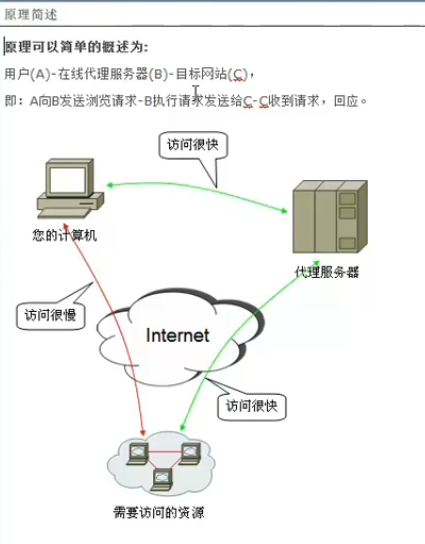
代理IP的使用与cookie的处理

# 代理IP的使用

# 原理简述



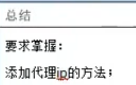
# 实现方法



# 具体实现



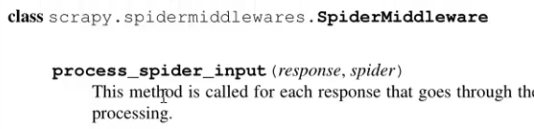
# 总结

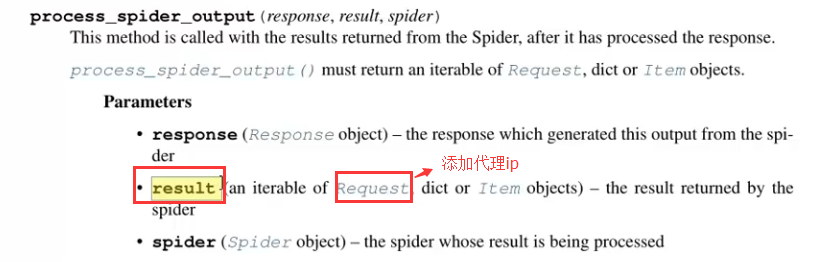


# 代码解读：

## 从哪里引入代理ip

### 重写自己的SpiderMiddleware的process\_spider\_output方法



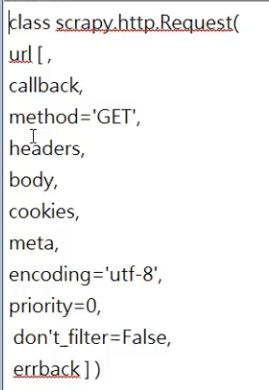


在SpiderMiddleware的**process\_spider\_output方法中有一个参数为result，可以通过result获取Request，从而添加ip的代理proxy。**

### 或者重写DownloaderMiddleware的process\_request方法

由于DownloaderMiddleware的process\_request方法具有参数request，所以在这里可以直接对request修改，主要是对request的meta进行修改，添加proxy。

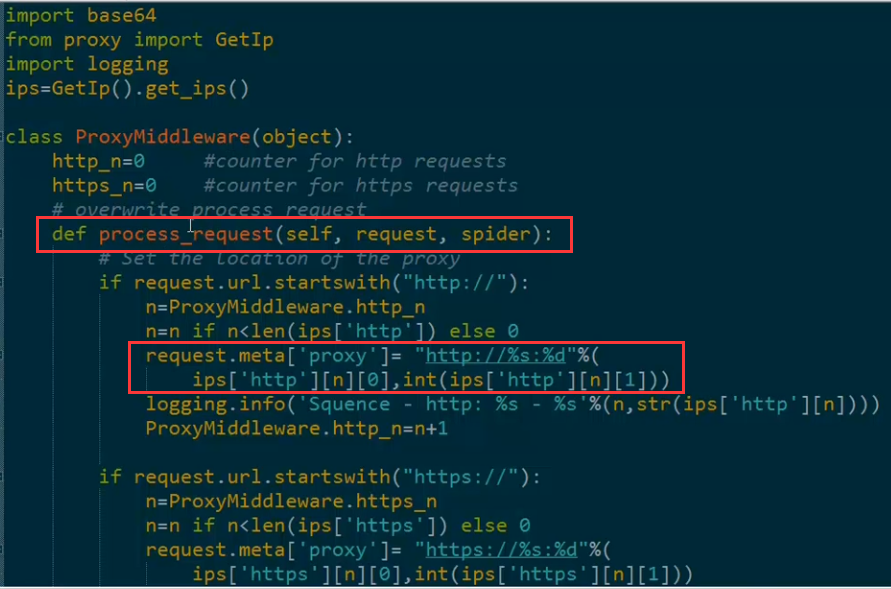
主要是***通过 request.meta[‘proxy’]=”***[***http://ip地址***](http://ip地址)***” 来实现***。

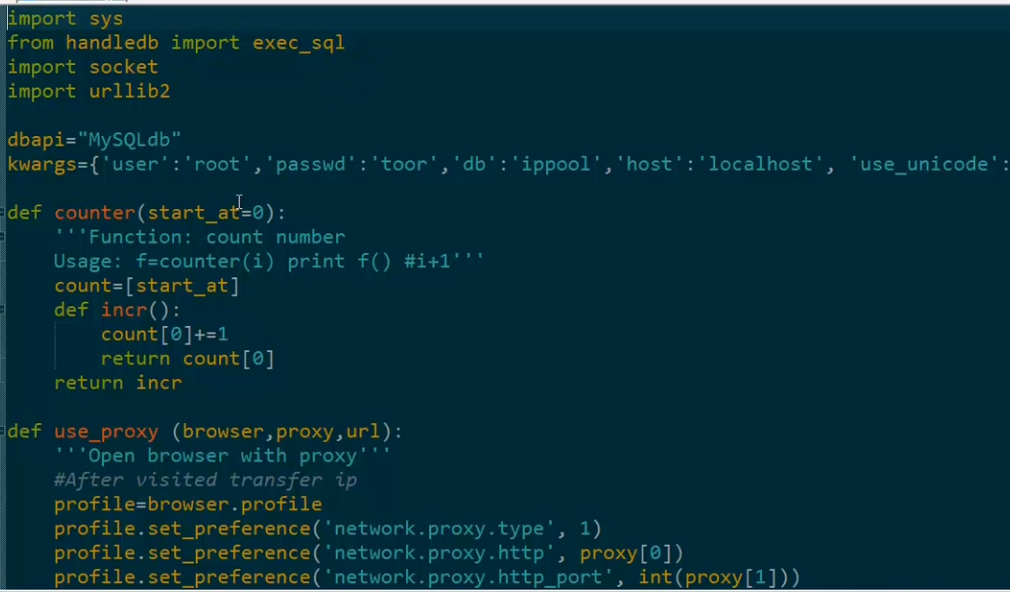


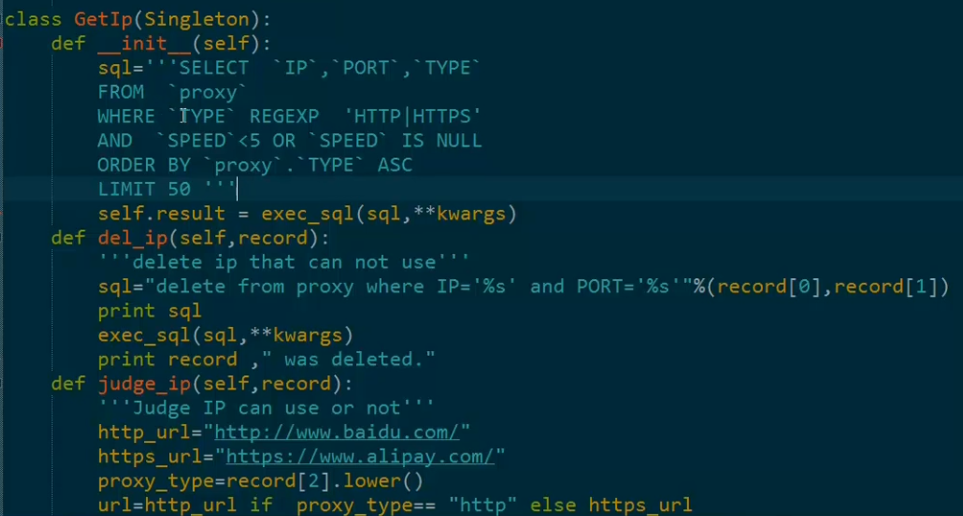
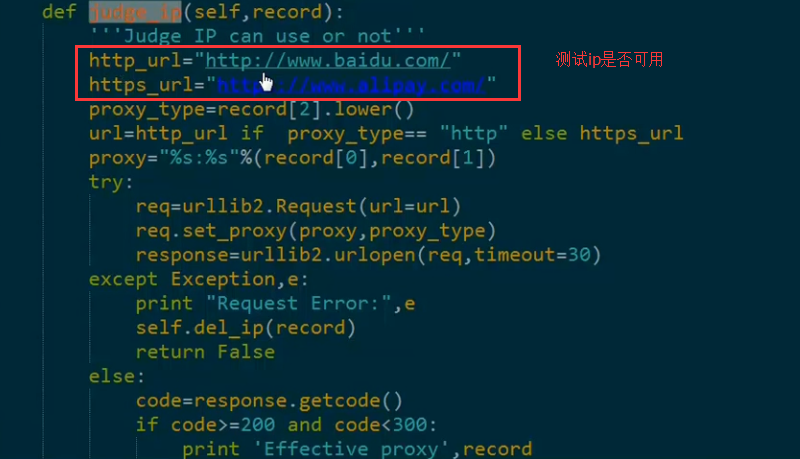
## 首先获取代理ip，从西刺网站获取代理ip。

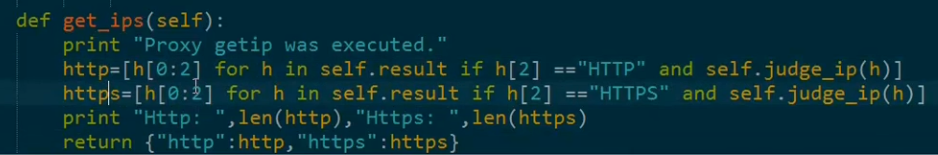
## 验证ip：获取的ip有些不能使用，所以在使用之前需要验证ip的可行性。

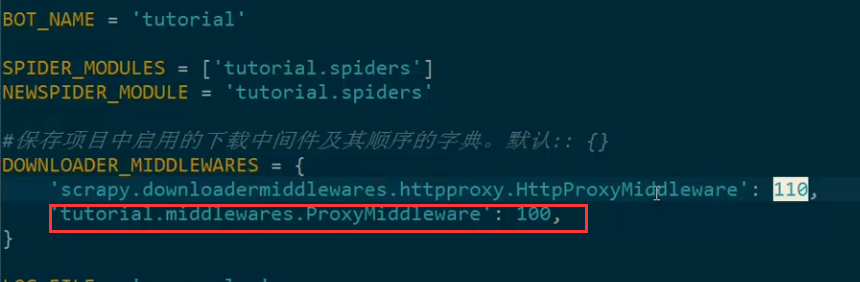
## 代码解读









# 代理ip的使用：

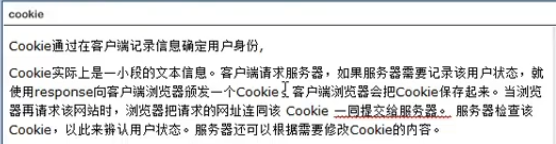
一般是找出一些代理ip，轮换使用这些ip。因为一些网站对ip具有查封功能，轮换使用ip，目的就是掩盖爬虫的行为。

# cookie的处理

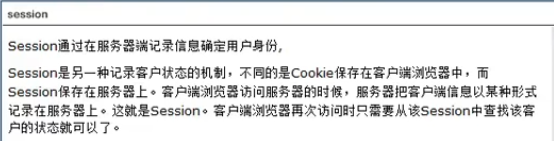
# 登录的原理



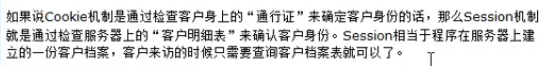
## cookie介绍



## session介绍



## cookie与session的区别

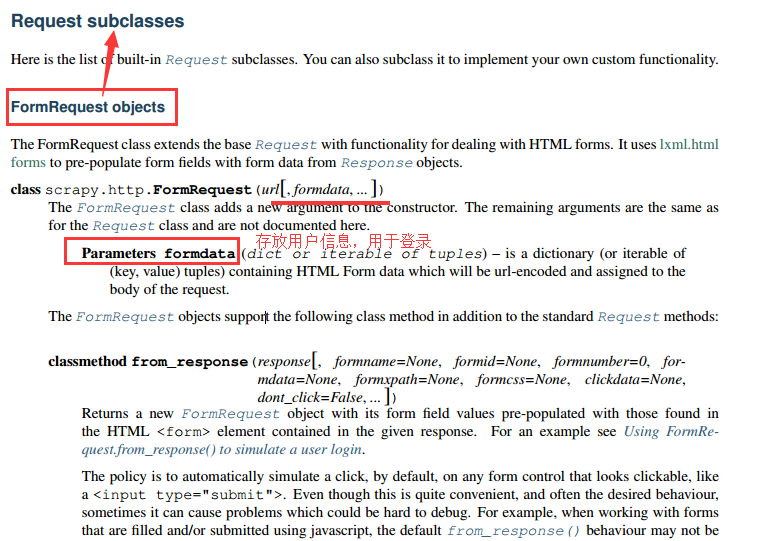


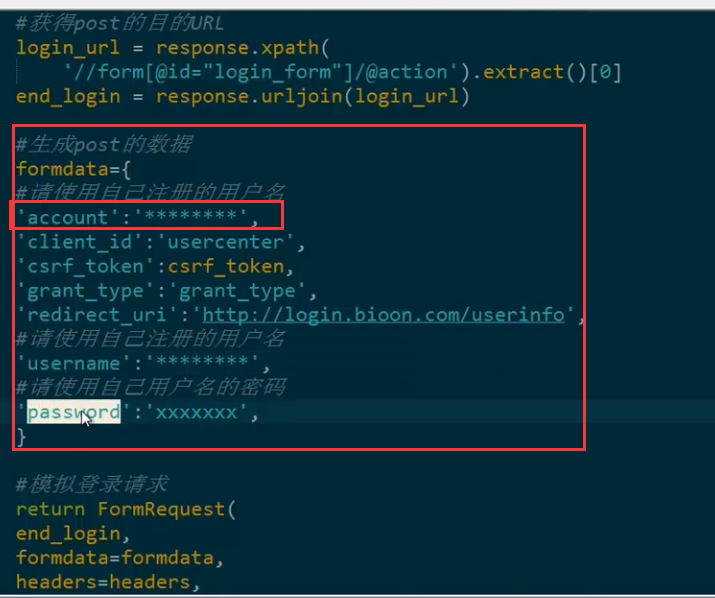
# 登录的方法



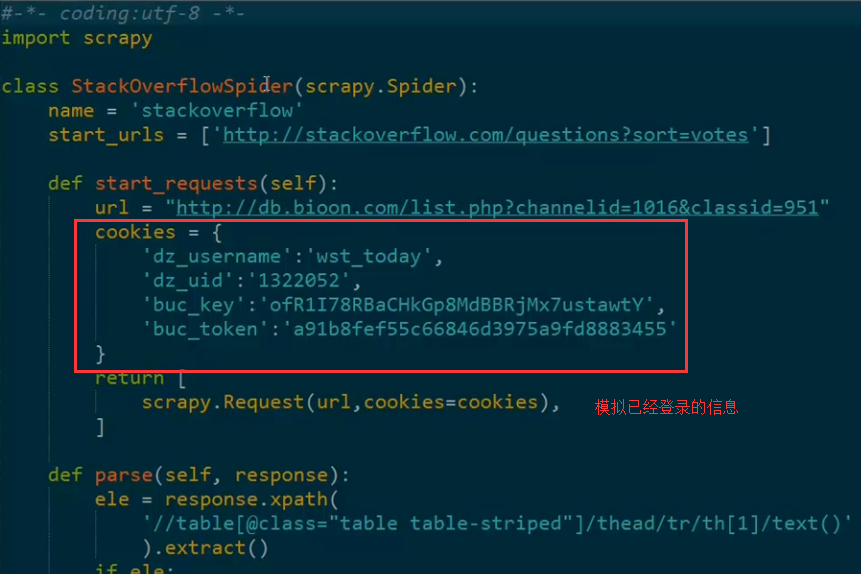
## FormRequest

**FormRequest是Request的一个子类**，可以用于提交表单数据，因此可以用于用户登录。

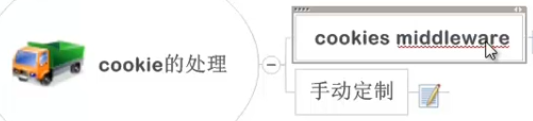




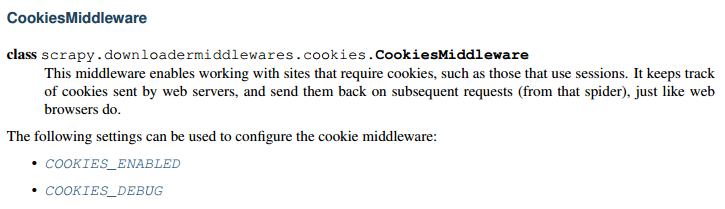
## 添加cookies到Request中，伪装成一个登录用户

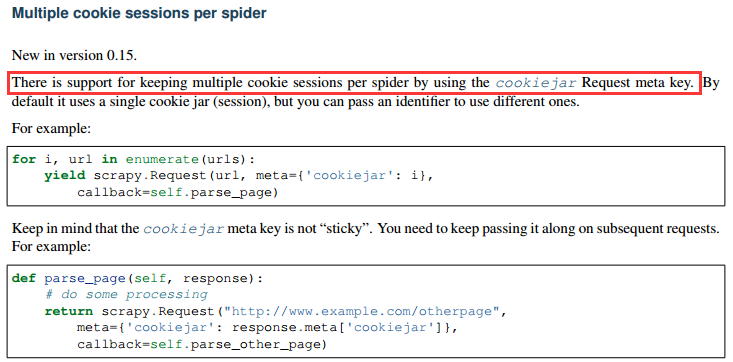


# cookie的处理



CookiesMiddleware是一个内置的中间件。用于对cookies信息的处理。





默认启用cookies。

