

## 反向代理

nginx只能处理静态文件例如：html、js、css、图片等，无法处理jsp、php、asp

那么nginx可以通过配置来处理jsp、php、asp这类请求

配置如下：

```
1. server {
2.     listen 80;
3.     server_name www.myhost2.com;
4.
5.     location ~* \.(jpg|jpeg|css|js) {
6.         root myhost2;
7.     }
8.
9.     location ~ \.jsp$ {
10.        proxy_pass 192.168.72.1:8080;
11.    }
12. }
```

nginx处理静态文件，发现是jsp文件就把 请求转发给 192.168.72.1:8080 的tomcat， 当tomcat响应完成的结果在交给nginx进行返回

## 正向代理

场景：Java程序爬取网站，但是同一个ip访问太频繁会封，那么用nginx做正向代理。

Java程序请求Nginx机器，nginx机器去请求网络，nginx机器将请求的结果返回Java程序

Nginx配置

```
1. #HTTP
2. server{
3.     resolver 8.8.8.8;
4.     access_log /usr/local/nginx/logs/access_proxy.log;
5.     listen 8118;
6.     location / {
7.         root html;
8.         index index.html index.htm;
9.         proxy_pass $scheme://$host$request_uri;
10.        proxy_set_header HOST $http_host;
11.        proxy_buffers 256 4k;
12.        proxy_max_temp_file_size 0k;
13.        proxy_connect_timeout 30;
```

```

14.             proxy_send_timeout 60;
15.             proxy_read_timeout 60;
16.             proxy_next_upstream error timeout
invalid_header http_502;
17.         }
18.         error_page 500 502 503 504 /50x.html;
19.         location = /50x.html {
20.             root html;
21.         }
22.
23. }
24.
25.
26. #HTTPS
27. server{
28.     resolver 8.8.8.8;
29.     access_log
/usr/local/nginx/logs/access_proxy_https.log ;
30.     listen 8119;
31.     location / {
32.         root html;
33.         index index.html index.htm;
34.         proxy_pass https://$host$request_uri;
35.         proxy_buffers 256 4k;
36.         proxy_max_temp_file_size 0k;
37.         proxy_connect_timeout 30;
38.         proxy_send_timeout 60;
39.         proxy_read_timeout 60;
40.         proxy_next_upstream error timeout
invalid_header http_502;
41.     }
42.     error_page 500 502 503 504 /50x.html;
43.     location = /50x.html {
44.         root html;
45.     }
46. }

```

## Java程序

```

1. import java.io.IOException;
2. import java.net.InetSocketAddress;
3. import java.net.Proxy;
4.
5. import org.jsoup.Connection;
6. import org.jsoup.Connection.Method;
7. import org.jsoup.Connection.Response;
8. import org.jsoup.Jsoup;
9.
10. public class DownLoader {
11.
12.
13.     public static Response download(String href) {
14.         String ip = "192.168.204.132";
15.         int port = 8118;
16.         Proxy proxy = new Proxy(Proxy.Type.HTTP, new
InetSocketAddress(ip, port));

```

```

17.
18.         Connection con = Jsoup.connect(href);
19.         con.referrer(href);
20.         con.timeout(30000); //时间怎么这么长
21.         con.method(Method.GET);
22.         con.ignoreHttpErrors(true); //忽略加载动态的js, 媒体资源
23.         con.ignoreContentType(true);
24.         con.header("Connection", "keep-alive");
25.         con.header("Accept-Language", "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3");
26.         con.userAgent("Mozilla/5.0 (Windows NT 10.0; WOW64; rv:44.0) Gecko/20100101 Firefox/44.0");
27.         con.header("Accept",
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
);
28.         int x = 0;
29.         Response res = null;
30.         while (x < 3) {
31.             try {
32.                 res = con.execute();
33.                 int sc = res.statusCode(); //返回的状态码
34.                 String sm = res.statusMessage();
35.                 if (sc == 404 || sc >= 500) {
36.                     System.err.println(href + " >> " + proxy
+ " >> " + sc + " >> " + sm);
37.                     return null;
38.                 }
39.                 break;
40.             } catch (Exception e) {
41.                 x++;
42.                 System.err.println(href + " >> " + proxy + "
>> " + x + " >> " + e.getMessage());
43.             }
44.         }
45.         return res;
46.     }
47.
48.
49.     public static void main(String[] args) {
50.         Response res = download("http://www.jd.com");
51.         try {
52.             System.out.println(res.parse().toString());
53.         } catch (IOException e) {
54.             e.printStackTrace();
55.         }
56.     }
57.
58. }

```

图解说明

