Atitit webservice attilax总结

[1.1. WebService服务的三种途径Endpoint Disco WSDL 有什么不同 1](#_Toc31151)

[1.2. 发现webservice 1](#_Toc10212)

[1.3. Ws disconvert 的组播地址和端口就是3702 1](#_Toc6295)

[2. 2](#_Toc15992)

## [WebService服务的三种途径Endpoint Disco WSDL 有什么不同](http://blog.csdn.net/xieyufei/article/details/11572839)

**Disco:** http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?disco     
通过这个可以找到该web服务。即利用了web服务发现机制中的一种发现机制disco

**WSDL:** http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?wsdl    
该web服务的描述性语言。即Web Services Description Language。用来描述你的web服务定义的方法和属性、binding协议部分、port端口以及服务URI。

## 发现webservice

## Ws disconvert 的组播地址和端口就是3702

[关于cxf报：Cannot create URL for this address soap.udp://239.255.255.250:3702的原因](http://sxlkk.iteye.com/blog/1920932)

 the ONVIF discovery process is based on<http://specs.xmlsoap.org/ws/2005/04/discovery> it use SOAP over UDP.

If you are using apache-cxf, this can be achieve using

org.apache.cxf.ws.discovery.WSDiscoveryClient

A simple sample code could be :

import java.util.List;

import javax.xml.ws.EndpointReference;

import org.apache.cxf.ws.discovery.WSDiscoveryClient;

public class Main

{

public static void main(String[] args)

{

WSDiscoveryClient client = new WSDiscoveryClient();

client.setVersion10(); // use WS-discovery 1.0

client.setDefaultProbeTimeout(1000); // timeout 1s

System.out.println("Probe:" + client.getAddress());

List<EndpointReference> references = client.probe();

System.out.println("Nb answsers:" + references.size());

for (EndpointReference ref : references)

{

System.out.println(ref.toString());

}

}

}

Probe is :soap.udp://239.255.255.250:3702

Nb answsers:1

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><EndpointReference xmlns="http://www.w3.org/2005/08/addressing"><Address>http://192.168.31.144:10080/onvif/device\_service</Address><ReferenceParameters/></EndpointReference>

[Media](http://192.168.15.240/onvif/Media" \t "http://blog.csdn.net/lvwx369/article/details/_blank)

http://192.168.31.144:10080/onvif/[Media](http://192.168.15.240/onvif/Media" \t "http://blog.csdn.net/lvwx369/article/details/_blank)

# 具体底层soap

向组播地址 239.255.255.250 发送探测请求，然后通过各个IP的应答信息(XML)解析结果

 udp://239.255.255.250:3702

请求数据如下：

**[html]** [view plain](http://blog.csdn.net/zhangkunbin/article/details/24007145" \o "view plain) [copy](http://blog.csdn.net/zhangkunbin/article/details/24007145" \o "copy)

1. "**<?xml** version=\"1.0\" encoding=\"utf-8\"**?>**"
2. "**<Envelope** xmlns:dn=\"http://www.onvif.org/ver10/network/wsdl\"xmlns=\"http://www.w3.org/2003/05/soap-envelope\"**>**"
3. "**<Header>**"
4. "**<wsa:MessageID** xmlns:wsa=\"http://schemas.xmlsoap.org/ws/2004/08/addressing\"**>**uuid:5aab9180-b87e-4238-8f4d-da32ab28e7c4**</wsa:MessageID>**"
5. "**<wsa:To** xmlns:wsa=\"http://schemas.xmlsoap.org/ws/2004/08/addressing\"**>**urn:schemas-xmlsoap-org:ws:2005:04:discovery**</wsa:To>**"
6. "**<wsa:Action** xmlns:wsa=\"http://schemas.xmlsoap.org/ws/2004/08/addressing\"**>**http://schemas.xmlsoap.org/ws/2005/04/discovery/Probe**</wsa:Action>**"
7. "**</Header>**"
8. "**<Body>**"
9. "**<Probe** xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" xmlns=\"http://schemas.xmlsoap.org/ws/2005/04/discovery\"**>**"
10. "**<Types>**dn:NetworkVideoTransmitter**</Types>**"
11. "**<Scopes** **/>**"
12. "**</Probe>**"
13. "**</Body>**"
14. "**</Envelope>**"

java使用soap方式简单实现webservice - 千里之行始于足下 - 博客频道 - CSDN.NET.html

组播技术 - jianchaolv的专栏 - 博客频道 - CSDN.NET.html