Enterprise Chat and Email Developer's Guide to Web Service APIs for Chat, Release 12.5(1)

For Unified Contact Center Enterprise and Packaged Contact Center Enterprise

First Published: January, 2020

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
https://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to https://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Enterprise Chat and Email Developer’s Guide to Web Service APIs for Chat: For Unified Contact Center Enterprise and Packaged Contact Center Enterprise. January 7, 2020

© 2016-2020 Cisco Systems, Inc. All rights reserved.

Contents

Welcome to the Enterprise Chat and Email (ECE) feature, which provides multichannel interaction software used by businesses all over the world as a core component to the Unified Contact Center Enterprise product line. ECE offers a unified suite of the industry's best applications for chat and email interaction management to enable a blended agent for handling of web chat, email and voice interactions.

1. About this Guide

The *Developers Guide to Web Service APIs for Chat* describes how customers can use the Chat Web Services APIs to show the Chat link on their web sites based on queue depth and the availability of agents to handle new chats.

2. Related Documents

The latest versions of all Cisco documentation can be found online at https://www.cisco.com

Subject	Link
Complete documentation for Enterprise Chat and Email, for both Cisco Unified Contact Center Enterprise (UCCE) and Cisco Packaged Contact Center Enterprise (PCCE)	https://www.cisco.com/c/en/us/support/customer-collaboration/cisco-enterprise-chat-email/tsd-products-support-series-home.html

3. Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

4. Field Alerts and Field Notices

Cisco products may be modified or key processes may be determined to be important. These are announced through use of the Cisco Field Alerts and Cisco Field Notices. You can register to receive Field Alerts and Field Notices through the Product Alert Tool on Cisco.com. This tool enables you to create a profile to receive announcements by selecting all products of interest.

Log into www.cisco.com and then access the tool at https://www.cisco.com/cisco/support/notifications.html

5. Documentation Feedback

To provide comments about this document, send an email message to the following address: mailto:contactcenterproducts_docfeedback@cisco.com

We appreciate your comments.

6. Document Conventions

This guide uses the following typographical conventions.

Convention	Indicates
Italic	Emphasis. Or the title of a published document.
Bold	Labels of items on the user interface, such as buttons, boxes, and lists. Or text that must be typed by the user.
`Monospace`	The name of a file or folder, a database table column or value, or a command.
Variable	User-specific text; varies from one user or installation to another.

7. Introduction

Enterprise Chat and Email customers can use the Chat Web Services APIs to show the Chat link on their web sites based on the availability of agents to handle new chats. The APIs can be used for following purposes:

- To enable or disable the Chat button on web sites depending on available agents.

- To get the amount of time a customer might have to wait before an agent is available to chat.
- To find the position of customer in a queue to estimate how long a customer might have to wait before an agent is available to chat.
- Write new custom surveys for chat sessions to capture additional data.

The API implementation follows standards for REST-based APIs and provides anonymous access. The API returns data in XML format.

Supported Versions

- ECE 12.5(1)

8. Schema Definitions

The schema definitions used in web services are available as XSD files in the distribution. You can use the PackIt tool to extract the schema definitions. They are located in the `eService.ear` file at the location `lib/int/egpl_application_server.jar/com/egain/live/framework/bosh/xsd`.

9. Display chat option based on agent availability

Check if there are any available agents (defined as the Available for chat check box being selected in the Agent Console) who can handle chats that start from a specific entry point. Use this API to decide to show or hide the chat link on the website based on agent availability.

Request

A request to check agent availability has the following form:

Method	URL
GET	/egain/chat/entrypoint/agentAvailability/*id*

URL Parameters

Name	Description	Type	Required	Default value for optional parameters
ID	The ID of the entry point for which you want to check the agent availability.	long	Yes	

Request Headers

Not applicable.

Request Body

None

XSD for Request Body

Not applicable.

SAMPLE REQUEST XML

Not applicable.

Response

The response includes HTTP status code and a Response Body.

STATUS CODES

Success codes:

- **200:** The agent availability status is returned. A True response means an agent is available. A False response means no agent is available.

Error codes:

- **500:** Unable to retrieve agent availability information.

RESPONSE HEADERS

Not applicable.

RESPONSE BODY

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<agentAvailability available="true" xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind"/>
```

XSD FOR RESPONSE BODY

The XML schema is defined in the `agentAvailability` node in the `eGainBosh.xsd` file.

10. Display chat option based on agent capacity for chats

Fetches the capacity of all agents to work on new chat activities in the queue mapped to the given entry point. This API returns the difference between the maximum load that all agents can take and the current load of all agents in the queue of the entry point. Use this API to decide to show the Chat button or offer new chats to customers only when there are agents available to receive more work.

Request

A request to check agent capacity has the following form:

Method	URL
GET	/egain/chat/entrypoint/capacity/*id*

URL Parameters

Name	Description	Type	Required	Default value for optional parameters
ID	The ID of the entry point for which you want to check the agent capacity.	long	Yes	

Request Headers

Not applicable.

Request Body

Not applicable.

XSD for Request Body

Not applicable.

Sample Request XML

Not applicable.

Response

The response includes HTTP status code and a Response Body.

Status codes

Success codes:

- 200: The remaining capacity of agents for the entry point is returned. A positive number reflects the remaining capacity. Zero means either there are no agents available in the system, or all agents are working to their maximum load.

Error codes:

- 500: Unable to retrieve the information.

Response Headers

Not applicable.

Response Body

Output is generated in an XML format.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<availableSlots xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind">
  <ns2:count>4</ns2:count>
</availableSlots>
```

XSD for Response Body

The XML schema is defined in the `availableSlots` node in the `eGainBosh.xsd` file.

11. Display chat option based on queue depth and wait time

Get details of the number of chats waiting in the queue for assignment, and the estimated wait time in the queue. Use this API to decide to show or hide the chat link on the website based on wait time and number of chats in the queue.

Request

A request to check queue wait time and number of chats in the queue has the following form:

Method	URL
GET	/egain/chat/entrypoint/liveSessionStatus/id

URL Parameters

Name	Description	Type	Required	Default value for optional parameters
ID	The ID of the entry point for which you want to check the queue depth and the wait time.	long	Yes	

REQUEST HEADERS

Not applicable.

REQUEST BODY

Not applicable.

XSD FOR REQUEST BODY

Not applicable.

SAMPLE REQUEST XML

Not applicable.

Response

The response includes HTTP status code and a Response Body.

STATUS CODES

Success codes:

- 200: Queue depth and wait time are returned.

Error codes:

- 500: Unable to retrieve the queue depth and wait time.

RESPONSE HEADERS

Not applicable.

RESPONSE BODY

Output is generated in an XML format.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sessionStatus xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind">
    <ns2:waitTime>3.0</ns2:waitTime>
    <ns2:queueDepth>1</ns2:queueDepth>
</sessionStatus>
```

XSD FOR RESPONSE BODY

The XML schema is defined in the `sessionStatus` node in the `eGainBosh.xsd` file.

12. Display chat option based on queue depth and agent availability

Checks eligibility of a chat entry point to handle new chat activities based on the following conditions:

1. If there are any agents available to work on new chat activities.
2. If the queue associated with that entry point has reached its configured maximum depth i.e. total number of chats being processed by the queue is equal to the maximum number of chats that the queue can process at any given point of time.

Use this API to decide to show or hide the chat link on the website based on agent availability and whether the queue associated with the given entry point can accept more chats.

Request

A request to check eligibility of chat entry point has the following form:

Method	URL
GET	/egain/chat/entrypoint/checkEligibility/*id*

URL PARAMETERS

Name	Description	Type	Required	Default value for optional parameters
ID	The ID of the entry point for which you want to check.	long	Yes	

REQUEST HEADERS

Not applicable.

REQUEST BODY

Not applicable.

XSD FOR REQUEST BODY

Not applicable.

SAMPLE REQUEST XML

Not applicable.

Response

The response includes HTTP status code and a Response Body.

STATUS CODES

Success codes:

- 200: Entry point eligibility is returned. This is identified by value of the attribute responseType. This attribute can have any one of the following values:
    - 0: The queue associated with this entry point can handle new chats.
    - 1: No agent is available to work on new chats.
    - 2: Maximum queue depth has reached for the queue associated with the given entry point and no new chats will be processed.

Error codes:

- 500: Unable to retrieve the eligibility of chat entry point.

RESPONSE HEADERS

Not applicable.

RESPONSE BODY

Output is generated in an XML format.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<checkEligibility xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind" responseType=0 />
```

XSD FOR RESPONSE BODY

The XML schema is defined in the `checkEligibility` node in the `eGainBosh.xsd` file.

13. Display chat option based on queue depth, agent availability, and entry point status

Checks if new chats can be processed by a given chat entry point based on the following conditions:

1. Chat entry point is active.
2. If there are any agents available to work on new chat activities.
3. If the queue associated with that entry point has reached its configured maximum depth i.e. total number of chats being processed by the queue is equal to the maximum number of chats that queue can process at any given point of time.

Use this API to decide to show or hide the chat link on the website based on the state of the chat entry point, agent availability, and whether the queue associated with the given entry point can accept more chats.

Request

A request to check if new chats can be processed by chat entry point has the following form:

Method	URL
GET	/egain/chat/entrypoint/chatAllowed/*id*

URL Parameters

Name	Description	Type	Required	Default value for optional parameters
ID	The ID of the entry point for which you want to check.	long	Yes	

Request Headers

Not applicable.

Request Body

Not applicable.

XSD for Request Body

Not applicable.

SAMPLE REQUEST XML

Not applicable.

Response

The response includes HTTP status code and a Response Body.

STATUS CODES

Success codes:

- 200: If new chats can be processed by entry point is returned. This is identified by value of the attribute allowed. This attribute can have value either true or false. If it is true, that means this entry point can handle new chats. If it is false, that means this entry point cannot handle new chats. If the value is false, attribute reason can have one of the below mentioned values to identify the cause if unavailability:
    - queue_depth_reached: Maximum queue depth has reached for the queue associated with the given entry point and no new chat will be processed.
    - agent_not_available: No agent is available to work on new chat.
    - service_not_running: Agent assignment service is not running.
    - invalid_entry_point: Entry point passed in the request is not valid.
    - entry_point_inactive: Entry point passed in the request is not active.
    - Other: This entry point cannot handle new chats due to other reasons.

Error codes:

- 500: Unable to retrieve whether this entry point can handle new chats.

RESPONSE HEADERS

Not applicable.

RESPONSE BODY

Output is generated in an XML format.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<chatAllowed xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind" allowed="true"/>
OR
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<chatAllowed xmlns:ns2="http://bindings.egain.com/chat"
xmlns:ns4="urn:ietf:params:xml:ns:xmpp-stanzas" xmlns:ns3="jabber:client"
xmlns:ns5="http://jabber.org/protocol/httpbind" allowed="false"
reason="entry_point_inactive"/>
```

XSD FOR RESPONSE BODY

The XML schema is defined in the `chatAllowed` node in the `eGainBosh.xsd` file.

14. Submit custom chat surveys

This API can be used to submit custom survey forms shown at the end of chat session.

Request

Method	URL
POST	/egain/chat/entrypoint/survey

URL PARAMETERS

Not applicable.

REQUEST HEADERS

Not applicable.

REQUEST BODY

The Request Body needs to be in XML format. The following table describes the required and optional attributes that can be used to submit a survey.

Name	Description	Type	Required
Question	Question which is part of the survey that is displayed to the customer.	String	Yes
Answer	Answer to the corresponding question.	String	Yes

XSD FOR REQUEST BODY

The XML schema is defined in the `eGainSurvey` node in the `eGainSurvey.xsd` file.

SAMPLE REQUEST XML

```
<egainSurvey sid="1000">
<survey> <question>Question 1</question> <answer>Answer 1</answer></survey>
<survey><question>Question 2</question> <answer>Answer 2</answer></survey>
<survey><question>Question 3</question> <answer>Answer 3</answer></survey>
<survey><question>Question 4</question> <answer>Answer 4</answer></survey>
</egainSurvey>
```

Response

The response includes HTTP status code.

STATUS CODES

Success codes:

- 204: Survey is submitted successfully.

RESPONSE HEADERS

Not applicable.

RESPONSE BODY

Not applicable.

XSD FOR RESPONSE BODY

Not applicable.