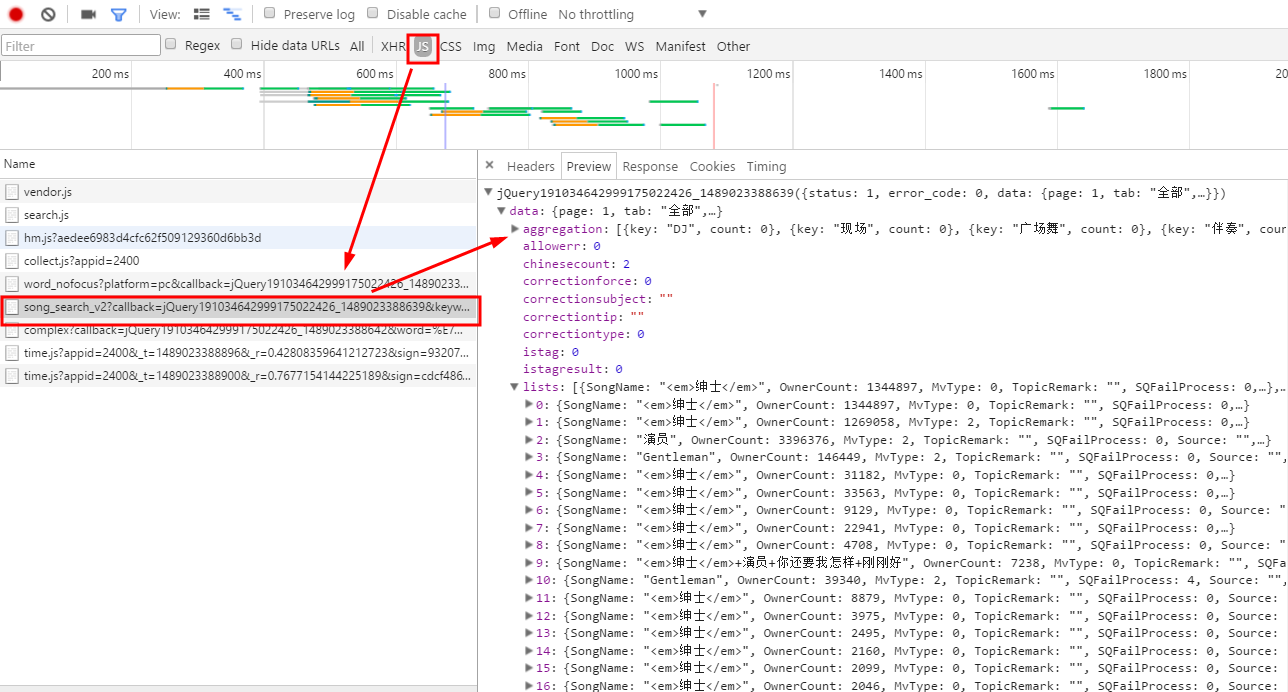
**这里我们会发现有歌词地址，和歌曲名称，歌词，还有一个hash值，查看请求地址.**



**这里我们可以看到，网页向数据库发送了一个hash的数据。我们从中知道，这个hash值是歌曲在数据库唯一编码**

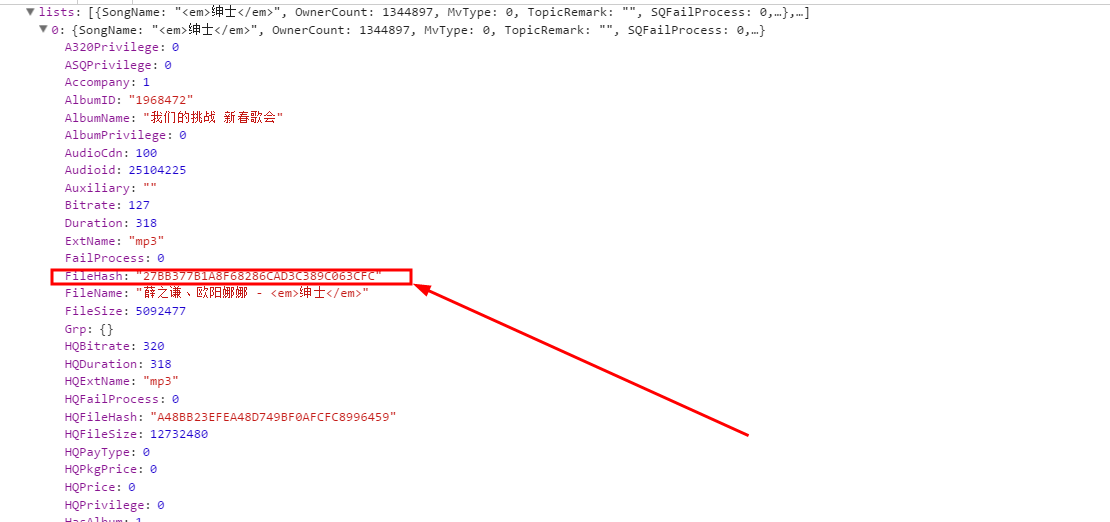
**接下来，我们只需要知道歌曲的hash值，便可以找到歌曲地址了。**

**再次打开搜索列表，用开发者工具拦截数据，筛选js，会找到其中这样的数据**



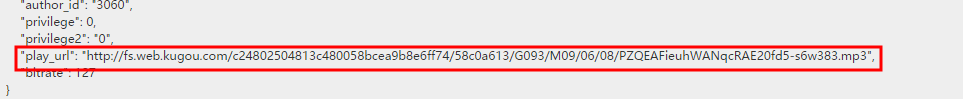
**随便打开其中一个列表我们会发现里面有一个filehash值，复制这段值**

**并且我们替换到之前查找歌曲时候，查找到的php文件，并替换那个hash值**



**这时候页面会返回一个json文件**

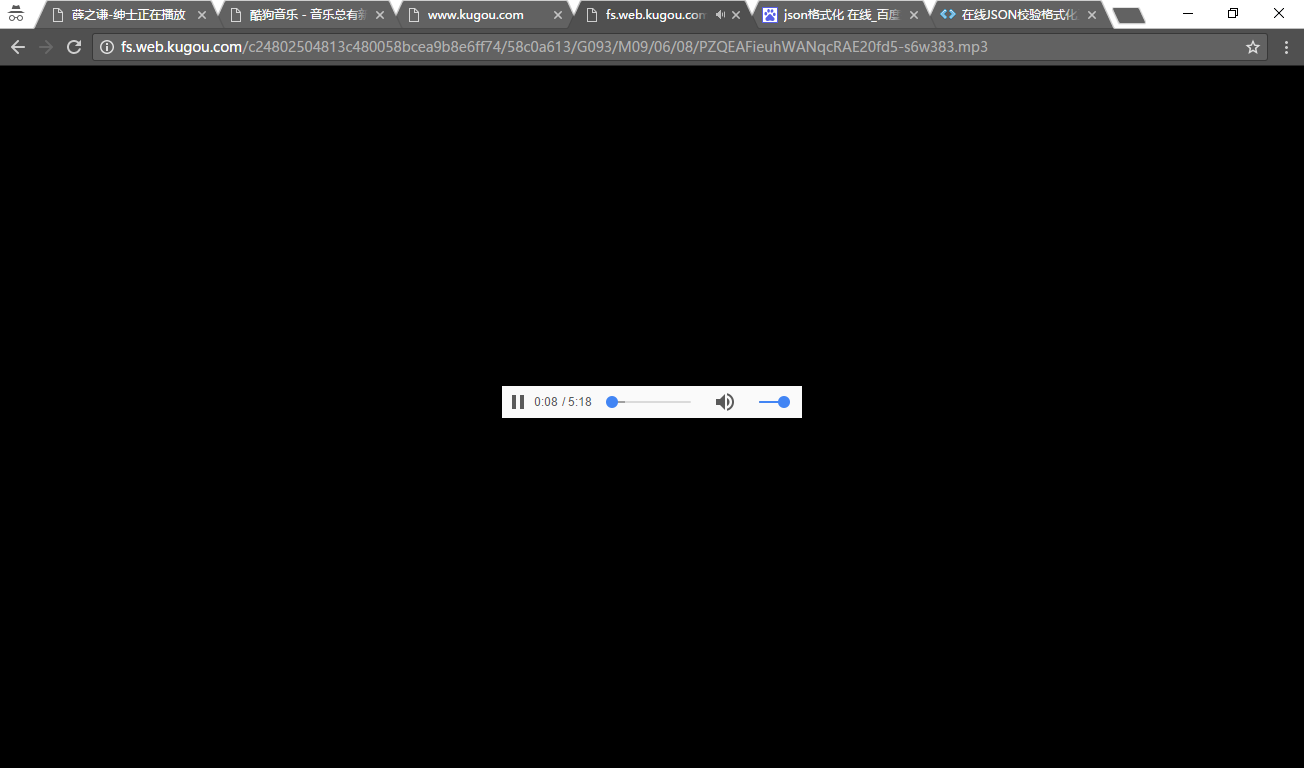
**我们可以通过在线json解析器，来格式化这个json**



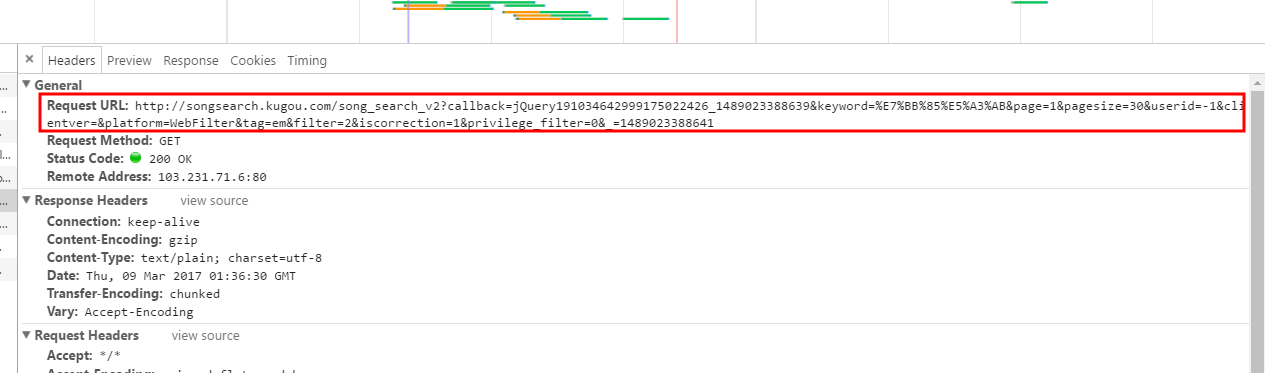
**这里面有个play\_url的参数，里面的值是一个地址**

**我们可以在新窗口打开这个地址**

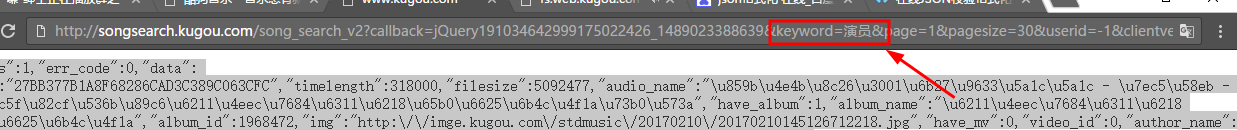
**这里我们发现打开的一个熟悉的窗口，一个MP3，**



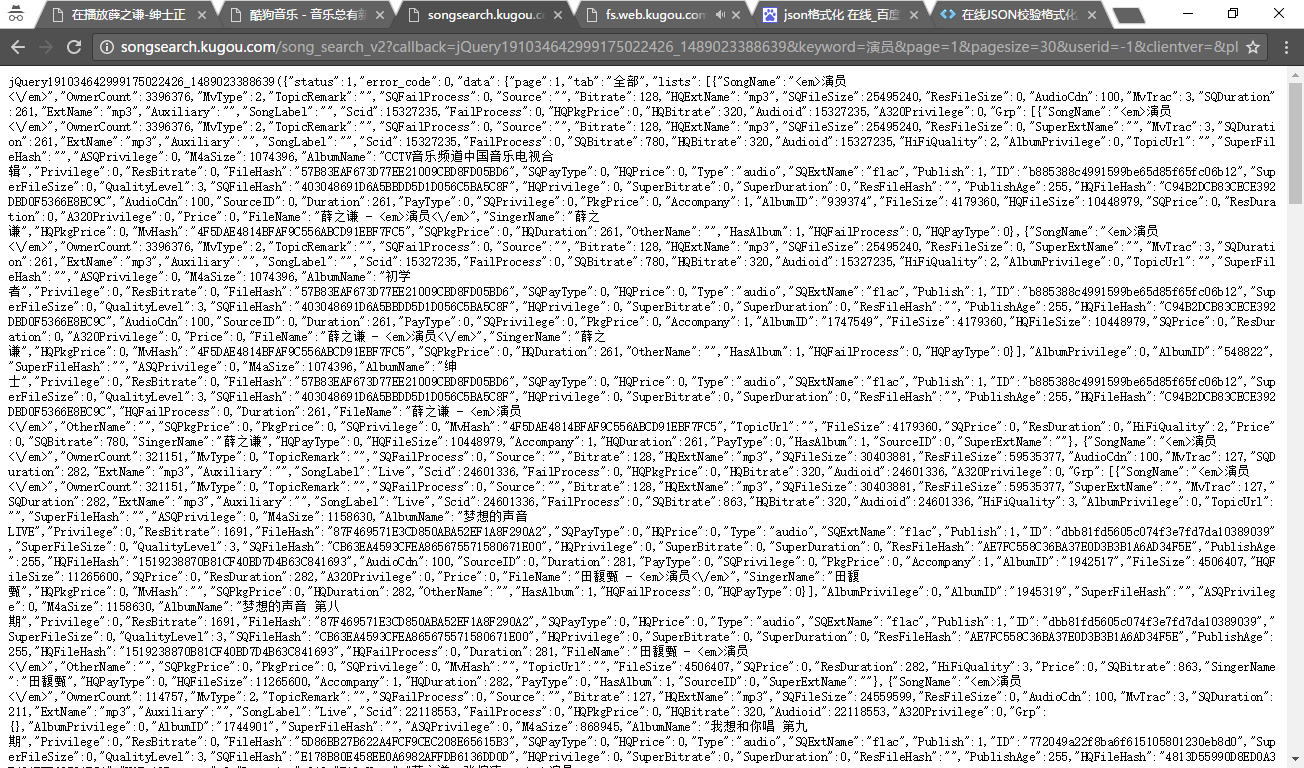
**我们查看歌词列表的请求地址**



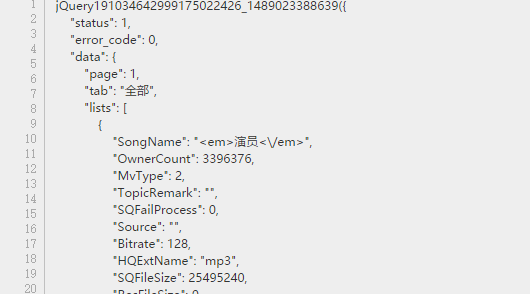
**这会有一个keyword的参数，后面是乱码，但是这是一种中文的转码方式，复制这个请求地址，并改变这个keyword的值。**



**返回的结果是一个json的文件，我们来格式化**



**我们观察后其实可以发现，这其实都是由搜索我们设置的那个关键词的歌曲**



**里面会有好多的歌曲**

**我们找到其中一首歌的filehash**



**用之前同样的方法，就可以获得这首歌的MP3的地址，**

**由此我们获得了这两个url的请求地址，第一个就是歌曲的查询列表，可以获取歌曲的hash值，第二个是歌曲的地址，可以根据一个查询列表中的hash值来获取歌曲的地址**

<http://songsearch.kugou.com/song_search_v2?callback=jQuery191034642999175022426_1489023388639&keyword>**={歌曲名称}&page=1&pagesize=30&userid=-1&clientver=&platform=WebFilter&tag=em&filter=2&iscorrection=1&privilege\_filter=0&\_=1489023388641**

<http://www.kugou.com/yy/index.php?r=play/getdata&hash>**={歌曲hash值}**