



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

12/13/2025

Contents

- 1 Chat API
 - 1.1 Overview
 - 1.2 Structures
 - 1.3 Quick Start Examples
 - 1.4 Chat Interaction APIs
 - 1.5 Basic Chat Service API
 - 1.6 Advanced Chat Service API

Chat API

Warning

Note that the Chat API Version 1 feature has been deprecated. Please see the related [deprecation notice](#) for additional details.

Introduced in release: **8.1.100.14**

Overview

Prerequisite: Before using the Chat API, you must [configure your Genesys Mobile Services deployment](#) correctly.

The Chat API is used by customer-facing applications to create and manage a chat session associated with contact center-related services. One service is associated with exactly one chat session. There are two types of supported services:

1. Basic Services (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 going to be 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
- routing logic associated with specified interaction endpoint (or the configured by default value) would be responsible for finding an appropriate agent
- both polling and async (CometD) modes of message delivery are supported

2. Advance Services (Genesys Mobile Services + ORS-Based)

- allows an application to define custom service processing logic
- an application can create a service session and pass business context data
- session logic can use agent availability information and notify the application when a chat-enabled resource with the appropriate business skills is available
- session logic can also engage in pre- and post-chat interaction dialogs with the customer by pushing dialog information to the client application (for example, in URL/HTML/XML form)

- advance-chat-pool session is provided as an example of the advanced service logic and can be customized to fit particular business needs
- both polling and async (CometD) modes of message delivery are supported

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

Structures

The Chat API uses the data structures described in this section (in JSON format) to exchange data. Requests are accepted in '**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 a 415 (Unsupported Media Type) error will occur.

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). Note: you need to create service session before you can create chat interaction.

Create Chat Interaction: /genesys/1/service/{sessionid}/ixn/chat?notify_by=comet&firstName=Buzz&lastName=Brain&ject=French&email=b.b%40gmail.com

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

Create Chat Interaction: /genesys/1/service/{sessionid}/ixn/chat?_verbose=true&ify_by=comet&firstName=Buzz&lastName=Brain&ject=French&email=b.b%40gmail.com

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

Attribute Descriptions:

- 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.

Refresh Chat Transcript: `/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"
}

```

Refresh chat transcript: /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": "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"
}
```

Attributes description:

- `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}`]
Where:
 - Event type: `{"Message.Text", "Notice.Joined", "Notice.Left", "Notice.TypingStart", "Notice.TypingStop", "Notice.PushUrl"}`
 - Type of user: `{"AGENT", "CLIENT", "EXTERNAL"}`

Service API 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"
}
```

Advanced Chat Poll: `genesys/1/service/advanced-chat-poll`

```
{
  "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001C"
}
```

Advanced Chat Poll: `genesys/1/service/advanced-chat-poll?_verbose=true`

```
{
  "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001C",
  "_session_comand_STATUS_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001C/status"
}
```

Advanced Chat Poll with poke enabled: `genesys/1/service/request-chat-poll?_device_notification_id=b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb`

```
{
  "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001C",
  "_session_comand_STATUS_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001C/status"
}
```

COMETD Based Chat API

CometD handshake request:

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

{"version":"1.0","minimumVersion":"0.9","channel":"/meta/handshake","id":"0"}
```

Response:

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

[{"id":"0","minimumVersion":"1.0","supportedConnectionTypes":["callback-polling","long-polling"],"successful":true,"channel":"/meta/handshake","ext":{"ack":true},"clientId":"dljw6puyen98clbaidgmlcivxs","version":"1.0"}]
```

CometD connect request:

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

{"channel":"/meta/connect","clientId":"dljw6puyen98clbaidgmlcivxs","id":"1","connectionType":"long-polling"}
```

Response:

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

[{"id":"1","successful":true,"advice":{"interval":0,"reconnect":"retry","timeout":60000},"channel":"/meta/connect"}]
```

CometD subscribe:

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

[{"channel":"/meta/subscribe","subscription":"/_genesys","clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"2"}]
```

Response:

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

[{"id":"3","successful":true,"channel":"/meta/connect"}]
```

CometD polling:

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

{"clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"4","channel":"/meta/connect","connectionType":"long-polling"}
```

Response:

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

[{"id":"4","successful":true,"channel":"/meta/connect"}]
```

... continue cometd polling ...

Create service session:

```
POST http://localhost:8080/genesys/1/service/request-chat
Content-Type:      application/x-www-form-urlencoded; charset=UTF-8
gms_user:          BuzzBrain

_verbose=true
```

Response:

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

{"_chatIxnAPI-CREATE-URL":"/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/
chat","_id":"0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f"}
```

Create chat interaction for session 0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f:

```
POST http://localhost:8080/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat
Content-Type:      application/x-www-form-urlencoded; charset=UTF-8
gms_user:          BuzzBrain

_verbose=true&ify_by=comet&firstName=Buzz&lastName=Brain&ject=French&email=b.b%40gmail.com
```

Response:

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

{"message":"Messages will sent over CometD service channel \/_genesys"}
```

... continue cometd polling ...:

CometD polling:

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

{"clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"32","channel":"/meta/
connect","connectionType":"long-polling"}
```

Response:

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

[{"data":{"id":"ccd229401d5f11e200000805691661bf","message":{"chatSessionId":"000Bwa7RECRK001D","transcriptPosi
service is
available","startedAt":"2012-10-23T22:19:42Z","chatIxnState":"TRANSCRIPT","transcriptToShow":[]},"tag":"service
connect"}}]
```

Polling agent 'Joined' message through CometD:

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

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

```
{"clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"33","channel":"/meta/connect","connectionType":"long-polling"}
```

Response:

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

```
[{"data":{"id":"3cc44da01d6011e200000805691661bf","message":{"chatSessionId":"000Bwa7RECRK001D","transcriptPosition":1,"service is available","startedAt":"2012-10-23T22:19:42Z","chatIxNState":"TRANSCRIPT","transcriptToShow":[{"Notice.Joined","joined the session","21","AGENT"}]},"tag":"service.chat.refresh.0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f"},"channel":"/_genesys/connect"}]]
```

Polling agent 'StartTyping' message through CometD:

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

```
{"clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"34","channel":"/meta/connect","connectionType":"long-polling"}
```

Response:

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

```
[{"data":{"id":"3cc44da01d6011e200000805691661bf","message":{"chatSessionId":"000Bwa7RECRK001D","transcriptPosition":2,"service is available","startedAt":"2012-10-23T22:19:42Z","chatIxNState":"TRANSCRIPT","transcriptToShow":[{"Notice.TypingStart","typing","212","AGENT"}]},"tag":"service.chat.refresh.0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f"},"channel":"/_genesys/connect"}]]
```

Polling agent chat message through CometD:

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

```
{"clientId":"eurw5vlezp5yn17z3cdrf3jxsc","id":"35","channel":"/meta/connect","connectionType":"long-polling"}
```

Response:

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

```
[{"data":{"id":"3fc6b8301d6011e200000805691661bf","message":{"chatSessionId":"000Bwa7RECRK001D","transcriptPosition":3,"service is available","startedAt":"2012-10-23T22:19:42Z","chatIxNState":"TRANSCRIPT","transcriptToShow":[{"Message.Text","connect"}]},"tag":"service.chat.refresh.0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f"},"channel":"/_genesys/connect"}]]
```

Send client chat message:

POST http://localhost:8080/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/refresh
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

_verbose=true&message=hello+agent

Response:

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

```
{"message": "Messages will sent over CometD service channel \/_genesys"}
```

Client message is being echoed back through CometD channel as a response to "refresh" or "send" request:

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

```
{"clientId": "eurw5vlezp5yn17z3cdrf3jxsc", "id": "36", "channel": "/meta/connect", "connectionType": "long-polling"}
```

Response:

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

```
[{"data": {"id": "472e2a901d6011e200000805691661bf", "message": {"chatServerLoadBalancerAlias": "349", "clientTimeZonSimpleSamples812/ChatHA/ChatLBServerInfo.jsp?chatServerLoadBalancerAlias=349", "_chatIxnAPI_REFRESH_FROM_START_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/refresh?transcriptPosition=1", "_chatIxnAPI_REFRESH_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/refresh", "chatSessionId": "000BWa7RECRK001D", "isTLsrequired": "false", "_chatIxnAPI_DISCONNECT_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/disconnect", "userId": "015D508717FE0027", "chatServiceMessage": "Chat service is available", "_chatIxnAPI_STOP_TYPING_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/stopTyping", "_chatIxnAPI_START_TYPING_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/startTyping", "transcriptToShow": [{"Message.Text", "BuzzB", "hello agent", "230", "CLIENT"}], "_chatIxnAPI_SEND_URL": "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/send", "chatServerHost": "demosrv.genesyslab.com"}, "tag": "service.chat.refresh.0da34a9d-0c5e-46a6-953c-6cc1aa82bfconnect"}]}
```

Disconnect chat session:

POST http://localhost:8080/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/disconnect
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

_verbose=true

Response:

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

```
{
  "chatIxnState" : "DISCONNECTED",
  "transcriptPosition" : "6",
  "chatServiceMessage" : "Chat was finished",
  "chatSessionId" : "000BWa7RECRK001D",
  "userId" : "015D508717FE0027",
  "secureKey" : "29a6758abacdd33e",
  "checkChatServiceLoadBalancerPath" : "/WebAPI812/SimpleSamples812/ChatHA/
```

```
ChatLBServerInfo.jsp?chatServerLoadBalancerAlias=349",
  "chatServerLoadBalancerAlias" : "349",
  "chatServerHost" : "demosrv.genesyslab.com",
  "chatWebApiPort" : "4853",
  "isTLSrequired" : "false",
  "clientTimeZoneOffset" : "-420",
  "_chatIxnAPI_SEND_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/send",
  "_chatIxnAPI_REFRESH_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/refresh",
  "_chatIxnAPI_START_TYPING_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/startTyping",
  "_chatIxnAPI_STOP_TYPING_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/stopTyping",
  "_chatIxnAPI_DISCONNECT_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/disconnect",
  "_chatIxnAPI_REFRESH_FROM_START_URL" : "/genesys/1/service/0da34a9d-0c5e-46a6-953c-6cc1aa82bf7f/ixn/chat/refresh?transcriptPosition=1"
}
```

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 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

Parameter Name	Mandatory	Description
		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 8.1.100.28. Nickname to be displayed in the chat conversation.
notify_by	false	If using a CometD connection for the asynchronous receiving of chat messages, then supply this parameter with the value "comet".

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
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 08:30:10 GMT
Content-Type: application/json
Content-Length: 119

{
  "chatIxnState" : "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

```
{  
  "chatIxnState" : "DISCONNECTED",  
  "transcriptPosition" : "9",  
  "chatServiceMessage" : "Chat was finished"  
}
```

Chat Interaction APIs

Start Chat

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

Note: In release **8.1.100.28**, the ixn/chat parameters are now optional, and added parameter

userDisplayName.

Operation

Method	POST		
URL	/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 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:			
• _verbose - This will allow the application to get all the detail attributes associated with the chat session in the corresponding response. • notify_by - If specified should be "comet". • firstName - User's first name. Optional. • lastName - User's last name. Optional. • email - User's email address. Optional. • subject - Subject of the chat conversation. • userDisplayName - Nickname displayed in the chat conversation. Optional.			

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.
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;ject=test
```

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
Server: Jetty(7.6.0.v20120127)
```

```
{
  "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"
}
```

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

Method	POST		
URL	/genesys/1/service/{ <i>service_id</i> }/ixn/chat/refresh		
Parameter	Type	Mandatory	Description

Method	POST		
URI Parameters			
{service_id}	string	yes	The identifier of the service that the chat session is associated with.
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:			
- transcriptPosition – This optional property indicates the current position in the chat session that the current user is in. - 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
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).
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/<code>x-www-form-urlencoded</code>
message=aaa
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 06:50:49 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
```

Content-Type: application/json

Content-Length: 1495

Server: Jetty(7.6.0.v20120127)

```
{
  "chatIxnState": "TRANSCRIPT",
  "chatSessionId": "000C2a7VVQRB001W",
  "transcriptPosition": "5",
  "chatServiceMessage": "Chat service is available",
  "transcriptToShow": [
    [
      "Notice.Joined",
      "ksippo",
      "has joined the session",
      "18",
      "AGENT"
    ],
    [
      "Notice.TypingStarted",
      "ksippo",
      "is typing",
      "21",
      "AGENT"
    ],
    [
      "Message.Text",
      "ksippo",
      "hello customer",
      "28",
      "AGENT"
    ],
    [
      "Message.Text",
      "VasyaP",
      "aaa",
      "172",
      "CLIENT"
    ]
  ],
  "startedAt": "2012-06-10T06:47:58Z",
  "userId": "015E4FD4431D0039",
  "secureKey": "4ba8838b7cf9afc4",
  "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/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat/send",
  "_chatIxnAPI_REFRESH_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat/refresh",
  "_chatIxnAPI_START_TYPING_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat/startTyping",
  "_chatIxnAPI_STOP_TYPING_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat/stopTyping",
  "_chatIxnAPI_DISCONNECT_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/ixn/chat/disconnect",
  "_chatIxnAPI_REFRESH_FROM_START_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001H/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

Method	POST		
URL	/genesys/1/service/{ <i>service_id</i> }/ixn/chat/startTyping		
Parameter	Type	Mandatory	Description
URI Parameters			
{ <i>service_id</i> }	string	yes	The identifier of the service that the chat session is suppose to be associated with.
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:			
_verbose - This 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
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 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/EKUJPKAQ197CFA6SJQKTJ03DBG00001J/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
Server: Jetty(7.6.0.v20120127)
{
  "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

Method	POST		
URL	/genesys/1/service/{service_id}/ixn/chat/stopTyping		
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 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: _verbose - This 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
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 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/EKUJPKAQ197CFA6SJQKTJ03DBG00001J/ixn/chat/stopTyping HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/<code>x-www-form-urlencoded</code>
```

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
Server: Jetty(7.6.0.v20120127)
{
  "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

Method	POST		
URL	/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.

Method	POST
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.	

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 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/EKUJPKAQ197CFA6SJQKTJ03DBG00001J/ixn/chat/
disconnect HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/<code>x-www-form-urlencoded</code>
```

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
Server: Jetty(7.6.0.v20120127)
{
  "chatIxnState" : "DISCONNECTED",
  "transcriptPosition" : "9",
  "chatServiceMessage" : "Chat was finished"
}
```

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
Server: Jetty(7.6.0.v20120127)
```

```
{
  "_chatIxnAPI-CREATE-URL": "/genesys/1/service/81f0ef4e-99dd-43ea-8366-8d27a2cbd605/ixn/chat",
  "_id": "81f0ef4e-99dd-43ea-8366-8d27a2cbd605"
}
```

Advanced Chat Service API

Create advanced chat service

This API allows the application to create advanced chat service session and then initiate chat interaction when agent is available or other enterprise side conditions are met.

Note: ORS 'request-chat-poll' sample session is available.

Operation

Method	POST		
URL	genesys/1/service/request-chat-poll		
Parameter	Type	Mandatory	Description
URI Parameters			
'request-chat-poll'	String	yes	Name of the preconfigured advanced 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. • <code>...</code> - 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-poll 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 08:01:58 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.6.0.v20120127)
```

```
{
  "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001K",
  "_session_comand_STATUS_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001K/status"
}
```

Execute 'status' request

This API allows the application to send data to the advanced service session and receive reply back immediately.

Operation

Method	POST		
URL	genesys/1/service/{ sessionid }/status		
Parameter	Type	Mandatory	Description
URI Parameters			
sessionid	String	yes	The identifier of the

Method	POST		
			service that the chat session is suppose to be associated with.
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 could 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/EKUJPKAQ197CFA6SJQKTJ03DBG00001K/status 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 08:06:44 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 364
Server: Jetty(7.6.0.v20120127)
{
  "message": {
    "_id": "EKUJPKAQ197CFA6SJQKTJ03DBG00001K",
    "_status": "available",
    "_dialog": "agent_available.html",
```

```

    "_chatIxnAPI_CREATE_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001K/ixn/
chat",
    "_session_comand_STATUS_URL": "/genesys/1/service/EKUJPKAQ197CFA6SJQKTJ03DBG00001K/
status"
  },
  "tag": "a2c.advanced.service.agentavailable.EKUJPKAQ197CFA6SJQKTJ03DBG00001K"
}

```

Execute 'internal-poke' request

Note: The internal-poke feature is available in GMS 8.1.100.28 or higher.

This API allows the agent desktop to send poke request to the advanced service session. When the session receives the internal-poke request the poke message is sent to the application as a push notification. Additional config options should be configured for the advanced GMS service to enable this API.

This API is available when,

- Poke related config options are configured and
- Service was created with device_os and device_notification_id parameters to enable native push notification are specified and
- Chat interaction is successfully routed to the agent by the advanced service (implies ixn/chat request was already executed)

'poke_url' is available as attached data in the chat interaction and should be used to invoke this request. Firewall can be configured to allow /internal-* requests to GMS from agent desktop.

Operation

Method	POST		
URL	genesys/1/service/{ sessionid }/internal-poke		
Parameter	Type	Mandatory	Description
URI Parameters			
sessionid	String	yes	The identifier of the service that the chat session is suppose to be associated with.
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:			
• poke_message - This message is sent as a push notification to the application. If not specified, default poke message configure in the service options is used • ... - any other business data attributes could be passed			

Response

HTTP code	200
HTTP message	OK
Body	OK
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/EKUJPKAQ197CFA6SJQKTJ03DBG00001K/internal-poke
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 08:06:44 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Content-Length: 364
Server: Jetty(7.6.0.v20120127)
```

OK