



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 iOS

5/12/2025

Contents

- [1 Callback Push Notifications for iOS](#)
 - [1.1 Push Notifications in iOS Applications](#)
 - [1.2 Push Notifications](#)

Callback Push Notifications for iOS

This article is for developers who wish to develop an iOS client application for Genesys Mobile Services (GMS) Callback Services.

GMS employs various mechanisms to achieve asynchronous messaging (push notifications). For iOS devices, Apple Push Notification service (APN) and Comet are used. For Android devices, GCM/C2DM and Comet are used. This article describes how push notifications should be handled in iOS devices. The [Genesys Mobile Services iOS Sample](#) provides example code for the concepts discussed.

Important

For Android devices, see [CallbackPushNotifications for Android](#).

Push Notifications in iOS Applications

Push notifications are used for two purposes in GMS iOS applications:

- Chat
- Callback

Chat

The GMS chat implementation uses Comet push notifications for the text message exchange. GMS implements a Comet server that the mobile client connects to when a chat session is opened.

The iOS sample code includes a Comet library. The library includes a Comet client class called `DDCometClient`. The iOS sample application `GMSChatViewController` class illustrates how to use `DDCometClient` to connect to the server and to send and receive chat messages. The default channel used for chat is `"/_genesys"`.

Callback

The GMS Callback functions utilize APN for push notifications to iOS applications. The processing of APN notifications operates as follows:

1. Obtain the Service ID and Action identifier from the notification “message” component (the “message” component is in JSON format).
2. Issue an HTTP POST to GMS with specified action. The response includes further action data in JSON format.
3. Execute action using data provided by GMS.

The APN notification contains several components. The “aps” component specifies the confirmation dialog to display to the user. You can refer to Apple developer documentation for more information on this topic. The iOS sample application `GMSAppDelegate` `didReceiveRemoteNotification` method provides an example. The “message” component provides the data from GMS and has the following format:

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

The `_id` and `_action` parameters are extracted and used to construct a URL for a POST to GMS. The iOS sample application `GMSAppDelegate` `processAPN` method provides an example of this.

```
$(ServerURL)$(URLPath)$(_id)/$(_action)
such as
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 data, which describes an action to perform and/or UI elements to display in the client application. Each of these requests will be referred to as Dialogs.

Dialogs (REST)

The following are example 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 [Callback section](#)).

Refer to the [Genesys Mobile Services iOS 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)/snooze",
                    "_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)/snooze",
        "_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