IMS语音用户拨测终端接口V2

**IMS语音用户拨测需求说明：**

1. IMS拨测是作为PON 112 WEB界面的一个单项测试功能来实现。
2. 用户在PON 112 WEB界面输入IMS号码，点击抓线，抓线成功后，有IMS语音仿真测试权限的用户显示IMS语音仿真测试按钮；
3. 用户点击IMS语音仿真测试
4. 系统显示IMS语音仿真测试界面。

**拨打测试过程：**

1. 用户输入被叫号码（系统可设置缺省的被叫号码，用户可编辑）；
2. 系统将抓线获取的用户账号、密码等用户侧参数缺省填入，可修改。
3. 用户点击仿真测试按钮发起仿真测试；
4. 系统发起IMS语音仿真测试，直到呼叫成功/呼叫失败/超时/异常
5. 系统在界面显示仿真测试结果（呼叫成功/呼叫失败/超时/异常）和本次仿真测试用时
6. 如呼叫不成功，显示问题产生的原因。（待扩展）
7. 用户点击详细按钮，可查看本次呼叫相关的详细参数。（待扩展）
8. 如可能，可让用户查询本次呼叫的详细协议交互过程。（待确认）

# 拨打测试接口：

界面与后台之间的接口方式为：WEBSERVICE接口；

界面通过http的get方式调用后台的webservice后，后台立即返回一个会话号。然后后台进行实际测试工作，并保存测试过程和结果。

前台可以通过get方式调用查询测试进度的方法，得到测试进度和结果。

后台返回的数据类型为JSON。

## 1.发起默认帐号测试（上层系统请求拨测模块）

用于使用默认帐号呼叫待测用户，检查其作为被叫的情况。

最短请求方式：

<http://192.168.203.21:78/calling?cmd_type=calling&called_party=12345678>

cmd\_type 参数为calling，指定请求的类型是发起呼叫测试

called\_party参数为被叫号码，一般为用户号码。

以上两个参数必填。

带上其他可选参数的请求方式：

<http://192.168.203.21:78/calling?cmd_type=calling&called_party=12345678&timeout=25&gateway_ip=118.85.214.1&gateway_port=5060>

timeout参数指定自动挂机的时间，如果不带该参数，则拨测终端使用配置文件中的值。携带该参数将在本次测试会话中覆盖配置文件中的值。

gateway\_ip和gateway\_port 参数指定在本次会话中使用的sip网关地址。如果不带该参数，则拨测终端使用配置文件中的值。

接到这个请求后，拨测终端将检查现有的默认帐号测试通道中是否有空闲通道，如有，则使用该通道发起测试，并回复成功应答；否则，回复失败应答：

**发起测试应答**

**{"progress": "NA", "reason": "0", "result": "NA","session\_no":"1021", "status": "1"}**

Status为1代表拨测终端已通过初步检查，正在发起异步拨测；

session\_no为会话号，由拨测终端服务分配，用来标识一通测试任务，以便界面后期用来指定特定测试的详细参数和信令流程。

其他参数无关。

**{"progress": "NA", "reason": "1999", "result": "NA", "session\_no": "0", "status": "0"}**

Status为0代表拨测请求遇到问题。此时session\_no固定为0。

Reason为1400，1500，1999中一项，具体意义参见reason编码表。

其他参数无关。

**Reason编码表**

|  |  |  |
| --- | --- | --- |
| Reason码 | 意义 | 备注 |
| 0 | 正常（无reason指示） |  |
| 1400 | 请求参数格式不正确 | 发起测试接口的应答中携带 |
| 1500 | 拨测终端内部错误（拨测模块离线） | 同上 |
| 1999 | 拨测资源忙（指定的账户正在测试中） | 同上 |
| **2480** | **对端失效（"480"）** | **查询测试进度的应答中携带** |
| **2486** | **被叫忙（" 486"）** | **同上** |
| **24xx** | **对端异常（4xx）** | **同上** |
| **2902** | **起呼失败（"Intial INVITE failed!"）** | **同上** |
| **3001** | **注册失败（"initial registerfailed!"）** | **同上** |
| **3002** | **鉴权失败（"Add\_authentication\_info failed!"）** | **同上** |
| **3003** | **注册失败（"send register msg failed!"）** | **同上** |
| **34xx** | **注册失败，代码4xx（"register receive 4XX msg failed!"）** | **同上** |
| **35xx** | **注册失败，代码5xx（"register receive 5XX msg failed!"）** | **同上** |
| **5001** | **请求失败（"request failure!"）** | **同上** |
| **5002** | **网关失败（"server failure!"）** | **同上** |
| **5003** | **全局错误（"global failure!"）** | **同上** |

## 2.发起指定帐号测试（上层系统请求拨测模块）

类似接口1，需要在参数中携带用户帐号信息。

<http://192.168.203.21:78/calling?cmd_type=calling&called_party=10000&account=33647934&password=123456>

也可以加入可选参数，可选参数用法和意义同接口1。

本接口的应答消息同接口1

## 3.查询测试进度（上层系统请求拨测模块）

<http://192.168.203.21:78/calling?session_no=1021&cmd_type=querycalling>

session\_no为会话号，用来标识一通测试任务；

cmd\_type为命令类型，querycalling代表请求查询指定主叫测试进度；

**后台返回测试进展数据**

**{"session\_no":"1021","progress":"register\_calling","status":"processing","result":"ok","reason":"0"}**

**progress为测试阶段，**

**完整的测试阶段为register\_calling\_ringtone\_answer\_hangon\_unregister**

**可能缺失其中一项或多项；**

Status为测试会话状态 processing, end

result为拨测结果，测试总结果 ok, fail, timeout；

reason为具体原因说明，具体意义参见上节reason编码表

## 4.请求详细参数（上层系统请求拨测模块）

<http://192.168.203.21:78/params?session_no=1021&cmd_type=params>

session\_no为会话号，用来标识一通测试任务；

cmd\_type为命令类型，params代表请求详细参数命令；

**后台返回呼叫相关的详细参数**

**{"session\_no":"131","params":[{"param\_name":"CallingNum","param\_value":"999"},{"param\_name":"CalledNum","param\_value":"555"}]}**

每个参数作为一个对象，分别有参数名和参数值。多个参数对象形成JSON的数组，归属于params属性。

## 5.请求协议过程（暂不实现）

[http://TestServer:8010/[packagepath]/processes?session\_no=131&cmd\_type=processes](http://testserver:8010/[packagepath]/processes?session_no=131&cmd_type=processes)

ession\_no为会话号，用来标识一通测试任务；

cmd\_type为命令类型，processes代表请求协议过程命令。

后台返回呼叫的详细协议交互过程

{"session\_no":"131",

"processes":[

{"process\_no":"1","process\_direction":"forward","process\_name":"invite","process\_time":"2014-05-05 hh:mm:ss.nnn","process\_describe":"xxxxx"},

{"process\_no":"2","process\_direction":"backward","process\_name":"200ok","process\_time":"2014-05-05 hh:mm:ss.nnn","process\_describe":"xxxxx"}

]}

每个协议过程作为一个对象，多个协议过程对象形成JSON的数组，归属于processes属性。

每个协议对象含有：

process\_no为协议交互的顺序号，

process\_direction为协议方向，

process\_name为协议消息的类别，

process\_time为协议消息时间戳，格式为YYYY-MM-DD hh:mm:ss.nnn，

process\_describe为协议消息具体内容