**视频下载**

之前我们爬取的内容都是文字和图片，我们从来没有试过爬取视频，我们今天就来学习如何爬取视频。

其实大致步骤都一样

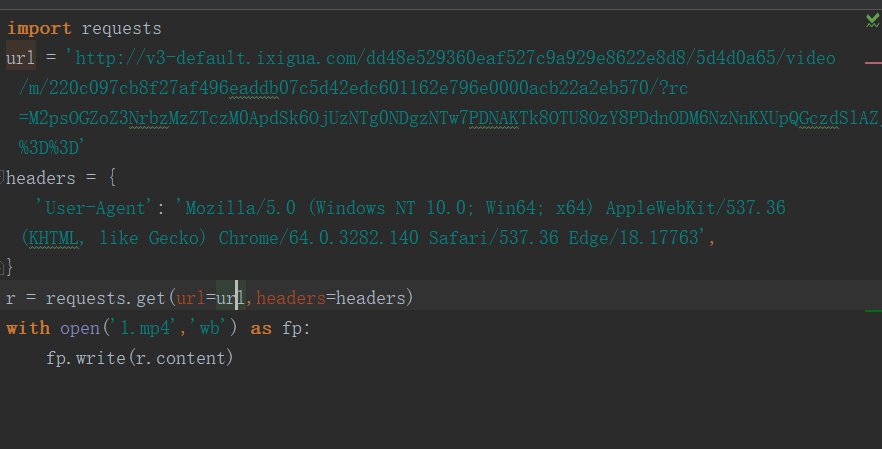
第一步：获取视频所在的网页

第二步：F12中找到视频真正所在的链接

第三步：获取链接并转换成机械语言

第四部：保存

我们先来看一个我写的小代码，来试试水先

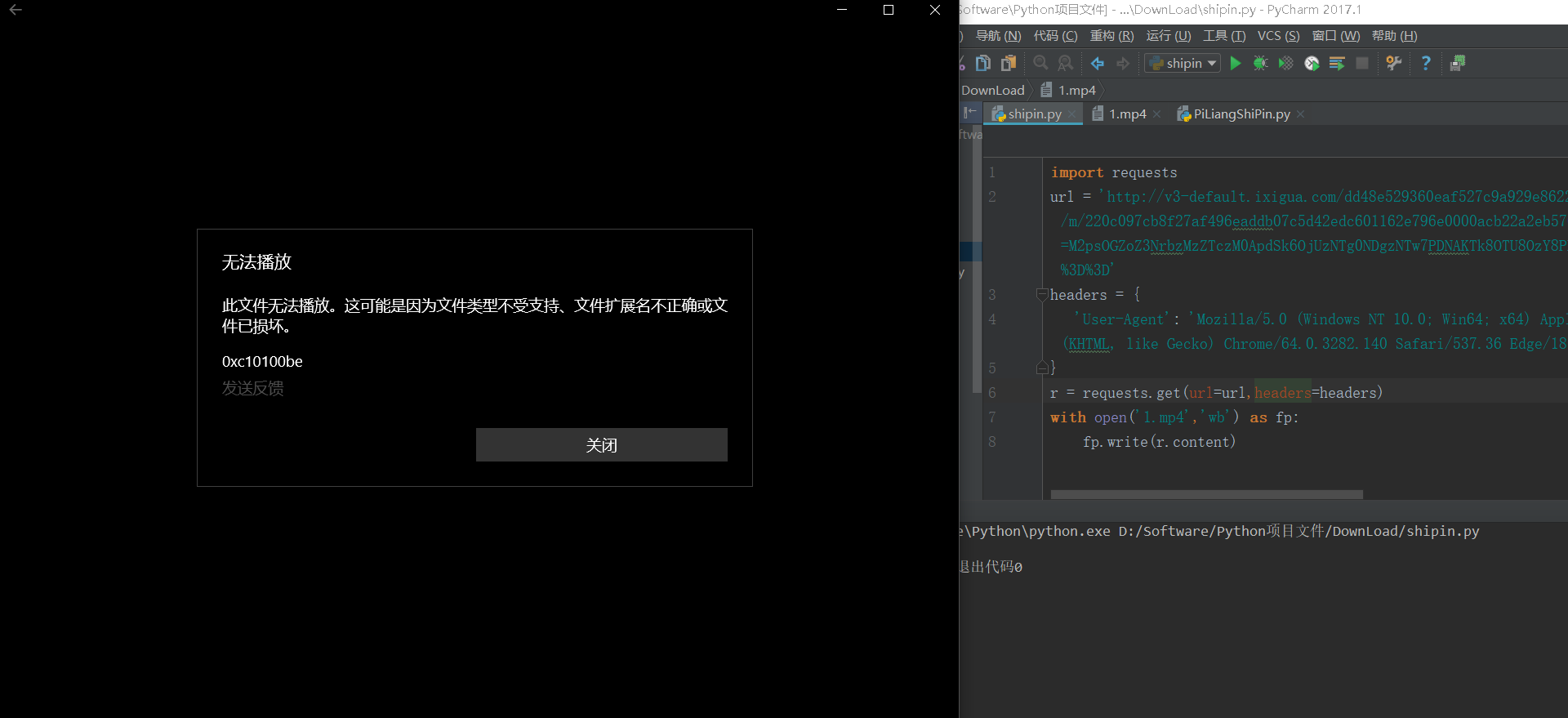


这是头条上的一个小视频，我已经在检查里面获取到了视频的src，获取到了他真正的链接，我们只需要向这个链接发送请求，下载就好了

分析代码：

* 先导入库，然后把我们获取到的src当作url写入
* 既然是爬取就要写上请求头，还是以前的
* 其实很简单，我们直接向这个src发送请求就完事了，我们这个时候已经获取到了响应内容
* 直接保存到本地

我们来看看运行结果：



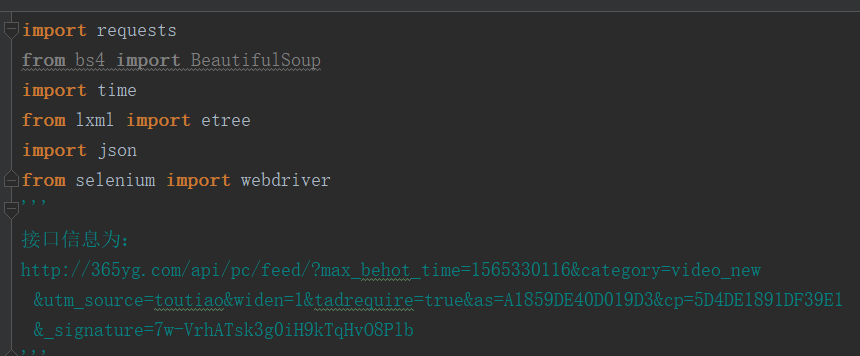
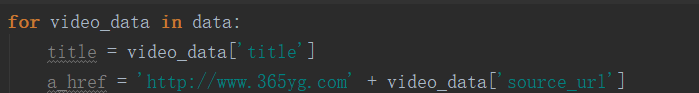
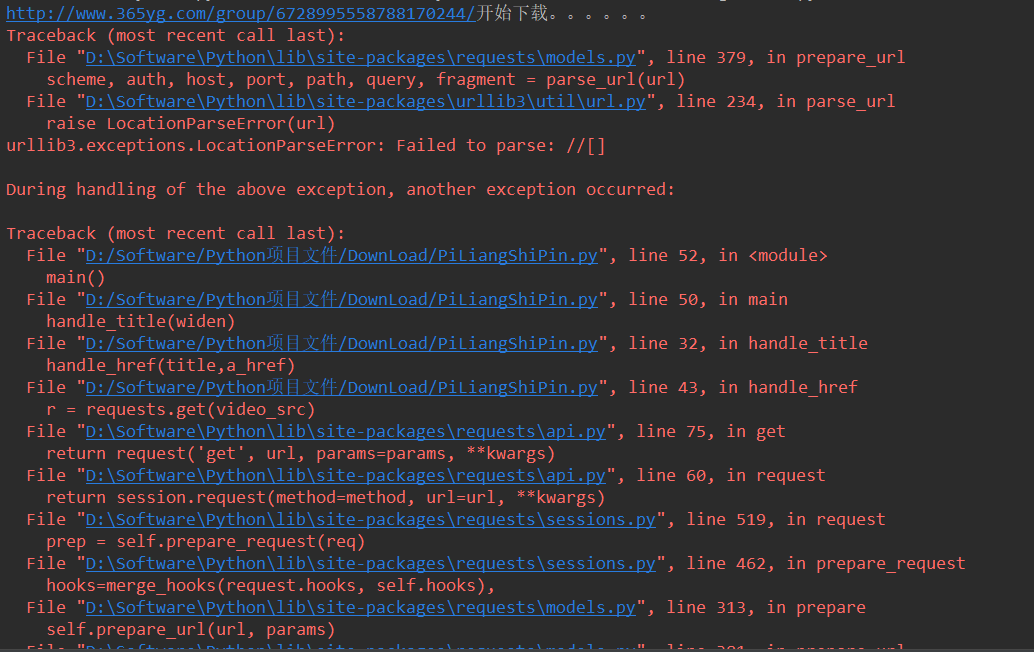
其实是没问题的，因为这个代码是我好久之前写的，当时是可以播放的，我们已经成功下载了，主要是我刚才看了看网站，这个视频已经被下架了，所以现在我再下载它，就这样了

我们接下来做一个大一点的吧，多爬取一点，批量视频下载，

我们来看代码：



分析代码：

* 先导入我们需要的库，还有我们捕获到的接口信息
* 
* 这次我们把请求头放在最前面，当作一个全局变量，方便使用
* 在数函数里面，我们先解析首页，返回所有的标题链接，我们先循环1到5就可以
* 
* 所以我们需要新建一个函数
* 来到这个函数
* 把我们刚才捕获到的视频接口信息拿过来，将刚才的页码和url（也就是接口信息）拼接起来，组成完整的url
* 然后我们向这个url发送请求，获取到相应内容之后，我们就要解析内容，因为返回的是json格式，所以我们直接解析json格式
* 
* 通过分析，需要data里面的title\source\_url，所以我们将json数据转化为python对象
* 
* 在这里我们取出所有和视频相关的数据，data是一个列表，里面存放的都是字典
* 我们接下来循环这个data列表，依次取出每一个视频信息
* 
* 我们现在已经获取到了视频的标题和href，所以在这里我们做的是，发送请求，获取内容，解析内容，获取src
* 我们这次用selenium控制浏览器直接下载，所以新建一个函数
* 来到这个函数
* 我们首先把phantomjs的位置写进去，通过它发送请求，我们这个时候需要获取源码，生成tree对象，然后查找video里面的src属性
* 我们怎么获取src属性呢，这个时候其实可以写正则提取，也可以写xpath提取，我更喜欢使用xpth，所以就用它，然后获取到之后，我们就开始下载，保存到本地
* 这个整体思路没问题，但是当我运行，遇到下面的错误信息
* 
* 提示我请求失败，按理说不应该啊，在上面已经获取到了src，都开始下载了，应该问题出现在了视频接口信息，我后来查阅了好多资料还是一样的问题，这个问题先留下来，我继续查一下！