



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Mobile Services API Reference

Callback Push Notifications for Android

12/13/2025

Contents

- 1 Callback Push Notifications for Android
 - 1.1 Procedure
 - 1.2 Dialogs (REST)
 - 1.3 Push Notifications

Callback Push Notifications for Android

Genesys Mobile Services (GMS) employs various mechanisms to achieve asynchronous messaging (push notifications). For Android devices, it is a combination of GCM/C2DM, and Comet. Likewise, iOS devices employ APNs and Comet. The scope of this article is limited to how push notifications are handled in Android devices, particularly for the Callback application.

Note: For iOS devices, see [Callback Push Notifications for iOS](#).

Procedure

[Implementing a GCM client](#) is well documented by Google. Alternately, you can also refer to the [Genesys Mobile Services Android Sample](#) for a Genesys implementation of a GCM BroadcastReceiver.

Push notifications can be divided into two distinct parts:

1. Chat (implements push notifications over Comet)
2. Callback (implemented over GCM)

For Chat, a Bayeux Client is created to listen to all push notifications related to Chat. The default channel for Chat is `/_genesys`. The format of Chat push notifications can be seen in the [Chat \(Comet\) section](#).

For Callback push notifications, refer to the [Genesys Mobile Services Android Sample](#) for reference.

Processing of GCM notifications is a three-step procedure:

1. Obtain Service ID and Action identifier from GCM Intent.
2. Issue HTTP POST to GMS with specified action to obtain action data as JSON.
3. Execute action using data provided by GMS.

The data contained within the GCM Intent is structured as follows:

```
Intent intent;  
Bundle extras = intent.getExtras();  
String message = extras.getString("message");  
System.out.println(message);
```

```
-----  
Result:  
{  
    "_id": "$(_id)",  
    "_action": "$(_action)",  
}
```

In the case of the Genesys sample client, the `GenesysCloudMessageReceiver` repackages the GCM Intent into an application-specific Intent:

```
Intent newIntent = new Intent(context, GenesysSampleActivity.class);
newIntent.setAction(Globals.ACTION_GENESYS_CLOUD_MESSAGE);
newIntent.putExtra(Globals.EXTRA_MESSAGE, extras.getString("message"));
newIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
context.startActivity(newIntent);
```

This intent is then handled by the `GenesysSampleActivity (handleIntent() > interpretCloudMessage())` where the encapsulated data is used to form an HTTP POST with the following URL:

```
$(ServerURL)$(URLPath)$( _id)/$( _action)
for example,
http://135.34.145.123:8080/genesys/1/service/3SQI3S31693JL9L3R00506T40C000U73/get-dialog-start-chat
```

Identifier	Description	Example Values
\$(ServerURL)	URL to Genesys Mobile Services host	http://135.34.145.123:8080
\$(URLPath)	Path to Services API	/genesys/1/service/
\$(_id)	GMS-issued Service ID	3SQI3S31693JL9L3R00506T40C000U73
\$(_action)	Callback action to perform	get-dialog-user-confirmation-provide_code-true get-dialog-user-confirmation-provide_code-false get-dialog-start-chat connect-inbound connect-outbound wait-for-agent

The response of the HTTP POST request contains JSON, which describes an action to perform and/or UI elements to display in the client application. Each of these requests are referred to as Dialogs.

Dialogs (REST)

The following examples are JSON structures returned by the GMS Callback service to the client application. The contents of the JSON response depends on the Callback action performed (as described in the [Procedure section](#)).

Refer to the [Genesys Mobile Services Android Sample](#) for examples on how these JSON responses can be interpreted as actions (for example: Call agent, Display menu, Display dialog) and/or UI elements (for example: Confirmation dialogs or Menu items).

get-dialog-user-confirmation-provide_code-true

```
{
  "_dialog_id": "0",
  "_label": "Agent is available right now",
  "_user_action_url": "$(ExtURLBase)/1/service/$(ServiceID)/not-used",
  "_method": "POST",
  "_action": "DisplayMenu",
  "_expires": "$(Date)",
  "_resource_url": "$(ExtURLBase)/1/service/$(ServiceID)/get-dialog-user-confirmation",
```

```

    "_content": [
      {
        "_group_name": "Are you ready?",
        "_group_content": [
          {
            "_dialog_id": "1",
            "_label": "Yes, I'm ready to talk",
            "_action": "MenuItem",
            "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/connect",
            "_method": "POST",
            "_id_to_jump_before": "exit://",
            "_confirmation_dialog": {
              "_text": "You will hear tones immediately
after call is connected. This is normal.",
              "_dialog_type": "Notification",
              "_dismiss_timeout": 2
            }
          }, {
            "_dialog_id": "2",
            "_label": "No, try again in 5 minutes",
            "_action": "MenuItem",
            "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/snooze",
            "_method": "POST",
            "_id_to_jump_before": "exit://"
          }, {
            "_dialog_id": "3",
            "_label": "Cancel, my problem has been solved",
            "_action": "MenuItem",
            "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/cancel",
            "_method": "POST",
            "_id_to_jump_before": "exit://"
          }
        ]
      }
    ]
  }
}

```

get-dialog-user-confirmation-provide_code-false

```

{
  "_dialog_id": "0",
  "_label": "Agent is available right now",
  "_user_action_url": "${ExtURLBase}/1/service/${ServiceID}/not-used",
  "_method": "POST",
  "_action": "DisplayMenu",
  "_expires": "${Date}",
  "_resource_url": "${ExtURLBase}/1/service/${ServiceID}/get-dialog-user-confirmation",
  "_content": [
    {
      "_group_name": "Are you ready?",
      "_group_content": [
        {
          "_dialog_id": "1",
          "_label": "Yes, I'm ready to talk",
          "_action": "MenuItem",
          "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/connect",
          "_method": "POST",
          "_id_to_jump_before": "exit://",
        }, {

```

```

        "_dialog_id": "2",
        "_label": "No, try again in 5 minutes",
        "_action": "MenuItem",
        "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/snooze",
        "_method": "POST",
        "_id_to_jump_before": "exit://"
    }, {
        "_dialog_id": "3",
        "_label": "Cancel, my problem has been solved",
        "_action": "MenuItem",
        "_user_action_url": "${ExtURLBase}/1/
service/${ServiceID}/cancel",
        "_method": "POST",
        "_id_to_jump_before": "exit://"
    }
]
}
}
}

```

get-dialog-start-chat

```

{
    "_dialog_id": "1",
    "_action": "StartChat",
    "_label": "Start Chat",
    "_start_chat_url": "${ExtURLBase}/1/service/${ServiceID}/ixn/chat",
    "_comet_url": "${CometURL}",
    "_user_header": "${GMSUser}",
    "_id_to_jump_before": "exit://",
    "_chat_parameters": {
        "subject": "None"
    },
    "_id": "${ServiceID}"
}

```

connect-inbound

```

{
    "_dialog_id": "0",
    "_label": "Connecting ...",
    "_action": "DialNumber",
    "_tel_url": "n/a",
    "_access_code": "n/a",
    "_id": "${ServiceID}"
}

```

connect-outbound

```

{
    "_dialog_id": "0",
    "_action": "ConfirmationDialog",
    "_text": "You will receive the call shortly",
    "_ok_title": "Ok",
    "_id": "${ServiceID}"
}

```

wait-for-agent

```

{

```

```
{
  "_dialog_id": "0",
  "_action": "ConfirmationDialog",
  "_text": "We will notify you when agent is available",
  "_ok_title": "Ok",
  "_id": "${ServiceID}"
}
```

Push Notifications

Chat (Comet)

Message Receipt

```
{
  "data": {
    "message": {
      "startedAt": "2014-05-02T16:27:38Z",
      "chatIxnState": "TRANSCRIPT",
      "chatSessionId": "000FRa9NYM9A001K",
      "transcriptToShow": [
        ["Message.Text", "Stan", "Hello.", "8", "CLIENT"]
      ],
      "transcriptPosition": "2",
      "chatServiceMessage": "Chat service is available"
    },
    "id": "b2e607a0d21611e3000010932938a0ff",
    "tag": "service.chat.refresh.3SQIS3S1693JL9L3R00506T40C000UL4"
  },
  "channel": "/_genesys"
}
```

Party Joined/Left

```
{
  "data": {
    "message": {
      "startedAt": "2014-05-02T16:27:38Z",
      "chatIxnState": "TRANSCRIPT",
      "chatSessionId": "000FRa9NYM9A001K",
      "transcriptToShow": [
        ["Notice.Joined", "Kristi Sippola", "has joined the session", "17", "AGENT"]
      ],
      "transcriptPosition": "3",
      "chatServiceMessage": "Chat service is available"
    },
    "id": "b7dd6460d21611e3000010932938a0ff",
    "tag": "service.chat.refresh.3SQIS3S1693JL9L3R00506T40C000UL4"
  },
  "channel": "/_genesys"
}
```

Typing Started/Stopped

```
{
  "data": {
    "message": {
      "startedAt": "2014-05-02T16:27:38Z",
      "chatIxnState": "TRANSCRIPT",
      "chatSessionId": "000FRa9NYM9A001K",

```

```

        "transcriptToShow": [{"Notice.TypingStarted", "Kristi Sippola", "is
typing", "20", "AGENT"}],
        "transcriptPosition": "4",
        "chatServiceMessage": "Chat service is available",
    },
    "id": "b91bd7d0d21611e3000010932938a0ff",
    "tag": "service.chat.refresh.3SQIS3S1693JL9L3R00506T40C000UL4"
},
"channel": "/_genesys"
}

```

Notes

Identifier	Description	Values
\$(TranscriptType)	Type of event to display in the chat log.	Message.Text Notice.TypingStarted Notice.TypingStopped Notice.Joined Notice.Left
\$(Timestamp)	UTC Time format	YYYY-MM-DDTHH:MM:SSZ
\$(TranscriptPosition)	Line Number	Some integer.
\$(ChatIxnState)	State of chat interaction.	TRANSCRIPT DISCONNECTED