



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.

Web Services API Reference

Web Services and Applications 8.5.0

12/31/2021

Table of Contents

Web Services API Reference	3
Request Parameters	4
Return Values	8
Features	11
Settings	13
Channels API	17
Query Agent State	18
Set Agent States Across all Channels	23
Set Agent State for a Particular Channel	26
CometD Notifications	28
Interactions API	29
Interactions API Operations	30
Interactions On E-mail Channel	40
Interactions On E-mail Channel Notifications	48
Interactions On Chat Channel	51
Interaction Notifications	57
Attachments API	67
Users API	68
Skills API	70
Devices API	75
Contacts API	80
Calls API	82
Asynchronous Operations	84
Change Password	85
Reset Password	87
Presence	88
Call Control	89
Device/Channel Management	91
Asynchronous Telephony Operations Notifications	92
Workbins	97

Web Services API Reference

RESTRICTED

NOTE: This is **restricted** documentation. The content of this document is subject to change without notice. Some of the information contained in this document might not apply to your deployment.

The API Reference covers all the representations and methods available through the Web Services API Reference. Each subcategory presents a set of operations, and the related resources.

Request Parameters

This topic discusses the request parameters for the Web Services API Reference.

Object Fields

When making "list" requests for any kind of object, Web Services returns a list of the corresponding object URI's.

For example:

Request:

```
GET .../api/v2/me/devices
```

Response:

```
{ "statusCode" : 0,
  "uris" : [ "http://127.0.0.1:8080/api/v2/devices/ba0f987f-15b4-42c7-bed0-5f302259f9db" ]
}
```

In order to receive a list of objects with their actual fields, you will need to provide the `fields` request parameter, and have it set either to `*`, or to a list of data fields of interest.

For example:

Request:

```
GET .../api/v2/me/devices?fields=*
```

Response:

```
{ "devices" : [ { "capabilities" : [ "ForwardCallsOn",
  "DoNotDisturbOn"
],
  "deviceState" : "Active",
  "doNotDisturb" : "Off",
  "e164Number" : "5001",
  "id" : "ba0f987f-15b4-42c7-bed0-5f302259f9db",
  "phoneNumber" : "5001",
  "telephonyNetwork" : "Private",
  "userState" : { "displayName" : "Ready",
    "id" : "9430250E-0A1B-421F-B372-F29E69366DED",
    "state" : "Ready"
  },
  "voiceEnvironmentUri" : "http://127.0.0.1:8080/api/v2/voice-environments/a481cd8e-7b6a-4466-af88-db3471ac909e"
} ],
  "statusCode" : 0
}
```

```
}
```

Subresources

The subresources feature allows you to read subresources of an object together with the object itself. For example, if you have a user object that has one or more skills and one or more devices, and you want to read all of those in one request, you need to do the following:

Request:

```
GET .../api/v2/users/<user_id>?subresources=*
```

Response:

```
{
  "id":<user_id>,
  "firstName":<first_name>,
  "...":
  "skills":[{
    "id":<skill_1_id>,
    ...
  },
  ...
  {
    "id":<skill_N_id>,
    ...
  }],
  "devices":[{
    "id":<device_1_id>,
    ...
  },
  ...
  {
    "id":<device_M_id>,
    ...
  }]
}
```

In comparison, if you do not include the `subresources` parameter in the request, you will get everything except the "skills" collection and "devices" collection.

Important

It is also possible to apply the subresources feature to object settings and request both an object and its settings in one request.

Selecting Subresources

In the example above, "subresources=*" was specified in order to get all available subresources. If the object you are interested in has several types of subresources, it is possible to choose whether you want all subresources to be returned or just some of them. This can be achieved by specifying a comma-separated list of subresources.

For example:

Example 1

Request:

```
GET .../api/v2/users/<user_id>?subresources=skills,devices
```

Response:

```
{
  "id":<user_id>,
  "firstName":<first_name>,
  ...
  "skills":[{
    "id":<skill_1_id>,
    ...
  },
  ...
  {
    "id":<skill_N_id>,
    ...
  }],
  "devices":[{
    "id":<device_1_id>,
    ...
  },
  ...
  {
    "id":<device_M_id>,
    ...
  }]
}
```

Example 2

Request:

```
GET .../api/v2/users/<user_id>?subresources=skills
```

Response:

```
{
  "id":<user_id>,
  "firstName":<first_name>,
  ...
  "skills":[{
    "id":<skill_1_id>,
    ...
  },
  ...
  {
    "id":<skill_N_id>,
    ...
  }]
}
```

```
    },
    ...
  {
    "id":<skill_N_id>,
    ...
  }
}
```

User Authentication

Basic HTTP Authentication is used. Please see [RFC 2617](#) Section 2 for reference.

Supported Requests

Important

Note that this is a Reference Guide for Pre-Release software. Therefore, some attributes and operations might not yet be supported for all resources. Some are marked throughout the document, but it is possible some were missed.

The following requests are supported at this time:

- /devices: fields=*
- /features: fields=*
- /me: subresources=*
- /me/calls: fields=*
- /me/devices: fields=*
- /me/skills: fields=*
- /skills: fields=*
- /system/features: fields=*
- /system/routing-templates: channel, version (these are query parameters), fields=*
- /users: fields=*, subresources=*
- /users/{id}: subresources=*
- /users/{id}/devices: fields=*
- /recordings: startTime, endTime, callerPhoneNumber, dialedPhoneNumber, userName, offset, limit (query parameters)

Return Values

All API methods return a result for the operation in addition to the HTTP status code. The results are different depending on the type of operation. Except as noted, the results will be as follows:

All Methods

All methods will **ALWAYS** return the attribute `statusCode` and, if an error has occurred, `statusMessage` on the top level of the response. The following status codes are supported:

Code	Description
0	The operation is successful. No <code>statusMessage</code> is provided.
1	A required parameter is missing in the request.
2	A specified parameter is not valid for the current state.
3	The operation is forbidden.
4	An internal error occurred. This might happen if an internal error occurred with Web Services or with one of the servers working with Web Services (such as Cassandra or a Genesys Framework component).
5	The user does not have permission to perform this operation.
6	The requested resource could not be found.

If an error occurred during an operation, the response includes `statusCode` and `statusMessage` clarifying the error. No other attributes will be included.

Note that if an error occurs during a request, with you can assume that the request failed to modify the data of the contact center.

GET

GET requests are used to retrieve a variety of information. The response body will depend on the requested and the request parameters. These are the possible scenarios:

1. If retrieving a collection of URI's, the response will include the array attribute `"uris"` which will hold the requested collection.
2. If retrieving a collection of resources, the response will include an array attribute named after the requested resource. For example, GET `/users?fields=*` will contain `"users": [{ ".user1.."},`

{...user2...}, etc].

3. If the URI is a "singular" resource such as `GET /users/{id}`, the response will include an attribute named after the singular of the requested resource which will contain the requested value. `GET /users/{id}` will return `"user": {...user..}`.

Examples

`GET /skills`

```
{
  "statusCode":0,
  "uris":["http://../api/v2/skills/123", "http://../api/v2/skills/456", etc]
}
```

`GET /users?fields=*`

```
{
  "statusCode":0,
  "users":[{"userName":"..", "firstName":"...", etc}, {"userName":"..", "firstName":"...",
etc}]
}
```

POST to Create Resource

When a POST request is successful, the following extra attributes will be included:

1. `id`—the ID of the newly-created object.
2. `uri`—the URI to access the newly-created object.

Examples

Request:

```
POST /users
{
  ... some user data
}
```

Response:

```
{
  "statusCode":0
  "id":"12345",
  "uri":"http://...api/v2/users/12345"
}
```

POST to Assign Resource

POST can also be used to assign one resource to another's collection such as when assigning a skill to a user. When this is the case, no extra attributes are returned and only `statusCode:0` will be returned on success.

DELETE

The DELETE operation does not have any extra attributes. Only `statusCode:0` will be returned on success.

DELETE to Unassign Resource

DELETE can also be used to unassign one resource from another's collection such as when unassigning a skill from a user. No extra attributes are returned and only `statusCode:0` will be returned on success.

PUT

The PUT operation does not have any extra attributes. Only `statusCode:0` will be returned on success.

Asynchronous Operations

Web Services supports many operations that are performed using POST on an existing resource and the response for which is sent via CometD. When POST is used to perform one of these operations, `statusCode:0` will be returned on success.

Hybrid Operations

In order to increase API usability and minimize network traffic, multi-step operations are occasionally implemented. For instance, it is possible to create a device and assign it to a user with one operation. When hybrid operations are implemented, the methods will return all of the values required for each operation being performed. For example, POST to create a resource requires a return value of `"uri"` and `"id"` whereas POST to assign does not have any extra return values. Implementing a multi-step "create and assign" POST returns `"uri"`, `"id"`, and `"statusCode"` on successful completion.

Features

The features resource allows the client application to determine which functionality is available in the current contact center. This data can then be used to draw the UI as appropriate for the feature set that is supported for the current contact center.

A *feature* represents a set of functionality that may include channels, services, resources, sets of operations, settings groups, and so on. Anything that is needed for the feature to function successfully should be available when a feature is enabled for the contact center. When a feature is disabled, the API will behave as if this set of functionality does not exist. This will return results such as 404 errors when relevant resources are being accessed, settings groups not being visible in lists, and operations returning with `invalid` operation errors.

Operations

Two resources are available in the API to support this functionality:

- `api/v2/system/features` represents all features available in the system.
- `api/v2/features` represents the set of features for a given contact center.

The following operations are available for **/features**

Operation	Description	Permissions
GET	Returns a list of URIs for the features assigned to this contact center. The parameter <code>fields=*</code> will cause full feature descriptions to be returned instead of URI's.	Contact Center Admin, Agent

The following operations are available for **/system/features**

Operation	Description	Permissions
GET	Returns a list of URIs for all of the features available in the system. The parameter <code>fields=*</code> will cause full feature descriptions to be returned instead of URI's.	Contact Center Admin

Important

The full feature set is defined by Web Services and is not modifiable.

Features

The following operations are available for **/features/{id}**

Operation	Description	Permissions
GET	Returns the full feature description.	Agent, Contact Center Admin
DELETE	Removes the feature from the contact center.	Cloud Admin

Attributes

The following attributes are supported for each feature:

Attribute	Type	Description	Access
id	String	The name of the feature (this is also the unique identifier and should be in a URI-compatible format).	GET
displayName	String	Name that describes the feature.	GET
description	String	Description of the feature.	GET

Supported Features

The following features are currently supported. If a feature is marked "assigned by default" all contact centers will have this feature assigned.

Name	Description	Assigned by default
api-provisioning-read	General provisioning read	Y
api-provisioning-write	General provisioning write	Y
api-voice	The API for the voice channel.	Y
ui-supervisor-provisioning-routing	UI to provision routing	Y
ui-supervisor-provisioning-skill	UI to provision skills	Y
ui-supervisor-provisioning-user	UI to provision users	Y
ui-supervisor-recording	UI to view call recordings	Y
ui-supervisor-reporting	UI to view contact center statistics	Y
api-multimedia	API for non-voice channels	N

Settings

The *settings* resource is intended for configuration tasks that modify the behavior of existing functionality. This resource contains a list of URI's that correspond to named settings groups. Each of these groups has its own attributes and security settings.

Operations

The following operations are supported by **/settings**:

Operation	Description	Permissions
GET	Returns a list of all available settings groups for the contact center	Contact Center Admin

Attributes

The following attributes are supported for each item that is returned by GET **/settings**:

Attribute	Type	Description	Access
uri	String	The URI to the settings group.	GET
displayName	String	Name that describes the settings group.	GET

Example

The following sample will return a list of all settings groups for the contact center:

GET /api/v2/settings

```
{
  settings:[{
    "displayName":"Agent States",
    "uri":"http://.../api/v2/settings/agent-states"
  },{
    "displayName":"Dispositions",
    "uri":"http://.../api/v2/settings/dispositions"
  }]
}
```

Important

Settings groups are different from **features**. Enabling a feature for the contact center may result in particular settings groups becoming available.

Each setting group may return the attribute "key" along with a setting array. This attribute will specify which of the setting attributes should be used as a key to identify the setting during modification (PUT) requests. If the "key" attribute is not present, "name" will be the default identifying attribute.

Supported Settings Groups

The supported Settings Groups are General and Agent States.

General

This group is available under the following URI: `http://<host:port>/api/v2/settings/general-settings`. It contains the following attributes:

Attribute	Type	Description
countryCode	String	A two-character country code for the Contact Center.
countryDigits	String	A numerical country prefix for phone numbers.
countryName	String	Country name.

Agent States

Agent state is accessible by using **/settings/agent-states**. It allows the Contact Center Admin to define custom agent states that include reason codes.

Operations

The following operations are supported for the **/settings/agent-states** resource:

Operation	Description	Permissions
GET	Returns a list of all available agent states for the contact center	Contact Center Admin, Agent
POST	Creates a new agent state description.	Contact Center Admin
PUT	Modifies an existing agent state.	Contact Center Admin
DELETE	Removes an agent state description from the system.	Contact Center Admin

Attributes

The following attributes are supported for each agent state descriptor:

Attribute	Type	Description	Access	Required
id	String	The unique ID (GUID) for the agent state. This ID is included in the userState of device change messages when an agent state is matched.	GET	Y
operationName	String	The unique operation name that will be used to set this state (for example, OutToLunch).	GET, POST, PUT	Y
displayName	String	Name for the state.	GET, POST, PUT	Y
state	Enum	The actual T-Server state (Ready/NotReady).	GET, POST, PUT	Y
workMode	Enum	An after call work mode. Note that modes are applicable to particular states. For Ready: ManualIn/ AutoIn/ReturnBack. For NotReady: AfterCallWork/AuxWork/ LegalGuard/ NoCallDisconnect/ WalkAway. This should be enforced by the API.	GET, POST, PUT	N
reason	String	The reason for the agent's state (if specified, it must be unique as it will be used as a reason code).	GET, POST, PUT	N

Examples

Each contact center will initially have five agent state descriptions for the basic Ready/Not Ready/ Offline operations that are currently in use. These five operations are read only. They cannot be deleted or modified:

```
{ "key" : "operationName",
  "settings" : [ { "displayName" : "AfterCallWork",
    "id" : "D3663509-3D82-4DD3-A82E-2EA8EFA02AEF",
    "operationName" : "AfterCallWork",
    "state" : "NotReady",
    "workMode" : "AfterCallWork"
  },
  { "displayName" : "AuxWork",
    "id" : "2B36138D-C564-4562-A8CB-3C32D564F296",
    "operationName" : "AuxWork",
    "state" : "NotReady",
    "workMode" : "AuxWork"
  },
  { "displayName" : "Not Ready",
    "id" : "900D55CC-2BB0-431F-8BF9-D3525B383BE6",
    "operationName" : "NotReady",
    "state" : "NotReady"
  },
  { "displayName" : "Offline",
    "id" : "0F7F5003-EF26-4D13-A6Ef-D0C7EC819BEB",
    "operationName" : "Offline",
    "state" : "Logout"
  },
  { "displayName" : "Ready",
    "id" : "9430250E-0A1B-421F-B372-F29E69366DED",
    "operationName" : "Ready",
    "state" : "Ready"
  }
],
  "statusCode" : 0
}
```

To add a new "Not Ready" state called "Out to lunch":

POST /settings/agent-states

```
{
  "operationName":"OutToLunch",
  "displayName":"Not Ready - Out to lunch",
  "state":"NotReady",
  "reason":"OutToLunch"
}
```

To modify the display name of the "OutToLunch" state described above:

PUT /settings/agent-states

```
{
  "operationName":"OutToLunch",
  "displayName":"Not Ready - Lunch!"
}
```


Channels API

A **channel** typically represents a particular type of media such as e-mail or chat, but it can also represent specific online platforms such as Salesforce. Channels are linked to **Features** in a one-to-many relationship: a feature can have multiple channels, while each channel is only associated with one feature.

The following operations are supported on **/me/subresources=*** :

Operation Name	HTTP Operation	Description	Permissions
Overall User State	GET	Returns all of an Agent's channel and device states.	Agent

The following operations are supported on **/me/devices**:

Operation Name	HTTP Operation	Description	Permissions
Get All Devices	GET	Returns all of an Agent's device states.	Agent
Get a Particular Device	GET	Returns all of an Agent's specific device states.	Agent

The following operations will be supported on **/me/channels**:

Operation Name	HTTP Operation	Description	Permissions
Get All Multimedia Channels	GET	Returns all of an Agent's channel states.	Agent
Get a Particular Multimedia Channel	GET	Returns all of an Agent's specific channel state.	Agent
Ready Channel	POST	Sets an Agent's channel to Ready.	Agent
NotReady Channel	POST	Sets an Agent's channel to NotReady.	Agent

The following operations will be supported on **/me**:

Operation Name	HTTP Operation	Description	Permissions
Ready for all Channels	POST	Sets all user channels to 'Ready'	Agent
NotReady for all Channels	POST	Sets all user channels to 'NotReady'	Agent
DNDOn	POST	Activates Do Not Disturb.	Agent
DNDOff	POST	Deactivates Do Not Disturb.	Agent

Query Agent State

Use the following to retrieve information about agent states.

Important

The Output shown in the following examples might not be applicable to your deployment.

Overall User State

Returns all of a User's device and channel states.

Input Parameters

GET on /me?subresources=*

N/A

Output

Success

```
{
  devices: [{
    ...
    id: idOne,
    userState: {
      state: "NotReady",
      reason: "Lunch",
      displayName: "Out to Lunch"
      workMode: "AuxWork"
    }
    ... // other device properties
  }],
  channels [{
    channel: "email",
    userState: {
      state: "NotReady",
      reason: "Lunch"
      displayName: "Out to Lunch"
    },
    ... // other channel properties
  }, {
    channel: "chat",
    userState: {
```

```
state: "NotReady",
reasonCode: "Lunch"
displayName: "Out to Lunch"
},
...
}]
...
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

Get All Devices

Returns all of a User's device states.

Input Parameters

GET on /me/devices

N/A

Output

Success

```
{
  devices: [{
    ...
    id: idOne
    userState: {
      state: "NotReady",
      reason: "Lunch",
      displayName: "Out to Lunch"
      workMode: "AuxWork"
    }
    ... // other device properties
  },
    {
      ...
      id: idTwo
      userState: {
        state: "NotReady",
        reason: "Lunch",
        displayName: "Out to Lunch"
        workMode: "AuxWork"
      }
      ... // other device properties
    }
  ]
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

Get a Particular Device

Returns a User's specific device state.

Input Parameters

GET on /me/devices/{id}

N/A

Output

Success

```
{
  device: {
    ...
    id: idOne
    userState: {
      state: "NotReady",
      reason: "Lunch",
      displayName: "Out to Lunch"
      workMode: "AuxWork"
    }
    ... // other device properties
  }
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

Get All Multimedia Channels

Returns all of a User's channel states.



Pre-Release Notice: The information contained in this section is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This section provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

Input Parameters

GET on /me/channels

N/A

Output

Success

```
{
  channels: [{
    channel: "email",
    userState: {
      state: "NotReady",
      reason: "Lunch"
      displayName: "Out to Lunch"
    },
    ... // other channel properties
  }, {
    channel: "chat",
    userState: {
      state: "NotReady",
      reason: "Lunch"
      displayName: "Out to Lunch"
    },
    ...
  }]
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

Get a Particular Multimedia Channel

Returns a User's channel state, given a `channelId`.



Pre-Release Notice: The information contained in this section is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This section provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

Input Parameters

GET on /me/channels/{channelId}

N/A

Output

Success

```
{
  channel: {
    channel: "email",
    userState: {
      state: "NotReady",
      reason: "Lunch",
      displayName: "Out to Lunch"
    },
    ... // other channel properties
  }
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

Set Agent States Across all Channels

Use the following to set Agent States across all channels.

Ready for All Channels

Sets all of an Agent's channel statuses to 'Ready', meaning that the Agent is ready to receive interaction invitations for all the Agent's media channels.

Input Parameters

POST on /me

```
{
  "operationName": "Ready"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

NotReady For All Channels

Sets all of an Agent's channel statuses to 'NotReady', meaning that the Agent will not receive interaction invitations for any of the Agent's media channels.

Input Parameters

POST on /me

```
{
  "operationName": "NotReady"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

DndOn

Activates the Do Not Disturb status. This applies to all devices and channels, and overrides statuses such as 'Ready'.

Input Parameters

POST on /me

```
{
  "operationName": "DoNotDisturbOn"
}
```

Output

Success

```
{
  "status": "ok"
}
```

See [Cometd Notifications](#).

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

DndOff

Deactivates the Do Not Disturb status. This applies to all devices and channels, and overrides statuses such as 'Ready'.

Input Parameters

POST on /me

```
{
  "operationName": "DoNotDisturbOff"
}
```

Output

Success


```
{
  "status": "ok"
}
```

See [Cometd Notifications](#).

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

Set Agent State for a Particular Channel

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

Updates Agent States for a Channel

Method	POST		
URL	By channel: /me/channels/{channelId}		
Name	Type	Mandatory	Description
Parameters			
operationName	string	Y	For example: NotReady, Ready, OutToLunch

Output

Ready Channel

Puts a User's channel ('chat', 'email', and so on) into a 'Ready' state, meaning that the User is open to receiving interaction invitations for that multimedia channel.

Input Parameters

POST on /me/channels/{channelId}

```
{
  "operationName": "Ready"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

NotReady Channel

Puts a User's multimedia channel ('chat', 'email', and so on) into a 'NotReady' state, meaning that the User will not even view an interaction invitation for that multimedia channel.

Input Parameters

POST on /me/channels/{channelId}

```
{
  "operationName": "NotReady"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failure

```
{
  "status": "error",
  "errorDescription": details
}
```

CometD Notifications

A client can subscribe to `/v2/me/channels` to receive asynchronous CometD notifications.

Sample Notifications

```
{
  "data": {
    "channels": [
      {
        "userState": {
          "state": "Ready",
          "displayName": "Ready",
          "id": "9430250E-0A1B-421F-B372-F29E69366DED"
        },
        "channel": "chat"
      }
    ]
  }
}

{
  "data": {
    "channels": [
      {
        "dndState": "On",
        "userState": {
          "state": "NotReady",
          "displayName": "Not Ready",
          "id": "900D55CC-2BB0-431F-8BF9-D3525B383BE6"
        }
      }
    ]
  }
}
```

Interactions API




Pre-Release Notice: The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

An **interaction** represents activity on a particular channel, and thus is very tightly coupled with **Channels**. Activity in this case records not just the name of the action, but all of the relevant data as well such as the reply text of an email or even a reply e-mail's **Attachments** (or Documents). An Interaction can be associated with other Interactions through the `parentInteractionId` (also seen as `parentId`) attribute.

See the following topics for details about the Interactions API:

- [Interactions API Operations](#)
- [Interactions on E-mail Channel](#)
- [Interactions on Chat Channel](#)

Interactions API Operations

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

Synchronous

The following operations are supported on **/interactions**:

HTTP Operation	Description	Permissions
GET	Returns the list of all currently active interactions for this Contact Center.	<ul style="list-style-type: none"> Contact Center Admin Agent

The following operations are supported on **/me/interactions**:

HTTP Operation	Parameters	Description	Response Sample	Permissions
GET	fields=*	Returns the list of interactions that are currently being handled by the agent.	{"statusCode":0,"interactions":[same message format as comes via cometD notifications for interactions]}	Agent
GET	N/A	Returns a list of interaction URIs that are currently being handled by the agent.	{"statusCode":0,"uris":[" http://.../api/v2/interactions/00009a8T8Y130123 "]}	Agent

The following operations are supported on **/interactions/{id}**:

HTTP Operation	Description	Permissions
GET	Returns interaction details for a given interaction.	Agent
PUT	Updates a given interaction. The actual operation depends on the interaction's channel.	Agent

The following operations are supported on **me/interactions/{id}**:

Operation Name	HTTP Operation	Description	Permissions
Accept	POST	This operation can be performed after a new	Agent

Operation Name	HTTP Operation	Description	Permissions
		interaction notification arrives. Accepting an interaction allows Agents to start working on it.	
Reject	POST	This operation can be performed after a new interaction notification arrives. Rejecting an interaction allows an Agent the opportunity to work on a different one.	Agent
StopProcessing	POST	This operation can be performed after an agent has finished working on an interaction.	Agent
PlaceInQueue	POST	This operation can be performed on an interaction so that another person can continue processing it.	Agent
UpdateProperties	POST	This operation can be performed on an interaction that an agent is currently working on.	Agent
PlaceInWorkbin	POST	This operation places an interaction into a specified workbin.	Agent
GetWorkbinContent	POST	This operation will get the contents of a given workbin.	Agent
Transfer	POST	This operation transfers interaction ownership from one Agent to another Agent.	Agent
StartConference	POST	This operation allows Agents to initiate a conference with another Agent on a specific interaction.	Agent
EndConference	POST	This operation lets Agents leave a currently active conference.	Agent

Accept

This operation can be performed after a new Interaction invitation notification arrives. It is essentially one of two options an agent (**User**) has regarding an Interaction invitation: accepting it or **rejecting** it. Choosing this action will associate the Interaction to the agent. This association or "ownership" will continue until that agent performs a Complete, Transfer, or Re-direct action on the aforementioned Interaction. A successful request will result in an empty http-200 response. In most cases, the true response will arrive on the cometd topic (channel) **/v2/me/interactions/**.

Please refer to **Email Accepted** for details on the actual response for email Interactions.

Input Parameters

POST on **/me/interactions/{id}**

```
{
  "operationName": "Accept"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

Reject

This operation can be performed after a new Interaction invitation notification arrives. It is essentially one of two options an agent has regarding an Interaction invitation: **accepting** it or rejecting it. A successful request will result in an empty http-200 response, and compels the System to present the Interaction to another Agent.

Input Parameters

POST on **/me/interactions/{id}**

```
{
  "operationName": "Reject"
}
```


Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

StopProcessing

This operation can be performed after an Agent **accepts** an interaction. This marks the end of the Agent's ownership of the interaction.

This is typically called when an Agent (**User**) has finished working on an interaction

Input Parameters

POST on **/me/interactions/{id}**

```
{
  "operationName": "StopProcessing"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

PlaceInQueue

This operation can be performed after an Agent **accepts** an Interaction. This marks the end of the Agent's ownership of the interaction.

This is typically called when an Agent (**User**) has finished working on an interaction

Input Parameters

POST on **/me/interactions/{id}**

```
{
  "operationName": "PlaceInQueue"
  "queue": "queue-name"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```

UpdateProperties

This operation can be performed on an interaction that agent is working on.

Input Parameters

POST on **/me/interactions/{id}**

```
{
  "operationName": "UpdateProperties"
  "properties": {
    {
      "field1" : value1,
      "field2" : value2,
      "field3" : {
        {
          "fieldA" : valueA,
          "fieldB" : valueB
        }
      }
    }
  }
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

PlaceInWorkbin

This operation will place an interaction into a specified workbin.

Input

```
{
  "operationName": "PlaceInWorkbin",
  "workbin": workbin-identifiers,
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

GetWorkbinContent

This operation will return the contents of a given workbin. The contents will come as a cometD message.

Input

```
{
  "operationName": "GetWorkbinContent",
  "workbin": "workbin-identifier"
}
```

Output

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

Notifications

A list of `InteractionStateMessages`; it will contain the same information as found in new notification messages.

Transfer

This operation transfers interaction ownership from one Agent to another Agent. Functionality may vary from different media types. Further details can be found within the media type sub-topics.

Input

```
{
  "operationName": "Transfer",
  "targetId": "a unique Agent identifier"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

StartConference

This operation allows Agents to initiate a conference with another Agent on a specific interaction; it allows two Agents to work on the same Interaction simultaneously. To leave a conference, users must make an **EndConference** request.

Currently, functionality is limited to conferencing. Monitoring and coaching functionality is currently not available. Further details can be found within the media type sub-topics.

HTTP Request

```
{
  "operationName": "StartConference",
  "agentId": "a unique Agent identifier"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```

Cometd Response Message

Important

The full response will vary depending on the media type.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "state": "InvitedConference"
}
```

Cometd Notification Messages

This notifies the remaining Agents on the conference that an Agent has left the conference.

Important

The full response will vary depending on the media type.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "state": "LeftConference"
}
```

EndConference

This operation allows Agents to leave a currently active conference.

When Agents leave a conference, the last remaining Agent will retain Interaction ownership.

HTTP Request

```
{
  "operationName": "LeaveConference",
  "agentId": "a unique Agent identifier"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```


See also

Refer to the following topics for more information about the Interactions API:

- [E-mail Interactions](#)
- [Notifications](#)

- Chat Interactions
 - Inbound Operations
 - Notifications

Interactions On E-mail Channel

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

Operations

The following operations are available on **me/interactions/{interaction_id}**:

Operation Name	HTTP Operation	Description	Type	Permissions
Accept	POST	This operation can be performed after a new interaction notification arrives. Accepting an interaction allows Agents to start working on it.	E-mail Specific	Agent
Reject	POST	This operation can be performed after a new interaction notification arrives. Rejecting an interaction allows an Agent the opportunity to work on a different one.	General Interaction	Agent
Reply	POST	This operation can be performed after a new e-mail interaction has been accepted.	E-mail Specific	Agent
ReplyAll	POST	This operation can be performed after a new e-mail interaction has been accepted.	E-mail Specific	Agent
Forward	POST	This operation can be performed after a new e-mail interaction has been accepted.	E-mail Specific	Agent

Operation Name	HTTP Operation	Description	Type	Permissions
CancelReply	POST	This operation can be performed to cancel an outbound e-mail.	E-mail Specific	Agent
Send	POST	This operation can be performed to send an outbound e-mail.	E-mail Specific	Agent
Transfer	POST	This operation can be performed while working on an interaction. The interaction can be transferred to another agent or to a group of agents associated with a skill.	General Interaction	Agent
StopProcessing	POST	This operation can be performed after an agent has finished working on an interaction.	General Interaction	Agent
PlaceInQueue	POST	This operation can be performed on an interaction to continue processing by another process.	General Interaction	Agent

Accept

This operation can be performed after a new Interaction invitation notification arrives. It is essentially one of two options an agent (**User**) has regarding an Interaction invitation: accepting it or **rejecting** it. Choosing this action will associate the Interaction to the agent. This association or "ownership" will continue until that agent performs a Complete, Transfer, or Re-direct action on the aforementioned Interaction. A successful request will result in an empty http-200 response. The true response will arrive on the cometd topic (channel) /me/interactions/

Please refer to **E-mail Accepted** for details on the actual response.

HTTP Request

POST on **/me/interaction/{id}**

```
{
  "operationName": "Accept"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

Cometd Response

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": alpha-numeric-string,
  "channel": "email",
  "email_object" {
    "ToAddress": to-address,
    "FromAddress": from-address,
    "CCAddresses": cc-addresses,
    "Subject": subject-of-the-email-in-string-representation,
    "Text": text-of-email-as-text,
    "StructuredText": text-of-email-formatted,
    "MimeType": mime-type-of-text,
    "StructuredTextMimeType": mime-type-of-structuredtext,
    "TheComment": notes-on-interaction,
    "ContactID": contactid-mapped-to-sender,
    "attachments": [{
      "id": attachmentId,
      "name": attachment-name,
      "size": attachment-size,
      "mimetype": attachment-mime-type
    },
    ...
  ]
},
  "interactionType": "Email",
  "interactionSubType": "InboundNew",
  "userData": complete-map-of-user-data-excluding-binary-values
  "currentQueue": value-of-attr_itx_queue,
  "inQueues": value-of-attr_in_queues,
  "outQueues": value-of-attr_out_queues,
  "receivedAt": value-of-attr_itx_received_at
  "state": "Accepted",
  "capabilities": ["Transfer", "Reply", "ReplyAll", "StopProcessing", "PlaceInQueue", "PlaceInWorkbin", "UpdateProperties"]
}
```

Reply

This operation can be performed after an agent (**User**) has **Accepted** an e-mail interaction. A Reply operation creates an interaction in Interaction Server and Universal Contact Server (UCS). The

operation is asynchronous, meaning that cometD notification will be sent upon successful completion of the operation (see [E-mail Reply Created](#)). The newly-created interaction will have:

- ToAddress equal to ReplyToAddress-from-original-email
- FromAddress equal to ToAddress-from-original-Email
- Subject equal to subjectPrefix+original-email-subject
- The body will be empty unless quoteOriginal is set to true and will be pre-pended by replyToStartLine and modified with indentCharacter, if present

HTTP Request

POST on **/me/interactions/{id}**

```
{
  "operationName": "Reply",
  "subjectPrefix": "prefix-to-add-to-copy-of-subject-from-inbound-email (optional)",
  "replyToStartLine": "line-to-add-before-quoting-original-text (optional)",
  "indentCharacter": "indent-character-to-add-before-each-newline-of-original-text (optional)",
  "quoteOriginal": "boolean-to-tell-whether-to-add-original-text",
  "queueName": "name-of-the-queue-where-to-place-interaction-while-agent-is-composing-it"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```

Cometd Response

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "ToAddress": "to-address",
    "FromAddress": "from-address",
    "CCAddresses": "cc-addresses",
    "Subject": "subject-of-the-email-in-string-representation",
    "Text": "text-of-email-as-text",
    "StructuredText": "text-of-email-formatted",
    "MimeType": "mime-type-of-text",
    "StructuredTextMimeType": "mime-type-of-structuredtext",
  },
  "interactionType": "Email",
  "interactionSubType": "OutboundReply",
}
```

```
"receivedAt:value-of-attr_itx_received_at
"state":"ReplyCreated",
"capabilities":["Cancel","Send","PlaceInWorkbin","UpdateProperties","LinkAttachment"]
}
```

ReplyAll

This operation can be performed after an agent (**User**) has **accepted** an e-mail interaction. The ReplyAll operation creates an interaction in Interaction Server and UCS. The operation is asynchronous, meaning that cometD notification will be sent upon successful completion of operation (see **E-mail Reply Created**). The newly-created e-mail interaction will have:

- ToAddress equal to ReplyToAddress-from-original-email
- CCAddresses equal to CCAddresses-from-original-email
- FromAddress equal to ToAddress-from-original-Email
- Subject equal to subjectPrefix+original-email-subject
- The body will be empty unless quoteOriginal is set to true and will be pre-pended by replyToStartLine and modified with indentCharacter, if present

HTTP Request

POST on **/me/interactions/{id}**

```
{
"operationName":"ReplyAll",
"subjectPrefix":prefix-to-add-to-copy-of-subject-from-inbound-email (optional),
"replyToStartLine":line-to-add-before-quoting-original-text (optional),
"indentCharacter":indent-character-to-add-before-each-newline-of-original-text (optional),
"quoteOriginal":boolean-to-tell-whether-to-add-original-text -- default is true
"queueName":name-of-the-queue-where-to-place-interaction-while-agent-is-composing-it
}
```

HTTP Response

Success

```
{
"status":"ok",
"replyInteractionId":interactionId
}
```

Failed

```
{
"status":"error",
"errorDescription":details
}
```

Cometd Response

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "ToAddress": "to-address",
    "FromAddress": "from-address",
    "CCAddresses": "cc-addresses",
    "Subject": "subject-of-the-email-in-string-representation",
    "Text": "text-of-email-as-text",
    "StructuredText": "text-of-email-formatted",
    "MimeType": "mime-type-of-text",
    "StructuredTextMimeType": "mime-type-of-structuredtext",
  },
  "interactionType": "Email",
  "interactionSubType": "OutboundReply",
  "receivedAt": "value-of-attr_itx_received_at",
  "state": "ReplyCreated",
  "capabilities": ["Cancel", "Send", "PlaceInWorkbin", "UpdateProperties", "LinkAttachment"]
}
```

Forward

This operation can be performed after an agent (**User**) has **Accepted** an e-mail interaction. The only difference between the Forward action and the **Reply** action is that Forward sends an e-mail to an e-mail address that is provided by an Agent. This should not be confused with the **Transfer** operation, where the owner of the interaction changes.

HTTP Request

POST on **/me/interaction/{id}**

```
{
  "operationName": "Forward"
}
```

HTTP Response

Success

```
{
  "status": "ok",
  "replyInteractionId": "interactionId"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```

Send

This operation is used to send the contents of the e-mail that an agent has created.

HTTP Request

POST on **/me/interactions/{id}**

```
{
  "operationName": "Send",
  "queueName": "queue-name-to-place-interaction-in",
  "email_object": {
    "toAddress": "to-address",
    "subject": "email-subject",
    "text": "text-of-email",
    "fromAddress": "from-address",
    "ccAddress": "cc-addresses, //optional",
    "bccAddress": "bcc-addresses //optional"
  }
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": "details"
}
```

CancelReplyEmail

This operation can be performed after a client successfully completes a create **reply** or **reply all** operation. This is typically called after an agent is finished writing the contents of the reply e-mail.

HTTP Request

POST on **/me/interactions/{id}**

```
{
  "operationName": "CancelReplyEmail"
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```

Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

CreateEmail

This operation can be performed to create a new interaction of type email. The new interaction will come to the agent via the cometD notification described in [E-mail Reply Created](#).

HTTP Request

POST on **/me/interactions/**

```
{
  "operationName": "CreateEmail",
  "queueName": name-of-queue-where to store the interaction,
  "interactionType": type of the interaction to be created (e.g. Outbound),
  "interactionSubType": subtype of the interaction to be created (e.g. OutboundNew)
}
```

HTTP Response

Success

```
{
  "status": "ok"
}
```


Failed

```
{
  "status": "error",
  "errorDescription": details
}
```

See also:

- [Interactions on E-mail Channel Notifications](#)
- [Interaction Notifications](#)

Interactions On E-mail Channel Notifications

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

To begin receiving e-mail channel-related notifications from the System through the cometd topics (cometd channels), an agent needs to set his agent status for the e-mail channel to Ready. He then needs to subscribe to the **/me/interactions** cometd topic and listen. To prevent the System from sending more e-mail interaction notifications, the user needs to set his e-mail channel state to NotReady.

When a user sets his status for the e-mail channel to Ready, the following notifications may occur:

New E-mail Interaction

This notification invites an Agent to either **Accept** or **Reject** an incoming e-mail Interaction.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "subject": "subject-of-the-email-in-string-representation",
    "interactionId": "unique id of email interaction"
  },
  "interactionType": "Email",
  "interactionSubType": "InboundNew",
  "userData": "complete-map-of-user-data-excluding-binary-values",
  "currentQueue": "value-of-attr_itx_queue",
  "state": "Invited",
  "capabilities": ["Accept", "Reject"]
}
```

messageTypeName—denotes the type of cometd notification. This is a fixed value. It will always be `InteractionStateMessage`.

id—the interaction's unique identifier.

channel—fixed value. This will always be `email`.

email_object—contains e-mail related data.

- **subject**—the e-mail's topic.

state—identifies the interaction's current state. This is a fixed value. It will always be Invited.

capabilities—this array displays possible operations that may occur after this operation.

E-mail Revoked

After a predetermined period of Agent inactivity, this notifies the Agent that the System has removed an e-mail interaction from him or her.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "subject": "subject-of-the-email-in-string-representation"
  },
  "state": "Revoked",
  "capabilities": []
}
```

E-mail Accepted

After an Agent performs an **Accept**, the following message will be sent via cometD.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "ToAddress": "to-address",
    "FromAddress": "from-address",
    "CCAddresses": "cc-addresses",
    "Subject": "subject-of-the-email-in-string-representation",
    "Text": "text-of-email-as-text",
    "StructuredText": "text-of-email-formatted",
    "MimeType": "mime-type-of-text",
    "StructuredTextMimeType": "mime-type-of-structuredtext",
    "TheComment": "notes-on-interaction",
    "ContactID": "contactid-mapped-to-sender",
    "attachments": [{
      "id": "attachmentId",
      "name": "attachment-name",
      "size": "attachment-size",
      "mimetype": "attachment-mime-type"
    },
    ...
  ]
},
  "interactionType": "Email",
  "interactionSubType": "InboundNew",
  "userData": "complete-map-of-user-data-excluding-binary-values",
  "currentQueue": "value-of-attr_itx_queue",
  "inQueues": "value-of-attr_in_queues",
  "outQueues": "value-of-attr_out_queues",
  "receivedAt": "value-of-attr_itx_received_at"
}
```


```
"state": "Accepted",
"capabilities": ["Transfer", "Reply", "ReplyAll", "StopProcessing", "PlaceInQueue", "PlaceInWorkbin", "UpdateProperties"]
}
```

E-mail Reply Created

After an Agent performs either **Reply** or **ReplyAll**, the following message will be sent via cometD.

```
{
  "messageTypeName": "InteractionStateMessage",
  "id": "alpha-numeric-string",
  "channel": "email",
  "email_object" {
    "ToAddress": "to-address",
    "FromAddress": "from-address",
    "CCAddresses": "cc-addresses",
    "Subject": "subject-of-the-email-in-string-representation",
    "Text": "text-of-email-as-text",
    "StructuredText": "text-of-email-formatted",
    "MimeType": "mime-type-of-text",
    "StructuredTextMimeType": "mime-type-of-structuredtext",
  },
  "interactionType": "Email",
  "interactionSubType": "OutboundReply",
  "receivedAt": "value-of-attr_itx_received_at",
  "state": "ReplyCreated",
  "capabilities": ["Cancel", "Send", "PlaceInWorkbin", "UpdateProperties", "LinkAttachment"]
}
```

Interactions On Chat Channel

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

The following operations are available:

Operation Name	HTTP Operation	Description	Type	Permissions
Accept	POST	This operation can be performed after a new interaction notification arrives. Accepting an interaction allows Agents to start working on it.	Chat Specific	Agent
Reject	POST	This operation can be performed after a new interaction notification arrives. Rejecting an interaction allows an Agent the opportunity to work on a different one.	General Interaction	Agent
SendMessage	POST	This operation can be performed whenever an agent wants to send a message to other chat participants.	Chat Specific	Agent
SendNotification	POST	This operation can be performed to send notifications to other chat participants (for example, 'agent typing').	Chat Specific	Agent
Transfer	POST	This operation can be performed while working on an interaction. The interaction can be transferred to another agent or	General Interaction	Agent

Operation Name	HTTP Operation	Description	Type	Permissions
		to a group of agents associated with a skill.		
StopProcessing	POST	This operation can be performed after an agent has finished working on an interaction.	General Interaction	Agent
RequestChatHistory	POST	This operation can be performed to get the chat history. Only agents who are participating in the chat can execute this operation.	Chat Specific	Agent
InitiateConsulation	POST	This operation can be performed to initiate a consultation.	Chat Specific	Agent
CompleteTransfer	POST	This operation transforms a consultation to a transfer.	Chat Specific	Agent
CompleteConference	POST	This operation transforms a consultation to a conference.	Chat Specific	Agent

Accept

After successfully executing this operation, a cometD notification will come with complete chat history up until this point.

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "Accept",
  "nickname": "agent-nickname-to-be-displayed-in-chat"
}
```

HTTP Response

Success

```
{
  "statusCode": 0
}
```

Failure

```
{
  "statusCode": an integer value above 0,
  "statusMessage": details
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

SendMessage

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "SendMessage",
  "text": string-representation-of-text
}
```

HTTP Response

Success

```
{
  "statusCode": 0
}
```

Failure

```
{
  "statusCode": an integer value above 0,
  "statusMessage": details
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

SendNotification

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "SendNotification",
  "notification": TypingStarted, TypingStopped
}
```

HTTP Response

Success

```
{
  "statusCode": 0
}
```

Failure

```
{
  "statusCode": an integer value above 0,
  "statusMessage": details
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

RequestChatHistory

After this request is received, the chat history will be sent via cometD notification.

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "RequestChatHistory",
  "eventId", event-id-from-which-to-start-chat-history (optional)
}
```

HTTP Response

Success

```
{
  "statusCode": 0
}
```

InitiateConsultation

After this request is received, a chat consultation with an agent will be started. This operation should also be used for Two-Step Transfer and Two-Step Conference

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "StartChatConsultation",
  "agentId": "unique alpha-numeric identifier for user"
}
```

HTTP Response

```
{
  "status": 0
}
```

CompleteTransfer

After this request is received, a chat interaction will be transferred to the agent with whom consultation was initiated using [InitiateConsultation](#). This will complete the Two-Step Transfer.

HTTP Request

POST on /me/interactions/{id}

```
{
  "operationName": "CompleteTransfer"
}
```

HTTP Response

```
{
  "status": 0
}
```

CompleteConference

After this request is received, a chat interaction will be conferenced with the agent with whom consultation was initiated using [InitiateConsultation](#). This will complete the Two-Step Conference.

HTTP Request

POST on /me/interactions/{id}

```
{  
  "operationName": "LeaveChatConsultation"  
}
```


HTTP Response

```
{  
  "status": 0  
}
```

See also:

- [Interaction Notifications](#)

Interaction Notifications

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

The following is a list of Cometd notifications that are not tied to a specific interaction operation.

To begin receiving e-mail channel related notifications from the System through the Cometd topics (also known as cometd channels), an agent needs to set his agent status for the email channel to Ready. He then needs to subscribe to the **/me/interactions** cometd topic and listen. To prevent the System from sending more e-mail interaction notifications, the user needs to set his e-mail channel state to NotReady.

When a user sets his status for the email channel to Ready, the following notifications may occur:

New Email Interaction Invite

This notification invites an Agent to either **Accept** or **Reject** an incoming email Interaction.

```
[
  {
    u'data': {
      u'userData': {
        u'Header_Content-Type': u'multipart/
mixed;boundary="=====1840393583==",
        u'RTargetTypeSelected': u'2',
        u'RTargetObjectSelected': u'?': 2>1',
        u'_ContainsAttachment': u'false',
        u'CBR-actual_volume': u'',
        u'RVQID': u'',
        u'To': u'support@ci-vm106',
        u'Header_Date': u'Fri,
28Jun2013 13: 30: 34-0700',
        u'_AttachmentsSize': u'0',
        u'_AutoReplyCount': 0,
        u'RTargetObjSelDBID': u'',
        u'Header_MIME-Version': u'1.0',
        u'Mailbox': u'support',
        u'CBR-Interaction_cost': u'',
        u'CBR-contract_DBIDs': u'',
        u'ContactId': u'0000Ha8S2A7X000N',
        u'RTargetAgentSelected': u'Agent1',
        u'CBR-IT-path_DBIDs': u'',
        u'RTargetAgentGroup': u'?': 2>1',
        u'RTargetRuleSelected': u'',
        u'FromPersonal': u'',
        u'RTargetPlaceSelected': u'Agent1',
        u'_AttachmentFileNames': u'',
        u'RTenant': u'Environment',
        u'RRequestedSkills': None,
```

```

        u'Origination_Source': u'Email',
        u'RRequestedSkillCombination': u'',
        u'RVQDBID': u'',
        u'RStrategyDBID': u'134',
        u'CustomerSegment': u'default',
        u'PegAG?: 2>1': 1,
        u'ServiceType': u'default',
        u'FromAddress': u'Customer1@ci-vm106',
        u'ServiceObjective': 0,
        u'Header_Message-ID': u'<4fdz6n7aiqcqyw3.280620131330@ci-vm106>',
        u'RTargetRequested': u'?': 2>1',
        u'EmailAddress': u'Customer1@ci-vm106',
        u'RStrategyName': u'SimpleEmailInStrategy',
        u'Subject': u'test_reply'
    },
    u'email_object': {
        u'interactionId': u'0000Ja8V814907JM'
    },
    u'currentQueue': u'SimpleEmailInQueue',
    u'interactionSubType': u'InboundNew',
    u'capabilities': [
        u'Accept',
        u'Reject'
    ],
    u'state': u'Invited',
    u'messageType': u'InteractionStateMessage',
    u'id': u'0000Ja8V814907JM',
    u'channel': u'email',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'25',
    u'channel': u'/meta/connect'
}
]

```

Email Revoked

After a predetermined period of Agent inactivity, this notifies the Agent that the System has removed an e-mail Interaction from him or her.

```

[
  {
    u'data': {
      u'userData': {
        u'Header_Content-Type': u'multipart/
mixed;boundary="=====1840393583==",
        u'RTargetTypeSelected': u'2',
        u'RTargetObjectSelected': u'?': 2>1',
        u'_ContainsAttachment': u'false',
        u'CBR-actual_volume': u'',

```

```

    u'RVQID': u'',
    u'To': u'support@ci-vm106',
    u'Header_Date': u'Fri,
28Jun2013 13: 30: 34-0700',
    u'_AttachmentsSize': u'0',
    u'_AutoReplyCount': 0,
    u'RTargetObjSelDBID': u'',
    u'Header_MIME-Version': u'1.0',
    u'Mailbox': u'support',
    u'CBR-Interaction_cost': u'',
    u'CBR-contract_DBIDs': u'',
    u'ContactId': u'0000Ha8S2A7X000N',
    u'RTargetAgentSelected': u'Agent1',
    u'CBR-IT-path_DBIDs': u'',
    u'RTargetAgentGroup': u'?': 2>1',
    u'RTargetRuleSelected': u'',
    u'FromPersonal': u'',
    u'RTargetPlaceSelected': u'Agent1',
    u'_AttachmentFileNames': u'',
    u'RTenant': u'Environment',
    u'RRequestedSkills': None,
    u'Origination_Source': u'Email',
    u'RRequestedSkillCombination': u'',
    u'RVQDBID': u'',
    u'RStrategyDBID': u'134',
    u'CustomerSegment': u'default',
    u'PegAG?: 2>1': 1,
    u'ServiceType': u'default',
    u'FromAddress': u'Customer1@ci-vm106',
    u'ServiceObjective': 0,
    u'Header_Message-ID': u'<4fdz6n7aiqcqyw3.280620131330@ci-vm106>',
    u'RTargetRequested': u'?': 2>1',
    u'EmailAddress': u'Customer1@ci-vm106',
    u'RStrategyName': u'SimpleEmailInStrategy',
    u'Subject': u'test_reply'
  },
  u'email_object': {
    u'interactionId': u'0000Ja8V814907JM'
  },
  u'currentQueue': u'SimpleEmailInQueue',
  u'interactionSubType': u'InboundNew',
  u'capabilities': [],
  u'state': u'Revoked',
  u'messageType': u'InteractionStateMessage',
  u'id': u'0000Ja8V814907JM',
  u'channel': u'email',
  u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
  u'successful': True,
  u'advice': {
    u'interval': 0,
    u'timeout': 30000,
    u'reconnect': u'retry'
  },
  u'id': u'25',
  u'channel': u'/meta/connect'
}
]

```

Email Accepted

```
[
  {
    u'data': {
      u'userData': {
        u'Header_Content-Type': u'multipart/
mixed;boundary="=====1840393583==",
        u'RTargetTypeSelected': u'2',
        u'RTargetObjectSelected': u'?': 2>1',
        u'_ContainsAttachment': u'false',
        u'CBR-actual_volume': u'',
        u'RVQID': u'',
        u'To': u'support@ci-vm106',
        u'Header_Date': u'Fri,
28Jun2013 13: 30: 34-0700',
        u'_AttachmentsSize': u'0',
        u'_AutoReplyCount': 0,
        u'RTargetObjSelDBID': u'',
        u'Header_MIME-Version': u'1.0',
        u'Mailbox': u'support',
        u'CBR-Interaction_cost': u'',
        u'CBR-contract_DBIDs': u'',
        u'ContactId': u'0000Ha8S2A7X000N',
        u'RTargetAgentSelected': u'Agent1',
        u'CBR-IT-path_DBIDs': u'',
        u'RTargetAgentGroup': u'?': 2>1',
        u'RTargetRuleSelected': u'',
        u'FromPersonal': u'',
        u'RTargetPlaceSelected': u'Agent1',
        u'_AttachmentFileNames': u'',
        u'RTenant': u'Environment',
        u'RRequestedSkills': None,
        u'Origination_Source': u'Email',
        u'RRequestedSkillCombination': u'',
        u'RVQDBID': u'',
        u'RStrategyDBID': u'134',
        u'CustomerSegment': u'default',
        u'PegAG?: 2>1': 1,
        u'ServiceType': u'default',
        u'FromAddress': u'Customer1@ci-vm106',
        u'ServiceObjective': 0,
        u'Header_Message-ID': u'<4fdz6n7aiqcqyw3.280620131330@ci-vm106>',
        u'RTargetRequested': u'?': 2>1',
        u'EmailAddress': u'Customer1@ci-vm106',
        u'RStrategyName': u'SimpleEmailInStrategy',
        u'Subject': u'test_reply'
      },
      u'email_object': {
        u'text': u'This is email body.\r\nTesting,
hopefully this will work',
        u'interactionId': u'0000Ja8V814907JM',
        u'fromAddress': u'Customer1@ci-vm106',
        u'toAddress': u'support@ci-vm106',
        u'subject': u'test_reply'
      },
      u'currentQueue': u'SimpleEmailInQueue',
      u'interactionSubType': u'InboundNew',
      u'inQueues': {
        u'__STOP__': u''
      },
      u'capabilities': [
```

```

        u'Transfer',
        u'PlaceInQueue',
        u'StopProcessing',
        u'PlaceInWorkbin',
        u'UpdateProperties',
        u'Reply',
        u'ReplyAll'
    ],
    u'outQueues': {
        u'SimpleEmailOutQueue': u''
    },
    u'state': u'Accepted',
    u'messageType': u'InteractionStateMessage',
    u'id': u'0000Ja8V814907JM',
    u'channel': u'email',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'26',
    u'channel': u'/meta/connect'
}
]

```

Chat Invited

```

[
    {
        u'data': {
            u'userData': {
                u'IdentifyCreateContact': u'3',
                u'RTargetTypeSelected': u'2',
                u'RTargetObjectSelected': u'?': 2>1',
                u'RVQID': u'',
                u'ChatServerAppName': u'es_chat',
                u'ChatServerHost': u'hpe-voicevm-76.genesyslab.com',
                u'RTargetObjSelDBID': u'',
                u'ChatServerPort': u'7160',
                u'CBR-Interaction_cost': u'',
                u'CBR-contract_DBIDs': u'',
                u'RTargetAgentSelected': u'Agent1',
                u'CBR-IT-path_DBIDs': u'',
                u'RTargetRuleSelected': u'',
                u'RTargetPlaceSelected': u'Agent1',
                u'CBR-actual_volume': u'',
                u'RTenant': u'Environment',
                u'ChatServerDBID': u'155',
                u'RRequestedSkills': None,
                u'RRequestedSkillCombination': u'',
                u'RVQDBID': u'',
                u'RStrategyDBID': u'125',
                u'CustomerSegment': u'default',
                u'PegAG?: 2>1': 1,
            }
        }
    }
]

```

```

        u'ServiceType': u'default',
        u'ServiceObjective': 0,
        u'RTargetRequested': u'?: 2>1',
        u'RTargetAgentGroup': u'?: 2>1',
        u'RStrategyName': u'SimpleChatInStrategy',
        u'Subject': u'Customersupport'
    },
    u'currentQueue': u'SimpleChatInQueue',
    u'interactionSubType': u'InboundNew',
    u'capabilities': [
        u'Accept',
        u'Reject'
    ],
    u'state': u'Invited',
    u'messageType': u'InteractionStateMessage',
    u'id': u'0000Ja8V814907KS',
    u'channel': u'chat',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'15',
    u'channel': u'/meta/connect'
}
]

```

Chat Revoked

```

[
    {
        u'data': {
            u'userData': {
                u'IdentifyCreateContact': u'3',
                u'RTargetTypeSelected': u'2',
                u'RTargetObjectSelected': u'?: 2>1',
                u'RVQID': u'',
                u'ChatServerAppName': u'es_chat',
                u'ChatServerHost': u'hpe-voicevm-76.genesyslab.com',
                u'RTargetObjSelDBID': u'',
                u'ChatServerPort': u'7160',
                u'CBR-Interaction_cost': u'',
                u'CBR-contract_DBIDs': u'',
                u'RTargetAgentSelected': u'Agent1',
                u'CBR-IT-path_DBIDs': u'',
                u'RTargetRuleSelected': u'',
                u'RTargetPlaceSelected': u'Agent1',
                u'CBR-actual_volume': u'',
                u'RTenant': u'Environment',
                u'ChatServerDBID': u'155',
                u'RRequestedSkills': None,
                u'RRequestedSkillCombination': u'',
                u'RVQDBID': u'',
                u'RStrategyDBID': u'125',
            }
        }
    }
]

```

```

        u'CustomerSegment': u'default',
        u'PegAG?: 2>1': 1,
        u'ServiceType': u'default',
        u'ServiceObjective': 0,
        u'RTargetRequested': u'?: 2>1',
        u'RTargetAgentGroup': u'?: 2>1',
        u'RStrategyName': u'SimpleChatInStrategy',
        u'Subject': u'Customersupport'
    },
    u'currentQueue': u'SimpleChatInQueue',
    u'interactionSubType': u'InboundNew',
    u'capabilities': [],
    u'state': u'Revoked',
    u'messageType': u'InteractionStateMessage',
    u'id': u'0000Ja8V814907KS',
    u'channel': u'chat',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'15',
    u'channel': u'/meta/connect'
}
]

```

Chat Accepted

```

[
    {
        u'data': {
            u'chat_notifications': [
                {
                    u'EventId': 1,
                    u'NoticeText': None,
                    u'NoticeType': None,
                    u'MessageText': None,
                    u'UserId': u'009B51CDF64818CA',
                    u'MessageType': u'PARTY_JOINED',
                    u'UserType': u'Client',
                    u'Visibility': u'All',
                    u'NickName': u'CustomerName!L'
                },
                {
                    u'EventId': 2,
                    u'NoticeText': None,
                    u'NoticeType': None,
                    u'MessageText': u'testmessage1fromclient',
                    u'UserId': u'009B51CDF64818CA',
                    u'MessageType': u'NEW_MESSAGE',
                    u'UserType': None,
                    u'Visibility': u'All',
                    u'NickName': None
                }
            ]
        }
    }
]

```

```

    {
      u'EventId': 3,
      u'NoticeText': None,
      u'NoticeType': None,
      u'MessageText': u'testmessage2fromclient',
      u'UserId': u'009B51CDF64818CA',
      u'MessageType': u'NEW_MESSAGE',
      u'UserType': None,
      u'Visibility': u'All',
      u'NickName': None
    },
    {
      u'EventId': 4,
      u'NoticeText': None,
      u'NoticeType': None,
      u'MessageText': u'testmessage3fromclient',
      u'UserId': u'009B51CDF64818CA',
      u'MessageType': u'NEW_MESSAGE',
      u'UserType': None,
      u'Visibility': u'All',
      u'NickName': None
    },
    {
      u'EventId': 5,
      u'NoticeText': None,
      u'NoticeType': None,
      u'MessageText': None,
      u'UserId': u'009B51CDF65418CC',
      u'MessageType': u'PARTY_JOINED',
      u'UserType': u'Agent',
      u'Visibility': u'All',
      u'NickName': u'TestName'
    }
  ],
  u'userData': {
    u'IdentifyCreateContact': u'3',
    u'RTargetTypeSelected': u'2',
    u'RTargetObjectSelected': u'?': 2>1',
    u'RVQID': u'',
    u'ChatServerAppName': u'es_chat',
    u'ChatServerHost': u'hpe-voicevm-76.genesyslab.com',
    u'RTargetObjSelDBID': u'',
    u'ChatServerPort': u'7160',
    u'CBR-Interaction_cost': u'',
    u'CBR-contract_DBIDs': u'',
    u'RTargetAgentSelected': u'Agent1',
    u'CBR-IT-path_DBIDs': u'',
    u'RTargetRuleSelected': u'',
    u'RTargetPlaceSelected': u'Agent1',
    u'CBR-actual_volume': u'',
    u'RTenant': u'Environment',
    u'ChatServerDBID': u'155',
    u'RRequestedSkills': None,
    u'RRequestedSkillCombination': u'',
    u'RVQDBID': u'',
    u'RStrategyDBID': u'125',
    u'CustomerSegment': u'default',
    u'PegAG?: 2>1': 1,
    u'ServiceType': u'default',
    u'ServiceObjective': 0,
    u'RTargetRequested': u'?': 2>1',
    u'RTargetAgentGroup': u'?': 2>1',
    u'RStrategyName': u'SimpleChatInStrategy',
  }

```



```

        u'Subject': u'Customersupport'
    },
    u'currentQueue': u'SimpleChatInQueue',
    u'interactionSubType': u'InboundNew',
    u'inQueues': {
        u'SimpleChatTranscriptQueue': u''
    },
    u'capabilities': [
        u'Transfer',
        u'PlaceInQueue',
        u'StopProcessing',
        u'PlaceInWorkbin',
        u'UpdateProperties',
        u'Reply'
    ],
    u'state': u'Accepted',
    u'messageType': u'InteractionStateMessage',
    u'sessionId': u'0000Ja8V814907KS',
    u'id': u'0000Ja8V814907KS',
    u'channel': u'chat',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'16',
    u'channel': u'/meta/connect'
}
]

```

Chat Notifications

```

[
    {
        u'data': {
            u'chat_notifications': [
                {
                    u'EventId': 3,
                    u'NoticeText': None,
                    u'NoticeType': None,
                    u'MessageText': u'testmessagefromclient',
                    u'UserId': u'009B51CDF65818CD',
                    u'MessageType': u'NEW_MESSAGE',
                    u'UserType': None,
                    u'Visibility': u'All',
                    u'NickName': None
                }
            ],
            u'userData': {
                u'IdentifyCreateContact': u'3',
                u'RTargetTypeSelected': u'2',
                u'RTargetObjectSelected': u'?>1',
                u'RVQID': u'',
            }
        }
    }
]


```

```

        u'ChatServerAppName': u'es_chat',
        u'ChatServerHost': u'hpe-voicevm-76.genesyslab.com',
        u'RTargetObjSelDBID': u'',
        u'ChatServerPort': u'7160',
        u'CBR-Interaction_cost': u'',
        u'CBR-contract_DBIDs': u'',
        u'RTargetAgentSelected': u'Agent1',
        u'CBR-IT-path_DBIDs': u'',
        u'RTargetRuleSelected': u'',
        u'RTargetPlaceSelected': u'Agent1',
        u'CBR-actual_volume': u'',
        u'RTenant': u'Environment',
        u'ChatServerDBID': u'155',
        u'RRequestedSkills': None,
        u'RRequestedSkillCombination': u'',
        u'RVQDBID': u'',
        u'RStrategyDBID': u'125',
        u'CustomerSegment': u'default',
        u'PegAG?: 2>1': 1,
        u'ServiceType': u'default',
        u'ServiceObjective': 0,
        u'RTargetRequested': u'?: 2>1',
        u'RTargetAgentGroup': u'?: 2>1',
        u'RStrategyName': u'SimpleChatInStrategy',
        u'Subject': u'Customersupport'
    },
    u'currentQueue': u'SimpleChatInQueue',
    u'interactionSubType': u'InboundNew',
    u'inQueues': {
        u'SimpleChatTranscriptQueue': u''
    },
    u'capabilities': [
        u'Transfer',
        u'PlaceInQueue',
        u'StopProcessing',
        u'PlaceInWorkbin',
        u'UpdateProperties',
        u'Reply'
    ],
    u'state': u'Accepted',
    u'messageType': u'InteractionStateMessage',
    u'sessionId': u'0000Ja8V814907KU',
    u'id': u'0000Ja8V814907KU',
    u'channel': u'chat',
    u'interactionType': u'Inbound'
},
u'channel': u'/v2/me/interactions'
},
{
    u'successful': True,
    u'advice': {
        u'interval': 0,
        u'timeout': 30000,
        u'reconnect': u'retry'
    },
    u'id': u'17',
    u'channel': u'/meta/connect'
}
]

```

Attachments API

 **Pre-Release Notice:** The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

A Web Services attachment is the same as an e-mail attachment. In PSDK terms, an attachment is simply a document. Attachments are associated with one Interaction, while one Interaction can own many Attachments.

Important

These operations are not for AJAX, but should be performed as regular HTTP requests.

Operations

The following operations are supported on **/me/interactions/{id}/attachments**:

HTTP Operation	Description	Output	Permissions
POST	Creates a new Attachment and associates it with an Interaction. Note: The file should be bound to the 'attachment' variable.	<code>{"statusCode":0,"documentId":"00009a8T8Y13002S"}</code>	Agent

The following operations are supported on **/me/interactions/{id}/attachments/{attachmentId}**:

HTTP Operation	Description	Output	Permissions
GET	Returns an Attachment, given an attachmentId (documentId).	Output will be byte content of the attachment, with MimeType set accordingly	Agent
DELETE	Deletes an Attachment, given an attachmentId (documentId).	<code>{"statusCode":0}</code>	Agent

Users API

Attributes

Attribute	Type	Description	Access Level
userName	String	The user's e-mail address.	GET, POST, DELETE
password	String	The user's password (if allowed).	POST, PUT
firstName	String	The user's first name.	GET, POST, PUT
lastName	String	The user's last name.	GET, POST, PUT
roles	Array of Strings	User roles. Currently 'ROLE_ADMIN' and/or 'ROLE_AGENT'. Note that on PUT, the roles specified will be the new role set. It is not a "diff" operation.	GET, POST, PUT
skills	Array of Skill objects	The list of skills that are assigned to a user.	GET (subresources=* must be specified)
devices	Array of Device objects	The list of devices that are assigned to a user.	GET (subresources=* must be specified)
doNotDisturb	FlagV2 Enum	Denotes whether Do Not Disturb status is on or off for this user.	GET
changePasswordOnFirstLogin	Boolean	Possible values are True and False. If set to True, the User will be required to change their password when they log in. Default value is False.	GET, POST, DELETE

Operations

The following operations are supported for the **/users** resource:

Operation	Description	Permissions	Arguments
GET	Returns URIs for all agents in the contact center.	<ul style="list-style-type: none"> Contact Center Admin 	<ul style="list-style-type: none"> fields=*: Will return all fields for every

Operation	Description	Permissions	Arguments
		<ul style="list-style-type: none"> Agent 	user instead of URI <ul style="list-style-type: none"> subresources=*: Will return all fields and all subresources (for example, devices, skills) for every user.
POST	Creates a user.	Contact Center Admin	

The following operations are supported for the **/users/{id}** resource:

Operation	Description	Permissions	Arguments
GET	Returns the user specified by the ID.	<ul style="list-style-type: none"> Contact Center Admin Agent 	subresources=*: Will return all subresources (for example, devices, skills) assigned to the given user.
POST	Updates the user specified by the ID.	<ul style="list-style-type: none"> Contact Center Admin Agent 	
DELETE	Deletes the user specified by the ID.	<ul style="list-style-type: none"> Contact Center Admin Agent 	

Skills API

Attributes

The following attributes are supported for Skill objects.

Attribute	Type	Description	Access level
name	String	The name of the skill.	POST, GET
description	String	A detailed description.	POST, GET

Operations

Read contact center skills

Returns uris of all skills of current contact center.

Method

GET

Request mapping

.../api/v2/skills

Permissions

Contact center administrator

Examples

Request:

GET .../api/v2/skills

Response:

```
{
  "statusCode": 0,
  "uris":
  [
```

```

    "http://localhost:8080/api/v2/skills/8e8fbe93-a8e5-4997-a939-0e245c30f97c",
    "http://localhost:8080/api/v2/skills/9d81193d-9f8a-4bce-9e58-1f586505328d"
  ]
}

```

If detail information is demanded, request parameter **?fields=*** should be used. Notice that the "description" field will be absent in the response if it wasn't created for the skill.

Request:

```
GET .../api/v2/skills?fields=*
```

Response:

```

{
  "statusCode": 0,
  "skills":
  [
    {
      "id": "8e8fbe93-a8e5-4997-a939-0e245c30f97c",
      "name": "Might",
      "description": "Brandishing of heavy and sharp long iron bars.",
      "level": "0"
    },
    {
      "id": "9d81193d-9f8a-4bce-9e58-1f586505328d",
      "name": "Magic",
      "description": "Declaiming of strange phrases and making funny gestures to cast
fireballs and etc.",
      "level": "0"
    }
  ]
}

```

Read a specified skill

Read the specified skill.

Method

```
GET
```

Request mapping

```
.../api/v2/skills/{id}
```

Permissions

Contact center administrator

Examples

Request:

GET .../api/v2/skills/<skill_id>

Response:

```
{
  "statusCode":0
}
```

Read a user skill

Reads the specified skill of the specified user.

Method

GET

Request mapping

.../api/v2/users/{user_id}/skills/{skill_id}

Permissions

Contact center administrator

Examples

Request:

GET .../api/v2/users/{user_id}/skills/{skill_id}

Response:

```
{
  "statusCode":0
}
```

Read user skills

Reads all skills of the specified user.

Method

GET

Request mapping

.../api/v2/users/{user_id}/skills

Permissions

Contact center administrator

Examples

Request:

```
GET .../api/v2/users/{user_id}/skills
```

Response:

```
{
  "statusCode":0
}
```

Read a skill of the current user

Reads the specified skill of the current user.

Method

GET

Request mapping

```
.../api/v2/me/skills/{skill_id}
```

Permissions

- Contact center administrator
- Contact center agent

Examples

Request:

```
GET .../api/v2/me/skills/{skill_id}
```

Response:

```
{
  "statusCode":0
}
```

Read all skills of the current user

Reads all skills of the current user.

Method

GET

Request mapping

.../api/v2/me/skills

Permissions

- Contact center administrator
- Contact center agent

Examples

Request:

GET .../api/v2/me/skills

Response:

```
{
  "statusCode":0
}
```

Devices API

Operations

The following operations are available for the **/devices** URI:

Operation	Description	Permissions
GET	Retrieves a list of all devices for the specified grouping (for example, /users/{id}/devices will retrieve a list of devices assigned to the specified user).	<ul style="list-style-type: none">• Contact Center Admin• Agent (only for objects owned by this agent)

In addition, the following operations are supported for individual devices specified by the **/devices/{id}** URI:

Operation	Description	Permissions
GET	Retrieves information about the specified device.	<ul style="list-style-type: none">• Contact Center Admin• Agent (only for objects owned by this agent)

List Devices

The devices can be listed with the following groupings:

- all devices in contact center. The path to be used is **.../api/v2/devices** and the Contact Center Admin role is required.
- all devices assigned to given user. The path to be used is **..api/v2/users/{userid}/devices** and the Contact Center Admin role is required.
- all devices assigned to current user. The path to be used is **...api/v2/me/devices** and the Contact Center Agent or Login role is required.

The request can return the list of device URIs or the list of devices. To list URIs:

```
GET .../devices
return sample:
{
  "statusCode":0,
  "uris":[
    "http://172.21.16.123:8080/api/v2/devices/0c754c1f-7a65-4a7f-9a2b-c5bb5c983653",
    "http://172.21.16.123:8080/api/v2/devices/406ff680-63c8-4eeb-8c23-f39f538059d1",
    "http://172.21.16.123:8080/api/v2/devices/d6186ffc-af3d-4925-835e-a9e8dad98051"
  ]
}
```

```
}

```

To list devices, use the **fields=*** query parameter:

```
GET .../devices?fields=*
```

```
return sample:
```

```
{
  "statusCode":0,
  "devices":[
    {
      "id":"0c754c1f-7a65-4a7f-9a2b-c5bb5c983653",
      "deviceState":"Inactive",
      "phoneNumberUri":"http://172.21.16.123:8080/api/v2/phone-numbers/
6658d431-9195-4184-a462-b8a6fd901de8",
      "telephonyNetwork":"Public",
      "model":"CloudDevice",
      "vendor":"Genesys",
      "phoneNumber":"+16509870000",
      "country":{"
        "name":"United States",
        "code":"US",
        "callingCode":"1"
      },
      "location":"California",
      "e164Number":"+16509870000",
      "voiceEnvironmentUri":"http://172.21.16.123:8080/api/v2/voice-environments/
b40eb01e-199d-4db3-a91f-a0cf104ab2bf"
    },
    {
      "id":"1fd21300-303d-401f-9b4c-0d9c2801433e",
      "deviceState":"Inactive",
      "phoneNumberUri":"http://172.21.16.123:8080/api/v2/phone-numbers/
812a7b7b-0e5b-4025-9443-79363b85773c",
      "telephonyNetwork":"Public",
      "model":"CloudDevice",
      "vendor":"Genesys",
      "phoneNumber":"3001",
      "country":{"
        "name":"",
        "code":"",
        "callingCode":""
      },
      "e164Number":"3001",
      "voiceEnvironmentUri":"http://172.21.16.123:8080/api/v2/voice-environments/
b40eb01e-199d-4db3-a91f-a0cf104ab2bf"
    },
    ...
  ]
}
```

Attributes

Attribute	Type	Description	Access Level
model	String	Optional. The device model.	GET, POST

Attribute	Type	Description	Access Level
phoneNumber	String	Mandatory. The phone number assigned to this device.	GET, POST
userState	AgentState	The current contact center state of the user to whom the device is assigned represented by an agent state object as described in Agent_States. link to be updated	GET
country	String	The phone number country information: "country":{"name":"United States","code":"US","callingCode":"1"}	GET
location	String	The geographical location of this phone number.	GET
localNumber	String	The phone number in the format of its home country.	GET
e164number	String	The phone number in E.164 format.	GET
doNotDisturb	String	The do not disturb state on this device. Valid values are "On" and "Off".	GET
forwardTo	String	The number to which calls are forwarded (if forwarding is enabled).	GET
capabilities	String Array	<p>A list of operations currently available on the device. Possible values for users with role ROLE_AGENT:</p> <p>DoNotDisturbOn, DoNotDisturbOff, ForwardCallsOn, ForwardCallsOff</p> <p>In addition, users who have the role ROLE_SUPERVISOR may see one or more of the following:</p> <p>ListenIn, Bargeln, Coach, CancelSupervisorMonitoring</p> <p>The exact combination depends on the particular contact center configuration as well as the current supervisorMonitoringState</p>	GET

Attribute	Type	Description	Access Level
		and supervisorMonitoringMode (see below) set for the device. In addition, if another supervisor is already monitoring the device, these functions will not be available as only one supervisor at a time may monitor a given device.	
supervisorMonitoringState	String	The current supervisor monitoring state on this device. Possible values are "ListenIn", "Bargeln", and "Coach". If monitoring is not in progress, this attribute is not present in the response.	GET
supervisorMonitoringMode	String	Possible values: NextCall—Supervisor will monitor the next call to arrive on this device. AllCalls—Supervisor will monitor all calls on this device	GET
supervisorMonitoringScopeString		Call: Calls arriving on this device will be monitored by the supervisor from beginning to end, even when the call is transferred Agent: Only calls handled by this agent are monitored. If the call is transferred, the supervisor will no longer be monitoring it.	GET
telephonyNetwork	String	This property denotes the type of telephony network the device is associated with. Valid values are "Public" or "Private". Public—this would be set for a device that is connected over the PSTN via SIP Server. Private—other deployment scenarios such as local endpoints or IP phones connected to SIP Server or PBX hardphones connected to Avaya.	GET, POST

Attribute	Type	Description	Access Level
		If no value is provided for this property on device creation, the default value "Private" is assigned.	

Contacts API

Operations

The following operations are available for the **/contacts** URI:

Operation	Description	Permissions
GET	Retrieves a list of all contacts in the system. Currently this includes queues, users, as well as any "custom" contacts added via the provisioning API.	<ul style="list-style-type: none">Contact Center AdminAgent
POST	Creates a new contact of "custom" type.	Contact Center Admin
PUT	Modifies a contact of "custom" type (other types are read-only).	Contact Center Admin
DELETE	Removes a contact of "custom" type.	Contact Center Admin

Attributes

The following attributes are available for the contact object. Note that the presence or absence of each attribute is determined by resource type (as described in the table below). Note also that any of the attributes can be used to filter the results of the query and can be specified together in a logical "AND" operation. For instance, GET /contacts&type=User will return all users. GET /contacts?firstName=Joe&lastName=Johnson&userName=joej will return all contacts named "Joe Johnson" with the user name "joej."

Attribute	Type	Description	Access Level	Resource Type
id	String	The unique ID of the contact record.	GET	Custom, Queue, User
name	String	The name of this contact (for example, this could be the name of a queue or the first name/last name of a user).	GET for all PUT, POST, DELETE for contacts of type "Custom" only	Custom, Queue, User
type	String	Possible values are: Custom, Queue, User	GET	Custom, Queue, User
phoneNumber	String	The phone number	GET for all	Custom, Queue,

Attribute	Type	Description	Access Level	Resource Type
		at which this contact can be reached.	PUT, POST, DELETE for contacts of type "Custom" only	User
userName	String	The contact's user name.	GET	User
firstName	String	The contact's first name.	GET	User
lastName	String	The contact's last name.	GET	User
employeeId	String	The contact's employee ID.	GET	User

Example

The following creates a new "custom" contact:

```
POST /contacts
{
  "name": "Some Contact"
  "type": "Custom"
  "phoneNumber": "xxx-xxx-xxxx"
}
```

The following retrieves only "custom" contacts:

```
GET /contacts?type=custom
{
  contacts:[
    {
      "id": "Some unique id",
      "name": "Some Contact",
      "type": "Custom",
      "phoneNumber": "xxx-xxx-xxxx"
    }, etc
  ]
}
```

Calls API

Operations

The following operations are available for the **/me/calls** URI:

Operation	Description	Permissions
GET	Retrieves a list of all calls visible to the current user.	<ul style="list-style-type: none">• Contact Center Admin• Agent (only for objects owned by this agent)

In addition, the following operations are supported for individual calls specified by the **/me/calls/{id}** URI:

Operation	Description	Permissions
GET	Retrieves information about the specified call.	<ul style="list-style-type: none">• Contact Center Admin• Agent (only for objects owned by this agent)

Attributes

The following attributes will be present for "voice" channel interactions:

Attribute	Type	Description	Access Level
state	String	The current call state (Dialing, Held, Established, and so on).	GET
capabilities	Array of Strings	A list of capabilities at the current state. For example, if the current state is "Dialing", the list might be ["HangUp", "Hold"].	GET
deviceUri	URI	Link to the device for which this state is applicable.	GET
participants	Array of Strings	A collection of participants.	GET

Attribute	Type	Description	Access Level
ani	String	If this is an external call, the ani attribute will hold the number from which this call originated.	GET
participantsInfo	Array of Objects	<p>If the participant number contains +country_code, this will contain a list of structures, each of which will be:</p> <p>"country"—country information of the phone number in the JSON format:</p> <pre>"country":{"name":"United States","code":"US","callingCode":"1"}</pre> <p>"location"—region of the phone number.</p> <p>"localPhoneNumber"—phone number in domestic format.</p> <p>"E164digits"—phone number in e.164 format.</p>	GET
mute	Off	Shows whether the call is muted.	GET

Asynchronous Operations

Telephony operations do not quite fit into the REST model. A telephony state change is not just a synchronous resource modification but instead is a message to the Genesys system to take some action. Consequently, telephony operations are done in a way that is somewhat different from the rest of the API. Instead of a “PUT” update to the resource (for example, PUT new call state) we will POST an operation object to the URI of the resource on which it is being performed. To determine which operations are available at any given time in the call's lifecycle the "capabilities" property of the call may be checked. This property holds a collection of operation names that may be performed during the call's current state. The result of each asynchronous operation will be provided via CometD when it becomes available.

For instance, the below operation will release an interaction of type "call" identified by "id":

```
POST /me/call/{id}
{"operationName": "Hangup"}
```

The following topics provide information about the possible operations and their parameters.

- [Change Password](#)
- [Presence](#)
- [Call Control](#)
- [Device and Channel Management](#)
- [Notifications](#)

Change Password

Operation	Attributes	Description	Valid URIs
ChangePassword	<p>oldPassword—the user's old password, encoded using base64.</p> <p>newPassword—the user's new password, encoded using base64.</p>	Verifies that the value of "oldPassword" matches the logged in user's existing password and updates it to the value of "newPassword." Configuration Server will be responsible for applying the password rules.	<p>/me</p> <p>/users/{id} (Supervisors and Admins only)</p>

Request parameters

Following request parameters are available for the "ChangePassword" operation.

Parameter	Type	Description	Access Level
firstLogin	String	Allow user to change own password, when he/she tries to login first time after user's creation or changing of user's password by an administrator. Valid values: true false Default value: false	POST

Examples

The following changes password of a user.

Request:

```
POST api/v2/me
{
  "operationName": "ChangePassword",
  "oldPassword": "MQ==",
  "newPassword": "Mg=="
}
```

Response:

```
{
  "statusCode": 0
}
```

The following changes password, when a user makes first login.

Request:

```
POST api/v2/me?firstLogin=true
```

```
{
  "operationName": "ChangePassword",
  "oldPassword": "MQ==",
  "newPassword": "Mg=="
}
```

Response:

```
{
  "statusCode": 0,
}
```

Reset Password

Operation	Attributes	Description	Valid URIs
ResetPassword	newPassword: The user's new password, encoded using base64	Updates it to the value of "newPassword." Configuration Server will be responsible for applying the password rules.	/users/{id} (admins only!)

Examples

Request:

```
POST ...api/v2/users/5804abe284374b5bb0a18f57f5c9043d
{
  "operationName": "ResetPassword",
  "newPassword": "MQ=="
}
```

Response:

```
{
  "statusCode": 0
}
```

Presence

Operation	Attributes	Description	Valid URIs
StartContactCenterSession	<p>place—optional attribute that specifies the place name in Configuration Server.</p> <p>loginCode—optional attribute that specifies the agent login to use in the operation.</p> <p>queue—optional attribute that specifies the queue number.</p> <p>channels—string array that specifies a set of channels to use for the operation (for example, "voice").</p>	<p>This operation will perform the following steps:</p> <ol style="list-style-type: none"> 1. Find the first device associated with the DN on the specified place. 2. If any device is currently assigned to another user, return an error. 3. Assign all devices from the place to the current user 4. If 'loginCode' is not specified, select an agent login as follows: <ol style="list-style-type: none"> a. For each device, find the associated switch. b. For currently logged in users, find a list of all agent logins (in Configuration Server). c. For the first device, select the first login from the above list that is assigned to the same switch as the device. 5. For the first device, send a RequestAgentLogin operation to SIP Server with either the selected or the specified code. Include queue if specified. 	/me
EndContactCenterSession	n/a	<p>For each device:</p> <ol style="list-style-type: none"> 1. Send RequestAgentLogout. 2. Unassign device from user. 3. Invalidate the HTTP session. 	/me

Call Control

The following call control operations are available:

Operation	Required Attributes	Description	Valid URIs
Dial	destination —the phone number to dial.	Creates a new phone call.	/devices/{id}/calls —the device on which to create the phone call.
Answer	n/a	Answers a phone call.	/me/calls/{id} —an interaction of type "call".
Hangup	n/a	Releases a phone call.	/me/calls/{id} —an interaction of type "call".
Hold	n/a	Places the phone call on hold.	/me/calls/{id} —an interaction of type "call".
Retrieve	n/a	Retrieves the phone call from hold.	/me/calls/{id} —an interaction of type "call".
SingleStepTransfer	destination —the number to transfer to.	Transfers the phone call to the specified number.	/me/calls/{id} —an interaction of type "call".
SingleStepConference	destination —the number of the party to add to conference.	Initiates conference with the specified number.	/me/calls/{id} —an interaction of type "call".
InitiateConference	destination —the number of the party to add to conference. 'destination': {'phoneNumber': 'XXX'}	Initiates a conference with specified destination.	/me/calls/{id} —an interaction of type "call".
CompleteConference	consultCallUri	Completes the conference and adds consult call to it.consult call.	/me/calls/{id} —an interaction of type "call".
SendDtmf	digits —a string where each character is a digit 0..9. A dial tone will be produced for each of the digits.	Sends request to produce dial tones for the specified digits.	/me/calls/{id} —an interaction of type "call".
RemoveParticipantFromConference	participant —the phone number of the participant to remove from the conference.	Removes the specified participant from the conference. This will result in a "ParticipantsUpdated" notification which will contain the updated list	/me/calls/{id} —an interaction of type "call".

Operation	Required Attributes	Description	Valid URIs
		of participants.	
MuteCall		Mutes the call specified by the URI.	/me/calls/{id} —an interaction of type "call".
UnmuteCall		Unmutes the call specified by the URI.	/me/calls/{id} —an interaction of type "call".
SwapCalls	otherCallUri —the URI of the call to retrieve.	Places the call on which this operation is called on hold and retrieves the call specified by otherCallUri.	/me/calls/{id} —an interaction of type "call".
MergeWithOtherCall	otherCallUri —the URI of the call to merge with.	Merges the call on which this operation is called with the call specified by otherCallUri.	/me/calls/{id} —an interaction of type "call".

Device/Channel Management

Operation	Attributes	Description	Valid URIs
DoNotDisturbOn		Sets "do not disturb" on.	<p>/me—set "do not disturb" for all of a user's devices and channels.</p> <p>/devices/{id}—set "do not disturb" on a specific voice device.</p> <p>/channels/{id}—set "do not disturb" on a specific channel.</p>
DoNotDisturbOff		Sets "do not disturb" off	<p>/me—turn off "do not disturb" for all of a user's devices and channels.</p> <p>/devices/{id}—turn off "do not disturb" on a specific voice device.</p> <p>/channels/{id}—turn off "do not disturb" on a specific channel.</p>
ForwardCallsOn	destination —the number to which incoming calls should be forwarded.	Starts forwarding incoming calls to the specified number.	/devices/{id} —the device that should start forwarding calls.
ForwardCallsOff		Stops forwarding incoming calls.	/devices/{id} —a device that's currently forwarding calls.

Asynchronous Telephony Operations Notifications

User

/me

Currently the following notification will be available via the **/v2/me** topic:

Attribute	Description	Applies to notification type
notificationType	The type of notification that has been received. Possible values: <i>PasswordChangeSuccessful</i> : A password change operation has been executed successfully for the current user. User should use the new password upon next login. <i>PasswordChangeError</i> : An error occurred while changing the user's password	always present
errorMessage	An error message describing why the password change failed	PasswordChangeError
errorCode	The error code describing the reason for failure. Possible values are: 1: Password does not meet security guidelines 2: Did not enter old password correctly 3: Internal server error	PasswordChangeError

/users

NOTE: This topic is only available to supervisor and administrator users.

Currently the following notification will be available via the **/v2/users** topic:

Attribute	Description	Applies to notification type
notificationType	The type of notification that has been received. Possible values: <i>PasswordChangeSuccessful</i> : A password change operation has been executed successfully for the referenced user. User should use the new password upon next login. <i>PasswordChangeError</i> : An error occurred while changing the referenced user's password	always present
errorMessage	An error message describing why the password change failed	PasswordChangeError
errorCode	The error code describing the reason for failure. Possible values are: 1: Password does not meet security guidelines 2: Did not enter old password correctly 3: Internal server error	PasswordChangeError
userUri	The uri identifying the user for whom this notification is received	always present
userId	The id identifying the user for whom this notification is received.	always present

Calls

The following notifications are available via the **/v2/me/calls** topic. This topic should be used to subscribe to call notifications. A call notification will have the following json attributes:

Attribute	Description
call	An object describing the current call. See the voice interactions section for a list of its attributes.
phoneNumber	The phone number digits assigned to the device on which the call is active.
notificationType	Describes the reason the for the current notification. Can have one of the following values: StatusChange—the call's status has changed.

Attribute	Description
	ParticipantsUpdated—the participants list for the current call have been updated.
	AttachedDataChanged—the list of attached data has changed.
	DtmfSent—DTMF digits have been successfully sent for this call.

In addition, if an error occurs during an attempted call control operation, the following attributes will be present:

Attribute	Description
deviceUri	The URI for the device on which the call is active.
errorMessage	The error message describing the error that occurred.
connectionId	Genesys ConnID
callUuid	Genesys UUID

The following example shows the structure of a **CallStateChangedMessage**:

```
{
  "data": {
    "messageType": "CallStateChangeMessage",
    "call": {
      "id": "010a0229f0cbd001",
      "state": "Dialing",
      "callUuid": "RC5FN2QK1H3IFD3CV136FELT80000001",
      "deviceUri": "http://127.0.0.1:8080/api/v2/devices/5aceae61-0b5c-4578-9477-cd352d43cb17",
      "uri": "http://127.0.0.1:8080/api/v2/me/calls/010a0229f0cbd001",
      "participants": [
        "2"
      ],
      "participantsInfo": [
        {
          "location": null,
          "country": {
            "name": "",
            "code": "",
            "callingCode": ""
          },
          "formattedPhoneNumber": "2",
          "isValidNumber": false,
          "E164digits": "2"
        }
      ],
      "ani": "1",
      "capabilities": [
        "DeleteUserDataPair",
        "UpdateUserData",
        "AttachUserData",
        "Hangup",
        "DeleteUserData",
        "SendDtmf"
      ],
      "duration": "0",
    }
  }
}
```

```

    "mute": "Off"
  },
  "phoneNumber": "1",
  "notificationType": "StatusChange"
},
"channel": "/v2/me/calls"
}

```

Devices

The topic **/v2/me/devices** should be used to monitor the availability of a user's devices. The following json attributes will be present:

Attribute	Description
id	The ID of the device that is changing state.
deviceState	The state of the device. Either Active, or Inactive.
userState	A structure describing the current user state for the device. This will include the state (for example, Ready/NotReady), workMode, and reason (if applicable). If the combination of state, workMode, and reason match an agent state definition, the id and displayName of that agent state definition will also be included for convenience.
doNotDisturb	Off
forwardTo	If forwarding calls, the number to which calls will be forwarded.

The example below shows the structure and attribution of a **DeviceStateChangeMessage**:

```

{
  "data": {
    "messageType": "DeviceStateChangeMessage",
    "devices": [
      {
        "deviceState": "Active",
        "phoneNumberUri": "http://127.0.0.1:8080/api/v2/phone-numbers/2b36d336-d502-4e46-bc92-b727913a7754",
        "id": "5aceae61-0b5c-4578-9477-cd352d43cb17",
        "userState": {
          "id": "9430250E-0A1B-421F-B372-F29E69366DED",
          "displayName": "Ready",
          "state": "Ready"
        },
        "type": "PSTNPhone",
        "model": "FakeDeviceModel",
        "vendor": "FakeDeviceVendor",
        "phoneNumber": "1",
        "country": {
          "name": "",
          "code": "",
          "callingCode": ""
        },
        "e164Number": "1"
      }
    ]
  }
}

```

```
    }  
  ],  
  },  
  "channel": "/v2/me/devices"  
}
```


Workbins



Pre-Release Notice: The information contained on this page is not considered final and is managed under the terms and conditions found in the Pre-release Agreement. This page provides the most up-to-date reference information available for this pre-release version and is restricted for use by those who have signed the Pre-release Agreement with Genesys to acquire an early version of the software.

The following operations are allowed on **/me/workbins**

Operation Name	HTTP Operation	Description	Permissions
GetWorkbinTypesInfo	GET	Returns a list of workbins and the information associated with the workbins.	Agent

GetWorkbinTypesInfo

Input

None

Output

Success

```
{
  "status": 0,
  "workbins": [
    {
      u'workbinId': u'RmFjZWJvb2sgV29ya2JpbiBJblByb2dyZXNzL1ByaXZhdGVWawV3',
      u'active': 1,
      u'workbinName': u'FacebookWorkbinInProgress',
      u'viewName': u'FacebookWorkbinInProgress/PrivateView',
      u'workbinType': 1
    },
    {
      u'workbinId': u'VHdpdHRlciBXb3JrYmIuIERyYWZ0L1ByaXZhdGVWawV3',
      u'active': 1,
      u'workbinName': u'TwitterWorkbinDraft',
      u'viewName': u'TwitterWorkbinDraft/PrivateView',
      u'workbinType': 1
    },
    {
      u'workbinId': u'VHdpdHRlciBXb3JrYmIuIEluUHJvZ3Jlc3MvUHJpdmF0ZVZpZXc=',
      u'active': 1,
      u'workbinName': u'TwitterWorkbinInProgress',
      u'viewName': u'TwitterWorkbinInProgress/PrivateView',
      u'workbinType': 1
    },
    {
      u'workbinId': u'VGVzdFdvcmthW4vUHJpdmF0ZVZpZXc=',
      u'active': 1,

```

```

    u'workbinName': u'TestWorkbin',
    u'viewName': u'TestWorkbin/PrivateView',
    u'workbinType': 1
  },
  {
    u'workbinId': u'RmFjZWJvb2sgV29ya2JpbiBEcmFmdC9Qcm12YXRlVmlldw==',
    u'active': 1,
    u'workbinName': u'FacebookWorkbinDraft',
    u'viewName': u'FacebookWorkbinDraft/PrivateView',
    u'workbinType': 1
  }
]
}

```

Failure

```

{
  "status":integer above 0
  "description":error-code
}

```

Operations

In addition, the following operations are allowed:

Operation Name	HTTP Operation	Description	Permissions
GetWorkbinContent	GET	Returns workbin information and content list.	Agent
AddInteractionToWorkbin	POST	Adds an interaction to the specified workbin.	Agent
PullInteraction	POST	Pulls an interaction from a workbin.	Agent
SubscribeWorkbinEvents	POST	Subscribes the agent to listen to events from the workbin.	Agent
UnsubscribeWorkbinEvents	POST	Unsubscribes the agent, so he or she stops receiving events from the workbin.	Agent

GetWorkbinContent

Returns the content of the specified workbin. URI: **/me/workbins/{workbinid}/interactions?fields=***

Input

None

Output

Success

```
{
  "statusCode": 0,
  "interactions": [
    {
      "InQueues": {
        "__STOP__": ""
      },
      "RRequestedSkillCombination": "",
      "MediaType": "email",
      "InteractionId": "0000Aa8WRME43VG6",
      "RTargetAgentGroup": "?:2>1",
      "ServiceType": "default",
      "IsHeld": 0,
      "IsOnline": 0,
      "SubmitSeq": "41895540",
      "RTenant": "Environment",
      "DeliveredAt": "2013-07-20T00:41:33Z",
      "_ContainsAttachment": "false",
      "RVQID": "",
      "_AttachmentsSize": "0",
      "PegAG?:2>1": 2,
      "SubmittedBy": "es_esj",
      "Header_Date": "Sat, 20 Jul 2013 02:02:57 +0300",
      "EmailAddress": "qwerty@mcr812trywe",
      "Header_Message-ID": "<r4mo20o0o6jlniq.190720131550@mcr812trywe>",
      "RTargetRequested": "?:2>1",
      "Queue": "Facebook Workbin InProgress/PrivateQueue",
      "InteractionType": "Inbound",
      "MovedToQueueAt": "2013-07-20T00:41:33Z",
      "_AutoReplyCount": 0,
      "PlacedInQueueAt": "2013-07-20T00:41:33Z",
      "IsLocked": 0,
      "CustomerSegment": "default",
      "RTargetRuleSelected": "",
      "RTargetPlaceSelected": "MaratTest",
      "To": "saas54@mcr812trywe",
      "ReceivedAt": "2013-07-19T23:03:02Z",
      "RTargetObjSelDBID": "",
      "RTargetAgentSelected": "MaratTest",
      "CBR-contract_DBIDs": "",
      "InteractionSubtype": "InboundNew",
      "Header_Content-Type": "multipart/mixed; boundary=\"=====0302739529==\\\"",
      "CBR-IT-path_DBIDs": "",
      "CBR-actual_volume": "",
      "Workbin": "Facebook Workbin InProgress",
      "RRequestedSkills": null,
      "OutQueues": {
        "SimpleEmailOutQueue": ""
      },
      "RTargetObjectSelected": "?:2>1",
      "AssignedAt": "2013-07-20T00:41:33Z",
      "_AttachmentFileNames": "",
      "Mailbox": "saas54",
      "TenantId": 1,
      "PegDEF": 188,
      "WorkbinAgentId": "MaratTest",
      "FirstName": "qwerty",
      "RVQDBID": ""
    }
  ]
}
```

```

    "RStrategyName": "SimpleEmailInStrategy",
    "RTargetTypeSelected": "2",
    "SubmittedAt": "2013-07-19T23:03:16Z",
    "Subject": "test_receive_invite_and_accept_updateproperites",
    "ServiceObjective": 0,
    "AssignedTo": "MaratTest",
    "Origination_Source": "Email",
    "PlaceInQueueSeq": "41904672",
    "CBR-Interaction_cost": "",
    "Header_MIME-Version": "1.0",
    "FromAddress": "qwerty@mcr812trywe",
    "ContactId": "000001a8QWK4A003P",
    "RStrategyDBID": "188",
    "FromPersonal": "",
    "InteractionState": 0
  }
}

```

Failure

```

{
  "statusCode": an integer value above 0,
  "statusMessage": details
}

```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

AddInteractionToWorkbin

Adds interactions to a workbin. URI: **/me/workbins/{workbinId}/interactions**

Input

```

{
  "interactionId": id
}

```

Output

Success

```

{
  "status": 0
  uri: "http://host:port/api/v2/me/workbins/{workbinId}/interactions/{interactionid}"
}

```

Failure

```

{
  "statusCode": an integer value above 0,
  "statusMessage": details
}

```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values

section.

PullInteraction

This request will pull an interaction from a workbin and assign the interaction to the user. URI: **/me/interactions** Upon successful execution, a cometD message will be returned with interaction state "Pulled".

Input

```
{
  "operationName": "PullInteractionFromWorkbin"
  "uri": "http://host:port/api/v2/me/workbins/{workbinId}/interactions/{interactionid}"
}
```

Output

Success

```
{
  "status": 0
}
```

Failure

```
{
  "statusCode": an integer value above 0,
  "statusMessage": details
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

SubscribeWorkbinEvents

This request will subscribe the agent to events from the workbin.

Input

```
{
  "operationName": "Subscribe"
}
```

Output

Success

```
{
  "status": 0
}
```

Failure

```
{
  "statusCode": an integer value above 0,
  "statusMessage": details
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.

CometD Notifications

Topic: **/v2/me/workbins**

Message:

```
{
  u'messageType': u'workbinNotificationResource',
  u'operation': u'Create',
  u'workbinContentOperation': u'PlacedIn',
  u'interaction': {
    u'userData': {
      u'Sublist': {
        u'subkey': u'subkeyvalue'
      }
    },
    u'currentQueue': u'TestWorkbin/PrivateQueue',
    u'interactionSubtType': u'InboundNew',
    u'state': u'Queued',
    u'parentId': u,
    u'id': u'0000Ba8Y08X9001E',
    u'channel': u'email',
    u'interactionType': u'Inbound'
  },
  u'workbinInfo': {
    u'workbinTypeId': u'TestWorkbin',
    u'workbinGroupId': None,
    u'workbinPlaceGroupId': None,
    u'workbinAgentId': u'Agen1',
    u'workbinPlaceId': None
  },
  u'actorInfo': {
    u'agentId': None,
    u'actorMediaServerId': u'test',
    u'actorType': u'MediaServer',
    u'actorPlaceId': None
  },
  u'reasonInfo': {
    u'reason': 1,
    u'reasonDescription': abcd,
    u'reasonName': ddddd
  }
}
```

UnsubscribeWorkbinEvents

This request will unsubscribe the agent from events from the workbin.

Input

```
{  
  "operationName": "Unsubscribe"  
}
```

Output

Success

```
{  
  "status": 0  
}
```

Failure

```
{  
  "statusCode": an integer value above 0,  
  "statusMessage": details  
}
```

For details on the statusCode value, please refer to the [All Methods](#) sub-section of the Return Values section.