

Chat REST API Developer Guide

Version 56.0, Spring '23



'23

CONTENTS

Chapter 1: Overview	1
Chapter 2: Getting Started with the Chat REST API	2
Chapter 3: Request Headers	5
Chapter 4: Your Message Long Polling Loop	6
Chapter 5: Using Estimated Wait Time Instead of Queue Position for a Chat Session (Beta)	7
Chapter 6: Chat REST API Resources	8
Create a Chat Session	8
SessionId	8
Create a Chat Visitor Session	9
ChasitorInit	9
ReconnectSession	10
ChasitorResyncState	12
Monitor Chat Activity	13
ChasitorNotTyping	13
ChasitorSneakPeek	14
ChasitorTyping	14
ChatEnd	15
ChatMessage	16
CustomEvent	16
Messages	17
MultiNoun	23
Customize the Chat Visitors' Experience	24
Settings	24
Availability	25
Breadcrumb	26
VisitorId	27
Chapter 7: Request Bodies for Chat REST API	28
Chapter 8: Response Bodies for Chat REST API	33
Chapter 9: Chat REST API Data Types	46
Chapter 10: Status Codes and Error Responses	54
Index	55

CHAPTER 1 Overview

Take Chat to a native mobile app or a custom client using the Chat REST API.

 **Note:** This guide explains how to use the Chat REST API. If you're not already using Chat, check out [Messaging for In-App and Web](#), a newer chat solution for websites and mobile apps.

You don't have to rely on Visualforce to develop customized chat windows. With the REST resources in this guide, you can extend the functionality of chat windows beyond simple HTML and JavaScript environments that merge seamlessly into your company's own applications.

 **Note:** Chat REST API doesn't support building custom clients that work with Einstein Bots. See [Einstein Bots API and SDK](#).

SEE ALSO:

[Embedded Service SDK for iOS Developer Guide](#)

[Embedded Service SDK for Android Developer Guide](#)

EDITIONS


Available in: Salesforce Classic and Lightning Experience

Available in: **Performance** Editions and in **Developer** Edition orgs that were created after June 14, 2012

Available in: **Essentials**, **Unlimited**, and **Enterprise** Editions with Service Cloud or Sales Cloud

CHAPTER 2 Getting Started with the Chat REST API

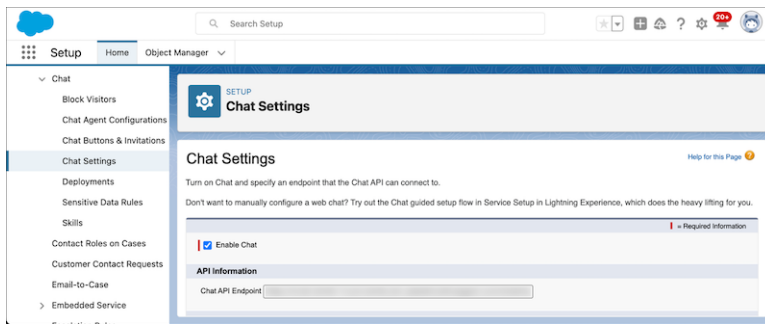
Learn how to start, confirm, and end a Chat session with the Chat REST API.

 **Note:** This guide explains how to use the Chat REST API. If you're not already using Chat, check out [Messaging for In-App and Web](#), a newer chat solution for websites and mobile apps.

Retrieve Your Chat API Endpoint

Your Chat API endpoint is a unique URL that lets you access data from your organization's Chat sessions. To find your organization's Chat API endpoint:

1. From Setup, in the Quick Find box, enter *Chat Settings*, and then select **Chat Settings**.
2. Retrieve the hostname from the Chat API Endpoint. The hostname is the URL without `/chat/rest/` at the end, for example, `"https://yourChatApiEndpoint.com"`. Substitute *hostname* in the Chat API endpoints with this URL.



Start a Chat Session

To start a Chat session, send a [SessionId](#) request. Replace *hostname* with the URL that you retrieved from Chat API Endpoint.

```
GET https://hostname/chat/rest/System/SessionId/
```

Use these [Request Headers](#).

- X-LIVEAGENT-AFFINITY
- X-LIVEAGENT-API-VERSION

Confirm the Chat Session Started

A [ChatRequestSuccess](#) response tells you that the Chat session started.

```
{
  queuePosition: 1,
  estimatedWaitTime: 120,
  geoLocation: {
    countryCode: "US",
```

```
        countryName: "United States of America",
        region: "CA",
        city: "San Francisco",
        organization: Salesforce,
        latitude: 37.793880,
        longitude: -122.395114
    },
    url: "http://yoursite",
    oref: "http://www.google.com?q=yoursite",
    postChatUrl: "http://yoursite/postchat",
    customDetails: [
        {
            label: "E-mail Address",
            value: "jon@example.com",
            transcriptFields: [
                "c__EmailAddress"
            ],
            displayToAgent: true
        }
    ],
    visitorId: "acd47048-bd80-476e-aa33-741bd5cb05d3"
}
```

Then wait for a [ChatEstablished](#) on page 21 response. That response tells you that an agent accepted the chat session.

```
{
    name: "Andy L.",
    userId: "f1dda237-57f8-4816-b8e8-59775f1e44c8",
    sneakPeekEnabled: true
}
```

Now you're ready to send, for example, [Messages](#) requests. Before you send further requests, wait until you receive the [ChatRequestSuccess](#) and [ChatEstablished](#) responses, otherwise the API throws a `NullPointerException`, and you receive a 500 error.

End the Chat Session

The Chat session ends when you send a [ChatEnd](#) request or send a `DELETE SessionId` request. In both request types, `X-LIVEAGENT-SESSION-KEY` is the unique ID of the Chat session that you want to end.

Here's the [ChatEnd](#) request.

```
https://hostname/chat/rest/Chasitor/ChatEnd
```

Use these [Request Headers](#).

- X-LIVEAGENT-AFFINITY
- X-LIVEAGENT-API-VERSION
- X-LIVEAGENT-SESSION-KEY
- X-LIVEAGENT-SEQUENCE

Here's the [SessionId](#) request.

```
DELETE https://hostname/chat/rest/System/SessionId/X-LIVEAGENT-SESSION-KEY
```

Getting Started with the Chat REST API

Use these [Request Headers](#).


- X-LIVEAGENT-AFFINITY
- X-LIVEAGENT-API-VERSION

CHAPTER 3 Request Headers

Each Chat REST API resource requires one or more headers to make a request.

Not all resources require all of the available request headers. Each resource indicates which headers are required to make a request.

The following headers are available:

Header Syntax	Description
X-LIVEAGENT-API-VERSION	The Salesforce API version for the request.
X-LIVEAGENT-AFFINITY	The system-generated ID used to identify the Chat session on the Chat servers. This affinity token is included in the response body of the SessionId request .
X-LIVEAGENT-SESSION-KEY	The unique ID associated with your Chat session.  Note: Your session key shouldn't be shared or sent over insecure channels, as it allows access to potentially sensitive chat information.
X-LIVEAGENT-SEQUENCE	The sequence of messages you have sent to the Chat server to help the Chat server avoid processing duplicate messages. This number should be increased by one with every new request.

CHAPTER 4 Your Message Long Polling Loop

Message long polling notifies you of events that occur on the Chat server for your Chat session.

When you start a request, all pending messages will be immediately delivered to your session. If there are no pending messages, the connection to the server will remain open. The connection will return messages continuously as they are received on the server.

If your session expires, you will receive a 200 ("OK") response code and a resource that contains an array of the remaining messages. If no messages were received, you will receive a 204 ("No Content") response code.

When you receive a 200 ("OK") or 204 ("No Content") response code, immediately perform another `Messages` request to continue to retrieve messages that are registered on the Chat server.



Warning: If you don't make another `Messages` request to continue the messaging loop, your session will end after a system timeout on the Chat server.

If you don't receive a response within the number of seconds indicated by the `clientPollTimeout` property in your `SessionId` request, your network connection to the server is likely experiencing an error, so you should terminate the request.

To initiate a long polling loop, perform a `Messages` request.

SEE ALSO:


[Messages](#)

[SessionId](#)

[Status Codes and Error Responses](#)

CHAPTER 5 Using Estimated Wait Time Instead of Queue Position for a Chat Session (Beta)

By default, the Chat API returns queue position information that you can relay to customers. However, you can also receive the estimated wait time in addition to the queue position. Sometimes, the estimated wait time more effectively conveys the right information to customers than a queue position number. This feature is available in API version 47.0 and later.

 **Note:** As a beta feature, Estimated Wait Time is a preview and isn't part of the "Services" under your main subscription agreement with Salesforce. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Salesforce doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. This feature is for evaluation purposes only, not for production use. It's offered as is and isn't supported, and Salesforce has no liability for any harm or damage arising out of or in connection with it. All restrictions, Salesforce reservation of rights, obligations concerning the Services, and terms for related Non-Salesforce Applications and Content apply equally to your use of this feature.

The following algorithm is used to calculate the wait time:

$$A = (0.9 * A') + (0.1 * W)$$

Where:

- A' is the previous value for A . If no previous value exists, this value is W .
- W is the wait time of the chat that has most recently been accepted.

The value returned is the value of A minus the time already spent waiting.

Additional algorithm details:

- This value is calculated separately for each chat button.
- A is recalculated each time a chat is accepted.
- 0 is returned if the result is less than 0.
- -1 is returned when the value cannot be calculated.

To use this feature, specify that you want the estimated wait time from the [Settings](#) request (by setting `Settings.needEstimatedWaitTime` to 1) and the [Availability](#) request (by setting `Availability.needEstimatedWaitTime` to 1). When this value is set to 1, the response includes the estimated wait time for each button ID requested.

If `receiveQueueUpdates` is set when initializing the session, [ChatRequestSuccess](#) and [QueueUpdate](#) will both contain the estimated wait time (in seconds) in their responses.

CHAPTER 6 Chat REST API Resources

To perform a POST or GET request, create and send an HTTP request with the appropriate parameters or request body.

The Chat REST API requests let you begin new chat sessions between agents and chat visitors and monitor the chat activity that occurs.

IN THIS SECTION:

[Create a Chat Session](#)

To create a new Chat session, you must call the `SessionId` request.

[Create a Chat Visitor Session](#)

To create or reestablish a chat visitor session using the Chat REST API, you must make certain requests.

[Monitor Chat Activity](#)

Chat requests indicate when certain activities occurred during a chat session.

[Customize the Chat Visitors' Experience](#)

With the Chat visitor REST API resources, you can establish your chat visitors' experience with Chat in custom mobile applications.

Create a Chat Session

To create a new Chat session, you must call the `SessionId` request.

IN THIS SECTION:

[SessionId](#)

Establishes a new Chat session. The `SessionId` request is required as the first request to create every Chat session.

SessionId

Establishes a new Chat session. The `SessionId` request is required as the first request to create every Chat session.

Syntax

URI

`https://hostname/chat/rest/System/SessionId/X-LIVEAGENT-SESSION-KEY` (where you only need `/X-LIVEAGENT-SESSION-KEY` if you use the DELETE method)

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

GET—Creates a session. Don't pass a session key to the URL.

DELETE—Deletes the session. Pass a session key to the URL.

Request headers

X-LIVEAGENT-AFFINITY
X-LIVEAGENT-API-VERSION

Request body

None

Request parameters

None

Response Body

[SessionId](#) response

SEE ALSO:

[Your Message Long Polling Loop](#)

Create a Chat Visitor Session

To create or reestablish a chat visitor session using the Chat REST API, you must make certain requests.

IN THIS SECTION:

[ChasitorInit](#)

Initiates a new chat visitor session. The `ChasitorInit` request is always required as the first POST request in a new chat session.

[ReconnectSession](#)

Reconnect a customer's chat session on a new server if the session is interrupted and the original server is unavailable.

[ChasitorResyncState](#)

Reestablishes the chat visitor's state, including the details of the chat, after a `ReconnectSession` request is completed.

ChasitorInit

Initiates a new chat visitor session. The `ChasitorInit` request is always required as the first POST request in a new chat session.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChasitorInit`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION
X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

[ChasitorInit](#) request

Response body

None

ReconnectSession

Reconnect a customer's chat session on a new server if the session is interrupted and the original server is unavailable.

This request should only be made if you receive a 503 response status code, indicating that the affinity token has changed for your Chat session. When you receive a 503 response status code, you must cancel any existing inbound or outbound requests.

The data in outbound requests will be temporarily stored and resent once the session is reestablished. Upon receiving the response for the `ReconnectSession` request, you can start polling for messages.

The first response will be a `ChasitorSessionData` message containing the data from the previous session that will be restored once the session is reestablished. After receiving that message, you can proceed to send the existing messages that were canceled upon receiving the 503 response status code.

Syntax

URI

`https://hostname/chat/rest/System/ReconnectSession`

Available since release

This resource is available in API versions 37.0 and later.

Formats

JSON

HTTP methods

GET

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

Request parameters

Name	Type	Description
<code>ReconnectSession.offset</code>	Number	The event offset from the most recent Messages request that your client received.

Query parameters

None

Request body

None

Response body

ReconnectSession



Example: Your REST client can get a 503 Invalid Affinity Token response, for example, to a long poll request (/chat/rest/System/Messages).

No matter which kind of request gets the 503 response, you must send a /chat/rest/System/ReconnectSession request to finish the handover process.

```
Method: GET
URL:
<!-- Change the live agent pool to the correct one for your org. -->
https://4.la2-stmfb1-0-prd.stmfb.stm.salesforceliveagent.com/chat/rest/System/
ReconnectSession?ReconnectSession.offset=54647226
Headers:
X-LIVEAGENT-AFFINITY:
null [the literal string "null"]
X-LIVEAGENT-API-VERSION:
42
X-LIVEAGENT-SESSION-KEY:
4eb90106-3410-4dd0-8f04-c4facf90a929!1519169434766!IbjEwmJkIIyqalZS3YBU8WO3nSM=
```

The `ReconnectSession.offset` query parameter has to be set to the “offset” parameter of the most recent long poll response that actually contained messages. Empty long poll responses don’t come with an “offset”.

The response to this `ReconnectSession` request looks like this:

```
{
  "messages": [
    {
      "type": "ReconnectSession",
      "message": {
        "resetSequence": true, [This may be undefined]
        "affinityToken": "efaelfa0"
      }
    }
  ]
}
```

The `resetSequence` is always set to true. Therefore, reset the sequence number of the next request and store the value in `affinityToken` to use in the `X-LIVEAGENT-AFFINITY` header for all future requests. Once another handover process occurs the `resetSequence` is updated again.

Testing

To test that your client handles this process correctly, check that your client sends a `ReconnectSession` request when it receives a 503 response from the server. You can use a proxy tool of your choice to mimic the 503 response or you can wait until the Salesforce server

sends one. When the proxy tool sends a 503 response, you can test that your client sends the `ReconnectSession` request and reconnects the chat session to a new server, as expected. To get an actual 503 response from the server, you can leave a session connected and wait until the server is restarted during scheduled maintenance. Then see if the chat session reconnects to a new server. However, the maintenance schedule is not announced in advance.

SEE ALSO:

[Status Codes and Error Responses](#)

[ChasitorSessionData](#)

[ChasitorResyncState](#)

[Status Codes and Error Responses](#)

ChasitorResyncState

Reestablishes the chat visitor's state, including the details of the chat, after a `ReconnectSession` request is completed.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChasitorResyncState`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

`X-LIVEAGENT-API-VERSION`

`X-LIVEAGENT-AFFINITY`

`X-LIVEAGENT-SESSION-KEY`

Request parameters

None

Query parameters

None

Request body

[ChasitorResyncState](#) request

Response body

None

SEE ALSO:

[ReconnectSession](#)

Monitor Chat Activity

Chat requests indicate when certain activities occurred during a chat session.

IN THIS SECTION:

[ChasitorNotTyping](#)

Indicates that the chat visitor is not typing in the chat window.

[ChasitorSneakPeek](#)

Provides a chat visitor's message that was viewable through Sneak Peek.

[ChasitorTyping](#)

Indicates that a chat visitor is typing a message in the chat window.

[ChatEnd](#)

Indicates that a chat visitor has ended the chat.

[ChatMessage](#)

Returns the body of the chat message sent by the chat visitor.

[CustomEvent](#)

Indicates a custom event was sent from the chat visitor during the chat.

[Messages](#)

Returns all messages that were sent between agents and chat visitors during a chat session.

[MultiNoun](#)

Batches multiple POST requests together if you're sending multiple messages at the same time.

ChasitorNotTyping

Indicates that the chat visitor is not typing in the chat window.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChasitorNotTyping`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

None

Response body

None

ChasitorSneakPeek

Provides a chat visitor's message that was viewable through Sneak Peek.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChasitorSneakPeek`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body[ChasitorSneakPeek request](#)**Response body**

None

ChasitorTyping

Indicates that a chat visitor is typing a message in the chat window.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChasitorTyping`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

None

Response body

None

ChatEnd

Indicates that a chat visitor has ended the chat.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChatEnd`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

ChatEndReason—Include the `ChatEndReason` parameter in the request body of your request to specify the reason that the chat ended. This parameter is required. For example: `{reason: "client"}`.

Response properties

attachedRecords—Includes attached record IDs. You can use this Visualforce component to display the attached record IDs in the post-chat page: `<apex:outputText value="{!$CurrentPage.parameters.attachedRecords}"/>
`.

ChatMessage

Returns the body of the chat message sent by the chat visitor.

Syntax

URI

`https://hostname/chat/rest/Chasitor/ChatMessage`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers`X-LIVEAGENT-API-VERSION``X-LIVEAGENT-AFFINITY``X-LIVEAGENT-SESSION-KEY``X-LIVEAGENT-SEQUENCE`**Request parameters**

None

Query parameters

None

Request body[ChatMessage request](#)**Response body**

None

CustomEvent

Indicates a custom event was sent from the chat visitor during the chat.

Syntax

URI

`https://hostname/chat/rest/Chasitor/CustomEvent`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

[CustomEvent](#) request

Response body

None

Messages

Returns all messages that were sent between agents and chat visitors during a chat session.

For a complete list of responses for the `Messages` resource, see [Chat REST API Messages Response Objects](#).

Syntax

URI

`https://hostname/chat/rest/System/Messages`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

GET

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

Request parameters

None

Query parameters

ack—The **ack** query parameter is a sequencing mechanism that allows you to poll for messages on the Live Agent server. The first time you make the **Messages** request, the **ack** parameter is set to **-1**. To guarantee that you receive the messages from the server in the correct order, update the **ack** value in the next request with the **sequence** value from the preceding response. You receive a **sequence** value only if the response code is **200**, which is the response if there are new messages. If the response code is **204**, there are no messages and the client doesn't provide a **sequence** value. In this case, run the **Messages** request with the same **ack** value as the previous request.

Request body

None

Response body[Messages response](#)

Troubleshooting

If your request doesn't receive an HTTP response and fails, retry the request. If you don't retry the request before the chat session times out, the session expires. The timeout value that determines how long you have to attempt to send requests before the server expires the session is configured in **Chat Deployment** in **Salesforce Setup**.

IN THIS SECTION:[Chat REST API Messages Response Objects](#)

The **Messages** request returns an array of objects that represent all the events that occurred during an agent's chat with a chat customer.

SEE ALSO:[Your Message Long Polling Loop](#)

Chat REST API **Messages** Response Objects

The **Messages** request returns an array of objects that represent all the events that occurred during an agent's chat with a chat customer.

This request can return several subtypes with unique response bodies, depending on the events that occurred within the chat.

Here is an example of the structure of a **Messages** response array:

```
{
  "messages": {
    "type": "array",
    "description": "The messages sent over the course of a chat.",
    "items": {
      "name": "Message",
      "type": "object",
      "properties": {
        "type": {
          "type": "string",
          "description": "The type of message that was received.",
          "required": true,
```

```

        "version":29.0
    },
    "message": {
        "type":"object",
        "description":"A placeholder object for the message that was received.
        Can return any of the responses available for the Messages request.",
        "required":true,
        "version":29.0
    }
},
"required":true,
"version":29.0
},
"sequence":{
    "type":"integer",
    "description":"The sequence of the message as it was received over
    the course of a chat.",
    "required":true,
    "version":29.0
}
}

```

IN THIS SECTION:

[AgentDisconnect](#)

Indicates that the agent has been disconnected from the chat.

[AgentNotTyping](#)

Indicates that the agent is not typing a message to the chat visitor.

[AgentTyping](#)

Indicates that the agent is typing a message to the chat visitor.

[ChasitorSessionData](#)

Returns the current chat session data for the chat visitor. This request is used to restore the session data for a chat visitor's chat session after a `ReconnectSessionrequest` is sent.

[ChatEnded](#)

Indicates that the chat has ended.

[ChatEstablished](#)

Indicates that an agent has accepted a chat request and is engaged in a chat with a visitor.

[ChatMessage](#)

Indicates a new chat message has been sent from an agent to a chat visitor.

[ChatRequestFail](#)

Indicates that the chat request was not successful.

[ChatRequestSuccess](#)

Indicates that the chat request was successful and routed to available agents.

[ChatTransferred](#)

Indicates the chat was transferred from one agent to another.

[CustomEvent](#)

Indicates a custom event was sent from an agent to a chat visitor during a chat.

[NewVisitorBreadcrumb](#)

Indicates the URL of the Web page the chat visitor is currently viewing.

[QueueUpdate](#)

Indicates the new position of the chat visitor in the chat queue when the visitor's position in the queue changes.

AgentDisconnect

Indicates that the agent has been disconnected from the chat.



Note: Though the agent has been disconnected from the chat, the chat session is still active on the server. A new agent may accept the chat request and continue the chat.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

None

Response properties

None

AgentNotTyping

Indicates that the agent is not typing a message to the chat visitor.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

None

Response properties

None

AgentTyping

Indicates that the agent is typing a message to the chat visitor.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

None

Response properties

None

ChasitorSessionData

Returns the current chat session data for the chat visitor. This request is used to restore the session data for a chat visitor's chat session after a `ReconnectSession` request is sent.

The `ChasitorSessionData` request is the first message sent after a `ReconnectSession` request is delivered.



Note: No messages should be sent after a 503 status code is encountered until this message is processed.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

`ChasitorSessionData` response

SEE ALSO:

[ReconnectSession](#)

[Status Codes and Error Responses](#)

[Status Codes and Error Responses](#)

ChatEnded

Indicates that the chat has ended.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Request Body

`ChatEndReason`—Include the `ChatEndReason` parameter in the request body of your request to specify the reason that the chat ended. For example: `{reason: "client"}`.

Response body

None

Response properties

None

ChatEstablished

Indicates that an agent has accepted a chat request and is engaged in a chat with a visitor.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

[ChatEstablished](#) response

SEE ALSO:

[ChatRequestSuccess](#)

ChatMessage

Indicates a new chat message has been sent from an agent to a chat visitor.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

[ChatMessage](#) response

ChatRequestFail

Indicates that the chat request was not successful.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

[ChatRequestFail](#) response

ChatRequestSuccess

Indicates that the chat request was successful and routed to available agents.



Note: The `ChatRequestSuccess` response only indicates that a request has been routed to available agents. The chat hasn't been accepted until the `ChatEstablished` response is received.

Syntax**Available since release**

This resource is available in API versions 29.0 and later.

Response body

[ChatRequestSuccess](#) response

SEE ALSO:

[ChatEstablished](#)

ChatTransferred

Indicates the chat was transferred from one agent to another.

Syntax

Available since release

This resource is available in API versions 29.0 and later.

Response body

[ChatTransferred](#) response

CustomEvent

Indicates a custom event was sent from an agent to a chat visitor during a chat.

Syntax

Available since release

This resource is available in API versions 29.0 and later.

Response body

[CustomEvent](#) response

NewVisitorBreadcrumb

Indicates the URL of the Web page the chat visitor is currently viewing.

Syntax

Available since release

This resource is available in API versions 29.0 and later.

Response body

[NewVisitorBreadcrumb](#) response

QueueUpdate

Indicates the new position of the chat visitor in the chat queue when the visitor's position in the queue changes.

Syntax

Available since release

This resource is available in API versions 29.0 and later.

Response body

[QueueUpdate](#) response

MultiNoun

Batches multiple POST requests together if you're sending multiple messages at the same time.

Syntax

URI

`https://hostname/chat/rest/System/MultiNoun`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

X-LIVEAGENT-AFFINITY

X-LIVEAGENT-SESSION-KEY

X-LIVEAGENT-SEQUENCE

Request parameters

None

Query parameters

None

Request body

[MultiNoun request](#)

Response body

None

Customize the Chat Visitors' Experience

With the Chat visitor REST API resources, you can establish your chat visitors' experience with Chat in custom mobile applications.

IN THIS SECTION:[Settings](#)

Retrieves all settings information about the Chat deployment that's associated with your chat session. The `Settings` request is required as the first request to establish a chat visitor's session.

[Availability](#)

Indicates whether a chat button is available to receive new chat requests.

[Breadcrumb](#)

Sets a breadcrumb value to the URL of the Web page that the chat visitor is viewing as the visitor chats with an agent. The agent can then see the value of the breadcrumb to determine the page the chat visitor is viewing.

[VisitorId](#)

Generates a unique ID to track a chat visitor when they initiate a chat request and tracks the visitor's activities as the visitor navigates from one Web page to another.

Settings

Retrieves all settings information about the Chat deployment that's associated with your chat session. The `Settings` request is required as the first request to establish a chat visitor's session.

Syntax

URI

`https://hostname/chat/rest/Visitor/Settings`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

GET

Request headers

`X-LIVEAGENT-API-VERSION`

Request parameters

None

Query parameters

org_id

The ID of the Salesforce organization that's associated with the Live Agent deployment.

deployment_id

The ID of the Chat deployment that the chat request was initiated from.

Settings.buttonIds

An array of chat button IDs for which to retrieve settings information.

Settings.needEstimatedWaitTime

Indicates whether the `estimatedWaitTime` property should be filled. Specify a value of `1` to request the estimated wait time.

Settings.updateBreadcrumb

Indicates whether to update the chat visitor's location with the URL of the Web page that the visitor is viewing.

Request body

None

Response body

[Settings response](#) on page 41

Availability

Indicates whether a chat button is available to receive new chat requests.

Syntax

URI

`https://hostname/chat/rest/Visitor/Availability`

Available since release

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

GET

Request headers

X-LIVEAGENT-API-VERSION

Request parameters

None

Query parameters**org_id**

The ID of the Salesforce organization that's associated with the Live Agent deployment.

deployment_id

The 15-digit ID of the Chat deployment that the chat request was initiated from.

Availability.ids

An array of object IDs for which to verify availability.

Availability.needEstimatedWaitTimeIndicates whether the `estimatedWaitTime` property should be filled. Specify a value of 1 to request the estimated wait time.**Request body**

None

Response body[Availability response](#)

Breadcrumb

Sets a breadcrumb value to the URL of the Web page that the chat visitor is viewing as the visitor chats with an agent. The agent can then see the value of the breadcrumb to determine the page the chat visitor is viewing.

Syntax

URI`https://hostname/chat/rest/Visitor/Breadcrumb`**Available since release**

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

POST

Request headers

X-LIVEAGENT-API-VERSION

Request parameters

None

Query parameters

None

Request body[Breadcrumb request](#)**Response body**

None

VisitorId

Generates a unique ID to track a chat visitor when they initiate a chat request and tracks the visitor's activities as the visitor navigates from one Web page to another.

Syntax

URI`https://hostname/chat/rest/Visitor/VisitorId`**Available since release**

This resource is available in API versions 29.0 and later.

Formats

JSON

HTTP methods

GET

Request headers`X-LIVEAGENT-API-VERSION`**Request parameters**

None

Query parameters**org_id**

The Salesforce organization ID

deployment_id

The ID of the Chat deployment that the chat request was initiated from

Request body

None

Response body[VisitorId response](#)

CHAPTER 7 Request Bodies for Chat REST API

To perform a POST or GET request, pass query parameters or create a request body that's formatted in JSON. Request bodies can contain one or more other request bodies that are nested inside. Each request body can contain unique request properties.

Breadcrumb

Request properties

Property Name	Type	Description	Available Versions
location	String	The URL of the web page that the chat visitor is viewing.	29.0

Request body

```
"location":{
  "type":"string",
  "description":"The current location or URL of the visitor",
  "required":true,
  "version":29.0
}
```

ChasitorInit

Request properties

Property Name	Type	Description	Available Versions
organizationId	String	The chat visitor's Salesforce organization ID.	29.0
deploymentId	String	The ID of the deployment from which the chat originated.	29.0
buttonId	String	The ID of the button from which the chat originated.	29.0
agentId	String	The ID of the agent of a direct-to-agent chat request. For normal chat requests, leave this field empty.	29.0
doFallback	Boolean	Specifies the fallback mode if agentId is present. If the value is false, it attempts to route the chat session back to that specific agent. If	29.0

Property Name	Type	Description	Available Versions
		the value is <code>true</code> , it attempts to route the chat session back to the specific agent first but, if the agent is unavailable, it attempts to route to the button next.	
<code>sessionId</code>	String	The chat visitor's Chat session ID.	29.0
<code>userAgent</code>	String	The chat visitor's browser user agent.	29.0
<code>language</code>	String	The chat visitor's spoken language.	29.0
<code>screenResolution</code>	String	The resolution of the chat visitor's computer screen.	29.0
<code>visitorName</code>	String	The chat visitor's custom name.	29.0
<code>prechatDetails</code>	Array of CustomDetail objects	The pre-chat information that was provided by the chat visitor.	29.0
<code>prechatEntities</code>	Array of Entity objects	The records created, searched for, or both depending on what EntityFieldsMaps on page 50 has enabled.	29.0
<code>buttonOverrides</code>	Array of Strings	<p>The button override is an ordered list of routing targets and overrides the <code>buttonId</code>, <code>agentId</code>, and <code>doFallback</code> modes. The possible options are:</p> <ul style="list-style-type: none"> • <code>buttonId</code>—Normal routing • <code>agentId</code>—Direct-to-agent routing with no fallback • <code>agentId_buttonId</code>—Direct-to-agent routing with fallback to the button <p>You can list one or more of these options, where the order specifies the routing target order. The second or third target is attempted only if the previous one fails.</p>	29.0
<code>receiveQueueUpdates</code>	Boolean	Indicates whether the chat visitor receives queue position updates (<code>true</code>) or not (<code>false</code>).	29.0
<code>isPost</code>	Boolean	Indicates whether the chat request was made properly through a POST request (<code>true</code>) or not (<code>false</code>).	29.0

Request body

```
{
  organizationId: "00DD00000000JvXs",
  deploymentId: "572D000000000J6",
  buttonId: "573D000000000OC",
  agentId: "005B00000000F3b2",
  doFallback: true,
  sessionId: "5503f854-0203-4324-8ed5-f793a367426f",
  userAgent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/28.0.1500.95 Safari/537.36",
  language: "en-US",
  screenResolution: "2560x1440",
  visitorName: "Jon A.",
  prechatDetails: [
    {
      label: "E-mail Address",
      value: "jon@example.com",
      entityFieldMaps: [
        {
          entityName: "Contact",
          fieldName: "Email",
          isFastFillable: false,
          isAutoQueryable: true,
          isExactMatchable: true
        }
      ],
      transcriptFields: [
        "c__EmailAddress"
      ],
      displayToAgent: true
    }
  ],
  prechatEntities: [],
  buttonOverrides: [
    "573D000000000OD"
  ],
  receiveQueueUpdates: true,
  isPost: true
}
```

ChasitorResyncState**Request properties**

Property Name	Type	Description	Available Versions
organizationId	String	The chat visitor's Salesforce organization ID.	29.0

Request body

```
{
  organizationId: "00DD0000000JVXs"
}
```

ChasitorSneakPeek

Request properties

Property Name	Type	Description	Available Versions
position	integer	The position of the Sneak Peek update in the chat.	29.0
text	String	The text that the chat visitor is typing in the text input area of the chat window.	29.0

Request body

```
{
  position: 3,
  text: "Hi there."
}
```

ChatMessage

Request properties

Property Name	Type	Description	Available Versions
text	String	The text of the chat visitor's message to the agent.	29.0

Request body

```
{
  text: "I have a question about my account."
}
```

CustomEvent

Request properties

Property Name	Type	Description	Available Versions
type	String	The type of custom event that occurred, used for adding the event listener on the agent's side.	29.0

Property Name	Type	Description	Available Versions
data	String	Data that's relevant to the event that was sent to the agent.	29.0

Request body

```
{
  type: "PromptForCreditCard",
  data: "Visa"
}
```

MultiNoun**Request properties**

Name	Type	Description	Available Versions
nouns	Array of NounWrapper objects	An array of noun objects and their properties that are batched in the <code>MultiNoun</code> request.	29.0

Request body

```
{
  nouns: [
    {
      prefix: "Chasitor",
      noun: "ChatMessage",
      object: {
        text: "Goodbye"
      }
    },
    {
      prefix: "Chasitor",
      noun: "ChatEnd",
      object: {}
    }
  ]
}
```

CHAPTER 8 Response Bodies for Chat REST API

A request to a Chat REST API resource returns a response code. The successful execution of a resource request can also return a response body in JSON format.

Availability

Response Properties

Property Name	Type	Description	Available Versions
results	Array of Result objects	A list of Salesforce IDs that correspond to agents and chat buttons and their respective availability to receive new chat requests.	29.0

Response body

```
{
  "results": {
    "type": "array",
    "description": "List of ids and their availability",
    "items": {
      "name": "result",
      "type": "object",
      "properties": {
        "id": {
          "type": "string",
          "description": "The id of the entity",
          "required": true,
          "version": 29.0
        },
        "isAvailable": {
          "type": "boolean",
          "description": "Whether or not the entity is available for chat",
          "version": 29.0
        }
      }
    },
    "required": true,
    "version": 29.0
  }
}
```

ChasitorSessionData

Response Properties

Property Name	Type	Description	Available Versions
queuePosition	integer	The position of the chat visitor in the chat queue.	29.0
geoLocation	GeoLocation object	The chat visitor's location, based on the IP address from which the request originated.	29.0
url	String	The URL that the chat visitor is visiting.	29.0
oref	String	The original URL that the chat request came from.	29.0
postChatUrl	String	The URL to which to redirect the chat visitor after the chat has ended.	29.0
sneakPeekEnabled	Boolean	Whether Sneak Peek is enabled for the agent who accepts the chat.	29.0
chatMessages	Array of TranscriptEntry objects	The chat message structure that's synchronized across the agent.js and chasitor.js files.	29.0

Response body

```
{
  queuePosition: 1,
  geoLocation: {
    countryCode: "US",
    countryName: "United States of America",
    region: "CA",
    city: "San Francisco",
    organization: Salesforce,
    latitude: 37.793880,
    longitude: -122.395114
  },
  url: "http://yoursite",
  oref: "http://www.google.com?q=yoursite",
  postChatUrl: "http://yoursite/postchat",
  sneakPeekEnabled: true,
  chatMessages: [
    {
      type: "Agent",
      name: "Andy L.",
      content: "Hello, how can I help you?",
    }
  ]
}
```

```

        timestamp: 1376596367548,
        sequence: 1
      },
      {
        type: "Chasitor",
        name: "Jon A.",
        content: "I have a question for you.",
        timestamp: 1376596349132
        sequence: 2
      }
    ]
  }
}

```

ChasitorIdleTimeoutWarningEvent

Response properties

Property Name	Type	Description	Available Versions
idleTimeoutWarningEvent	String	Informs the server when a warning is triggered or cleared. Possible values: <code>triggered</code> and <code>cleared</code> .	35.0

ChatEstablished

Response properties

Property Name	Type	Description	Available Versions
name	String	The name of the agent who is engaged in the chat.	29.0
userId	String	The user ID of the agent who is engaged in the chat.	29.0
sneakPeekEnabled	Boolean	Whether Sneak Peek is enabled for the agent who accepts the chat.	29.0
chasitorIdletimeout	ChasitorIdleTimeoutSettings	Gives the settings for chat visitor idle time-out.	35.0

Response body

```

{
  name: "Andy L.",
  userId: "f1dda237-57f8-4816-b8e8-59775f1e44c8",

```

```

    sneakPeekEnabled: true
  }

```

ChatMessage

Response properties

Property Name	Type	Description	Available Versions
name	String	The name of the agent who is engaged in the chat.	29.0
text	String	The text of the chat message that the agent sent to the chat visitor.	29.0

Response body

```

{
  name: "Andy L."
  text: "Hello, how can I help you?"
}

```

ChatRequestFail

Response properties

Property Name	Type	Description	Available Versions
reason	String	The reason why the chat request failed—for example, no agents were available to chat or an internal error occurred.	29.0
postChatUrl	String	The URL of the post-chat page to which to redirect the chat visitor after the chat has ended.	29.0

Response body

```

{
  reason: "Unavailable",
  postChatUrl: "http://yoursite/postChat"
}

```


ChatRequestSuccess

Response properties

Property Name	Type	Description	Available Versions
queuePosition	integer	The position of the chat visitor in the chat queue.	29.0
estimatedWaitTime	number	The estimated wait time for the button in seconds. If the server cannot retrieve the wait time, this property returns -1.	47.0
geoLocation	GeoLocation object	The chat visitor's location, based on the IP address from which the request originated.	29.0
url	String	The URL that the chat visitor is visiting.	29.0
oref	String	The original URL that the chat request came from.	29.0
postChatUrl	String	The URL to which to redirect the chat visitor after the chat has ended.	29.0
customDetails	Array of CustomDetail objects	The custom details of the deployment from which the chat request was initiated.	29.0
visitorId	String	The ID of the chat visitor.	29.0

Response body

```
{
  queuePosition: 1,
  estimatedWaitTime: 120,
  geoLocation: {
    countryCode: "US",
    countryName: "United States of America",
    region: "CA",
    city: "San Francisco",
    organization: Salesforce,
    latitude: 37.793880,
    longitude: -122.395114
  },
  url: "http://yoursite",
  oref: "http://www.google.com?q=yoursite",
  postChatUrl: "http://yoursite/postchat",
  customDetails: [
    {

```

```

        label: "E-mail Address",
        value: "jon@example.com",
        transcriptFields: [
            "c__EmailAddress"
        ],
        displayToAgent: true
    },
    visitorId: "acd47048-bd80-476e-aa33-741bd5cb05d3"}"

```

ChatTransferred

Response properties

Property Name	Type	Description	Available Versions
name	String	The name of the agent to whom the chat was transferred.	29.0
userId	String	The ID of the chat visitor.	29.0
sneakPeekEnabled	Boolean	Whether Sneak Peek is enabled for the agent to whom the chat was transferred.	29.0
chasitorIdleTimeout	ChasitorIdleTimeoutSettings	Gives the settings for chat visitor idle time-out.	35.0

Response body

```

{
    name: "Ryan S.",
    userId: "edacfa56-b203-43d5-9e1b-678278b61263",
    sneakPeekEnabled: false
}

```

CustomEvent

Response properties

Property Name	Type	Description	Available Versions
type	String	The type of custom event that occurred, used for adding the event listener on the chat visitor's side.	29.0

Property Name	Type	Description	Available Versions
data	String	Data that's relevant to the event that was sent to the chat visitor.	29.0

Response body

```
{
  type: "CreditCardEntered",
  data: "5105105105105100"
}
```

Messages

Response Properties

Property Name	Type	Description	Available Versions
messages	Array of Message objects	The messages that was sent over the course of a chat.	29.0
offset	integer	An internal number to be used with a ReconnectSession request that tracks which messages your client has received.	29.0
sequence	integer	The sequence of the message as it was received over the course of a chat.	29.0

Response body

```
{
  messages: [
    {
      type: "ChatEstablished",
      message: {
        name: "Andy L.",
        userId: "f1dda237-57f8-4816-b8e8-59775f1e44c8",
        sneakPeekEnabled: true
      }
    }
  ],
  sequence: 1,
  offset: 1234567890
}
```

NewVisitorBreadcrumb

Response properties

Property Name	Type	Description	Available Versions
location	String	The URL of the web page that the chat visitor is viewing.	29.0

Response body

```
{
  location: "http://yoursite/page2"
}
```

QueueUpdate

Response properties

Property Name	Type	Description	Available Versions
position	integer	The updated position of the chat visitor in the chat queue.	29.0
estimatedWaitTime	number	The estimated wait time for the button in seconds. If the server cannot retrieve the wait time, this property returns -1.	47.0

Response body

```
{
  position: 3,
  estimatedWaitTime: 120
}
```

ReconnectSession

Response properties

Property Name	Type	Description	Available Versions
resetSequence	Boolean	If true, the sequence for the next request should be reset.	37.0
affinityToken	String	The affinity token for the session that's passed in the header for all future requests.	37.0

Response body

```
{
  resetSequence: true,
  affinityToken: "73061fa0"
}
```

SessionId

Response Properties

Property Name	Type	Description	Available Versions
id	String	The session ID for the new session.	29.0
key	String	The session key for the new session.	29.0
affinityToken	String	The affinity token for the session that's passed in the header for all future requests.	29.0
clientPollTimeout	integer	The number of seconds before you must make a <code>Messages</code> request before your <code>Messages</code> long polling loop times out and is terminated.	29.0

Response body

```
{
  id: "241590f5-2e59-44b5-af89-9cae83bb6947",
  key:
    "f6c1d699-84c7-473f-b194-abf4bf7cccf8!b65b13c7-f597-4dd2-aa3a-cbe01e69f19c",
  affinityToken: "73061fa0"
  clientPollTimeout: "30"
}
```

Settings

Response Properties

Property Name	Type	Description	Available Versions
pingrate	number	The rate at which the visitor must ping the server to maintain the Chat visitor session.	29.0
contentServerUrl	String	The URL of the content server.	29.0

Property Name	Type	Description	Available Versions
buttons	Array of button objects	A list of chat buttons, along with their settings information, that were specified when you made the Settings request.	29.0

Response body

```
{
  "pingRate": {
    "type": "number",
    "description": "The rate at which the visitor should ping the server to maintain presence",
    "required": true,
    "version": 29.0
  },
  "contentServerUrl": {
    "type": "string",
    "description": "The content server URL",
    "required": true,
    "version": 29.0
  },
  "buttons": {
    "type": "array",
    "description": "The list of buttons",
    "items": {
      "name": "button",
      "type": "object",
      "properties": {
        "id": {
          "type": "string",
          "description": "The id of the button",
          "required": true,
          "version": 29.0
        },
        "type": {
          "type": "string",
          "description": "The type of the button",
          "required": true,
          "version": 29.0,
          "enum": ["Standard", "Invite", "ToAgent"]
        },
        "endpointUrl": {
          "type": "string",
          "description": "The custom chat window url of the button",
          "required": false,
          "version": 29.0
        },
        "prechatUrl": {
          "type": "string",
          "description": "The prechat url of the button",
          "required": false,

```

```

        "version":29.0
    },
    "language":{
        "type":"string",
        "description":"The language setting of the button",
        "required":false,
        "version":29.0
    },
    "isAvailable":{
        "type":"boolean",
        "description":"Whether or not the button is available for chat",

        "version":29.0
    },
    /* Invite related settings */
    "inviteImageUrl":{
        "type":"string",
        "description":"The image of the button",
        "required":false,
        "version":29.0
    },
    "inviteImageWidth":{
        "type":"number",
        "description":"The width of the button image",
        "required":false,
        "version":29.0
    },
    "inviteImageHeight":{
        "type":"number",
        "description":"The height of the button image",
        "required":false,
        "version":29.0
    },
    "inviteRenderer":{
        "type":"string",
        "description":"The animation option of the invite",
        "required":false,
        "version":29.0,
        "enum":["Slide","Fade","Appear","Custom"]
    },
    "inviteStartPosition":{
        "type":"string",
        "description":"The start position of the animation",
        "required":false,
        "version":29.0,
        "enum":["TopLeft","TopLeftTop","Top","TopRightTop","TopRight",
            "TopRightRight","Right","BottomRightRight","BottomRight",
"BottomRightBottom","Bottom","BottomLeftBottom","BottomLeft",
            "BottomLeftLeft","Left","TopLeftLeft"]
    },
    "inviteEndPosition":{
        "type":"string",

```

```

        "description": "The end position of the animation",
        "required": false,
        "version": 29.0,
    "enum": ["TopLeft", "Top", "TopRight", "Left", "Center", "Right", "BottomLeft", "Bottom", "BottomRight"]
    },
    "hasInviteAfterAccept": {
        "type": "boolean",
        "description": "Whether or not invite will trigger after accepting",
        "required": false,
        "version": 29.0
    },
    "hasInviteAfterReject": {
        "type": "boolean",
        "description": "Whether or not invite will trigger after rejecting",
        "required": false,
        "version": 29.0
    },
    "inviteRejectTime": {
        "type": "number",
        "description": "The auto reject setting of the invite",
        "required": false,
        "version": 29.0
    },
    "inviteRules": {
        "type": "object",
        "description": "The rules of the invite",
        "required": false,
        "version": 29.0
    }
    /* Invite related settings */
}
},
"required": true,
"version": 29.0
}
}

```

SwitchServer

This response is returned for requests to Visitor resources if the Live Agent instance URL is not correct for the Organization ID provided.

Response Properties

Property Name	Type	Description	Available Versions
newUrl	String	The new Chat API endpoint for your org if your org is moved. It can be moved due to a planned	29.0

Property Name	Type	Description	Available Versions
		org migration or during a Site Switch to a different instance.	

Response body

```
{
  "messages": [
    {
      "type": "SwitchServer"
      "message": {
        "newUrl": "https://LiveAgentPool.salesforceliveagent.com/chat"
      }
    }
  ]
}
```

VisitorId

Response Properties

Property Name	Type	Description	Available Versions
sessionId	String	The session ID for the new session.	29.0

Response body

```
"sessionId":{
  "type":"string",
  "description":"The session id of the visitor",
  "required":true,
  "version":29.0
}
```

SEE ALSO:

[Status Codes and Error Responses](#)

CHAPTER 9 Chat REST API Data Types

A request to a Chat REST API resource returns a response code. The successful execution of a resource request can also return a response body in JSON format. Some response bodies return data types that contain their own properties. All property values that refer to a name of an entity or field are case-sensitive.

Button

Properties

Property Name	Type	Description	Required	Available Versions
id	String	The ID of the chat button object.	TRUE	29.0
type	String	The button type. Valid values are: <ul style="list-style-type: none">• Standard• Invite• ToAgent	TRUE	29.0
endpointUrl	String	The URL of the custom chat window that's assigned to the chat button.	FALSE	29.0
prechatUrl	String	The URL of the pre-chat form that's assigned to the button.	FALSE	29.0
language	String	The chat button's default language.	FALSE	29.0
isAvailable	Boolean	Specifies whether the chat button is available to receive new chat requests (<code>true</code>) or not (<code>false</code>). If you don't see this property, the value is <code>false</code> .	FALSE	29.0
inviteImageUrl (for automated chat invitations only)	String	The URL to the automated invitation's static image resource.	FALSE	29.0

Property Name	Type	Description	Required	Available Versions
inviteImageWidth (for automated chat invitations only)	number	The width in pixels of the automated chat invitation's image.	FALSE	29.0
inviteImageHeight (for automated chat invitations only)	number	The height in pixels of the automated chat invitation's image.	FALSE	29.0
inviteRenderer (for automated chat invitations only)	String	<p>The animation option that's assigned to the automated chat invitation. Valid values are:</p> <ul style="list-style-type: none"> • Slide • Fade • Appear • Custom 	FALSE	29.0
inviteStartPosition (for automated chat invitations only)	String	<p>The position at which the automated chat invitation begins its animation. Valid values are:</p> <ul style="list-style-type: none"> • TopLeft • TopLeftTop • Top • TopRightTop • TopRight • TopRightRight • Right • BottomRightRight • BottomRight • BottomRightBottom • Bottom • BottomLeftBottom • BottomLeft • BottomLeftLeft • Left • TopLeftLeft 	FALSE	29.0
inviteEndPosition (for automated chat invitations only)	String	The position at which the automated chat invitation begins its	FALSE	29.0

Property Name	Type	Description	Required	Available Versions
		animation. Valid values are: <ul style="list-style-type: none"> • TopLeft • Top • TopRight • Left • Center • Right • BottomLeft • Bottom • BottomRight 		
hasInviteAfterAccept (for automated chat invitations only)	Boolean	Specifies whether the automated chat invitation can be sent again after the customer accepted a previous chat invitation (<code>true</code>) or not (<code>false</code>).	FALSE	29.0
hasInviteAfterReject (for automated chat invitations only)	Boolean	Specifies whether the automated chat invitation can be sent again after the customer rejected a previous chat invitation (<code>true</code>) or not (<code>false</code>).	FALSE	29.0
inviteRejectTime (for automated chat invitations only)	number	The amount of time in seconds that the invitation appears on a customer's screen before the invitation is automatically rejected.	FALSE	29.0
inviteRules (for automated chat invitations only)	Object	The custom rules that govern the behavior of the automated chat invitation, as defined in your custom Apex class.	FALSE	29.0
estimatedWaitTime	number	The estimated wait time for the button in seconds. If the server cannot retrieve the wait	FALSE	47.0

Property Name	Type	Description	Required	Available Versions
		time, this property returns -1.		

CustomDetail

Properties

Property Name	Type	Description	Required	Available Versions
label	String	The customized label for the detail.	TRUE	29.0
value	String	The customized value for the detail.	TRUE	29.0
transcriptFields	Array of Strings	The names of fields to which to save the customer's details on the chat transcript.	TRUE	29.0
displayToAgent	Boolean	Specifies whether to display the customized detail to the agent (<code>true</code>) or not (<code>false</code>).	FALSE	29.0

Entity

Properties

Property Name	Type	Description	Required	Available Versions
entityName	String	The record to search for or create.	TRUE	29.0
showOnCreate	Boolean	Specifies whether to display the record after it's created (<code>true</code>) or not (<code>false</code>).	FALSE	29.0
linkToEntityName	String	The name of the record to which to link the detail.	FALSE	29.0
linkToEntityField	String	The field within the record to which to link the detail.	FALSE	29.0

Property Name	Type	Description	Required	Available Versions
saveToTranscript	String	The name of the transcript field to which to save the record.	FALSE	29.0
entityFieldsMaps	Array of EntityFieldsMaps objects	The fields to which to associate the detail on a record.	TRUE	29.0

EntityFieldsMaps

Properties

Property Name	Type	Description	Required	Available Versions
fieldName	String	The name of the field to which to associate the detail.	TRUE	29.0
label	String	The customized label for the detail.	TRUE	29.0
doFind	Boolean	Specifies whether to use the field <code>fieldName</code> to perform a search for matching records (<code>true</code>) or not (<code>false</code>).	TRUE	29.0
isExactMatch	Boolean	Specifies whether to only search for records that have fields that exactly match the field <code>fieldName</code> (<code>true</code>) or not (<code>false</code>).	TRUE	29.0
doCreate	Boolean	Specifies whether to create a record based on the field <code>fieldName</code> if one doesn't exist (<code>true</code>) or not (<code>false</code>).	TRUE	29.0

GeoLocation

Properties

Property Name	Type	Description	Required	Available Versions
countryCode	String	The ISO 3166-1 alpha-2 country code for the chat visitor's location.	TRUE	29.0
countryName	String	The name of the country that's associated with the chat visitor's location.	TRUE	29.0
region	String	The principal administrative division associated with the chat visitor's location—for example, the state or province.	FALSE	29.0
city	String	The name of the city associated with the chat visitor's location.	FALSE	29.0
organization	String	The name of the organization associated with the chat visitor's location.	FALSE	29.0
latitude	number	The latitude associated with the chat visitor's location.	FALSE	29.0
longitude	number	The longitude associated with the chat visitor's location.	FALSE	29.0

Message

Properties

Property Name	Type	Description	Required	Available Versions
type	String	The type of message that was received.	TRUE	29.0
message	Object	A placeholder object for the message that was received. Can return any of the responses that are	TRUE	29.0

Property Name	Type	Description	Required	Available Versions
		available for the Messages request.		

NounWrapper

Properties

Property Name	Type	Description	Required	Available Versions
prefix	String	The prefix of the resource.	TRUE	29.0
noun	String	The name of the resource.	TRUE	29.0
data	String	The data to post to the resource.	FALSE	29.0

Result

Properties

Property Name	Type	Description	Required	Available Versions
id	String	The Salesforce ID of the agent or chat button.	TRUE	29.0
isAvailable	Boolean	Indicates whether the entity that's associated with the Salesforce ID <code>id</code> is available to receive new chat requests (<code>true</code>) or not (<code>false</code>). If you don't see this property, the value is <code>false</code> .	FALSE	29.0
estimatedWaitTime	number	The estimated wait time for the button in seconds. If the server cannot retrieve the wait time, this property returns -1.	FALSE	47.0

TranscriptEntry

Properties

Property Name	Type	Description	Required	Available Versions
type	Enumeration of type String	The type of message in the chat transcript. Valid values are: <ul style="list-style-type: none">• <code>Agent</code>: a message from an agent to a chat visitor• <code>Chasitor</code>: a message from a chat visitor to an agent• <code>OperatorTransferred</code>: A request to transfer a chat to another agent	TRUE	29.0
name	String	The name of the person who sent the chat message.	TRUE	29.0
content	String	The body of the message.	TRUE	29.0
timestamp	number	The date and time when the message was sent.	TRUE	29.0
sequence	number	The sequence in which the message was received in the chat.	TRUE	29.0

SEE ALSO:

[Status Codes and Error Responses](#)

CHAPTER 10 Status Codes and Error Responses

Each request returns a status code or error response to indicate whether the request was successful.

When an error occurs or when a response is successful, the response header contains an HTTP code, and the response body usually contains:

- The HTTP response code
- The message accompanying the HTTP response code

HTTP response code	Description
200	"OK" success code.
202	"Accepted" success code, for POST request.
204	"No Content" success code for Message request; resend the request as part of the message loop.
400	The request couldn't be understood, usually because the JSON body contains an error.
403	The request has been refused because the session isn't valid.
404	The requested resource couldn't be found. Check the URI for errors.
405	The method specified in the Request-Line isn't allowed for the resource specified in the URI.
409	A duplicate long poll using the same session ID has caused the chat to terminate. Reestablish the chat in a new session.
500	An error has occurred within the Chat server, so the request couldn't be completed. Contact Customer Support.
503	The affinity token has changed. Make a <code>ReconnectSession</code> request to get a new affinity token, then make a <code>ChasitorSessionData</code> request to reestablish the chat visitor's data within the new session.

SEE ALSO:

[Your Message Long Polling Loop](#)

[ReconnectSession](#)

[ChasitorSessionData](#)

[Response Bodies for Chat REST API](#)

[Chat REST API Data Types](#)

[ReconnectSession](#)

[ChasitorSessionData](#)

INDEX

A

AgentDisconnect [20](#)
AgentNotTyping [20](#)
AgentTyping [20](#)
Availability [25](#)

B

Breadcrumb [26](#)

C

ChasitorNotTyping [13](#)
ChasitorResyncState [12](#)
ChasitorSessionData [21](#)
ChasitorSneakPeek [14](#)
ChasitorTyping [14](#)
ChatEnd [15](#)
ChatEnded [21](#)
ChatEstablished [21](#)
ChatMessage [16, 22](#)
ChatRequestFail [22](#)
ChatRequestSuccess [22](#)
ChatTransferred [22](#)
CustomEvent [16, 23](#)

D

Data Types [46](#)

E

Error responses [54](#)

M

Messages [17](#)
MultiNoun [23](#)

N

NewVisitorBreadcrumb [23](#)

Q

QueueUpdate [23](#)

R

Requests
 Headers [5](#)

Resources

AgentDisconnect [20](#)
AgentNotTyping [20](#)
AgentTyping [20](#)
Availability [25](#)
Breadcrumb [26](#)
ChasitorNotTyping [13](#)
ChasitorResyncState [12](#)
ChasitorSessionData [21](#)
ChasitorSneakPeek [14](#)
ChasitorTyping [14](#)
ChatEnd [15](#)
ChatEnded [21](#)
ChatEstablished [21](#)
ChatMessage [16, 22](#)
ChatRequestFail [22](#)
ChatRequestSuccess [22](#)
ChatTransferred [22](#)
CustomEvent [16, 23](#)
Data Types [46](#)
Long Polling [6](#)
Messages [17–18](#)
MultiNoun [23](#)
NewVisitorBreadcrumb [23](#)
QueueUpdate [23](#)
Request headers [5](#)
Requests [8–9, 13, 24, 28](#)
SessionId [8](#)
Settings [24](#)
VisitorId [27](#)

Responses

Data Types [46](#)

S

SessionId [8](#)
Settings [24](#)
Status codes [54](#)

V

VisitorId [27](#)