爬虫03-浅谈Response对象

我们来讲讲请求之后，服务器返回的响应。

首先我们来看下下面这段代码，想来大家应该都不陌生。

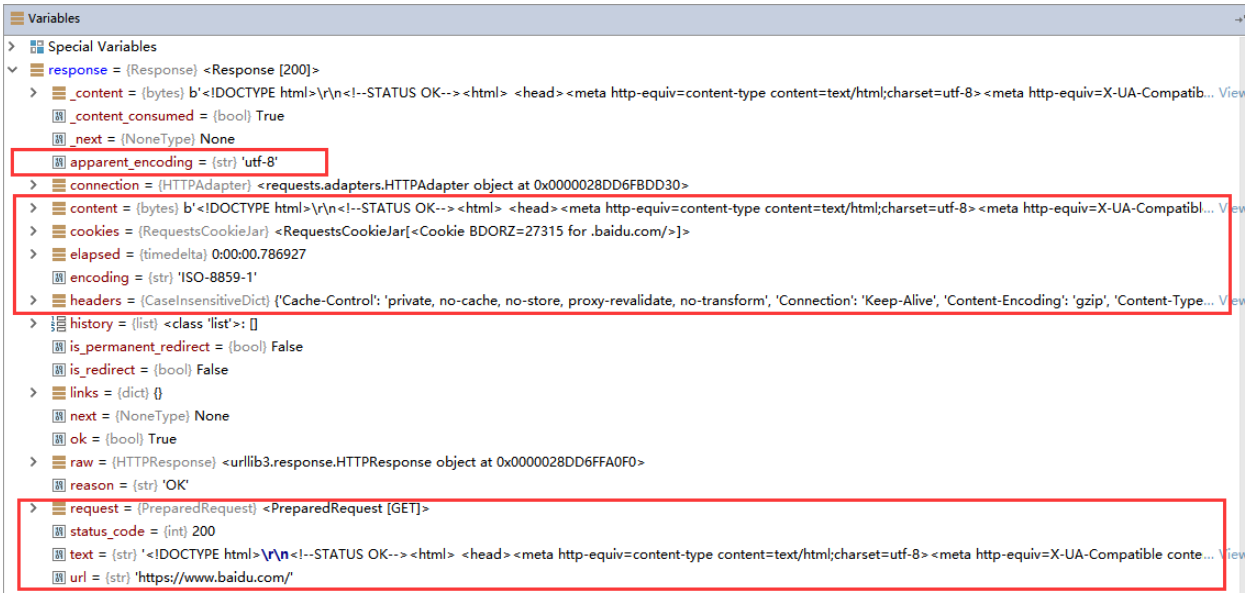
```
1 import requests
2 response = requests.get('https://www.baidu.com')
3 print(response.text)
4 print(response.content)
```

这里的response其实是一个Response对象，text和content分别表示取这个对象的text属性和content属性。

其中text表示文本数据形式的网页源代码，content表示字节型数据的网页源代码。

那么Response对象就只有这两个属性吗？还有其他的很有用的属性或方法吗？

当然有，接下来我们先看下Response对象都有些什么内容。



以上就是Response对象所有属性，有的我们暂时不需要了解，先来看看哪些是能够为我们所用的。

其中content和text属性我们已经学过了，分别表示网页源码的字节型数据和文本数据。

第一个要了解的是apparent_encoding和encoding属性，其中encoding是从HTTP header中猜测的响应内容编码方式，apparent_encoding是根据响应内容分析出的编码方式。

我们应该重点关注apparent_encoding，这个属性能够很好地帮助我们确认网页源码的编码方式，避免获取到的内容乱码。所以在获取网页源码时，为避免乱码，我们可以这么做。

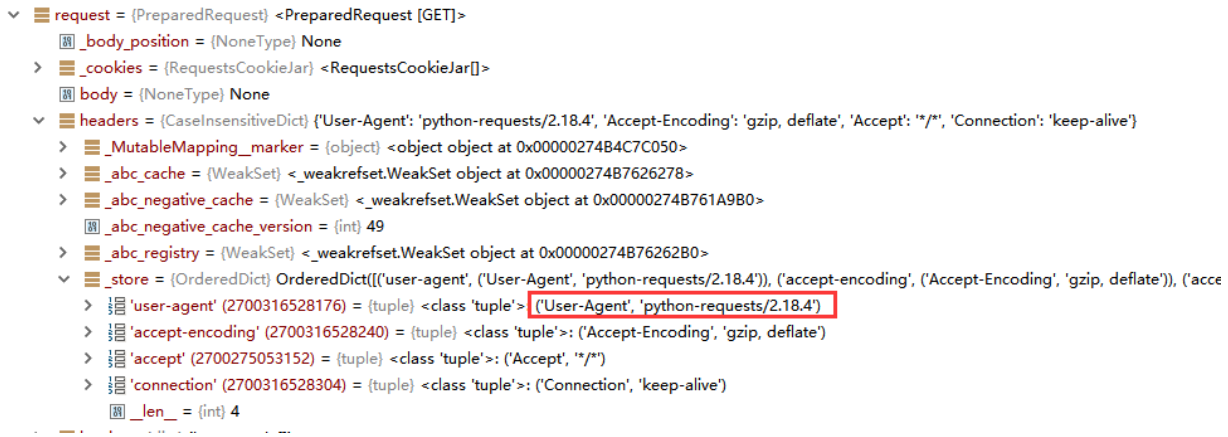
```
# 实现功能，避免获取到的网页源码是乱码
import requests

res = requests.get('https://www.baidu.com')
res.encoding = res.apparent_encoding
# 这样，我们就不需要去关心获取到的网页源码的编码格式具体是什么
print(res.text)
f = open('baidu.com', 'w', encoding=res.encoding)
f.write(res.text)
 f.close()
```

第二个要了解的是cookies属性，该属性保存了用户的cookie值，我们有的时候可以通过获取到上一个请求的cookie，作为请求头的一个cookie参数传入到请求中。

第三个要了解的是headers属性，这个属性中记录了响应头中的相关内容，虽然不怎么会用到，但还是要理解这是什么。

第四个要了解的是request属性，这个属性记录了请求的相关信息，这其中有一个请求头需要了解一下，通过了解这个请求头，我们能够更加深入地理解我们在编写爬虫的时候，为什么要在请求头中添加User-Agent参数，如果不加这个参数，这个headers属性中的User-Agent值是什么呢，我们可以来看一下：



可以看到，如果请求时不添加User-Agent参数，User-Agent默认值为python-requests/2.18.4，这样就容易被网站识别出是一个爬虫，从而限制我们的爬取。

最后一个需要了解的是status_code属性，这个属性表示响应的状态码，当我们一次性爬取的url数量过多的时候，就需要用status_code来过滤掉请求失败的url，否则在后面的数据解析中容易报错，从而导致程序运行的终止。

```
所以当爬取的url数量过多的话，经常会这样处理：
import requests
url_list = ['https://www.baidu.com', 'https://www.zhihu.com', 'https://www.douban.com']

for url in url_list:
 res = requests.get(url)
 if res.status_code != 200:
 continue
 print(res.text)
```