

# IP-PBX WebSocket API – User Guide

## Introduction

WebSocket API allows the user to establish a persistent connection with a Grandstream IPPBX. This bidirectional communication allows querying from the client side and pushing event notifications from the server once an event occurs.

## Transaction Headers

To send requests to the PBX, such as logging in, the user should send the queries to the following URL:

wss://[PBX IP]:8089/websockify

### Example:

Assuming that the IP address of the PBX is 192.168.129.49, the URL to use is `wss://192.168.129.49:8089/websockify`

### Request Headers:

```
GET wss://192.168.129.49:8089/websockify HTTP/1.1
Host: 192.168.129.49:8089
Connection: Upgrade
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36
Upgrade: websocket
Origin: https://192.168.129.49:8089
Sec-WebSocket-Version: 13
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: TRACKID=ldb3900441892dddd27618e23f1b4948; CookieName=CookieValue; session-identify=sid1103415366-1622163971;
Sec-WebSocket-Key: Uh+T3El0eIYba6xPNUFyRA==
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
```

### Response Headers:

```
Connection: upgrade
Content-Security-Policy: default-src 'self' 'unsafe-eval' 'unsafe-inline';
Date: Tue, 29 Nov 2022 06:16:59 GMT
Sec-WebSocket-Accept: VMAYaz9KdVgFkCk1lJJ2aSj1aaw=
Server: nginx
Set-Cookie: CookieName=CookieValue;HttpOnly;Secure;SameSite=Lax
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Upgrade: WebSocket
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1
```

## Session Control

### Challenge

To log in, the application/user needs to send a challenge request to generate the challenge number.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxa",
    "action": "challenge",
    "username": "cdrapi",
    "version": "1"
  }
}
```

The field “user” needs to match the username configured on the API on the IPPBX, otherwise, the challenge will not be achieved successfully.

Once the request above is sent, the response should look similar to the response below.

```
{
  "response": {
    "challenge": "0000001652831717"
  },
  "status": 0
}
```

## Login

WebSocket uses a long-period logging session which remains open until the user logs out. When logging in, the cookie indicates the session needs to be generated. If a cookie has been previously generated using HTTPS API, that same cookie can be used to send the queries.

## Heartbeat

Heartbeat is a mechanism to keep the WebSocket session open. The heartbeat should be sent from the client within a time interval, otherwise, the server will consider the session as terminated and will require logging in again to open the session.

### Request Example

The query format is as follows:

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxe",
    "action": "heartbeat"
  }
}
```

### Response Example

The response format is as follows

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxe",
    "action": "heartbeat",
    "status": "0"
  }
}
```

## Log Out

A log-out query will instruct the server to terminate the session.

### Request Example

The query format is as follows:

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxe",
    "action": "logout"
  }
}
```

### Response Example

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxe",
    "action": "logout",
    "status": "0"
  }
}
```

# Subscription

## Subscribe

Subscribe request allows sending certain event messages to the client from the IPPBX. The event messages will be pushed upon the event happening, e.g., network interface going down, trunk call established, extension creation, etc...

### Request Parameters

| Keywords   | Value | Type   | Description                                                                                                                                                                                                |
|------------|-------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eventnames |       | string | to be subscribed is composed of an array. You can subscribe to multiple fields at the same time. For example, "eventnames":["dd","ee","ff"] means subscribing to events dd , ee , and ff at the same time. |

### Request Example

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "ExtensionStatus",
      "ConferenceStatus",
      "TrunkStatus"
    ]
  }
}
```

### Response Example

- Successful Response

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "status": 0
  }
}
```

- Failed Response

If there is no corresponding event, or there is no subscription permission, a list of incorrect event names is returned:

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "xxxx"
    ] "status": -9
  }
}
```

## Tiered Subscriptions

Some events support hierarchical subscription, which means you can subscribe to the events of an extension or a range of extensions, such as ExtensionStatus.

### Request Example

Only subscribes to events from extension 1006

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "subscribe",
    "eventnames": [
      "ExtensionStatus/1006"
    ]
  }
}
```

Events from extension 1005 to extension 1007

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "subscribe",
    "eventnames": [
      "ExtensionStatus/1005~1007"
    ]
  }
}
```

Subscribe to all extension events

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "subscribe",
    "eventnames": [
      "ExtensionStatus"
    ]
  }
}
```

## Unsubscribe

An unsubscribe request is used to stop receiving events to which the client had previously subscribed.

### Request Parameters

| Keywords   | Value | Type   | Description                                                                                                                                                                                                                                                               |
|------------|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eventnames |       | string | to be unsubscribed is composed of an array. You can unsubscribe from multiple events at once. For example, "eventnames":["dd","ee","ff"] means unsubscribing from events dd , ee , and ff at the same time . For specific supported event types, see [Event Notification] |

### Request Example

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxd",
    "action": "unsubscribe",
    "eventnames": [
      "ExtensionStatus",
      "ConferenceStatus",
      "TrunkStatus"
    ]
  }
}
```

#### o Successful Response

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxd",
    "action": "unsubscribe",
    "status": 0
  }
}
```

#### o Failed Response

```
{
  "type": "response",
  "message": {
    "transactionid": "123456789zxd",
    "action": "unsubscribe",
    "eventnames": [
      "xxxx"
    ],
    "status": -9
  }
}
```

## Tiered Unsubscription

### Request Example

Unsubscribe only from events for extension 1006

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "unsubscribe",
    "eventnames": [
      "ExtensionStatus/1006"
    ]
  }
}
```

Unsubscribe from events from extension 1005 to extension 1007

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "unsubscribe",
    "eventnames": [
      "ExtensionStatus/1005~1007"
    ]
  }
}
```

Unsubscribe all extension events

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "unsubscribe",
    "eventnames": [
      "ExtensionStatus"
    ]
  }
}
```

## Notify

All events are reported by UCM proactively , so the type is requested. The action type of event notification is notified. After the subscription is successful, each time an event is generated, the page will receive the corresponding subscribed event.

| Keywords | Value | Type | Illustrate |
|----------|-------|------|------------|
|----------|-------|------|------------|



|            |  |        |                                                                                                                                                                                                                                             |
|------------|--|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eventnames |  | string | The event name. For specific supported event type, see [Event Notification]                                                                                                                                                                 |
| eventbody  |  | string | Event message body, generally global events, such as InterfaceStatus event body is {}, other eventswith multiple members belonging to the event, such as ExtensionStatus use [] array format. For details, see [Event Notification Details] |
| action     |  | string | For member, the value is add, update, or delete, indicating that the member is added, updated, or deleted.                                                                                                                                  |

Response Example

Event reporting message:

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ExtensionStatus",
    "eventbody": [
      {
        "extension": "6000",
        "status": "Idle"
      }
    ]
  }
}
```

## Event Notification Details

### PbxStatus

Subscribing to the PbxStatus will notify the user when an extension is registered, a call is made, or when a parking lot is used.

Event Parameters

| Item               | Value                                                                                        |
|--------------------|----------------------------------------------------------------------------------------------|
| calls_num          | Current call number                                                                          |
| extension_register | Successfully registered on UCM, including SIP extensions, IAX extensions, and FXS extensions |
| meetroom_inuse     | Meeting rooms in use                                                                         |
| parking_inuse      | Number of parking spaces currently in use                                                    |
| extension_count    | Number of extensions created                                                                 |
| meetroom_count     | Number of meeting rooms                                                                      |
| queue_count        | Number of call queues                                                                        |
| parking_count      | Number of parking lots                                                                       |
| dynamic_defense    | Dynamic Defense status                                                                       |

|                          |                                          |
|--------------------------|------------------------------------------|
| auto_sync                | Automatic Sync Status                    |
| auto_clean               | Automatic cleanup status                 |
| regular_backup           | Automatic backup status                  |
| fail2ban                 | Fail2ban status                          |
| available_trunk_number   | Number of available trunks               |
| unavailable_trunk_number | Number of unavailabe trunks              |
| busy_trunk_number        | Number of trunks in use                  |
| unmonitored_trunk_number | Number of trunks which are not monitored |

Example

subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "PbxStatus"
    ]
  }
}
```

When the number of registered extensions changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PbxStatus",
    "eventbody": {
      "extension_register": 8
    }
  }
}
```

When a call is established or terminated, the event will be reported.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PbxStatus",
    "eventbody": {
      "extension_register": 8
    }
  }
}
```

When the number of used meeting rooms changes.



```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "videoconfroom_inuse": 1
    }
  }
}
```

When the number of used parking lots changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "parking_inuse": 1
    }
  }
}
```

When the number of created extensions changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "extension_count": 8
    }
  }
}
```

When the number of created meeting rooms changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "meetroom_count": 3
    }
  }
}
```

When the number of call queues changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "queue_count": 3
    }
  }
}
```

When the number of extensions of the parking lot changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "parking_count": 25
    }
  }
}
```

When the status of Dynamic Defense changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "dynamic_defense": "yes"
    }
  }
}
```

When the data sync status changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "auto_sync": "yes"
    }
  }
}
```

When the automatic cleaner status changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "auto_clean": "yes"
    }
  }
}
```

When the scheduled backup status changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "regular_backup": "yes"
    }
  }
}
```

When Fail2ban status changes.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "fail2ban": "yes"
    }
  }
}
```

When the trunk status changes (A SIP trunk is added)

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PBXStatus",
    "eventbody": {
      "available_trunk_number": 0,
      "unavailable_trunk_number": 3,
      "busy_trunk_number": 0,
      "unmonitored_trunk_number": 6
    }
  }
}
```

## EquipmentCapacityStatus

When the client subscribes to the equipment capacity status, the client will be notified when the USB flash drive or the SD card is plugged into the device.

### Event Parameters

| Item       | Value                                                                                                                            |
|------------|----------------------------------------------------------------------------------------------------------------------------------|
| disk-total | External memory parameter. The value is stored in the value of the internal storage.                                             |
| diskname   | The name of the storage disk. Generally sdaX indicates a USB flash drive, and mmcblk1pX indicates an SD card.                    |
| value      | The total of available space on the storage device.                                                                              |
| disk-avail | Indicates the available space of the external memory storage device. The value is stored in the value of the internal structure. |
| disk-usage | The used space of the external memory device, the value is stored in the value if the internal structure.                        |

Example

Subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "EquipmentCapacityStatus"
    ]
  }
}
```

A USB flash drive is inserted, an event will be triggered reporting the existing internal and external storage units.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "EquipmentCapacityStatus",
    "eventbody": {
      "disk-total": [
        {
          "diskname": "sda1",
          "value": "14896"
        }
      ],
      "disk-avail": [
        {
          "diskname": "sda1",
          "value": "5656"
        }
      ],
      "disk-usage": [
        {
          "diskname": "sda1",
          "value": "62"
        }
      ]
    }
  }
}
```

## InterfaceStatus

When the client is subscribed to InterfaceStatus, the event will be sent every time the hardware interface status is updated. If it is an O port or an S port, there will be a chan parameter whose value is the number., starting from 1. For example, for UCM6308, the O port chan is 1~8, and the S port number is also 1~8. The content of the hardware interface status is as follows:

### Event Parameters

| Item        | Value                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------|
| status      | The possible values are: Idle, InUse                                                                                  |
| LAN         | LAN status, the value is linked or unlink (routing or switching mode)                                                 |
| WAN         | WAN status, the value is linked or unlink (only in Route mode)                                                        |
| LAN1        | LAN1 status, the value is linked or unlink (only in Dual mode)                                                        |
| LAN2        | LAN2 status, the value is linked or unlinked (only in Dual mode)<br>Note: Only on the UCM6304, UCM6308, and UCM6308A. |
| HBT         | Heartbeat port status, the value is linked or unlinked (only for the UCM6300/A Series)                                |
| NetHDLC1    | The possible value is linked and unlinked (only for the UCM6300/A Series)                                             |
| power-poe   | LAN PoE, the value is 0 or 1.                                                                                         |
| power-port1 | Power port 1, the value is Work, Disconnected, Abnormal, or Unknown.                                                  |
| power-port2 | Power port 1, the value is Work, Disconnected, Abnormal, or Unknown.                                                  |

### Example

#### Subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "InterfaceStatus"
    ]
  }
}
```

Analog relay incoming (port O is chan 4)

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-fxo": [
        {
          "status": "InUse",
          "chan": 4
        }
      ]
    }
  }
}
```

FXS port number FXS1 rings

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-fxs"{
        "status": "InUse",
        "chan": 1
      }
    }
  }
}
```

Remove the USB drive

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-usbdisk": {
        "status": "false"
      }
    }
  }
}
```

Network interface, Route mode

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-network": {
        "LAN": "unlink",
        "WAN": "linked",
        "HBT": "unlink",
        "NetHDLC1": "unlink"
      }
    }
  }
}
```



Network interface, Dual mode

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-network": {
        "LAN1": "unlink",
        "LAN2": "linked",
        "HBT": "unlink",
        "NetHDLCl": "unlink"
      }
    }
  }
}
```

Network interface, Switch mode

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-network": {
        "LAN": "linked",
        "HBT": "unlink",
        "NetHDLCl": "unlink"
      }
    }
  }
}
```

Power interface

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "power-poe": 0,
      "power-port1": "Work",
      "power-port2": "Disconnected"
    }
  }
}
```

ISDN port

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "InterfaceStatus",
    "eventbody": {
      "interface-pri": [
        {
          "span": 3,
          "span_type": "E1",
          "lbo": 0,
          "alarm": 1800,
          "tx_level": 0,
          "rx_level": 0,
          "framing_errors": 0,
          "crc_errors": 0,
          "code_violations": 0,
          "ebit_count": 0,
          "general_errorred_seconds": 1
        }
      ]
    }
  }
}
```

# Call Status

## ActiveCallStatus

When subscribed to ActiveCallStatus.

### Event Parameter

| Item          | Value                                                                                       |
|---------------|---------------------------------------------------------------------------------------------|
| uniqueid      | Channel identifier                                                                          |
| state         | Channel status (Up, Ringing...)                                                             |
| service       | Channel Type                                                                                |
| callername    | Caller Name                                                                                 |
| callernum     | Caller ID                                                                                   |
| connectednum  | The number of connected peer (when the channel is not bridged, it is displayed as s)        |
| connectedname | Connected peer name                                                                         |
| linkedid      | LinkedID is based on uniqueID, but extends to other channels such as call transfers, etc... |
| chantype      | Current channel type (bridge, unbridge)                                                     |
| bridge_id     | Identifier of the channel bridge                                                            |
| alloc_time    | Channel change time (the first time is the channel creation time)                           |



|        |                                                                                                              |
|--------|--------------------------------------------------------------------------------------------------------------|
| action | The value is add, update, or delete. Indicating that the channel operation is adding, updating, or deleting. |
|--------|--------------------------------------------------------------------------------------------------------------|

Example

subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "ActivityCallStatus"
    ]
  }
}
```

4000 dials 4001

The display of the call status occurs in two steps. First, two unbridged channels are added, then they are correspondingly updated. Therefore, four events are pushed at the same time. The order of the events for each channel may change.

Added calling channel PJSIP/4000-000000d and called channel PJSIP/4001-000000e

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "add",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "",
          "callernum": "4000",
          "callid": "",
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge",
          "connectedname": null,
          "connectednum": "4001",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Ring",
          "uniqueid": "1669809315.28"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0033297016"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "add",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "BBB CV",
          "callernum": "4001",
          "callid": "",
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge",
          "connectedname": null,
          "connectednum": "s",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Down",
          "uniqueid": "1669809315.29"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0033297016"
    }
  ],
  "type": "request"
}

```

Update the calling channel status to ring and the called channel status to ringing

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "update",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "",
          "callernum": "4000",
          "callid": "",
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge",
          "connectedname": "BBB CV",
          "connectednum": "4001",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Ring",
          "uniqueid": "1669809315.28"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0033297016"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "update",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "BBB CV",
          "callernum": "4001",
          "callid": "",
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge",
          "connectedname": "4000",
          "connectednum": "4000",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Ringing",
          "uniqueid": "1669809315.29"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0033297016"
    }
  ],
  "type": "request"
}

```

## 4001 connected

Connecting a call occurs in two steps; First, both unbridged channels are updated, and then add a bridged channel and delete both unbridged channels. Five events are pushed at the same time. The order of the may change.

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "update",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "4000",
          "callernum": "4000",
          "callid": "",
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge",
          "connectedname": "BBB CV",
          "connectednum": "4001",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Up",
          "uniqueid": "1669809315.28"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "1273412080"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "update",
          "alloc_time": "2022-11-30 19:55:15",
          "callername": "BBB CV",
          "callernum": "4001",
          "callid": "",
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge",
          "connectedname": "4000",
          "connectednum": "4000",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "1669809315.28",
          "outbound_trunk_name": "",
          "service": "normal",
          "state": "Up",
          "uniqueid": "1669809315.29"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "1273412080"
    }
  ],
  "type": "request"
}

```

Then add a new bridge channel and delete the two unbridged channels

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "add",
          "bridge_id": "014f7481-3f4e-44af-889d-ec11e562eb24",
          "bridge_time": "2022-11-30 19:55:26",
          "callerid1": "4000",
          "callerid2": "4001",
          "channel1": "PJSIP/4000-0000000d",
          "channel2": "PJSIP/4001-0000000e",
          "chantype": "bridge",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "have_send": 1,
          "inbound_trunk_name": "",
          "name1": "4000",
          "name2": "BBB CV",
          "outbound_trunk_name": "",
          "uniqueid1": "1669809315.28",
          "uniqueid2": "1669809315.29"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "1273412080"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "delete",
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "1273412080"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "delete",
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "1273412080"
    }
  ],
  "type": "request"
}

```

## 4000 hangs up the call

Hanging up a call occurs in two steps: First, Unbridge the bridged channel, and then hang up the two unbridged channels. Five events are pushed at the same time. The order may change.

The bridged channel is disassembled. The channel changes from bridged to unbridged (in case of a call transfer, the channel may not be hung up directly after it becomes unbridged, so hanging up the call is divided into two steps): A channel with action “delete” and chantype “bridge” will be pushed; at the same time, two channels with action “add” and chantype “unbridge” will be pushed, including the specific content of the channel (caller, called party, time, channel name, etc.)

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "delete",
          "channel": "PJSIP/4000-0000000d",
          "channel2": "PJSIP/4001-0000000e",
          "chantype": "bridge"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0999433754"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "add",
          "alloc_time": "2022-11-30 19:55:26",
          "callername": "BBB CV",
          "callernum": "4001",
          "callid": null,
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge",
          "connectedname": "4000",
          "connectednum": "4000",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "",
          "outbound_trunk_name": "",
          "service": "macro-dial",
          "state": "Ring",
          "uniqueid": "1669809315.29"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0999433754"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "add",
          "alloc_time": "2022-11-30 19:55:26",
          "callername": "4000",
          "callernum": "4000",
          "callid": null,
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge",
          "connectedname": "BBB CV",
          "connectednum": "4001",
          "dial_service": "normal",
          "feature_calleename": "",
          "feature_calleenum": "",
          "feature_name": "",
          "feature_num": "",
          "inbound_trunk_name": "",
          "linkedid": "",
          "outbound_trunk_name": "",
          "service": "macro-dial",
          "state": "Ring",
          "uniqueid": "1669809315.28"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0999433754"
    }
  ],
  "type": "request"
}

```

Hang up 2 unbridged channels, and push two more channels with the action “delete” and chantype “unbridge”



```
{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "delete",
          "channel": "PJSIP/4001-0000000e",
          "chantype": "unbridge"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0999433754"
    },
    {
      "action": "notify",
      "eventbody": [
        {
          "action": "delete",
          "channel": "PJSIP/4000-0000000d",
          "chantype": "unbridge"
        }
      ],
      "eventname": "ActiveCallStatus",
      "transactionid": "0999433754"
    }
  ],
  "type": "request"
}
```

ExtensionsStatus

When the client is subscribed to the extension status, this event will be sent every time the extension status is updated. The event name is ExtensionStatus, exten indicates the extension number, and the extension status content is as follows:

Event Parameters

| Item   | Value                                                                                                                                                                             |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| status | The status of the extension, the values are <b>Idle</b> , <b>InUse</b> , <b>Busy</b> , <b>Unavailable</b> , <b>Ringing</b> .                                                      |
| addr   | The IP port of the extension. When one port is used for multiple machines, it may be composed of multiple IP and port numbers, such as: 192.168.124.168:5062, 192.168.124.77:5060 |

Example

Subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "ExtensionStatus"
    ]
  }
}
```

Extension 1121 is ringing

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ExtensionStatus",
    "eventbody": [
      {
        "extension": "1121",
        "status": "Ringing"
      }
    ]
  }
}
```

Extension 1002 registered

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ExtensionStatus",
    "eventbody": [
      {
        "extension": "1002",
        "status": "Idle",
        "addr": "192.168.124.168:5062"
      }
    ]
  }
}
```

TrunkStatus

When the client is subscribed to the trunk status, this event will be pushed every time a trunk status is updated. The event name is TrunkStatus. The subscription ID is trunk\_id, and the trunk status content is as follows:

Event Parameters

| Item   | Value                                                                                                                                                                                      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| status | The status of the relay. The values are <b>Unknown, Unmonitored, Reachable, Unreachable, Registered, Unregistered, Lagged, Failed, Request Sent, Rejected, Timeout, No Authentication.</b> |

Example

Registered trunk trunk\_6 changes to unregistered.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "TrunkStatus",
    "eventbody": [
      {
        "trunk_index": 6,
        "status": "Unregistered"
      }
    ]
  }
}
```



ConferenceForVideoStatus

When the client is subscribed to multimedia meeting room status, this event will be sent every time the meeting room is updated. The event name is ConferenceStatus, and exten is the meeting room number. The meeting room status content is as follows:

| Item         | Value                                                                                                                                         |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| attend_count | Number of attendees, including admin members and ordinary members.                                                                            |
| start_time   | Meeting start time, that is, the time when the first member enters.                                                                           |
| user_no      | 1 for the first member, 2 for the second number, etc...                                                                                       |
| is_locked    | Whether the meeting room is locked, 1 for locked, 0 for unlocked.                                                                             |
| caller_id    | CallerID number of the member extension.                                                                                                      |
| caller_name  | CallerID name of the member.                                                                                                                  |
| channel_name | Member channel name.                                                                                                                          |
| join_time    | Time for members to enter the meeting room.                                                                                                   |
| is_admin     | Whether the member is an administrator, 1 indicates that the member is an administrator, 0 indicates that the member is not an administrator. |
| is_talking   | Whether the member is speaking, 1 means the member is speaking, 0 means that the member has stopped speaking.                                 |
| is_muted     | Whether the member is banned, 1 means the member is banned, 0 means that the member is either not banned or the ban has been lifted.          |

Example

4001 calls meeting room 6302 and reports the number of members, number of administrators, and conference room status time.

Membership:

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "attend_count": 1,
          "conf_number": "6302",
          "meet_type": 1,
          "member": [
            {
              "action": "add",
              "channel_name": "PJSIP/4001-00000003",
              "is_admin": 0,
              "join_time": "2022-11-30 18:07:11",
              "media_status": "audio",
              "member_name": "BBB CV",
              "member_number": "4001",
              "user_no": 1
            }
          ],
          "start_time": "2022-11-30 18:07:06"
        }
      ],
      "eventname": "ConferenceForVideoStatus",
      "transactionid": "0908360792"
    }
  ],
  "type": "request"
}

```

4000 banned:

To mute, set is\_audio\_muted to yes; to unmute, set is\_audio\_muted to no

```

{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "admin_count": 1,
          "attend_count": 2,
          "conf_number": "20708001",
          "member": [
            {
              "action": "update",
              "is_audio_muted": "yes",
              "member_number": "4000"
            }
          ]
        }
      ],
      "eventname": "ConferenceForVideoStatus",
      "transactionid": "0853032737"
    }
  ],
  "type": "request"
}

```

4000 leaves the meeting room:

Report the number of members and administrators at the same time:

```
{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "admin_count": 1,
          "attend_count": 1,
          "conf_number": "20708001",
          "member": [
            {
              "action": "delete",
              "member_number": "4000"
            }
          ]
        }
      ],
      "eventname": "ConferenceForVideoStatus",
      "transactionid": "1634825916"
    }
  ],
  "type": "request"
}
```

Leave

4001 end the meeting and set the meeting room to unlock:

```
{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "admin_count": 0,
          "attend_count": 0,
          "conf_number": "20708001",
          "is_locked": 0,
          "member": [
            {
              "action": "delete",
              "member_number": "4001"
            }
          ],
          "start_time": ""
        }
      ],
      "eventname": "ConferenceForVideoStatus",
      "transactionid": "1304950212"
    }
  ],
  "type": "request"
}
```

VoiceMailStatus

When a client is subscribed to voicemail status, an event will be pushed every time the voicemail status is updated. The event name is VoiceMailStatus and the status content is as follows.

Event Parameters

| Item      | Value                       |
|-----------|-----------------------------|
| extension | Voicemail extension number. |
| urgemsg   | Urgent message.             |

|        |                  |
|--------|------------------|
| newmsg | Unread messages. |
| oldmsg | Read messages.   |

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "VoiceMailStatus",
    "eventbody": [
      {
        "extensions": "1000",
        "urgemsg": 1,
        "newmsg": 0,
        "oldmsg": 0
      }
    ]
  }
}
```

When a new message is received in the voicemail inbox of extension 1000

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "VoiceMailStatus",
    "eventbody": [
      {
        "extensions": "1000",
        "urgemsg": 0,
        "newmsg": 1,
        "oldmsg": 0
      }
    ]
  }
}
```

After the message in the voicemail inbox 1000 is listened to.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "VoiceMailStatus",
    "eventbody": [
      {
        "extensions": "1000",
        "urgemsg": 0,
        "newmsg": 0,
        "oldmsg": 1
      }
    ]
  }
}
```

CallQueueStatus

CallQueueStatus event notification includes two parts, one is the configuration information of the queue, and the other is the status of its members (i.e., agents)

Queue-related Configuration Information

Event Parameters

| Item               | Value                                                                                                                                                                                                                                                                     |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| extension          | Indicates the extension number of the queue.                                                                                                                                                                                                                              |
| strategy           | This field indicates the ringing strategy of the queue. There are ringall (all ringing), linear (linear ringing), leastrecent (longest idle priority ringing), fewestcalls (least calls priority ringing), random (random ringing), rrmemory (memory round-robin ringing) |
| queue_chairman     | This field indicates the chairman of the queue (that is, the administrator of the queue). It can be empty.                                                                                                                                                                |
| enable_agent_login | The values of this field are: "yes" and "no". When it is "yes", static agents added to the queue need to log in before joining the queue and dynamic agents cannot join the queue.                                                                                        |

Example

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "CallQueueStatus",
    "eventbody": [
      {
        "extension": "6500",
        "strategy": "leastrecent",
        "queue_chairman": "10615",
        "enable_agent_login": "no"
      }
    ]
  }
}
```

Real-time Status Information of The Queue Agents

Event Parameters

| Item             | Value                                                                              |
|------------------|------------------------------------------------------------------------------------|
| extension        | Indicates the extension number of the queue.                                       |
| abandon          | The number of abandoned calls.                                                     |
| answer           | The number of answered calls.                                                      |
| first_name       | The first name of the agent.                                                       |
| last_name        | The last name of the agent.                                                        |
| logintime        | The login time of the agent.                                                       |
| member_extension | Indicates the agent extension number in the extension queue.                       |
| membership       | The type of the agent.<br><ul style="list-style-type: none"><li>• Static</li></ul> |

|              |                                                                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | <ul style="list-style-type: none"><li>• Dynamic</li></ul>                                                                                                            |
| pause_reason | <p>The pause reason.</p> <ul style="list-style-type: none"><li>• Lunchtime</li><li>• Hourly Break</li><li>• Backoffice</li><li>• Send Email</li><li>• Wrap</li></ul> |
| pausetime    | Time and date of the pause.                                                                                                                                          |
| status       | The status of agent.                                                                                                                                                 |
| talktime     | Total talk time.                                                                                                                                                     |

Example

```
{
  "message": [
    {
      "action": "notify",
      "eventbody": [
        {
          "extension": "6500",
          "idlecount": 1,
          "member": [
            {
              "abandon": 0,
              "answer": 0,
              "first_name": null,
              "last_name": null,
              "logintime": "--",
              "member_extension": "1004",
              "membership": "static",
              "pause_reason": "Lunch",
              "pausetime": "2024-12-18 14:35:30",
              "queue_action": "CallQueueUpdateMember",
              "status": "Paused",
              "talktime": 0
            }
          ]
        }
      ],
      "eventname": "CallQueueStatus",
      "transactionid": "0965057392"
    }
  ],
  "type": "request"
}
```

The Current Call

Event Parameters

| Item       | Value                                        |
|------------|----------------------------------------------|
| extension  | Indicates the extension number of the queue. |
| callerid   | Caller number.                               |
| callername | Caller name.                                 |
| starttime  | The call started.                            |



|              |                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------|
| position     | Indicates the position of the current call in the queue.                                                        |
| queuechannel | Caller channel. This channel can be used to implement queue insertion, forced removal and monitoring functions. |

Example

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "CallQueueStatus",
    "eventbody": [
      {
        "extension": "6500",
        "callerid": "10616",
        "callername": "",
        "starttime": "2016-12-30 11:32:39",
        "position": 1,
        "queuechannel": "PJSIP/10616-00000010"
      }
    ]
  }
}
```

# Eventlist Status

## EventListStatus

When subscribed to the EventListStatus, if the status of an extension changes, an EventListStatus event will be reported. When the number of subscribers changes, an EventListStatus event will also be sent.

Event Parameters

| Item           | Value                 |
|----------------|-----------------------|
| extension      | Extension number      |
| status         | Extension status      |
| location       | Channel type          |
| trunk          | Relay type            |
| subscriber_num | Number of subscribers |

Example

Subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "EventListStatus",
      "EventlistSubscriberStatus"
    ]
  }
}
```

Trigger event

When the extension status changes, such as when the extension is ringing or in a call.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "EventListStatus",
    "eventbody": [
      {
        "uri": "hytkiu",
        "member": [
          {
            "extension": "1005",
            "location": "local",
            "trunk": "-",
            "status": "terminated"
          }
        ]
      }
    ]
  }
}
```

When the number of eventlist subscribers changes

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "EventListStatus",
    "eventbody": [
      {
        "uri": "hytkiu",
        "subscriber_num": 1
      }
    ]
  }
}
```

EventlistSubscriberStatus

When the EventListSubscriberStatus is subscribed to, if a new extension subscribes to the Eventlist or unsubscribes from it, the EventlistSubscriberStatus event is reported.

Event Parameter

| Item | Value         |
|------|---------------|
| uri  | Eventlist uri |



|           |                                      |
|-----------|--------------------------------------|
| extension | Subscribed extension number          |
| action    | Add or delete Eventlist subscription |

Example

When an extension subscribes or unsubscribes from EventList, report the event.

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "EventlistSubscriberStatus",
    "eventbody": [
      {
        "uri": "hytkiu",
        "members": [
          {
            "extension": "1005",
            "action": "add"
          }
        ]
      }
    ]
  }
}
```

SCA Status

ScaUserStatus

When the ScaUserStatus is subscribed to, the event will be sent every time the SCA shared number or private number has a status update. The event name is ScaUserStatus. Every time this event is sent, status, contact, and subscribed will be pushed to the front end. The user\_id is the SCA number. Both shared numbers and private numbers are represented by this field. The SCA user status content is as follows:

Event Parameters

| Item       | Value                                                    |
|------------|----------------------------------------------------------|
| status     | SCA user, the value is Idle, InUse, Unavailable, Ringing |
| contact    | SCA user's IP and port, such as: 192.168.124.156:5060    |
| subscribed | Whether to subscribe, the value is yes, no               |

Example

SCA user 1005 ringing

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ScaUserStatus",
    "eventbody": [
      {
        "user_id": "1005",
        "status": "Ringing",
        "contact": "192.168.124.156:5060",
        "subscribed": "yes"
      }
    ]
  }
}
```

## ScaLineStatus

When the ScaLineStatus is subscribed to, this event will be sent every time the SCA line status is updated. The event name is ScaLineStatus. Every time this event is sent, line\_index, sca\_status, local\_info, and remote\_info will all be pushed to the front end. Shared line is the SCA shared number. The SCA line status content is as follows.

### Event Parameters

| Item        | Value                                                                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| line_index  | The code starts from 1, and there are up to 10 lines under one shared number.                                                                |
| sca_status  | Line status. values possible: "idle", "seized", "progressing", "alerting", "active", "held", "held-private", "bridge-active", "bridge-held". |
| local_info  | Local number information, such as 1005.                                                                                                      |
| remote_info | Remote number information, such as 3005.                                                                                                     |

### Example

When SCA shared number 1005 calls another extension 3005

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ScaLineStatus",
    "eventbody": [
      {
        "sharedline": "1005",
        "line_index": "1",
        "sca_status": "progressing",
        "local_info": "1005",
        "remote_info": "3005"
      }
    ]
  }
}
```

## ScaLineStatusClean

When ScaLineStatusClean is subscribed to, this event will be sent each time, and all line statuses under the SCA shared number are cleared. The event name is ScaLineStatusClean. Each time this event is sent, sharedline will be pushed to the front end. Sharedline is the SCA shared number.

### Event Parameters

| Item       | Value                                                   |
|------------|---------------------------------------------------------|
| sharedline | To clear all line-related shared numbers, such as 1005. |

### Example

Clearing the status of all lines under the 1005 shared number

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "ScaLineStatusClean",
    "eventbody": [
      {
        "sharedline": "1005"
      }
    ]
  }
}
```

## Email Status

### sendMailStatus

When the user is subscribed to the email sending status, the corresponding extension information will be pushed when the sending status of the extension information email, remote registration email, and Wave welcome email is updated.

### Event Parameters

| Item       | Value                                                                                                        |
|------------|--------------------------------------------------------------------------------------------------------------|
| extension  | Indicates the extension number                                                                               |
| Result     | Email sending result information                                                                             |
| date       | Send time                                                                                                    |
| msgID      | To distinguish emails.                                                                                       |
| sendStatus | The status of the mail server response, Values: sent, deferred, bounced, deferral, reject.                   |
| emailType  | Email type. Values: account (extension information), register (remote registration), welcome (Wave welcome). |

### Example

subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "sendMailStatus",
      "EmailToUserStatus"
    ]
  }
}
```

When the response status of the extension information email of extension 2000 is updated (sendStatus is sent, it indicates that it has been delivered), this event will be reported as follows:

```
{
  "type": "request",
  "message": [
    {
      "transactionid": "0895990446",
      "action": "notify",
      "eventname": "sendMailStatus",
      "eventbody": {
        "date": "2022-04-12 17:59:03",
        "msgID": "1649757544.3529620",
        "sendStatus": "sent",
        "emailType": "account",
        "extension": "2001"
      }
    }
  ]
}
```

EmailToUserStatus

If the existing information email sent successfully notifies the corresponding extension user, the corresponding message will be pushed.

Event Parameters

| Item          | Value                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Extension     | Indicates the extension number                                                                                                                        |
| email_to_user | Extension information email sending result information, used to indicate whether the user has been notified. Value: yes (notified), no (not notified) |

Example

When the extension information email of extension 1000 is successfully notified to the user, this event will be reported.

```
{
  "type": "request",
  "message": [
    {
      "transactionid": "0895990446",
      "action": "notify",
      "eventname": "EmailToUserStatus",
      "eventbody": {
        "extension": "1000",
        "email_to_user": "yes"
      }
    }
  ]
}
```

# HA Status

## HaStatus

Subscribing to HaStatus will send the event notification when the HA operation status is updated. The Event name is “HaStatus” and the status content is as follows:

### Event Parameters

| Item     | Value                                                                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HO       | HA working status, 0 represents a single machine, 1 represents dual machines.                                                                                          |
| haERR    | HA standalone functioning status, 0 or empty means no problem detected, 1 means the version                                                                            |
| bkp4web  | HA backup status, 1 means not backed up, 2 means the primary PBX is backing up, 3 means the backup PBX is restoring, 4 means idle, 5 means file backup is in progress. |
| schedule | Full backup progress percentage.                                                                                                                                       |

### Example

#### Subscription

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zxc",
    "action": "subscribe",
    "eventnames": [
      "HaStatus"
    ]
  }
}
```

#### HA working status changes

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "HaStatus",
    "eventbody": {
      "haOL": "1"
    }
  }
}
```

HA stand-alone operation changes

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "HaStatus",
    "eventbody": {
      "haERR": "1"
    }
  }
}
```

The HA backup status changes

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "HaStatus",
    "eventbody": {
      "bkp4web": "1"
    }
  }
}
```

HA full backup 35% completed

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "HaStatus",
    "eventbody": {
      "schedule": "35"
    }
  }
}
```

# PMS Status

## PMSRoomStatus

When the client is subscribed to the PMSRoomStatus, this event will be sent when the room status is updated. The event name is PMSRoomStatus, and the status content is as follows:

Event Parameters



| Item       | Value                                                                                                        |
|------------|--------------------------------------------------------------------------------------------------------------|
| address    | Room unique identifier                                                                                       |
| room       | room number                                                                                                  |
| extension  | Extension number                                                                                             |
| first_name | First name of the guest                                                                                      |
| last_name  | Last name of the guest                                                                                       |
| datein     | Check-in date                                                                                                |
| dateout    | Check-out date                                                                                               |
| status     | Room status, value 1 indicates that the guest has checked in. 0 Indicates that the guest has not checked in. |
| account    | Customer account                                                                                             |
| vipcode    | Customer VIP level                                                                                           |
| credit     | Customer credit amount                                                                                       |
| maid       | Housekeeper number                                                                                           |

Example

Report incidents when checking in

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSRoomStatus",
    "eventbody": [
      {
        "address": "1000",
        "extension": "1000",
        "first_name": null,
        "last_name": null,
        "room": "1000",
        "status": "1",
        "user_name": "1000",
        "account": null,
        "maid": null,
        "credit": null,
        "vipcode": null,
        "datein": null,
        "dateout": null
      }
    ]
  }
}
```

Report incident when checking out

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSRoomStatus",
    "eventbody": [
      {
        "address": "1000",
        "extension": "1000",
        "first_name": null,
        "last_name": null,
        "room": "1000",
        "status": "0",
        "user_name": "1000",
        "account": null,
        "maid": null,
        "credit": null,
        "vipcode": null,
        "datein": null,
        "dateout": null
      }
    ]
  }
}
```

Report an event when a name is modified (Mitel only)

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSRoomStatus",
    "eventbody": [
      {
        "address": "1000",
        "first_name": "",
        "last_name": "john"
      }
    ]
  }
}
```

When a customer’s credit balance is updated (Mitel only)

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSRoomStatus",
    "eventbody": [
      {
        "address": "1000",
        "credit": "23456"
      }
    ]
  }
}
```

**PMSWakeupStatus**

When the PMSWakeupStatus is subscribed to, the event will be sent every time the execution status and answering status change. The event name is PMSWakeupStatus, and the status content is as follows.

**Event Parameters**



| Item        | Value                                                                                                                                                                                                                                                                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| w_action    | This field indicates the wake-up service execution status, the value is 0, 1, and 2.<br>When the value is 0, it means that the wake-up service has been canceled.<br>When the value is 1, it means that the wake-up service has been set but not executed.<br>When the value is 2, it means that the wake-up service has been executed. |
| w_status    | The client's answering status can be 1 or 2. When the value is 2, it means the wake-up service has not been answered, and the value is 1, it means the wake-up service has been answered.                                                                                                                                               |
| address     | Pms room number                                                                                                                                                                                                                                                                                                                         |
| send_status | Report wakeup status                                                                                                                                                                                                                                                                                                                    |
| w_time      | Wakeup time                                                                                                                                                                                                                                                                                                                             |
| w_date      | Wakeup date                                                                                                                                                                                                                                                                                                                             |
| w_type      | Types of wakeups                                                                                                                                                                                                                                                                                                                        |
| action      | Wakeup operation, such as add, delete, update.                                                                                                                                                                                                                                                                                          |

Example

When a wake-up call is executed

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSWakeupStatus",
    "eventbody": [
      {
        "address": "1000",
        "w_action": "2",
        "w_status": "1"
      }
    ]
  }
}
```

When the wake-up call is answered by the customer

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "PMSWakeupStatus",
    "eventbody": [
      {
        "address": "1000",
        "w_action": "2",
        "w_status": "2"
      }
    ]
  }
}
```

When the wake-up service is changed on the page

```
{
  "type": "request",
  "message": {
    "transactionid": "1812866259",
    "action": "notify",
    "eventname": "PMSWakeupStatus",
    "eventbody": [
      {
        "address": "1001",
        "send_status": "1",
        "w_time": "1059",
        "w_date": "20170728",
        "w_type": "1",
        "w_action": "1",
        "action": "update"
      }
    ]
  }
}
```

# CRM User Status

## CRMUserStatus

When subscribing to CRM user status, this event will be sent every time the user status changes. The event name is CRMUserStatus, and the status content is as follows:

### Event Parameters

| Item         | Value                                                |
|--------------|------------------------------------------------------|
| extension    | Configure the extension of the CRM account.          |
| user_name    | CRM account name                                     |
| login_status | User login status, the value is "login" or "logout". |

### Example

```
{
  "type": "request",
  "message": {
    "transactionid": "123456789zx",
    "action": "notify",
    "eventname": "CRMUserStatus",
    "eventbody": [
      {
        "extension": "1000",
        "user_name": "admin",
        "login_status": "login"
      }
    ]
  }
}
```