response 报400或者403错误，基本判定为请求未被授权，往往是cookie或者token未设置。

**可以通过以下3种方法设置cookie**

**Request headers传参**

class MySpider(scarpy.Spider):

name = 'myspider'

start\_url = 'http://xxx.com'

headers = {'Cookie': 'mycookie=cookieVal'}

def start\_request(self):

yield Request(url=self.start\_url, headers=self.headers)

**Request cookies传参**

class MySpider(scarpy.Spider):

name = 'myspider'

start\_url = 'http://xxx.com'

cookies = {'mycookie': 'cookieVal'}

def start\_request(self):

yield Request(url=self.start\_url, cookies=self.cookies)**)**

**写在自定义middlewares里面**

class CookiesMiddleware(object):

def process\_request(self, request, spider):

request.headers['Cookie'] = 'cookieVal'

**遇到的坑**

使用headers传参时，即使Scrapy的headers参数 和 postman 或者 自己通过request库写的request请求 中header的参数一模一样，后面二者能得到200正确返回，前者却始终返回400错误

原因：使用headers传参方法设置cookie的优先级特别低，会被scrapy自带的cookiesMiddleware覆盖，导致cookie为空、未正确配置。

这几种方法存在各自的优先级顺序，优先级低的Cookie会被覆盖

首先注意一点：

Scrapy会自带一个cookie中间件scrapy.downloadermiddlewares.cookies.CookiesMiddleware

自定义的middlewares在settings中其优先值应当比scrapy自带cookieMiddlewares**大**（否则会被覆盖）

\*\* 通过命令行 scrapy settings --get=DOWNLOADER\_MIDDLEWARES\_BASE来查看scrapy自带的middlewares优先级值，一般scrapy.downloadermiddlewares.cookies.CookiesMiddleware的优先值为700

写在header里面的cookie会被cookiesMiddleware（不论是自己设置的还是scrapy自带的）所覆盖

\*\* 如果只使用了scrapy自带的cookiesMiddleware，写在header里面的cookie会直接被清除，因此要在settings里面将其禁用掉

而Request cookies传参的优先级基于自定义CookiesMiddleware与scrapy自带的CookiesMiddleware之间

**因此，在自定义CookiesMiddleware优先值大于scrapy自带的CookiesMiddleware优先值情况下**

**这几个cookie配置的优先级为：**

自定义CookiesMiddleware > Request cookies传参 > scrapy自带CookiesMiddleware > Request headers传参

注意：某些网站在验证cookie的时候是只查看header里面的cookie值是否存在，因此，在写入cookie的时候最好写两份

1、Request headers传参

2、Request cookies传参 或者 自定义CookiesMiddleware