爬虫程序中怎么加入动态代理

相信很多人都用过代码写过不同的爬虫程序吧，来获取互联网上自己需要的信息，这比自己手动的去一个一个复制来的容易。但是，居然是用程序来获取某个网站里面的信息，可以知道，在很短的时间内，这个程序会访问某个网站很多次，很多网站都会对这样的情况进行屏蔽；比如，隔几分钟才能正常访问。这对于我们的爬虫程序来说是个大麻烦。我们知道，当我们访问一个网站的时候，对方服务器是会记下我们电脑的IP地址，有没有方法来动态改变自己的IP地址呢？答案是有的，那就是用代理。这样我们就可以在程序中加入代理功能，只要对方服务器屏蔽了我们的IP，我们就用程序自动的去换别的IP地址嘛，这样不就可以不断的访问某个服务器吗？可以利用Java的HttpClient包，来加入动态代理功能。

好了，说了这么多，程序怎么实现呢？具体的思路是：当我们可以正常访问一个页面的时候（给服务器发送一条HTTP请求），服务器一般是会返回2XX的HTTP响应码给我们。当服务器返回诸如403（被禁止访问了；当然，这个页面正常情况下是可以访问的，正常情况下都返回403 的代码，那就是人家本来就不让你访问的啦，我也哀莫能及）HTTP相应码的时候，我们就可以知道，服务器是把我们屏蔽了。这时候，当我们的程序检测到返回的403代码的时候，就可以换一个IP地址，再重新请求刚刚被屏蔽的页面不就实现了动态代码的程序吗？

第一步：先声明一个代理类

```
package com.wyp.utils;

/**
 * @author 过往记忆
 * Create Data: 2012-8-10
 * Email: wyphao.2007@163.com
 * Blog:
 * 版权所有，翻版不究，但在修改本程序的时候必须加上这些注释！
 * 仅用于学习交流之用
 *
 * 代理服务器IP以及端口
 */
public class proxyServer {
 public static String proxyIP [] = {"ec2-23-22-95-3.compute-1.amazonaws.com",
   "211.68.70.169", "202.203.132.29",
   "218.192.175.84", "ec2-50-16-197-120.compute-1.amazonaws.com",
   "50.22.206.184-static.reverse.softlayer.com",};
 public static int proxyPort[] = {8000, 3128, 3128, 3128, 8001, 8080};
}
```

这个代理类很简单吧，就加了几个代理的IP和端口。注意，proxyPort中的端口号要一一对应proxyIP中的IP地址。
第二步：声明一个HttpClient 对象，设置好超时时间。

```
HttpClient httpClient = null;
httpClient = new HttpClient();
// 设置 Http 连接超时 5s
httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(5000);
```

第三步：设置代理

```
/**
 * 设置代理
*/
private void setProxy(String proxyIP, int hostPort) {
 System.out.println("正在设置代理：" + proxyIP + ":" + hostPort);
 // TODO Auto-generated method stub
 httpClient.getHostConfiguration().setHost(Hosturl, hostPort, "http");
 httpClient.getParams().setCookiePolicy(CookiePolicy.BROWSER_COMPATIBILITY);
 httpClient.getHostConfiguration().setProxy(proxyIP, proxyPort);
       Credentials defaultcreds = new UsernamePasswordCredentials("",  "");
       httpClient.getState().setProxyCredentials(
   new AuthScope(proxyIP, proxyPort, null), defaultcreds);
}
```

第四步：测试当前的代理是否有用

```
/**
* @param string
* @param i
*
* 测试代理服务器的可用性
*
* 只有返回HttpStatus.SC_OK才说明代理服务器有效
* 其他的都是不行的
*/
private int testProxyServer(String url, String proxyIp, int proxyPort) {
 // TODO Auto-generated method stub
 setProxy(proxyIp, proxyPort);
 GetMethod getMethod = setGetMethod(url);
```

```
 if(getMethod == null){
  System.out.println("请求协议设置都搞错了，所以我无法完成您的请求");
  System.exit(1);
 }
 try {
  int statusCode = httpClient.executeMethod(getMethod);
  if (statusCode == HttpStatus.SC_OK) { //2XX状态码
   return HttpStatus.SC_OK;
  }else if(statusCode == HttpStatus.SC_FORBIDDEN){ //代理还是不行
   return HttpStatus.SC_FORBIDDEN;
  }else{ // 其他的错误
   return 0;
  }
 } catch (HttpException e) {
  // 发生致命的异常，可能是协议不对或者返回的内容有问题
  System.out.println("Please check your provided http address!");
  System.exit(1);
 } catch (IOException e) {
  // 发生网络异常
  System.exit(1);
 } finally {
  // 释放连接
  getMethod.releaseConnection();
 }
 return 0;
}
```

第五步
：得到服务器是否对我们进行屏蔽，如果返回的是SC_FORBIDDEN，代表被屏蔽的，那么我们就一个一个代理去试，也就是调用第四步的函数去判断当前的代理是否有用。

```
if(statusCode == HttpStatus.SC_FORBIDDEN){  //访问被人家限制了就设置代理
    //代理服务器的个数
    int proxySize = proxyServer.proxyIP.length;
    int i = 0;
    for(; i < proxySize; i++){        //我们一个一个测试代理服务器的有用性
        System.out.println("正在测试代理：" +
   proxyServer.proxyIP[i] + ":" + proxyServer.proxyPort[i]);
        int status = testProxyServer(url, proxyServer.proxyIP[i],
   proxyServer.proxyPort[i]);
        if(status == HttpStatus.SC_OK){//代理服务器找到了
            break;
        }else{ //其他情况你就继续去代理吧
```

```
            continue;
        }
    }

    if(i >= proxySize){
        System.out.println("唉，我把你设置的代理服务器都测试了，
好像没有发现有效的代理，我只有退出了！");
        return null;
    }

    System.out.println("代理：" + proxyServer.proxyIP[i] + ":" +
proxyServer.proxyPort[i] + "目前可用");
    proxyIP = proxyServer.proxyIP[i];
    proxyPort = proxyServer.proxyPort[i];
```

经过上面几步，我们就可以不要担心程序被人家屏蔽了。

源码下载