



链滴

狂虐小万网动态 IP

作者: [zaq369cde](#)

原文链接: <https://ld246.com/article/1447502861861>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h1 class="brush: java">功能：自动绑定 动态的ip到固定域名（
网DNS） </h1>
<p>我的博客服务器上的第一个小java应用。 </p>
<p>明天再贴第二个（狂虐小7牛） </p>
<p>ps:这个富文本编辑咱个这么难用。。。 </p>
<h1 class="brush: java">&nbsp</h1>
<pre class="brush: java">package com.zaq;

```
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.Inet6Address;
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.net.URL;
import java.net.URLConnection;
import java.net.UnknownHostException;
import java.util.Enumeration;
import java.util.Properties;

import com.aliyun.api.AliyunClient;
import com.aliyun.api.AliyunConstants;
import com.aliyun.api.DefaultAliyunClient;
import com.aliyun.api.dns.dns20150109.request.UpdateDomainRecordRequest;
import com.aliyun.api.dns.dns20150109.response.UpdateDomainRecordResponse;
import com.taobao.api.ApiException;

public class MainRun {
    private static String config = "conf.properties";
    private static AliyunClient client;
    private static Properties props;

    private static String localIP = "";//本地ip
    private static String outIP = "";//外网ip

    public static void init() throws IOException {
        props = new Properties();
        InputStream is = new BufferedInputStream(MainRun.class.getClassLoader().getResourceAsS
```

```

ream(config);
props.load(is);

/*
 * 详见
 * https://docs.aliyun.com/?spm=5176.776555948.1863381.554.7Lk02r#/pub/dns
 */

String serverUrl = "http://dns.aliyuncs.com/";//小万网云哥DNS解析
String accessKeyId = "不给你看";
String accessKeySecret = "*****";

client = new DefaultAliyunClient(serverUrl, accessKeyId, accessKeySecret, AliyunConstants.
ORMAT_JSON);

}

public static void main(String[] args) throws IOException, InterruptedException{
    init();
/*    DescribeDomainRecordsRequest request3=new DescribeDomainRecordsRequest();
    DescribeDomainRecordsResponse response3;
    request3.setDomainName("freshz.cn");
    try {
        response3 = client.execute(request3);

        System.out.println(response3.getBody()+response3.getMessage());
    } catch (ApiException e) {
        e.printStackTrace();
    }*/
    while(true){

        //绑定本地Ip到local
        String ipLocal=getLocalIP();
        System.out.println(ipLocal);

        if(!localIP.equals(ipLocal)){
            UpdateDomainRecordRequest request2=new UpdateDomainRecordRequest();
            UpdateDomainRecordResponse response2;
            request2.setRecordId("63500319");
            request2.setR("local");
            request2.setType("A");
            request2.setValue(ipLocal);
            try {
                response2 = client.execute(request2);
                System.out.println(response2.getBody());
                if(null==response2.getErrorCode()||"".equals(response2.getErrorCode())){
                    localIP=ipLocal;
                }
            }

            } catch (ApiException e) {
                e.printStackTrace();
            }
        }
}

```

```

        }

    }

//绑定外网Ip到xxx
String ip=getMyIP();
System.out.println(ip);
if(!outIP.equals(ip)){
    UpdateDomainRecordRequest request2=new UpdateDomainRecordRequest();
    UpdateDomainRecordResponse response2;
    request2.setRecordId("12345678");
    request2.setR("xxx");
    request2.setType("A");
    request2.setValue(ip);
    try {
        response2 = client.execute(request2);
        System.out.println(response2.getBody());
        if(null==response2.getErrorCode()||"".equals(response2.getErrorCode())){
            outIP=ip;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
//小睡10秒
Thread.sleep(1000*10);

}

/**
 * 获取外网IP
 * @return
 */
private static String getMyIP(){
InputStream ins = null;
try {
    URL url = new URL(props.getProperty("ipurl","http://1111.ip138.com/ic.asp"));
    URLConnection con = url.openConnection();
    ins = con.getInputStream();
    InputStreamReader isReader = new InputStreamReader(ins, "GB2312");
    BufferedReader bReader = new BufferedReader(isReader);
    StringBuffer webContent = new StringBuffer();
    String str = null;
    while ((str = bReader.readLine()) != null) {
        webContent.append(str);
    }
    int start = webContent.indexOf("[") + 1;
    int end = webContent.indexOf("]");
    return webContent.substring(start, end);
}catch(Exception e){
}
}

```

```
e.printStackTrace();
return "";
} finally {
    if (ins != null) {
        try {
            ins.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
//获取本地IP
public static String getLocalIP() {
    String ip = "";
    try {
        ip = InetAddress.getLocalHost().getHostAddress();
    } catch (UnknownHostException e1) {
        e1.printStackTrace();
    }

    if(!ip.startsWith("127.")){
        return ip;
    }
    //linux获取本地IP方法
    try {
        Enumeration<?> e1 = (Enumeration<?>) NetworkInterface
            .getNetworkInterfaces();
        while (e1.hasMoreElements()) {
            NetworkInterface ni = (NetworkInterface) e1.nextElement();
            if (!ni.getName().equals("eth0")) {
                continue;
            } else {
                Enumeration<?> e2 = ni.getInetAddresses();
                while (e2.hasMoreElements()) {
                    InetAddress ia = (InetAddress) e2.nextElement();
                    if (ia instanceof Inet6Address)
                        continue;
                    ip = ia.getHostAddress();
                }
                break;
            }
        }
    } catch (SocketException e) {
        e.printStackTrace();
    }
    return ip;
}
}

</pre>
```