



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

Chat API Version 1

5/9/2025

Contents

- 1 Chat API Version 1
 - 1.1 Prerequisites
 - 1.2 Getting Started
 - 1.3 Structures
 - 1.4 Basic Chat Service API
 - 1.5 Chat Interaction APIs
 - 1.6 Quick Start Examples
 - 1.7 CometD-Based Chat API
 - 1.8 Chat Push Notification and Background State
 - 1.9 CometD and Loss of Network Connection

Chat API Version 1

Important

This API is documented to support older deployments. Genesys recommends that you upgrade to **Chat API version 2** if you are still using Chat API version 1.

Prerequisites

Before using the Chat API described on this page, you must **configure your Genesys Mobile Services deployment** correctly. Then, before you can create Chat sessions, you must **create** your chat service (of type **ors** or **builtin**) with the Service Management UI. Then, you will be able to use this chat service to request chat session IDs.

Getting Started

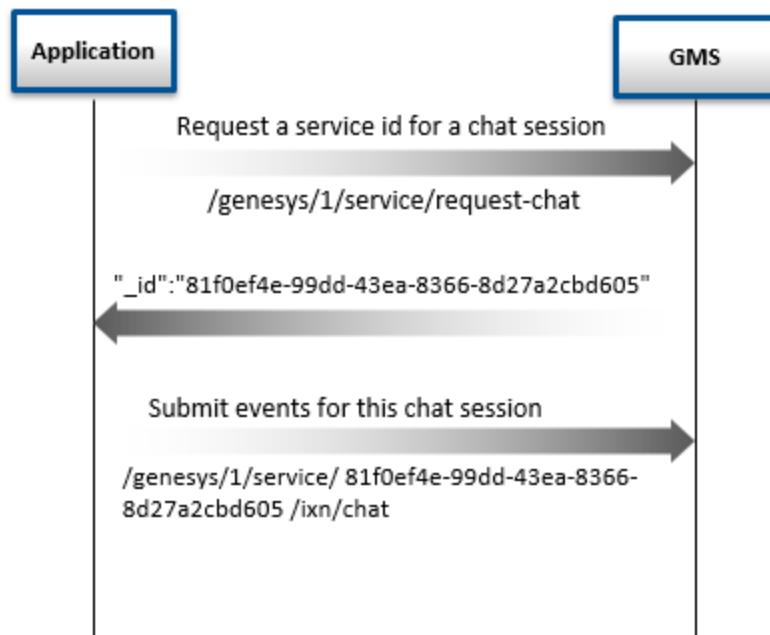
The Chat API is used by customer-facing applications to create and manage a chat session associated with contact center-related services.

Basic Chat

To start chat session:

- **Create** the built-in service of type request - chat that provides the Basic Chat Service API.
- Use the Basic Chat Service API to create a Chat Service ID for each chat session. The built-in request - chat service created in the Service Management UI provides the Basic Chat Service API.

Then, you can use the Chat Service ID to perform queries for your chat session.



Chat API Version 1 (Genesys Mobile Services-Based) allows an application to pass business context data in the service creation request, using the fixed service name `request-chat`.

- No corresponding Orchestration Server (ORS) session will be created.
- Data is preserved by Genesys Mobile Services using the specified time-to-live parameter (or the configured default value).
- Chat interaction could be initiated by an application at any point.
- The routing logic associated with the specified interaction endpoint (or the configured default value) would be responsible for finding an appropriate agent.
- Both polling and async (CometD) modes of message delivery are supported.
- Applications can handle background state through **chat push notifications**

Tip

When using asynchronous messaging with CometD, all HTTP headers must include the `gms_user` header.

ORS-Chat Services

Instead of creating a basic chat service to request chat session IDs, you can configure one of the following chat service of type `ors` in the Service Management UI:

- [Chat Immediate](#)
- [Chat Delayed](#)

Then, use this ors service to retrieve your new service ID associated with your chat session. For instance if you configured the chat-immediate service, you will post to the chat-immediate service:

```
POST http://localhost:8080/genesys/1/service/callback/chat-immediate
```

You can now use this ID in the [Chat Interaction APIs](#).

Structures

The Chat API uses the data structures described in this section (in JSON format) to exchange data. Requests are accepted in **application/json**, **'application/x-www-form-urlencoded'**, or **'multipart/form-data'** formats, and responses are returned in **'application/json'** format. If an expected value is missing or incorrect, then your application receives a 415 (Unsupported Media Type) error.

JSON-Encoding

For each method detailed below, if you wish to use JSON in your body, you must use JSON-encoding for body capabilities by setting the Content-Type to `application/json; charset=UTF-8`. For example, if you use the query for the chat session creation:

Request

```
POST http://localhost:8080/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json; charset=UTF-8

{
  "verbose": "true",
  "first_name": "John",
  "last_name": "Doe",
  "subject": "GMSDemo",
  "userDisplayName": "JohnDoe"
}
```

Response

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225

{
  "chatIxnState" : "CONNECTED",
  "transcriptPosition" : "1",
}
```

```
"chatSessionId" : "000F3aBC3NXB001F",
"userId" : "017356D8446D002A",
"secureKey" : "ffe843f41bfleec2",
"checkChatServiceLoadBalancerPath" : "null",
"chatServerLoadBalancerAlias" : "371",
"chatServerHost" : "bsdemo2012jac",
"chatWebApiPort" : "9002",
"isTLSrequired" : "false",
"clientTimeZoneOffset" : "60",
"_chatIxnAPI_SEND_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/send",
"_chatIxnAPI_REFRESH_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/refresh",
"_chatIxnAPI_START_TYPING_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/startTyping",
"_chatIxnAPI_STOP_TYPING_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/stopTyping",
"_chatIxnAPI_DISCONNECT_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/disconnect",
"_chatIxnAPI_REFRESH_FROM_START_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/refresh?transcriptPosition=1",
"_chatIxnAPI_SEND_CUSTOM_URL" : "/genesys/1/service/fb80ed7b-a164-46ef-a5b8-f14f99236042/ixn/chat/customNotice"
}
```

Warning

The current implementation supports simple key/value parameters. You cannot manage complex JSON-structured values such as maps and arrays. See the above code example above.

Chat Interaction API Resources

The chat interaction is used to represent the current state of the chat session and transcript. This information is returned in the HTTP response of each API request in poll mode or delivered asynchronously in push mode (CometD).

Important

You need to create the service session with the basic chat service before you can create a chat interaction. Since 8.5.1, you can specify push notification parameters to manage your application's background state (see sections below).

The Chat Interaction Attributes are the following:

- chatIxnState – The current state of the chat session.
- chatSessionId – Session ID associated with the chat.
- transcriptPosition – The current position in the chat dialog or transcript for this user.

- chatServiceMessage – A diagnostic message used for debugging.

The following are only returned if the `_verbose` parameter in the API request is true:

- userId – User ID assigned by the Genesys Chat Server.
- secureKey – The security key for this chat session.
- checkChatServiceLoadBalancer – Indicates that we should check the chat load balancer for the appropriate Chat Server to use.
- PathchatServerLoadBalancerAlias – The alias for the Chat Server that is assigned to this chat session by the Chat Server load balancer.
- chatServerHost – Host name for the Chat Server assigned to this chat session from the Chat Server load balancer.
- chatWebApiPort – Port number of the Chat Server load balancer
- isTLSrequired – Indicates whether a TLS connection is required for the Chat Server.
- clientTimeZoneOffset – Time zone offset specified by the user client. Could be used to convert UTC time returned by server into user local time.
- _chatIxnAPI_SEND_URL – URL used to send chat messages for this chat session.
- _chatIxnAPI_REFRESH_URL – URL used to refresh the chat transcript for this chat session.
- _chatIxnAPI_START_TYPING_URL – URL used to indicate that the user started typing a chat message for this chat session.
- _chatIxnAPI_STOP_TYPING_URL – URL used to indicate that the user stopped typing a chat message for this chat session.
- _chatIxnAPI_DISCONNECT_URL – URL used to disconnect the user from the chat session.
- _chatIxnAPI_REFRESH_FROM_START_URL – URL used to refresh the chat transcript from the beginning of the session.
- _chatIxnAPI_SEND_CUSTOM_URL – URL used to send a custom notice to the chat session.

Example of Chat Interaction Resources

```
/genesys/1/service/{sessionid}/ixn/chat
?notify_by=comet&firstName=Buzz&lastName=Brain
cject=French&email=b.b%40gmail.com
&push_notification_deviceid=deviceid
&push_notification_type=ios

{
  "chatIxnState": "CONNECTED",
  "chatSessionId": "000C2a7VVQRB001U",
  "transcriptPosition": 1,
  "chatServiceMessage": "Chat service is available"
}
```

```
/genesys/1/service/{sessionid}/ixn/chat?_verbose=true
-ify_by=comet&firstName=Buzz&lastName=Braincject=French
&email=b.b%40gmail.com&push_notification_deviceid=deviceid
&push_notification_type=ios
```

```
{
  "chatIxnState": "CONNECTED",
  "chatSessionId": "000C2a7VVQRB001U",
  "transcriptPosition": "1",
  "chatServiceMessage": "Chat service is available",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfalcf6",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp?
chatServerLoadBalancerAlias=350",
  "chatServerLoadBalancerAlias": "350",
  "chatServerHost": "135.225.51.225",
  "chatWebApiPort": "4856",
  "isTLSrequired": "false",
  "clientTimeZoneOffset": "-420",
  "_chatIxnAPI_SEND_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/send",
  "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/refresh",
  "_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/startTyping",
  "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/stopTyping",
  "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/disconnect",
  "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/refresh?transcriptPosition=1"
}
```

Chat Transcript Resource

The Chat Transcript Attributes are the following:

- **startedAt** - Chat interaction start time (in UTC).
- **transcriptToShow** - An ordered array of transcript events. Each event is represented by another array of the following format:
[{Event type}, {Agent nickname}, {Chat message}, {Number of seconds from interaction start}, {Type of user}, {message index}] where:
 - Event type: {"Message.Text", "Notice.Joined", "Notice.Left", "Notice.TypingStart", "Notice.TypingStop", "Notice.PushUrl"}
 - Type of user: {"AGENT", "CLIENT", "EXTERNAL"}

Starting in 8.5.105.xx, the message index field has been added to the chat message to indicate the position of each message in the transcript. Your chat client application can now manage the comet and the API response channel and merge the messages according to their index in the transcript. This will avoid duplicate messages on screen.

This example shows how to send messages with their transcript position:

```
$ curl --data "transcriptPosition=1&message=good"
http://localhost:8080/genesys/1/service
/9f704400-7967-4586-821e-2e2ad5c1585f/ixn/chat/send
{
  "chatIxnState" : "TRANSCRIPT",
  "transcriptPosition" : "3",
  "chatSessionId" : "000F7aBK86UC001F",
  "transcriptToShow" :
  [[{"Notice.Joined","Kristi Sippola","has joined the session",
    "21","AGENT","2"}],
  [{"Message.Text","127.0.0.1","good","35","CLIENT","3"}]],
  "startedAt" : "2016-06-01T14:51:37Z"
}

$ curl --data "transcriptPosition=4&message=right"
http://localhost:8080/genesys/1/service
/9f704400-7967-4586-821e-2e2ad5c1585f/ixn/chat/send
{
  "chatIxnState" : "TRANSCRIPT",
  "transcriptPosition" : "15",
  "chatSessionId" : "000F7aBK86UC001F",
  "transcriptToShow" :
  [[{"Notice.TypingStarted","Kristi Sippola",
    "user is typing","55","AGENT","5"},
  [{"Message.Text","Kristi Sippola","position 2","57","AGENT","6"}],
  [{"Notice.TypingStarted","Kristi Sippola",
    "user is typing","57","AGENT","7"},
  [{"Message.Text","Kristi Sippola","hello","57","AGENT","8"}],
  [{"Notice.TypingStarted","Kristi Sippola",
    "user is typing","58","AGENT","9"},
  [{"Message.Text","Kristi Sippola","allo","58","AGENT","10"}],
  [{"Notice.TypingStarted","Kristi Sippola",
    "user is typing","58","AGENT","11"},
  [{"Message.Text","Kristi Sippola","bonjour","58","AGENT","12"}],
  [{"Notice.TypingStarted","Kristi Sippola",
    "user is typing","58","AGENT","13"},
  [{"Message.Text","Kristi Sippola","Hallo","59","AGENT","14"}],
  [{"Message.Text","127.0.0.1","right","65","CLIENT","15"}]],
  "startedAt" : "2016-06-01T14:51:37Z"
}
```

Important

The index data set of the transcript allows missing numbers due to configurable filtering events system. For example, if you have index series like "1, 3, 4, 5, 7", the unavailable "2, 6" index numbers can be filtered events).

Refresh Chat Transcript Examples

```
/genesys/1/service/{sessionId}/ixn/chat/refresh?message=hello%20agent
```

```
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": 5,
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow":
  [
    ["Notice.Joined", "ksippo", "has joined the session", "35", "AGENT"],
    ["Notice.TypingStarted", "ksippo", "is typing", "42", "AGENT"],
    ["Message.Text", "ksippo", "hello customer", "48", "AGENT"],
    ["Message.Text", "VasyaP", "hello agent", "71", "CLIENT"]
  ],
  "startedAt": "2012-06-09T06:15:35Z"
}
```

```
/genesys/1/service/{sessionId}/ixn/chat/refresh?_verbose=true &message=hello%20agent
```

```
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [
    [
      "Notice.Joined",
      "ksippo",
      "has joined the session",
      "15",
      "AGENT"
    ],
    [
      "Message.Text",
      "VasyaP",
      "hello agent",
      "26",
      "CLIENT"
    ],
    [
      "Notice.TypingStarted",
      "ksippo",
      "is typing",
      "57",
      "AGENT"
    ],
    [
      "Message.Text",
      "ksippo",
      "hello customer",

```

```
        "61",
        "AGENT"
    ],
    "startedAt": "2012-06-09T22:26:17Z",
    "userId": "015E4FD3CD890036",
    "secureKey": "b306749dabfa1cf6",
    "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
    "chatServerLoadBalancerAlias": "350",
    "chatServerHost": "135.225.51.225",
    "chatWebApiPort": "4856",
    "isTLSrequired": "false",
    "clientTimeZoneOffset": "-420",
    "_chatIxnAPI_SEND_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/send",
    "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/refresh",
    "_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/startTyping",
    "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/stopTyping",
    "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/disconnect",
    "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f
/ixn/chat/refresh?transcriptPosition=1"
}
```

Basic Chat Service Resources

Basic Chat: genesys/1/service/request-chat

```
{
  "_id": "a7e6ed0b-0380-4223-97f8-b3c7d93205e8"
}
```

Basic Chat: genesys/1/service/request-chat?_verbose=true

```
{
  "_chatIxnAPI-CREATE-URL":
"/genesys/1/service/a7e6ed0b-0380-4223-97f8-b3c7d93205e8/ixn/chat",
  "_id": "a7e6ed0b-0380-4223-97f8-b3c7d93205e8"
}
```

Basic Chat Service API

Create basic chat service

This API allows the application to create basic chat service session and then initiate chat interaction

immediately or when user is ready. **Note:** If agent availability need to be checked before chat interaction is started - use one of the advanced sessions (for example: request-chat-poll)

Operation

Method	POST		
URL	/genesys/1/service/request-chat		
Parameter	Type	Mandatory	Description
URI Parameters			
'request-chat'	String	yes	Name of the preconfigured basic chat service
Body: The body will be x-www-form-urlencoded form consisting of different items representing the key/value pairs associated with the request.			
Body Properties: The following are the properties:			
• <code>_verbose</code> - This will allow the application to get all the detail attributes associated with the chat session in the corresponding response. • ... - Any other business data attributes can also be passed.			

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	None
HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is send if the service has not sent a notification to the application that an agent is available.

Example Request:

```
POST http://localhost:8080/genesys/1/service/request-chat HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
_verbose=true
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:49:46 GMT
Pragma: no-cache
Cache-Control: no-cache
```

```
Cache-Control: no-store
Content-Type: application/json
Transfer-Encoding: chunked
{
  "_chatIxnAPI-CREATE-URL":
  "/genesys/1/service/81f0ef4e-99dd-43ea-8366-8d27a2cbd605/ixn/chat",
  "_id": "81f0ef4e-99dd-43ea-8366-8d27a2cbd605"
}
```

Chat Interaction APIs

Start Chat

This API creates and initiates a Chat Session. It works with the service session created through Genesys Mobile Services.

Operation

POST /genesys/1/service/{service_id}/ixn/chat			
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The identifier of the service that the chat session is suppose to be associated with.
Body: The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.			
_verbose	boolean	yes if JSON	Allows the application to get all the detail attributes associated with the chat session in the corresponding response.
message	string	yes if JSON	Contains a custom notice message that replaces the standard system notice in the chat session/transcript.
notify_by	string	no	If specified, should be "comet".
firstName	string	no	User's first name. Optional.
lastName	string	yes	User's last name. Optional.
email	string	yes	User's email address. Optional.
subject	string	yes	Subject of the chat conversation.

POST /genesys/1/service/{service_id}/ixn/chat			
subscriptionID	string	yes	ID of the subscription created to receive specific events on Comet channel disconnection.
userDisplayName	string	yes	Nickname displayed in the chat conversation. Optional.
push_notification_deviceid	string	no	Device ID to use for chat push notifications (used to manage background state). If not specified, push notifications are disabled.
push_notification_type	string	no	Device Type to use for chat push notification. Possible values are ios, gcm, android, comet, httpcb, orscb.
push_notification_debug	boolean	no	Set to true to enable debug push notifications in the GMS server. Default value is false.
push_notification_language	string	no	Set the language to use for chat push notifications. (See localization instructions here).
push_notification_maxsize	integer	no	Limit the payload size of push notifications messages. If you do not add the <code>push_notification_maxsize</code> parameter to your query, the the payload size is set to 4096 bytes by default.

Tip
 For iOS version 7 and earlier, your iOS device limits the push notification payload size to 256 bytes. Set this parameter to 256 to ensure correct push notifications.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	The chat session id will be the service ID. The Genesys Mobile Services code for this API will keep track of the service ID to the chat server session.

Error

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	Returned if the service has not sent a notification to the application that an agent is available.

Example

Request:

```
POST http://localhost:8080/genesys/1/service
/9d6c31d3-1121-4ba9-91e1-b93c0fa6e32f/ixn/chat?_verbose=true HTTP/1.1
```

```
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
firstName=Vasya&lastName=Pupkin&email=Vasya.Pupkin@genesyslab.com
cject=test
```

Response (if transcriptPosition input parameter is null):

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "startedAt": "2012-06-09T22:26:17Z",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfalcf6",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
  "chatServerLoadBalancerAlias": "350",
  "chatServerHost": "localhost",
  "chatWebApiPort": "4856",
  "isTLSrequired": "false",
```

```
"clientTimeZoneOffset": "-420",
"_chatIxnAPI_SEND_URL":
"/genesys/1/service/{service_id}/ixn/chat/send",
"_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh",
"_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/startTyping",
"_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/stopTyping",
"_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/{service_id}/ixn/chat/disconnect",
"_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh?transcriptPosition=1"
}
```

Response (if the transcript input parameter is set [transcriptToShow output parameter is set]):

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [
    [
      "Notice.Joined",
      "ksippo",
      "has joined the session",
      "15",
      "AGENT"
    ],
    [
      "Message.Text",
      "VasyaP",
      "hello agent",
      "26",
      "CLIENT"
    ],
    [
      "Notice.TypingStarted",
      "ksippo",
      "is typing",
      "57",
      "AGENT"
    ],
    [
      "Message.Text",
      "ksippo",
      "hello customer",
      "61",
      "AGENT"
    ]
  ],
  "startedAt": "2012-06-09T22:26:17Z",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfa1cf6",
}
```

```

    "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
    "chatServerLoadBalancerAlias": "350",
    "chatServerHost": "localhost",
    "chatWebApiPort": "4856",
    "isTLSrequired": "false",
    "clientTimeZoneOffset": "-420",
    "_chatIxnAPI_SEND_URL":
"/genesys/1/service/{service_id}/ixn/chat/send",
    "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh",
    "_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/startTyping",
    "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/stopTyping",
    "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/{service_id}/ixn/chat/disconnect",
    "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh?transcriptPosition=1"
}

```

Refresh Chat

This API refreshes the users view with the latest updates to the Chat session. It can also be used to simultaneously send a user message to the chat session.

Operation

POST /genesys/1/service/{service_id}/ixn/chat/refresh			
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The identifier of the service that the chat session is associated with.
Body: The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.			
Body Properties: The following are the properties:			
- transcriptPosition - This optional property indicates the current position in the chat session that the current user is in. This property is ignored when notify_by = comet when starting the chat session (ixn/chat) - message - This optional property is a chat message that will be added to the chat session/transcript. _verbose - This optional property will allow the application to get all the detail attributes associated with the chat session in the corresponding response.			

Response

HTTP code	200
HTTP message	OK

HTTP code	200
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	The main property is the list of chat message that have been communicated (transcriptToShow).

Error

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is returned if the service has not sent a notification to the application that an agent is available.

Example

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001H
/ixn/chat/refresh?_verbose=true HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
message=aaa
```

Response (if transcriptPosition input parameter is null):

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "startedAt": "2012-06-09T22:26:17Z",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfalcf6",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
  "chatServerLoadBalancerAlias": "350",
  "chatServerHost": "localhost",
  "chatWebApiPort": "4856",
  "isTLSrequired": "false",
  "clientTimeZoneOffset": "-420",
  "_chatIxnAPI_SEND_URL":
"/genesys/1/service/{service_id}/ixn/chat/send",
  "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh",
  "_chatIxnAPI_START_TYPING_URL":
```

```
"/genesys/1/service/{service_id}/ixn/chat/startTyping",
  "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/stopTyping",
  "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/{service_id}/ixn/chat/disconnect",
  "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh?transcriptPosition=1"
}
```

Response (if transcript input parameter is set [transcriptToShow output parameter is set]):

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [
    [
      "Notice.Joined",
      "ksippo",
      "has joined the session",
      "15",
      "AGENT"
    ],
    [
      "Message.Text",
      "VasyaP",
      "hello agent",
      "26",
      "CLIENT"
    ],
    [
      "Notice.TypingStarted",
      "ksippo",
      "is typing",
      "57",
      "AGENT"
    ],
    [
      "Message.Text",
      "ksippo",
      "hello customer",
      "61",
      "AGENT"
    ]
  ],
  "startedAt": "2012-06-09T22:26:17Z",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfa1cf6",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
  "chatServerLoadBalancerAlias": "350",
  "chatServerHost": "localhost",
  "chatWebApiPort": "4856",
}
```

```

    "isTLSrequired": "false",
    "clientTimeZoneOffset": "-420",
    "_chatIxnAPI_SEND_URL":
"/genesys/1/service/{service_id}/ixn/chat/send",
    "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh",
    "_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/startTyping",
    "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/stopTyping",
    "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/{service_id}/ixn/chat/disconnect",
    "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh?transcriptPosition=1"
}

```

Start Typing

This API allows the application to indicate that the user started typing a chat message for the session.

Operation

POST /genesys/1/service/{service_id}/ixn/chat/startTyping			
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The identifier of the service that the chat session is suppose to be associated with.
Body: The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.			
_verbose	boolean	yes if JSON	This will allow the application to get all the detail attributes associated with the chat session in the corresponding response.
message	string	yes if JSON	This mandatory property must contain a custom notice message that will be added to the chat session/transcript.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	None

Error

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is returned if the service has not sent a notification to the application that an agent is available.

Example

Request:

```
POST http://localhost:8080/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG
00001J/ixn/chat/startTyping HTTP/1.1
```

```
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:38:38 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 246
{
  "chatIxnState": "TRANSCRIPT",
  "transcriptPosition": "8",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [
    [
      "Notice.TypingStarted",
      "VasyaP",
      "is typing",
      "57",
      "CLIENT"
    ]
  ],
  "startedAt": "2012-06-10T07:37:42Z"
}
```

Stop Typing

This API allows the application to indicate that the user has stopped typing a chat message for the session.

Operation

POST /genesys/1/service/{service_id}/ixn/chat/stopTyping			
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The identifier of the

POST /genesys/1/service/{service_id}/ixn/chat/stopTyping			
			service that the chat session is suppose to be associated with.
Body: The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.			
_verbose	boolean	yes if JSON	This will allow the application to get all the detail attributes associated with the chat session in the corresponding response.
message	string	yes if JSON	This mandatory property must contain a custom notice message that will be added to the chat session/transcript.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	None

Error

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is returned if the service has not sent a notification to the application that an agent is available.

Example

Request:

```
POST http://localhost:8080/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG
00001J/ixn/chat/stopTyping HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:38:58 GMT
Pragma: no-cache
Cache-Control: no-cache
```

```

Cache-Control: no-store
Content-Type: application/json
Content-Length: 251
{
  "chatIxnState": "TRANSCRIPT",
  "transcriptPosition": "9",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [ [
    "Notice.TypingStopped",
    "VasyaP",
    "stopped typing",
    "77",
    "CLIENT"
  ] ],
  "startedAt": "2012-06-10T07:37:42Z"
}

```

Disconnect from chat session

This API allows the application to disconnect user from the chat session.

Operation

POST /genesys/1/service/{service_id}/ixn/chat/disconnect			
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The identifier of the service that the chat session is suppose to be associated with.
Body: The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.			
_verbose	boolean	yes if JSON	This will allow the application to get all the detail attributes associated with the chat session in the corresponding response.
message	string	yes if JSON	This mandatory property must contain a custom notice message that will be added to the chat session/transcript.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	None

Error

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is returned if the service has not sent a notification to the application that an agent is available.

Example Request:

```
POST http://localhost:8080/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03D
BG00001J/ixn/chat/disconnect HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:43:07 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 114
{
  "chatIxnState" : "DISCONNECTED",
  "transcriptPosition" : "9",
  "chatServiceMessage" : "Chat was finished"
}
```

Send Custom Notice

This API query allows you to send a user notice that contains a custom message.

Operation

Method	POST		
URL	/genesys/1/service/{service_id}/ixn/chat/customNotice		
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	Yes	The identifier of the service associated with the chat session.
Body	The body can be either a MultiPart form, or x-www-form-urlencoded form, or JSON key/value pairs associated with the request.		
Body Properties			
_verbose	boolean	No	Set to true to allow the application to get all the detail attributes

Method	POST		
			associated with the chat session in the corresponding response.
message	string	Yes	Custom notice message that will replace the standard system notice in the chat session transcript.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for the details of the object properties. See the data structures for more details.

Error

If a problem occurs during subscription, the following status codes are returned:

HTTP code	503
HTTP Message	Service Unavailable
Body	None

Important

This error is returned if the service could not send a notification to the agent application.

Example

Request

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat
/customNotice?_verbose=true HTTP/1.1
```

```
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
message=photo.png
```

Response

```
HTTP/1.1 200 OK
Date: Sat, 09 Jun 2012 22:26:16 GMT
Pragma: no-cache
Cache-Control: no-cache
```

```
Cache-Control: no-store
Content-Type: application/json
Content-Length: 1225
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000BRa84KRFB00BK",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "startedAt": "2012-06-09T22:26:17Z",
  "userId": "015E4FD3CD890036",
  "secureKey": "b306749dabfalcf6",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=350",
  "chatServerLoadBalancerAlias": "350",
  "chatServerHost": "localhost",
  "chatWebApiPort": "4856",
  "isTLSrequired": "false",
  "clientTimeZoneOffset": "-420",
  "_chatIxnAPI_SEND_URL":
"/genesys/1/service/{service_id}/ixn/chat/send",
  "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/{service_id}/ixn/chat/refresh",
  "_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/startTyping",
  "_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/{service_id}/ixn/chat/stopTyping",
  "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/{service_id}/ixn/chat/disconnect",
  "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/{service_id}/ixn/chat
/refresh?transcriptPosition=1",
  "_chatIxnAPI_SEND_CUSTOM_URL" :
"/genesys/1/service/{service_id}/ixn/chat/customNotice"
}
```

Quick Start Examples

The following quick start examples show how you can establish a CometD connection to receive asynchronous notification, and how to create a service.

Using CometD to Receive Event Updates

If you are using CometD to get event updates on the chat session then you need to set up a CometD connection with a subscription for `/_genesys`. You also need to make sure 'gms_user' header in all cometd related requests is set to the value uniquely representing application end user. Typically this value would be setup (or at least verified) by security gateway located between the client application and GMS.

CometD handshake request

```
POST http://localhost:8080/genesys/cometd
Accept-Encoding: gzip,deflate
Content-Type: application/json;charset=UTF-8
gms_user: BuzzBrain
{"version": "1.0", "minimumVersion": "0.9",
"channel": "/meta/handshake", "id": "0"}
```

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 230
[{"id":"0","minimumVersion":"1.0","supportedConnectionTypes":
["websocket","callback-polling","long-polling"],
"successful":true,"channel":"/meta/handshake","ext":{"ack":true},
"clientId":"44xkkazwfabw73jrvjsvoy4ul","version":"1.0"}]
```

CometD /meta/connect subscription request

```
POST http://localhost:8080/genesys/cometd
Accept-Encoding: gzip,deflate
Content-Type: application/json;charset=UTF-8
gms_user: BuzzBrain
{"channel":"/meta/connect",
"clientId":"44xkkazwfabw73jrvjsvoy4ul","id":"1",
"connectionType":"long-polling"}
```

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 116
[{"id":"1","successful":true,"advice":
{"interval":0,"reconnect":"retry","timeout":60000},
"channel":"/meta/connect"}]
```

CometD /_genesys subscription request

```
POST http://localhost:8080/genesys/cometd Accept-Encoding: gzip,deflate
Content-Type: application/json;charset=UTF-8
gms_user: BuzzBrain
[{"channel":"/meta/subscribe","subscription":"/_genesys",
"clientId":"44xkkazwfabw73jrvjsvoy4ul","id":"2"}]
```

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 85
[{"id":"2","subscription":"/_genesys","successful":true,
"channel":"/meta/subscribe"}]
```

CometD long polling request

```
POST http://localhost:8080/genesys/cometd
Accept-Encoding: gzip,deflate
Content-Type: application/json;charset=UTF-8
gms_user: BuzzBrain
{"clientId":"44xkkazwfabw73jrvjsvoy4ul","id":"3",
"channel":"/meta/connect","connectionType":"long-polling"}
```

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 85
[{"id":"4","successful":true,"channel":"/meta/connect"}]
```

Creating a Genesys Mobile Services-Based Service Associated with a Chat Session

The following section illustrates the process of creating and using a service.

Create a Service:

Request:

```
POST http://localhost:8080/genesys/1/service/request-chat-poll HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
_verbose=false
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:23:29 GMT
Content-Type: application/json
{"_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001M"}
```

Use the `_id` field from the response to check service status until it changes to "available":

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/status HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:26:26 GMT
Content-Type: application/json
Content-Length: 185
{
  "message": {
    "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001M",
    "_status": "waiting",
    "_dialog": "waiting_for_agent.html"
  },
  "tag": "a2c.advanced.service.statuschanged
.EKUJPKAQ197CFA6SJQKTJ03DBG00001M"
}
```

Repeat request until status changes:

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/status HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
```

```
Date: Sun, 10 Jun 2012 08:28:25 GMT
Content-Type: application/json
Content-Length: 186
{
  "message": {
    "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001M",
    "_status": "available",
    "_dialog": "agent_available.html"
  },
  "tag": "a2c.advanced.service.agentavailable
.EKUJPKAQ197CFA6SJQKTJ03DBG00001M"
}
```

Create chat interaction using same sessionid:

To create a chat interaction that is associated with a service, a ixn/chat request is sent with the parameters to initiate the chat session.

Parameter Name	Mandatory	Description
firstName	no	First name of the user. If provided will be attached as fldnFirstName to the chat interaction.
lastName	no	Last name of the user. If provided will be attached as fldnLastName to the chat interaction.
email	no	e-Mail address of the subject. If provided will be attached as fldnEmailAddress to the chat interaction.
subject	yes	Subject of the service and chat session.
userDisplayName	no	Available since GMS 8.1.100.27. Nickname to be displayed in the chat conversation.
notify_by	no	If using a CometD connection for the asynchronous receiving of chat messages, then supply this parameter with the value "comet".
push_notification_deviceid	no	Device ID to use for chat push notifications (used to manage background state). If not specified, push notifications are disabled.
push_notification_type	no	Device Type to use for chat push notification. Possible values are: ios, gcm, android, comet, httpcb, orscb.
push_notification_debug	no	Set to true to enable debug push notifications in the GMS server. Default value is false.

Parameter Name	Mandatory	Description
push_notification_language	no	Set the language to use for chat push notifications. (See localisation instructions here)
push_notification_maxsize	no	Limit the payload size of push notifications messages. If you do not add the push_notification_maxsize parameter to your query, the payload size is set to 4096 bytes by default. Tip For iOS version 7 and earlier, your iOS device limits the push notification payload size to 256 bytes. Set this parameter to 256 to ensure correct push notifications.

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/ixn/chat HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
notify_by=comet&firstName=Vasya&lastName=Pupkin
&email=Vasya.Pupkin@genesyslab.com<ject=test
&push_notification_deviceid=deviceid&push_notification_type=ios
&push_notification_debug=true&push_notification_language=en
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 119
{
  "chatIxState" : "CONNECTED",
  "transcriptPosition" : "1",
  "chatServiceMessage" : "Chat service is available"
}
```

Refresh chat transcript and show messages to the user:**Request:**

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/ixn/chat/refresh HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:33:00 GMT
Content-Type: application/json
Content-Length: 367
{"_id": "B2FS3346K151548QMEAFD89TE8000EBJ",
```

```
"comet_channel":"/_genesys"}
```

Send user's message:

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/ixn/chat/refresh HTTP/1.1

Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
message=hello agent
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:34:38 GMT
Content-Type: application/json
Content-Length: 241
{"_id":"B2FS3346K151548QMEAFD89TE8000EBJ",
"comet_channel":"/_genesys"}
```

Disconnect user from chat:

Request:

```
POST http://localhost:8080/genesys/1/service
/EKUJPKAQ197CFA6SJQKTJ03DBG00001M/ixn/chat/disconnect HTTP/1.1

Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:43:07 GMT
Content-Type: application/json
Content-Length: 114
{
  "chatIxnsState" : "DISCONNECTED",
  "transcriptPosition" : "9",
  "chatServiceMessage" : "Chat was finished"
}
```

CometD-Based Chat API

CometD Handshake

Request

```
POST http://localhost:8080/genesys/cometd/handshake
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/json;charset=UTF-8
[{"version":"1.0","minimumVersion":"0.9",
"channel":"/meta/handshake",
```

```
"supportedConnectionType":["long-polling","callback-polling"],
"advice":{"timeout":60000,"interval":0},"id":"1"}}
```

Response

```
Content-Type: application/json;charset=UTF-8
[{"id":"1","minimumVersion":"1.0","supportedConnectionTypes":
["callback-polling","long-polling"],
"successful":true,"channel":"/meta/handshake","ext":{"ack":true},
"clientId":"3vym301sjdtc218qabm5w0z8yb","version":"1.0"}]
```

CometD Subscribe

Request

```
POST http://localhost:8080/genesys/cometd/
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673

Content-Type: application/json;charset=UTF-8
[{"channel":"/meta/subscribe","subscription":"/_genesys","id":"2",
"clientId":"3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8
[{"id":"2","subscription":"/_genesys","successful":true,
"channel":"/meta/subscribe"}]
```

CometD Connect

Request

Note, ext component containing transcriptPosition is optional, see section on network connection loss below.

```
POST http://localhost:8080/genesys/cometd/connect

gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673

Content-Type: application/json;charset=UTF-8
[{"channel":"/meta/connect","connectionType":"long-polling",
"advice":{"timeout":0},
"id":"3","clientId":"3vym301sjdtc218qabm5w0z8yb"},
"ext":{"transcriptPosition": "4"}]
```

Response

```
Content-Type: application/json;charset=UTF-8
[{"id":"3","successful":true,"advice":
{"interval":0,"reconnect":"retry","timeout":60000},
"channel":"/meta/connect"}]
```

Create Service Session

Request

```
POST http://localhost:8080/genesys/1/service/request-chat
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
```

```
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
```

```
_verbose=true
```

Response

```
Content-Type: application/json; charset=UTF-8
{
  "_chatIxnAPI-CREATE-URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn/chat",
  "_id": "4d1697a9-dda5-4742-8a6f-fbc01c25c640",
  "_data_id": "429-791eaae8-571e-4633-9a73-cc936336f8e2"
}
```

Create Chat Interaction for Session 4d1697a9-dda5-4742-8a6f-fbc01c25c640

Request

```
POST
http://localhost:8080/genesys/1/service
/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn/chat
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
```

```
_verbose=true&ify_by=comet&FirstName=John&LastName=Harry
cject=French&EmailAddress=j.h%40gmail.com
```

Response

```
Content-Type: application/json; charset=UTF-8
{
  "chatServerLoadBalancerAlias": "371",
  "_chatIxnAPI_SEND_CUSTOM_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn/chat/customNotice",
  "clientTimeZoneOffset": "120",
  "transcriptPosition": "1",
  "chatWebApiPort": "9002",
  "chatIxnState": "CONNECTED",
  "comet_channel": "/_genesys",
  "secureKey": "1b21478a91a7d1dc",
  "checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=371",
  "_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/refresh?transcriptPosition=1",
  "_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/refresh",
  "chatSessionId": "000E5aA2A40P000Q",
  "isTLSrequired": "false",
  "_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/disconnect",
  "chatServiceMessage": "Chat service is available",
  "userId": "0173542518870006",
}
```

```
"_chatIxnAPI_STOP_TYPING_URL":  
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640  
/ixn/chat/stopTyping",  
"_chatIxnAPI_START_TYPING_URL":  
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640  
/ixn/chat/startTyping",  
"_chatIxnAPI_SEND_URL":  
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640  
/ixn/chat/send",  
"chatServerHost": "demosrv.genesyslab.com"  
}
```

Polling Agent 'Joined' Message Through CometD

Request

```
POST http://localhost:8080/genesys/cometd/connect  
gms_user:  
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673  
Content-Type: application/json;charset=UTF-8
```

```
[{"channel":"/meta/connect","connectionType":"long-polling",  
"id":"4","clientId":"3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8  
[  
{  
  "data":{  
    "id":"7b239ff0455011e4b31cbb293fafa316",  
    "message":{"chatSessionId":"000E5aA2A40P000Q",  
    "transcriptPosition":"2",  
    "chatServiceMessage":"Chat service is available",  
    "startedAt":"2014-09-26T07:40:55Z",  
    "chatIxnState":"TRANSCRIPT",  
    "transcriptToShow":[ ["Notice.Joined","Kristi Sippola",  
    "has joined the session","14","AGENT"]]},  
    "tag":"service.chat.refresh.4d1697a9-dda5-4742-8a6f-fbc01c25c640"},  
    "channel":"/_genesys"},  
    {"id":"4","successful":true,"channel":"/meta/connect"}  
  ]  
}
```

Polling Agent 'StartTyping' Message Through CometD

Request

```
POST http://localhost:8080/genesys/cometd/connect  
gms_user:  
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673  
Content-Type: application/json;charset=UTF-8
```

```
[{"channel":"/meta/connect","connectionType":"long-polling",  
"id":"5","clientId":"3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8  
[  
{  
  "data":{  
    "id":"802c88e0455011e4b31cbb293fafa316",  
    "message":{"chatSessionId":"000E5aA2A40P000Q",
```

```
"transcriptPosition": "3",
"chatServiceMessage": "Chat service is available",
"startedAt": "2014-09-26T07:40:55Z",
"chatIxnState": "TRANSCRIPT",
"transcriptToShow":
[["Notice.TypingStarted", "Kristi Sippola",
"is typing", "22", "AGENT"]]],
"tag": "service.chat.refresh.4d1697a9-dda5-4742-8a6f-fbc01c25c640",
"channel": "/_genesys",
{"id": "5", "successful": true, "channel": "/meta/connect"}
]
```

Polling Agent Chat Message Through CometD

Request

```
POST http://localhost:8080/genesys/cometd/connect
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/json; charset=UTF-8
```

```
[{"channel": "/meta/connect", "connectionType": "long-polling",
"id": "6", "clientId": "3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json; charset=UTF-8
[
{"data": {"id": "816f9030455011e4b31cbb293fafa316",
"message": {"chatSessionId": "000E5aA2A40P000Q",
"transcriptPosition": "4",
"chatServiceMessage": "Chat service is available",
"startedAt": "2014-09-26T07:40:55Z",
"chatIxnState": "TRANSCRIPT",
"transcriptToShow": [{"Message.Text", "Kristi
Sippola", "Hello", "23", "AGENT"}]},
"tag": "service.chat.refresh.4d1697a9-dda5-4742-8a6f-fbc01c25c640",
"channel": "/_genesys"},
{"id": "6", "successful": true, "channel": "/meta/connect"}
]
```

Send Client Chat Message

Request

```
POST
http://localhost:8080/genesys/1/service
/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn/chat/refresh
gms_user:
"b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

_verbose=true&message=Hi%20Verbose
```

Response

```
Content-Type: application/json; charset=UTF-8
{
"chatServerLoadBalancerAlias": "371",
```

```
"_chatIxnAPI_SEND_CUSTOM_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/customNotice",
"clientTimeZoneOffset": "120",
"transcriptPosition": "5",
"chatWebApiPort": "9002",
"startedAt": "2014-09-26T07:40:55Z",
"chatIxnState": "TRANSCRIPT",
"comet_channel": "/_genesys",
"secureKey": "1b21478a91a7d1dc",
"checkChatServiceLoadBalancerPath":
"/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=371",
"_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/refresh?transcriptPosition=1",
"_chatIxnAPI_REFRESH_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/refresh",
"isTLSrequired": "false",
"chatSessionId": "000E5aA2A40P000Q",
"_chatIxnAPI_DISCONNECT_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/disconnect",
"chatServiceMessage": "Chat service is available",
"userId": "0173542518870006",
"_chatIxnAPI_STOP_TYPING_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/stopTyping",
"_chatIxnAPI_START_TYPING_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/startTyping",
"_chatIxnAPI_SEND_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640
/ixn/chat/send",
"chatServerHost": "demosrv.genesyslab.com"
}
```

Client Message is Being Echoed Back Through CometD Channel as a Response to "refresh" or "send" Request

Request

```
POST http://localhost:8080/genesys/cometd/connect
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/json;charset=UTF-8

[{"channel": "/meta/connect", "connectionType": "long-polling", "id": "7",
"clientId": "3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8
[
{"data": {"id": "867b3840455011e4b31cbb293fafa316",
"message": {"chatSessionId": "000E5aA2A40P000Q",
"transcriptPosition": "5",
"chatServiceMessage": "Chat service is available",
"startedAt": "2014-09-26T07:40:55Z",
```

```
        "chatIxnState": "TRANSCRIPT",
        "transcriptToShow": [{"Message.Text", "127.0.0.1",
"Hi Verbose", "32", "CLIENT"]]},
        "tag": "service.chat.refresh.4d1697a9-dda5-4742-8a6f-fbc01c25c640"},
        "channel": "/_genesys"},
        {"id": "7", "successful": true, "channel": "/meta/connect"}}
```

CometD Polling

Request

POST http://localhost:8080/genesys/cometd/connect
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/json;charset=UTF-8

```
[{"channel": "/meta/connect", "connectionType": "long-polling",
"id": "8", "clientId": "3vym301sjdtc218qabm5w0z8yb"}]
```

Response

Content-Type: application/json;charset=UTF-8
[{"id": "8", "successful": true, "advice": {"reconnect": "none"},
"channel": "/meta/connect"}]

Disconnect Chat Session

Request

POST
http://localhost:8080/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn/chat/disconnect

gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

_verbose=true

Response

Content-Type: application/json;charset=UTF-8
{
 "chatIxnState" : "DISCONNECTED",
 "transcriptPosition" : "5",
 "chatServiceMessage" : "Chat was finished",
 "chatSessionId" : "000E5aA2A40P000Q",
 "userId" : "0173542518870006",
 "secureKey" : "1b21478a91a7d1dc",
 "checkChatServiceLoadBalancerPath" :
 "/WebAPI812/SimpleSamples812/ChatHA/ChatLBServerInfo.jsp
?chatServerLoadBalancerAlias=371",
 "chatServerLoadBalancerAlias" : "371",
 "chatServerHost" : "demosrv.genesyslab.com",
 "chatWebApiPort" : "9002",
 "isTLSrequired" : "false",
 "clientTimeZoneOffset" : "120",
 "_chatIxnAPI_SEND_URL" :
 "_genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn

```
/chat/send",
"_chatIxnAPI_REFRESH_URL" :
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/refresh",
"_chatIxnAPI_START_TYPING_URL" :
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/startTyping",

"_chatIxnAPI_STOP_TYPING_URL" :
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/stopTyping",

"_chatIxnAPI_DISCONNECT_URL" :
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/disconnect",
"_chatIxnAPI_REFRESH_FROM_START_URL":
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/refresh?transcriptPosition=1",
"_chatIxnAPI_SEND_CUSTOM_URL" :
"/genesys/1/service/4d1697a9-dda5-4742-8a6f-fbc01c25c640/ixn
/chat/customNotice"
}
```

CometD Unsubscribe

Request

```
POST http://localhost:8080/genesys/cometd/
gms_user:
  b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673

Content-Type: application/json;charset=UTF-8

[{"channel":"/meta/unsubscribe","subscription":"/_genesys",
"id":"9","clientId":"3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8
[{"id":"9","subscription":"/_genesys","successful":true,
"channel":"/meta/unsubscribe"}]
```

CometD Disconnect

Request

```
POST http://localhost:8080/genesys/cometd/disconnect
gms_user:
b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Content-Type: application/json;charset=UTF-8

[{"channel":"/meta/disconnect","id":"10","clientId":
"3vym301sjdtc218qabm5w0z8yb"}]
```

Response

```
Content-Type: application/json;charset=UTF-8
[{"id":"10","successful":true,"channel":"/meta/disconnect"}]
```

Chat Push Notification and Background State

Mobile Application Background State Issues

If an iOS or Android mobile application is moved to the background state, the operating system does not close active TCP connections. So, if your app does not handle the foreground and background events, the server-side of your CometD connection isn't notified.

More specifically, if your app handles chat with GMS, the agent may send a new message which would result in sending the long poll response. Here is a list of the various scenarios which can happen according to your implementation.

- The Cometd server attempts to send long poll response but fails and closes socket of current long-poll request.
- The Cometd server sets a 10 second timer; if the app does not reconnect within this period, the app is considered to be gone and its context is destroyed.

Later, if the app returns to the foreground, it does not receive chat notifications from the Cometd server and the chat session appears to be broken.

To avoid these type of issues, your app should listen for events that both iOS and Android submit when your app enters the background (or foreground) state. See the Official documentation.

- Apple provides background instructions for iOS, [here](#).
- Android provides best background practices [here](#).

Then, your app should implement the Cometd Meta Disconnect message detailed below to handle background State issues.

Cometd Meta Disconnect Messages

To handle background state, your app must send Cometd /meta/disconnect messages to GMS and later, if your app returns to foreground, it should establish a new Cometd session as detailed below.

1. When your app enters the background state, it sends a Comet/meta/disconnect message to GMS which includes the highest transcript position received by your app, as for example:

```
{
  "channel": "/meta/disconnect",
  "clientId": "71k6evjfbffq9eir3jhn6udxz",
  "ext": {
    "transcriptPosition": "4"
  },
  "id": "4"
}
```

2. If GMS receives a `/meta/disconnect` message, it flags your app as disconnected and sends a chat Notice. Custom event to the agent. The default value is `customer is not online`. You can configure this text.
 1. Edit your GMS application (with Configuration Manager for example).
 2. Navigate to **Options > Service.<service_name>**.
 3. Edit the `_agent_timeout_notification_message` value.
3. If the agent sends a new message while your app is disconnected, GMS starts a configurable timer. You can set this timer value to zero
 1. Edit your GMS application (with Configuration Manager for example).
 2. Navigate to **Options > Service.<service_name>**.
 3. Edit the `_client_timeout_notification` value.

When the timer expires, GMS sends a push notification (iOS apns, Android gcm, or http). Typically, the notification results in a dialog which allows the user to optionally return the app to the foreground.

4. The chat session may have ended when the app returns to the foreground, either because the agent left the session or because the session timed out as a result of user inactivity. In that case, the app no longer receives Cometd notifications. To detect if the session has ended, call the Chat Refresh Chat API – it will return a 4xx reply to indicate that a session has ended.
5. When the app returns to the foreground and if the session has not ended, it must start a new Cometd session with GMS. GMS flags the app as connected and sends a Cometd notification that includes all chat transcript events that occurred after the app entered the background state, as for example:

```
[
  {
    "data":{
      "id":"fdf93730c1e411e4a5e8f1be76b28efd",
      "message":
        { "chatSessionId":"0001BaAFC4J8000N",
          "transcriptPosition":"6",
          "chatServiceMessage":"Chat service is available",
          "startedAt":"2015-03-03T20:36:12Z",
          "chatIxnState":"TRANSCRIPT", "transcriptToShow":
            [ ["Notice.TypingStarted","agentName","is typing","28","AGENT"],
              ["Message.Text","agentName","Hello","29","AGENT"] ] },
      "tag":"service.chat.refresh.180-655e104c-a6cf-447d-b94b-f132d49a35fe"
    },
    "channel":"/_genesys"
  },
  { "successful":true, "channel":"/meta/connect" }
]
```

Important

- If your app returns to foreground before the timer expires, no push notification is sent, but your app must still start a new Cometd session and it will receive a Cometd notification (which includes all new transcript data) from GMS.
- If your app does not return to foreground after the first push notification is sent, GMS sends additional push notification messages for each new agent event. Note that these

events are sent immediately.

Content of Push Notification Messages

GMS triggers push notification messages if the agent submits new chat events to the app flagged as disconnected. The following events can result in a notification:

- Notice.TypingStarted
- Notice.TypingStopped
- Notice.Joined
- Notice.Left
- Notice.PushUrl
- Notice.Custom
- Message.Text

Your application's configuration includes a filter to exclude events from triggering a push notification. The `filtering_chat_events` option in the push section sets the default value for this filter to `Notice.TypingStarted,Notice.TypingStopped`. You can still set a different value for your service.

1. Edit your GMS application (with Configuration Manager for example).
2. Navigate to **Options > Service.<service_name>**.
3. Edit the `_filtering_chat_events` option to add new event types to exclude.

For the Android and http push notification types, notification messages include the content of the filtered agent events.

The following example of the push notification message includes the content of the filtered agent events.

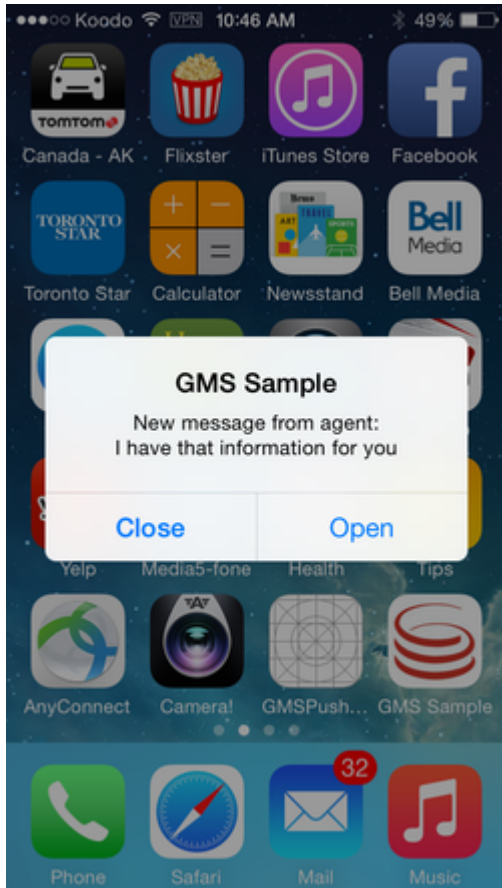
```
"message": {
  "message": "New message from Agent",
  "serviceId": "180-e941460d-1eb0-4f0b-9022-b99ca9ee41f7",
  "lastTranscript": [
    {"Message.Text": "A line of text."}
    {"Message.Text": "Another line of text."}
  ]
}
```

By default, the message attribute (or notification message) is set to `New message from Agent`, but you can change this text in your service options.

1. Edit your GMS application (with Configuration Manager for example).
2. Navigate to **Options > Service.<service_name>**.
3. Edit the value of the `_client_timeout_notification_message` option.

The notification is tagged as `chat.newagentmessage`. In addition to the notification message, the notification content provides the `lastTranscript` text that can be displayed to the user.

- You can display this the `lastTranscript` text in a popup a or banner notification to the user.



Specific Configuration for Chat Push Notification

To customize your GMS configuration for push notification messages (for both iOS and Android), you can create a `push.provider.event.chat.newagentmessage` section in the **Options** tab (with Configuration Manager for instance). You can add there any of your Android or iOS options.

- This additional configuration is not mandatory.

Tip

- Read more about options of the GMS push section [here](#).
- Read more about the OS specific capabilities associated with the notification message [here](#).

CometD and Loss of Network Connection

Mobile devices are subject to temporary loss of carrier or wifi network connectivity. If this occurs when a chat session is active, the mobile device may not receive CometD transcription notifications that occurred during or around the time of the loss.

The mobile device CometD client should detect when network connectivity has been lost; for example, the current long-poll request will return an error. When this occurs, the CometD client should establish a new session starting with a handshake sequence. To support the mobile client informing GMS on the transcription notifications already received, the first /meta/connect message that is sent after the handshake can optionally include the client's transcriptPosition in the ext component. An example connect message is shown below.

```
POST http://localhost:8080/genesys/cometd
Accept-Encoding: gzip,deflate
Content-Type: application/json;charset=UTF-8
gms_user: BuzzBrain
```

```
{
  "channel": "/meta/connect",
  "clientId": "44xkkazwfabw73jrvjsvoy4ul",
  "connectionType": "long-polling"
  "advice": {
    "timeout": 0
  },
  "ext": {
    "transcriptPosition": "14"
  },
  "id": "1"
}
```

If the transcriptPosition is provided this way the next CometD notification sent by GMS will include transcript content starting at the next position, so the client will only receive the content that was missed due to the network connection loss.

Note, If the transcriptPosition is not provided in the /meta/connect after a network loss, GMS will estimate the client transcriptPosition based on the timing of the loss of network connection. Typically this will result in the next CometD notification including some transcript content that was previously received, but could also result in the transcript content starting at an offset greater than what was received. If the client does not provide the transcriptPosition upon reconnecting after a network connection loss, it is recommended to call the refresh API function to reset the local transcript display.