目　录

postxml 总体概述

1.Web Service 地址:

http://v3.huiyee.com:8080/diws/ups 注明:前面是服务器地址 后面加上/diws/ups

2.Web Service 接口类型

管理收件人	<userspec>xxx</userspec>	批量导入收件人到某个组(支持删除,修改,创建新组)
发送邮件	<sendspec>xxx</sendspec>	批量发送邮件(支持指定组,指定收件人)
邮件报告	<reportspec>xxx</reportspec>	支持多种邮件报告形式,(打开,订阅,硬弹,软弹,退订,点击,总括,发送日志)
管理任务	<taskspec>xxx</taskspec>	支持查看任务的进行状态,暂停任务,重新开始任务等

postxml 具体阐述

1.理念

首先我们的接口只有一个方法,即一个API.

public String postxml(String xml).

用户只需要调用这一个接口即可,而功能的区分是通过传给系统的参数来实现的.每个类型的接口有固定的XML规范,用户只需根据规范将编写完的XML语句传给系统即可.

这样处理,抛开了以往接口的局限性,当需要加新功能时,只需要规定新的XML规范即可.当然,传给系统不同的XML,返回值也会不一样.

2.接口参数及返回值

2.1 管理收件人

public String postxml(String userspec);

2.1.1 参数的 xml 规范

```
<?xml version="1.0" encoding="UTF-8"?>
<userspec>
<taskname>xxx</taskname>
<cmd>insert-update</cmd><!-- or --><cmd>delete</cmd>
```

<grp id="xxx"/><!-- or --><grp name="xxx"/>

<users>

<user>

<name>xxx</name>

<email>12323213@qq.com</email>*

<company>xxx</company>

</user>

</users>

</userspec>

注:

调用此接口是做的异步处理,即在系统后台创建了一个上传收件人的任务.

其中,根节点**<userspec>**:是用来识别管理收件人功能的.

<taskname>:是用来命名此任务的名字,此名字主要用来在系统前台识别此任务.当然也可以不填,那样系统会自动创建名字,名字的规则是任务类型加当前时间.

<cmd>insert-update</cmd><!-- or --><cmd>delete</cmd>:用来规定此任务是做什么操作的,其中insert-update是做插入和修改操作的,如上传的收件人不存在则插入新的收件人,如果收件人存在,那么就是修改;delete是删除收件人的操作,只需要指明email就可以了.此命令只能选择一个.

<grp id="xxx"/><!-- or --><grp name="xxx"/>:是用来指明收件人上传到哪个收件人组,这两个也是只能选其一,其中如果选择<grp name="xxx"/>.那么首先寻找此name的收件人组,但是如果不存在此name,系统会以此name新建收件人组,然后把数据插到此组里面.

<users>:就是收件人的集合,里面一个<user>就是单个收件人信息,下面是几个固定的<user>字段:

<name>:代表收件人的名字;

<email>:收件人的email地址,此很重要一定要填写;

<company>:公司;

<gender>:性别,X 默认,M男,F 女;

<phone>:电话;

<cell>:手机;

<title>:职称;

<fax>:传真;

<reason>:备注;

<zipcode>:邮编.

当然,接口同样支持收件人多字段属性的定义,只要在前台高级设置-收件人字段定义处做好映射配置,填好相应的映射关系,然后收件人的属性里面加对应的节点就可以了

2.1.2 返回值的 xml 规范

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<status>xxx</status>
<comment>xxx</comment>
<taskid>xxx</taskid>
<taskname>xxx</taskname>
</result>
```

注:

<status>:代表此是否任务添加成功

<comment>:表明调用此接口的描述信息,如果是发生错误会写明错误信息

<taskid>:返回此接口添加的任务id,方便用户通过此id进行任务状态的查看

<taskname>:即此任务的名字.

2.2 发送邮件

public String postxml(String sendspec);

2.2.1 参数 sendspec 规范

```
<?xml version="1.0" encoding="UTF-8"?>
<sendspec>
<taskname>xxx</taskname>
<sendtime>2012-2-22T12:00:00</sendtime>
<catalog id = "xxx"/><!-- or --><catalog name = "xxx"/>
<name>xxx</name>
<target>
<allowed>
<grp id="123"/><!-- or --><grp name="123"/>
<users>
<user>
<name>花</name>
<email>12323213@qq.com</email>*
<company>会议</company>
```

```
</user>
<user>xxx</user>
</users>
</allowed>
<disallowed>
<grp id="123"/><!-- or --><grp name="123"/>
<users>
<user>
<name>花</name>
<email>12323213@qq.com</email>*
<company>会议</company>
</user>
<user>xxx</user>
</users>
</disallowed>
<unsub>true</unsub>
</target>
<emailspec>
<from>xxxx</from>
<reply>xxx</reply>
<subject>xxxx</subject>
<content>xxxx</content>
```

</emailspec>

</sendspec>

注:

<sendspec>:跟节点,用来识别接口的类型

<taskname>:用来创建任务的名字,没有此命令系统则会根据接口类型和时间自动创建名字

<sendtime>2012-2-22T12:00:00</sendtime>:用来规定发送时间,格式为yyyy-MM-dd'T'HH:mm:ss

<catalog id = "xxx"/><!-- or --><catalog name = "xxx"/>:用来规定邮件所在的主题名称,此命令只选其一即可.如名称不存在,系统会自动根据名称来创建新的主题.

<name>:是邮件的名字

<target>:里面放的是发送的目标

<allowed>:里面放的是准许发送的目标,里面包含组,和批量的联系人.这里面可以只写组,或者批量联系人,或者组和联系人都可以写.

<disallow>:即是排除这里面的收件人发送.当然里面也是可以放组和联系人的

<emailspec>:里面放就是邮件的内容了,

<from>:即是发件人的email地址

<reply>:回复地址

<subject>:邮件的主题

<content>:里面放的纯的html字符串,就是邮件的真正内容.系统会根据此html字符串,自动创建邮件模板

说明:此接口,也是产生任务在我们后台执行,所以也是属于异步执行的.

2.2.2 发送任务的返回值

<?xml version="1.0" encoding="UTF-8"?>

<result>

<status>xxx</status>

<comment>xxx</comment>

<taskid>xxx</taskid>

<taskname>xxx</taskname>

</result>

注:

<status>:代表此任务添加成功了没有

<comment>:表明调用此接口的描述信息,如果是发生错误会写明错误信息

<taskid>:代表返回的已添加任务id,用户可通过此id查看任务状态

<taskname>:此任务的名字.

2.3 邮件报告接口

public String postxml(String reportspec);

2.3.1 参数 reportspec 规范

<?xml version="1.0" encoding="UTF-8"?>

<reportspec>

<emailid>xxx</emailid>*<!-- or --><taskid>xxx</taskid>

<taskname>xxx</taskname>

<starttime>xxx</starttime>

<enddtime>xxx</endtime>

<type>XOP/XSU/XHB/XSB/XUS/XCK/XOW/XLG</type>

</reportspec>

注:

<emailid>xxx</emailid>*<!-- or --><taskid>xxx</taskid>:通过指明id查看目标,来获得相应的报告.这里的id,可以是邮件id,也可以是调用接口生成的taskid

<taskname>:生成报告也是任务的一种,故也可为此任务命名,同上,如果用户不填则系统自动以接口类型和时间创建任务名称

<starttime>:为数据开始时间.

<endtime>:为数据的结束时间.

<type>XOP/XSU/XHB/XSB/XUS/XCK/XOW/XLG</type>:用作指明报告的类型,只能选择其中一个.

XOP:打开数据

XSU:订阅数据

XHB:硬弹数据

XSB:软弹数据

XUS:退订数据

XCK:点击数据

XOW:总括数据

XLG:发送日志

系统的查看报告也是异步的形式,即调用此接口,需要给系统一个url地址,系统会以post的形式往此url地址push数据.

2.3.2 report 返回值

返回分为两种,其一是调用接口直接就能获得的返回值,即方法返回值,其二是通过url地址系统push过去的报告数据.

2.3.2.1. 方法返回值:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<status>xxx</status>
<comment>xxx</comment>
<taskid>xxx</taskid>
<taskname>xxx</taskname>
</result>
```

注:

<status>:代表此任务是否添加成功（NDP未处理,ERR出错,CMP处理完)

<comment>:表明调用此接口的描述信息,如任务发生错误会写明错误信息

<taskid>:代表返回的已添加任务id,用户可通过此id查看任务状态

<taskname>:即此任务的名字.

2.3.2.2push 返回值

push回去的数据,是根据传过来任务的type来区分的.这里以打开数据为例.

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<status>xxx</status>
```

<comment>xxx</comment>

<taskid>xxx</taskid>

<taskname>xxx</taskname>

<reports>

<index>xxx</index>

<pagesize>xxx</pagesize>

<opens>

<open>

<email>xx</email><opentime>xx</opentime><area>xx</area><ip>xx</ip>

</open>

<open>

<email>xx</email><opentime>xx</opentime><area>xx</area><ip>xx</ip>

</open>

</opens><!-- or -->

<clicks>

<click>

<email>xx</email><time>xx</time><area>xx</area><ip>xx</ip><url>xx</url>

</click>

</clicks>

...

</reports>

</result>

注:

<status>:代表处理的状态,其中(NDP未处理,ERR出错,INP处理中,CMP完成)

<comment>:表明调用此接口的描述信息,如果是发生错误会写明错误信息

<taskid>:代表返回的已添加任务id,用户可通过此id查看任务状态

<taskname>:即此任务的名字.

<reports>:里面就是报告的数据.

<index>:代表数据是从第几条记录开始的.

<pagesize>:代表包含多少条数据.

<opens>:就是打开数据的集合,里面每个<open>就是一个打开的数据报告

<eid>:邮件的id

<emailname>:邮件的名字

<email>:收件人的邮件地址

<opentime>:打开时间

<ip>:收件人打开的ip

<area>:打开的地区

<clicks>:就是点击的数据的集合,里面的每个<click>就是一个点击数据

<eid>:邮件的id

<emailname>:邮件的名字

<email>:收件人的邮件地址

<clicktime>:点击的时间

<ip>:点击的ip

<area>:点击的区域

<link>:点击的链接地址

<bounces>:就是弹回的数据集合,里面的每个<bounce>就是一个弹回数据

<eid>:邮件的id

<emailname>:邮件的名字

<email>:收件人的邮件地址

<bouncereason>:弹回的原因

<bouncesubtype>:软弹时的类型

<bouncetype>:弹回的类型,H是硬弹,S是软弹

<bouncetime>:弹回的时间

<emaillogs>:就是发送邮件的日志报告,每个<emaillog>就是一个发送日志

<eid>:邮件的id

<area>:收件人的区域

<areadetail>:区域的详细信息

<bouncereason>:弹回原因

<bouncesubtype>:软弹的类型

<bouncetype>:弹回的类型,H是硬弹,S是软弹

<clickcount>: 点击的次数

<domain>: 发送域名

<email>:收件人的email地址

<ip>:收件人点击的ip

<reason>:发送失败的原因

<sent>:表示是否发送,Y发送,N没发送

<bouncetime>:弹回的时间

<opentime>:打开时间

<senttime>:发送时间

<overviews>:代表总括的数据,其中每个<overview>就是一个总括的数据

<eid>:邮件的id

<emailname>:邮件的名字

<emailsenttotal>:发送的总数

<sentsuccessnum>:发送的成功数

<opennum>:打开数目

<hardbouncenum>:硬弹的总数

<softbouncenum>:软弹的总数

<clicknum>:点击的总数

<subs>:订阅的数据,<sub>是每个订阅数据

<created>:订阅的时间

<reason>:订阅的原因

<email>:订阅者的邮件地址

<name>:是订阅还是退订

<unsubs>:订阅的数据,<unsub>是每个退订数据

<created>:退订的时间

<reason>:退订的原因

<email>:退订者的邮件地址

<name>:表示订阅还是退订

这些数据,都是通过给定的url服务器地址,然后以post的形式push过去的,用户只需开发一个简单的类似servlet服务器端,就可以得到数据,继而根据节点得到相应的内容.传输数据的时候,数据都经过base64的encode,当用户得到数据时反解码就可以了.通过url传输的XML是经过加密的.在后面的安全性中详细介绍.

2.4 管理任务接口

public String postxml(String taskspec);

2.4.1 任务接口的 xml 规范

```
<?xml version="1.0" encoding="UTF-8"?>
<taskspec>
<taskid>xxx</taskid>
<type>getstatus/stoptask/starttask</type>
</taskspec>
```

注:

<taskid>:需要管理的任务的id,此每次调用接口的时候,都会返回此任务的id的.

<type>:处理任务的类型,有3种:

getstatus:获得此任务的状态

stoptask:取消此任务,当然此任务一般是未处理NDP的状态,才可以取消.

starttask:将取消的任务,重新运行,即重新变为未处理NDP状态.

一般来说发送邮件的任务,才可以停止任务和开始任务.而其他的任务一添加就开始执行.

2.4.2 管理任务的返回值

管理任务的返回值,根据不同的任务,会产生不同的返回值.而本身此接口不会添加任何一个任务,只是一个同步的执行语句.

任务分三种,上传收件人,发送邮件和查看报告.

管理收件人任务和查看报告任务的返回值规范相同,如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<processtaskid>xxx</processtaskid>
<processtaskname>xxx</processtaskname>
<status>xxx</status>
<comment>xxx</comment>
</result>
```

注:

<processtaskid>:处理的某个任务的id

<processtaskname>:处理某个任务的名字

<status>:处理的状态,NDP未处理,ERR出错,INP正在执行,CMP完成

<comment>:任务信息的描述,发送错误,会写明原因

2.4.3 发送邮件任务,特殊返回值

<processcontactnum>xxx</processcontactnum>

<processcontactnum>如是管理收件人的任务,此处返回值是已经处理了多少收件人.如是

报告任务,此节点就没有.

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<processtaskid>xxx</processtaskid>
<processtaskname>xxx</processtaskname>
<status>xxx</status>
<comment>xxx</comment>
<emailcampaign>
<eid>xxx</eid>
<emailsentstatus>xxx</emailsentstatus>
<emailtotal>xxx</emailtotal>
<emailsent>xxx</emailsent>
<emailfalse>xxx</emailfalse>
<emailsb>xxx</emailsb>
<emailhb>xxx</emailhb>
</emailcampaign>
</result>
```

注:

<processtaskid>: 处理的某个任务的id

<processtaskname>: 处理某个任务的名字

<status>: 处理的状态

<comment>: 任务信息的描述,发送错误,会写明原因发送邮件任务的返回值

<emailcampaign>：里面是放这封简要邮件发送情况

<eid>xxx</eid>：这封邮件的id

<emailsentstatus>：这封邮件的发送状态,EDT是编辑状态,SCD开始整理模板,CCG整理完模板,ECR整理完收件人,ECS发送中,CMP是完成发送

<emailtotal>：邮件发送的总数;

<emailsent>：发送成功的总数;

<emailfalse>：发送失败总数

<emailsb>：软弹的总数;

<emailhb>：硬弹的总数.

3.安全性

用Web Service本身就有安全机制,只要passwordcallback写好就可以了.系统会根据提交的用户名和密码来验证,同时得到对应的owner.

xml里面的值和属性,如<email>xxx</email>里面的值都是暗码,是用base64进行了encode过的,所以得到的值还需要反解析下.

关于调用报告接口的任务,传输数据是用push的形式,即是封装一个xml的字符串,然后通过post的形式传给指定的url地址.那么封装的xml字符串是经过加密的,加密代码如下:

```
byte[] plainText = xml.getBytes("UTF8");
Key key = new SecretKeySpec(bys, "AES");
byte[] cipherText = null;
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```

```
cipher.init(Cipher.ENCRYPT_MODE, key);

cipherText = cipher.doFinal(plainText);

Base64 b=new Base64();

byte[]_a=b.encode(cipherText);

String sb=new String(a,"utf-8");

String secret = "secret=" + sb +"&";

String content= "content=" + md5(xml);
```

用户只需得到两个值，一是secret,这是加密后而且通过base64编码过后的字符串,二是content,是通过MD5加密后生成的字节码.得到secret后,先通过base64反解码,再通过SecretKeySpec来反解,这样就得到真正的XML值,再用MD5加密XML真正的值,将字节码和传过来的字节码匹对,如一致则说明数据传输过程中没有掉包或被更改，可认定此数据是安全的,可以进行下一步的数据处理了.需对xml节点里面的值,进行base64反解析.

注:用户获得自身的密钥,在我们系统的前台可以得到,如图:

通过这种安全机制,衍生出新的接口形式.就是我们系统提供一个url地址,然后用户只需要按这种安全机制加密和编码,然后以post的形式传给我们就可以调用接口了,而传给我们的xml也就是web service里面的参数.url等上线的时候,客服会告诉客户的.

4.用户接口前台页面的配置

此处,专门查看通过接口创建的任务信息,还包括系统自动创建的任务.

注:

url:是用来传输报告的push的url地址,即是用户自己的服务器地址.系统用来push报告数据的url地址.

类型:现在只有一种即是获得报告类型,

Isactive:是否是激活的,如果没有激活则不push报告数据.

注:

这里面的配置是个新功能,即通过此配置,用户可以定义在发送完邮件之后是否push此邮件的报告.

开始时间,是指在邮件发送之后的多少个小时开始传输这封邮件的报告(建议是4个小时).

结束时间即这封邮件不能开始之后每天都push这封邮件的报告,都有截止日期,此就是说发送邮件之后传输多天的数据,建议填15天.即15天之后系统不自动push数据了.

传输的频率即是每隔多少时间传输一次报告,建议是24小时,即是每天传输一次这封邮件的报告

isactive即是否激活.不激活是不push数据的.

报告类型分很多种:打开数据,点击数据,弹回,退订,订阅,发送日志.总括.这些,用户可以根据自己的需要选择即可.