CounterPath Corporation
Suite 300, Bentall One Centre
505 Burrard Street Box 95
Vancouver BC V7X 1M3
Canada V6B1R8

Telephone: +1.604.320.3344
www.counterpath.com

Bria 3 API

Developer Guide

CounterPath Corporation
Suite 300, One Bentall Centre
505 Burrard Street, Box 95
Vancouver, BC V7X 1M3
Tel: 604.320.3344
sales@counterpath.com   www.counterpath.com

This manual corresponds to Bria 3.2.

Rev 4

About the Bria 3 API

The Bria 3 API provides functions that let another application place phone calls and obtain call history information.

For example, your application may have a web page that provides information about customers for your employees to phone.

- The employee clicks on the customer phone number, which invokes a function from the Bria 3 API to place the phone call via Bria.
- Bria executes the request and sends events to your application. These events provide information about the status of the request.
- The employee will manage the call using Bria: the employee can put the call on hold, transfer the call, mute the call, and so on.
- When the call ends, your application will receive notification that the call has ended.

The API provides some control over Bria but not complete control: the user may have to interact with the Bria user interface in order to manage the call: place on hold, transfer, and so on.

This API guide assumes that you are familiar with the standard functionality of Bria 3, as described in the Bria 3 user guides.

Connecting to Bria 3

Connect to Bria by establishing a pipe connection to the named pipe “apipipe”. This pipe is full-duplex and message-oriented.

You can make unlimited connections to the same instance of Bria, but all connections will receive the same information (events and responses) from Bria.

The Bria API supports two types of communications:

- Events from Bria. See page 5.
- Request/response messages. See page 7.

Events, requests and responses are in XML format with UTF-8 encoding.

Events

As soon as you connect, you must be prepared to start receiving events from Bria.

Phone Status Changed: POST /statusChange “phone”

The readiness of Bria has changed.

```
POST /statusChange
User-Agent: Bria 3
Content-Type: application/xml
Content-Length: 63
<?xml version="1.0" encoding="utf-8" ?>
<event type="phone"></event>
```

Next Action

Get details on the change by sending this request: Get /status “phone”

Call Status Changed: POST /statusChange “call”

The status of an existing call has changed.

```
POST /statusChange
User-Agent: Bria 3
Content-Type: application/xml
Content-Length: 62
<?xml version="1.0" encoding="utf-8" ?>
<event type="call"></event>
```

Next Action

Get /status "call".

Call History Change: POST /statusChange “callHistory”

The contents of the call history has changed: a call has been added or the user has manually deleted calls

```
POST /statusChange
User-Agent: Bria 3
Content-Type: application/xml
Content-Length: 69
<?xml version="1.0" encoding="utf-8" ?>
<event type="callHistory"></event>
```

Next Action

Get /status “callHistory”

Missed Call Occurred: POST /statusChange “missedCall”

The missed call count has changed.

```
POST /statusChange
User-Agent: Bria 3
Content-Type: application/xml
Content-Length: 68
<?xml version="1.0" encoding="utf-8" ?>
<event type="missedCall"></event>
```

Next Action

GET /status “missedCall” or GET /status “callHistory”

MWI Count Change: POST /statusChange “voicemail”

Posted when the Bria receives a MWI message from the voicemail server.

```
POST /statusChange
User-Agent: Bria 3
Content-Type: application/xml
Content-Length: 72
<?xml version="1.0" encoding="utf-8" ?>
<event type="voiceMail"></event>
```

Next Action

GET /status “voicemail” and GET /checkVoiceMail

Requests and Reponses

Request and Response Syntax

The format and content of requests are illustrated in the following pages. The request can include a Transaction-ID; if this is included, then the response will also include this ID.

The order of elements within a tag is not important.

All requests receive an immediate response from Bria:

- 200 OK: Bria understands the request. This response may include details; see the information for the individual requests.
- 400 Bad Request: Bria does not understand the request. Check that the xml format and the request syntax are correct.
- 503 Service Unavailable: Bria is in the notReady or notAllowed state. See Get /status “phone”.

If the response is 200 OK, then you will start receiving events that describe the progress of the request.

Bring Phone to Front: GET /bringToFront

Bring the main window of Bria to front and give it focus.

```
GET/bringToFront
User-Agent: MyApplication
Transaction-ID: AE26f998027
Content-Type: application/xml
Content-Length: 0
```

Response

```
HTTP/1.1 200 OK
Transaction-ID: AE26f998027
Content-Type: application/xml
Content-Length: 0
```

Bring the History Panel to Front: GET /showHistory

Bring the History panel to the front and give it focus.

- type: “all”, “missed”, “received” or “dialed”.
- text: Optional. The text to enter in the History panel. The contents of the History panel will be filtered to show only entries that match this text.

```
GET/showHistory
User-Agent: MyApplication
Transaction-ID: FE881337
Content-Type: application/xml
Content-Length: 72
<?xml version="1.0" encoding="utf-8" ?>
<filter>
<type>missed</type>
<text>Frank Chan</type>
</filter>
```

Response

```
HTTP/1.1 200 OK
Transaction-ID: FE881337
Content-Type: application/xml
Content-Length: 0
```

Get Status of Phone: GET /status “phone”

Request the current phone status.

```
GET/status
User-Agent: MyApplication
Transaction-ID: GF832137
Content-Type: application/xml
Content-Length: 71
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>phone</type>
</status>
```

Response

- status: phone
- state: “ready” or “notReady” (no accounts are enabled).
- call: “allow” or “notAllow” (there is no free line available to make a call).
- maxLines: The number of lines allowed in your brand of Bria. Typically 6.

```
HTTP/1.1 200 OK
Transaction-ID: GF832137
Content-Type: application/xml
Content-Length: 125
<?xml version="1.0" encoding="utf-8" ?>
<status type="phone">
<state>ready</state>
<call>allow</call>
<maxLines>6</maxLines>
</status>
```

Get Information on Bria Setup: GET /status "systemSettings”

Request the current configuration of Bria.

```
GET/status
User-Agent: MyApplication
Transaction-ID: BS398809
Content-Type: application/xml
Content-Length: 80
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>systemSettings</type>
</status>
```

Response

- defaultCallType: The value currently selected in Bria in Preferences > Application > Default Action: “audio”, “video” or “conference”.
- callRightAwayOnceNumberSelected: “true” or “false”. If false, then when a call is placed using GET /call, then the call entry field on Bria will be populated with the data you provide but the user will have to click the Call button on the Bria user interface in order to place the call. If true, the user will not have to intervene.

```
HTTP/1.1 200 OK
Transaction-ID: BS398809
Content-Type: application/xml
Content-Length: 184
<?xml version="1.0" encoding="utf-8" ?>
<status type="systemSettings">
<defaultCallType>audio</defaultCallType>
<callRightAwayOnceNumberSelected>true</callRightAwayOnceNumberSelected>
</status>
```

Place a Call: GET /call

Place a call of the specified type, to the specified number, and presenting the specified display name for the caller.

- dial type: “audio”, “video”, “conference”
- number: The phone number to dial. See “Details” below.
- displayName: Optional. The name of the local user; this is the name that will be presented to the other party.

```
GET/call
User-Agent: MyApplication
Transaction-ID: GF8002137
Content-Type: application/xml
Content-Length: 135
<?xml version="1.0" encoding="utf-8" ?>
<dial type="audio">
<number>1440@domainA.com</number>
<displayName>Joseph Santos</displayName>
</dial>
```

Response

```
HTTP/1.1 200 OK
Transaction-ID: GF8002137
Content-Type: application/xml
Content-Length: 0
```

Details

If the response is 200 OK, Bria populates the call entry field with the specified phone number. Depending on the value of callRightAwayOnceNumberSelected (GET /status “systemSettings”) the call is either placed immediately or only after the user clicks the Call button.

To select the account to place the call on, Bria uses the account selection method that is currently selected in Bria: Auto Select or a specific account. If Auto Select is used and dial plans are defined, the phone number may be modified (by the dial plan) before it is placed. However, the call history shows the phone number as specified in the request (not as modified).

Once the call attempt starts, POST /statusChange “call” events are posted.

Get Information about Current Calls: GET /status “call”

Get information on all existing calls.

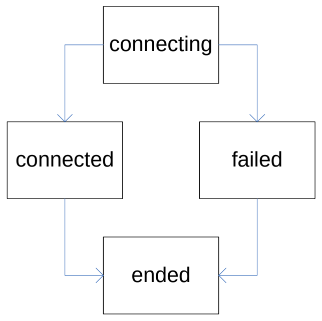
```
GET/status
User-Agent: MyApplication
Transaction-ID: EF855137
Content-Type: application/xml
Content-Length: 70
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>call</type>
</status>
```

Response

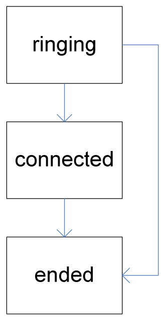
One or more <call> tags, each containing information about an incoming or outgoing call:

- id: the unique ID of this call.
- holdStatus: offHold, localHold, and remoteHold
- participants: One or more <participant> tags each containing:
  - number: The number of the other party
  - displayName: The display name of the other party
  - state: connected, connecting, ringing, failed, ended.

States for an outgoing call



States for an incoming call



  - timeInitiated: time the call attempt started, in UTC

If there are currently no existing calls, the <call> will be empty.

In the following example, the second call is a conference call.

```
HTTP/1.1 200 OK
Transaction-ID: EF855137
Content-Type: application/xml
Content-Length: 769
<?xml version="1.0" encoding="utf-8" ?>
<status type="call">
<call>
   <id>ea6c7664001343688013e3680013e368</id>
   <holdStatus>offHold</holdStatus>
   <participants>
      <participant>
         <number>1440@domainA.com</number>
         <displayName>Rita Santos</displayName>
         <state>connecting</state>
         <timeInitiated> 1303931552</timeInitiated>
      </participant>
   </participants>
</call>
<call>
   <id>ea6c7664001343688013e3680013e368</id>
   <holdStatus>onHold</holdStatus>
   <participants>
      <participant>
         <number>2758@domainA.com</number>
         <displayName>Kokila Perera</displayName>
         <state>connected</state>
         <timeInitiated> 1303931542</timeInitiated>
      </participant>
      <participant>
         <number>2764@domainA.com</number>
         <displayName>Frank Chan</displayName>
         <state>connected</state>
         <timeInitiated> 1303931531</timeInitiated>
      </participant>
   </participants>
</call>
</status>
```

End a Call: GET /endCall

Hang up the call that has the specified call ID. You can obtain the call ID using GET /status “call”.

```
GET/endCall
User-Agent: MyApplication
Transaction-ID: YW831137
Content-Type: application/xml
Content-Length: 104
<?xml version="1.0" encoding="utf-8" ?>
<endCall>
<callId>391A64292F0A413aB14CBAB341621A3D</callId>
</endCall>
```

Response

```
HTTP/1.1 200 OK
Transaction-ID: YW831137
Content-Type: application/xml
Content-Length: 0
```

Get Call History Data: GET /status “callHistory”

Obtain the call data for the most recent phone calls, not including calls that are still established.

- type: “callHistory”
- count: the number of calls to query, starting with the most recent.
- entryType: “all”, “missed”, “received” or “dialed”.

```
GET/status
User-Agent: MyApplication
Transaction-ID: CW839937
Content-Type: application/xml
Content-Length: 120
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>callHistory</type>
<count>10</count>
<entryType>all</entryType>
</status>
```

Response

One or more callHistory tags, each containing:

- type: “missed”, “received” or “dialed”
- number: The number dialed as displayed in the call history.
- displayname: If available.
- duration: In seconds
- timeInitiated: Time the call attempt started, in UTC.

```
HTTP/1.1 200 OK
Transaction-ID: CW839937
Content-Type: application/xml
Content-Length: 401
<?xml version="1.0" encoding="utf-8" ?>
<status type="callHistory">
<callHistory>
   <type>dialed</type>
   <number>6045553344</number>
   <displayName>Frank Chan</displayName>
   <duration>128</duration>
   <timeInitiated>1303932552</timeInitiated>
</callHistory>
<callHistory>
   <type>missed</type>
   <number>demo</number>
   <displayName>demo</displayName>
   <duration>1440</duration>
   <timeInitiated>1303932432</timeInitiated>
</callHistory>
</status>
```

Get Count of Missed Calls: GET /status “missedCall”

Obtain the current count for missed calls. A missed call is an incoming call that the user did not answer (if the call was picked up by voicemail, it is still considered to be a missed call).

The “current count” increments each time a call is missed.

The count is reset to zero when:

- The user clicks on the Missed Calls icon in the Bria dashboard, bringing the History panel to the front.
- This GET /status “missed calls” is sent with a 200 OK response.

The count is not reset when the History panel is brought to the front without the user clicking on the Missed Calls icon.

```
GET/status
User-Agent: MyApplication
Transaction-ID: CC8322937
Content-Type: application/xml
Content-Length: 76
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>missedCall</type>
</status>
```

Response

- count: the current number of missed calls

```
HTTP/1.1 200 OK
Transaction-ID: CC8322937
Content-Type: application/xml
Content-Length:86
<?xml version="1.0" encoding="utf-8" ?>
<status type="missedCall">
<count>2</count>
</status>
```

Get MWI Count: GET /status “voiceMail”

```
GET/status
User-Agent: MyApplication
Transaction-ID: PI834137
Content-Type: application/xml
Content-Length: 75
<?xml version="1.0" encoding="utf-8" ?>
<status>
<type>voiceMail</type>
</status>
```

Response

One or more <voiceMail> tags (one for each account configured for voicemail), containing:

- accountId
- accountName
- count: The number of voicemail items.
    - If voicemail is supported and MWI is supported, the voicemail count: 0 or more.
    - If voicemail is not set up on a specific account, then 0 is specified.
    - If the voicemail server does not support MWI with a count, then -1 is specified.

```

HTTP/1.1 200 OK
Transaction-ID: PI834137
Content-Type: application/xml
Content-Length: 167
<?xml version="1.0" encoding="utf-8" ?>
<status type="voiceMail">
<voiceMail>
   <accountId>0</accountId>
   <accountName>Account1</accountName>
   <count>3</count>
</voiceMail>
</status>
```

Check for Voicemail: GET /checkVoiceMail

Phone the voicemail server for the specified account ID. Bria must be configured with the voicemail phone number for the specified account (Bria > Account > Voicemail).

```
GET/checkVoiceMail
User-Agent: MyApplication
Transaction-ID: MI839837
Content-Type: application/xml
Content-Length: 93
<?xml version="1.0" encoding="utf-8" ?>
<checkVoiceMail>
<accountId>0</accountId>
</checkVoiceMail>
```

Response

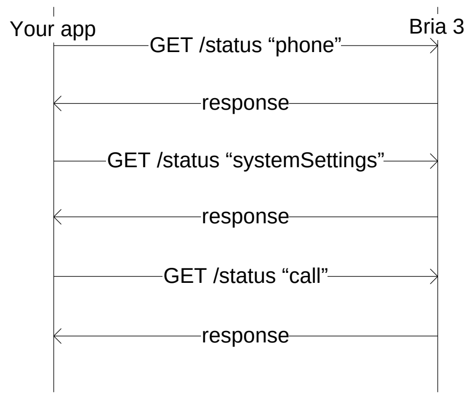
- 200 OK if the voicemail is configured in Bria on the specified account. Bria places a call to the voicemail server.
- 200 OK if voicemail is not configured on this account. However, no call is placed (which means that no call events are posted).

```
HTTP/1.1 200 OK
Transaction-ID: MI839837
Content-Type: application/xml
Content-Length: 0
```

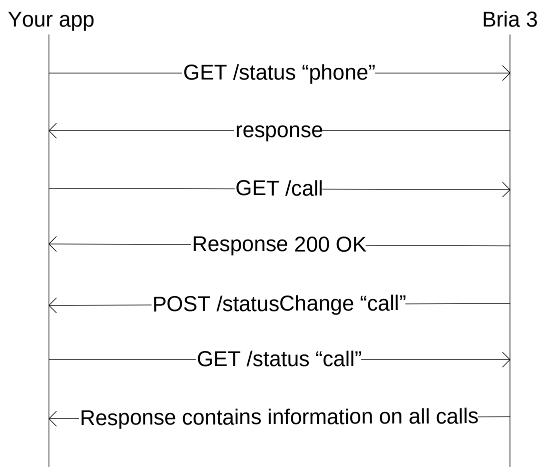
Typical Sequences

Startup

The following diagram shows a recommended startup sequence. Keep in mind that at any time before, during, or after this sequence, you may receive events completely independent of this sequence. For example, if a call ends.



Placing an Outgoing Call



Creating a Conference Call

When you create a conference call, the Bria call panel for the call starts in “conference call” mode. In this mode, an Add button is displayed to let the user quickly add participants. You cannot add participants to the conference call via the API.

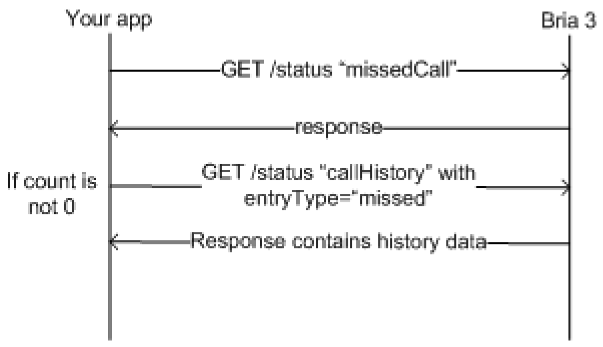
Monitoring Established Calls

You cannot use the API to handle the established call, but you can obtain information about the state of each call, for example, whether it is on hold, using GET /status "call". You can also hang up a call using GET /endCall.

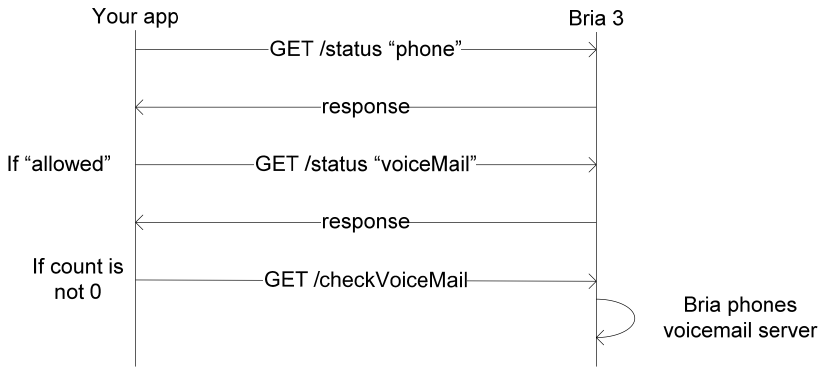
Obtaining the Missed Call Count and Details

When a call is missed on Bria, you receive the POST /statusChange "missedCall" event.

You can also query for missed calls independently of this event (for example, at startup), as follows:


Your app
Bria 3
GET /status "missedCall"
response
If count is not 0
GET /status "callHistory" with entryType="missed"
Response contains history data

Gettting MWI Information and Connecting to the Voicemail Server


Your app
Bria 3
GET /status "phone"
response
If "allowed"
GET /status "voiceMail"
response
If count is not 0
GET /checkVoiceMail
Bria phones voicemail server