Web Services API Reference

Web Services and Applications 8.5.1

1/5/2022

Table of Contents

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 resources and methods available through the Web Services API Reference. Each subcategory presents information about relevant operations, related resources, Request parameters and Return values.

Request Parameters

This topic outlines the request parameters for the Web Services API.

Object Fields

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

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.

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
}
```

When requesting an object from the Web Services server, it is possible to specify which data fields you receive by providing the `fields` request parameter.

For example:

Request:

```
GET .../api/v2/queues/<queue_id>?fields=id,name
```

Response:

```
{
        "id":<queue_id>,
        "name":<queue_name>
}
```

To request all fields of an object, set the `fields` property to `*`.

For example:

Request:

```
GET .../api/v2/queues/<queue_id>?fields=*
```

Response:

```
{
        "id":<queue_id>,
        "name":<queue_name>,
        "description":<queue_description>,
        ...
}
```

Note that when making "list" requests for any kind of object, Web Services returns a list of the corresponding object URIs.

For example:

Request:

```
 GET .../api/v2/queues
```

Response:

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

```
        "uris":[
                "http://.../api/v2/queues/<queue_1_id>",
                ...
                "http://.../api/v2/queues/<queue_N_id>"
        ]
 }
```

In order to receive a list of objects with their actual fields, you 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/queues?fields=id,name
```

Response:

```
{
        "statusCode":0,
        "queues":[{
                "id":<queue_1_id>,
                "name":<queue_1_name>
        },
        ...
        {
                "id":<queue_N_id>,
                "name":<queue_N_name>
        }]
}
```

Object Filtering

It is possible to filter objects using request parameters when doing "list" requests.

For example:

Request:

```
GET .../api/v2/queues?fields=id,name,channel&channel=voice
```

Response:

```
{
        "statusCode":0,
        "queues":[{
                "id":<queue_1_id>,
                "name":<queue_1_name>,
                "channel":"voice"
        },
        ...
        {
                "id":<queue_N_id>,
```

```
                "name":<queue_N_name>,
                "channel":"voice"
        }]
}
```

Note that the filtering parameter must be exactly the same as the name of the corresponding object field.

You can also combine several filtering parameters to make even more constraints, for example:

Request:

```
GET .../api/v2/system/routing-templates?fields=*&channel=voice&version=1.0.0
```

Response:

```
{
        "statusCode":0,
        "routingTemplates":[{
                "id":"00_RouteToSpecDestination",
                "name":"Route Call to Specified Destination",
                "description":"Routes calls to a skill or queue",
                "version":"1.0.0",
                "channel":"voice",
                "dependencies":["media", "destination"],
                "enabled":true,
                "schema": [...]
        },
        ...
        {
                "id":"07_SegmentCallerRouteToSpecDestination",
                "name":"Play Greeting, Segment Caller, and Route To Specified Destination",
                "description":"Plays a user-configured greeting, ...",
                "version":"1.0.0",
                "channel":"voice",
                "dependencies":["media", "destination", "data_record_type"],
                "enabled":false,
                "schema": [...]
        }]
}
```

Note that some "list" requests may make some of the filtering parameters mandatory.

Subresources

The subresources feature allows you to read subresources of an object together with the object itself. If you have a user object that has one or more skills and one or more devices, you can read all skills and devices of that user with the following request:

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>,
                ...
        }]
}
```

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.

Example 1

To receive a list of skills and devices associated with an agent, use the following.

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

To receive a list of skills associated with an agent, use the following.

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>,
                ...
        }]
}
```

Resolving URIs

Introduction

This feature is called "resource link resolution", which allows you to read an object and all other objects it is associated with in one request. For example, if we have a device object associated with a

phone number object and we want to read both of them in one request, we need to do the following:

Request:

```
GET .../api/v2/devices/<device_id>?resolveUris=*
```

Response:

```
{
        "id":<device_id>,
        "phoneNumberUri":"http://...",
        ...
        "phoneNumber":{
                "id":<phone_number_id>,
                ...
        }
}
```

In comparison, if you do not include the "resolveUris" parameter in the request, you will get everything except the "phoneNumber" object. In the example above, we specify "resolveUris=*" to resolve all URIs. It is possible to choose whether you want all URIs to be resolved or just some of them. This can be achieved by specifying a comma-separated list of property names referring to URIs.

Examples

Example 1

To resolve all URIs, use "`resolveUris=*`" as shown below.

Request:

```
GET .../api/v2/queues/<queue_id>?resolveUris=*
```

Response:

```
{
        "id":<queue_id>,
        "name":<queue_name>,
        ...
        "routingTemplateUri":"http://...",
        "phoneNumberUri":"http://...",
        ...
        "phoneNumber":{
                "id":<phone_number_id>,
                ...
        },
        "routingTemplate":{
                "id":<routing_template_id>,
                ...
        }
}
```

Example 2

To resolve a specific URI, use "`resolveUris=<uri>`" as shown below

Request:

```
GET .../api/v2/queues/<queue_id>?resolveUris=phoneNumberUri
```

Response:

```
{
        "id":<queue_id>,
        "name":<queue_name>,
        ...
        "routingTemplateUri":"http://...",
        "phoneNumberUri":"http://...",
        ...
        "phoneNumber":{
                "id":<phone_number_id>,
                ...
        }
}
```

Example 3

Request:

```
GET .../api/v2/queues/<queue_id>?resolveUris=phoneNumberUri,routingTemplateUri
```

Response:

```
{
        "id":<queue_id>,
        "name":<queue_name>,
        ...
        "routingTemplateUri":"http://...",
        "phoneNumberUri":"http://...",
        ...
        "phoneNumber":{
                "id":<phone_number_id>,
                ...
        },
        "routingTemplate":{
                "id":<routing_template_id>,
                ...
        }
}
```

User Authentication

Basic HTTP Authentication is used. Please see RFC 2617 Section 2 for reference.

Supported Requests

The following requests are supported at this time:

- /devices: fields=*

- /features: fields=*
- /me: subresources=*
- /me/calls: fields=*
- /me/devices: fields=*
- /me/skils: 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.

All Methods

All methods always return the `statusCode` attribute . If an error occurs, that is, if the `statusCode` is not `0`, the response includes error details in the `statusMessage` attribute.

The following status codes are supported:

Code	Description
0	The operation is successful. No `statusMessage` 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.
7	The operation was partially successful. Returned if at least one action in a bulk operation succeeded. See more information on partial success returns below.

If an error occurs during an operation, the response includes `statusCode` and `statusMessage` to clarify the error. No other attributes are included.

Note that if an error occurs during a request, 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 information and the request parameters.

Example

If retrieving a collection of URIs, the response will include the array attribute `"uris"` which will hold the requested collection.

```
GET .../skills

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

Example

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]`.

```
GET .../users?fields=*
{
    "statusCode":0,
    "users":[
              {
               "userName":"..",
               "firstName":"...",
                etc},
              }
              {
               "userName":"..",
               "firstName":"...",
                etc
              }
             ]
}
```

Example

If the URI is a "singular" resource such as `GET /users/{id}`, the response includes an attribute named after the singular form of the requested resource. This attribute contains the requested value. For example, `GET /users/{id}` will return `"user":{...user..}`.

```
GET /devices/{id}
{
    "statusCode": 0,
    "device": {
               "vendor": "...",
               "phoneNumber": "...",
                ...
               }
}
```

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.

Partial Success

Some operations may be considered successful if they are able to perform some of their work. These operations are considered "bulk" operations and are different from "transactions", which involve multiple steps that possibly use multiple servers. An example of a transaction is "create user" which involves creating some data in Cassandra as well as Configuration Server. If one of these actions fails, Web Services considers the whole operation a failure. In contrast, an operation such as "assign multiple skills to user" is a bulk operation which consists of a series of transactions (for example, each individual skill assignment is a transaction). The general rule is that if a step of a transaction fails, Web Services considers the whole operation a failure. If at least one transaction in a bulk operation succeeds, Web Services considers this a "partial success." Note that for bulk GETs (for example, GET /users) if the result is a partial list, the response includes statusCode:7 instead of 0. The rest of the result looks the same. For POST, PUT, and DELETE, the partial success returns have the following attributes:

Attribute	Value
statusCode	Always 7
succeeded	An array of resource descriptors (see below). Each represents a resource for which the transaction was successful.
failed	An array of failure descriptors (see below). Each represents a resource for which the transaction failed.

Attribute	Value
uri	The URI of a resource from request parameters for which the transaction succeeded. For example, if assigning multiple skills to a user, this is the URI of a skill).
id	The unique identifier of the resource above.

Attribute	Value
<uids>	The attributes which uniquely identify the resource for which this transaction failed. For example, if assigning skill uris, this will be "uri." If creating a user this will be "userName." If a resource has more than one identifying attribute all should be present.
statusCode	The status code describing the reason for failure.
statusMessage	The message describing the reason for failure.

Examples

Assign:

```
POST /users/{id}/skills
{
        uris:["uri1", "uri2", "uri3"]
}

{
        "statusCode":7 (partial success)
        "succeeded":[
                {
                        "id":<id1>,
                        "uri":"uri1"
                },
                {
                        "id":<id2>,
                        "uri":"uri2"
                }]
        "failed":[
                {
                        "statusCode":X,
                        "statusMessage":"msg",
                        "uri":"uri3"
                }
        ]
}
```

Create:

```
POST /users
{
        "users":[
                {
                        "firstName":"..",
                        "lastName":"..",
                        "userName":"u1", etc
                },
```

```
                {
                        "firstName":"..",
                        "lastName":"..",
                        "userName":"u2", etc
                },
                {
                        "firstName":"..",
                        "lastName":"..",
                        "userName":"u3", etc
                }

        ]
}

{
        "statusCode":7 (partial success)
        "succeeded":[
                {
                        "id":<id>,
                        "uri":"uri1"
                },
                {
                        "id":<id>,
                        "uri":"uri2"
                }]

        "failed":[
                {
                        "statusCode":3,
                        "statusMessage":"Operation forbidden, username already exists",
                        "userName":"u3"
                }]
}
```

Delete:

```
DELETE /users
{
        "uris":["uri1", "uri2", "uri3"]
}

{
        "statusCode":7 (partial success)
        "succeeded":[
                {
                        "id":<id1>,
                        "uri":"uri1"
                },
                {
                        "id":<id2>,
                        "uri":"uri2"
                }]

        "failed":[
                {
                        "statusCode":X,
                        "statusMessage":"...",
                        "uri":"uri3"
                }]
}
```

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 behaves as if this set of functionality does not exist. This returns results such as 404 errors when relevant resources are accessed, settings groups are not visible in lists, and operations return 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 `fields=*` causes full feature descriptions to be returned instead of URIs.	• 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 `fields=*` causes full feature descriptions to be returned instead of URIs.	• Contact Center Admin

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

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

Operation	Description	Permissions
GET	Returns the full feature description.	• Contact Center Admin • Agent
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

Services

The **services** resource provides a list of the services available in the system as well as their statuses and any information that is necessary to interact with the service (for example, a public SIP port for a "Voice" service). These services represent various aspects of Web Services that (in most cases) correspond to internal Genesys servers. For instance, each "Voice" service corresponds to a TServer, each "Reporting" service corresponds to a StatServer, and so on. A UI application can use this information to draw portions of the screen based on the status of a specific service. For instance, if the Provisioning service is "read only" the UI should disable all write operations but allow reading of provisioning data.

Operations

The following operations are available for **/services**:

Operation	Description	Permissions
GET	Returns a list of service URIs or actual service resources if the `fields` parameter is specified. The list is populated based on the features currently enabled for the contact center.	Agent, Supervisor, Contact Center Admin

The following operations are available for **/services/{id}**:

Operation	Description	Permissions
GET	Returns information about the specified service.	Agent, Supervisor, Contact Center Admin

Attributes

The following attributes are currently available for each service resource:

Attribute	Type	Description	Access	Applies To
id	String	A unique string identifying the service	GET	All Services
name	String	The service name. This will be equal to the name of the corresponding	GET	All Services

Attribute	Type	Description	Access	Applies To
		server's application object in Configuration Server. Note that in case there is a primary/backup pair, the primary server's application name will be used regardless of which instance is currently running. For the "Provisioning" service, the value will always be set to `Provisioning`.		
type	String	One of the following: Provisioning (CME + Cassandra) | Voice (T-Server) | Reporting (Stat Server) | Media (Interaction Server)	GET	All Services
state	String	The service's current state. Possible values are: Active | Inactive | ReadOnly (where applicable)	GET	All Services

Notifications

The client application can subscribe to the topic **/notifications/services** in order to receive service state change notifications. The following attributes will be present in a service state change notification:

Attribute	Value Type	Description
messageType	String	Will always be `ServiceStateChangeMessage`
service	Service Resource	The service resource for which the state has been changed

Examples

```
GET /services?fields=*
{
        "statusCode": 0,
        "services": [{
                "id": "..",
                "name": "Provisioning",
                "type": "Provisioning",
                "state": "Active"
        },
        {
                "id": "..",
                "name": "SIPS1",
                "type": "Voice",
                "state": "Active"
        },
        {
                "id": "..",
                "name": "SIPS2",
                "type": "Voice",
                "state": "Active"
        },
}
```

```
GET /services/<service_id>
{
                "statusCode": 0,
                "service": [{
                       "id": <service_id>,
                       "name": "Provisioning",
                       "type": "Provisioning",
                       "state": "Active" }
}
```

Settings

The *settings* resource is intended for configuration tasks that modify the behavior of existing functionality. This resource contains a list of URIs 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
POST	Creates a new settings group for the contact center	Contact Center Admin
DELETE	Removes the settings group from the contact center. Note that only settings groups that have been created via the API (for example, using `POST /settings`) can be deleted.	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, POST
name	String	A URI-compatible name for the settings group. This name will be used as part of the URI used to access the group: (for example, `GET /settings/my-settings-group`)	GET, POST
key	String	The name of the key attribute for this group's settings. Whenever an	POST

Attribute	Type	Description	Access
		individual setting needs to be modified, this key attribute will be used to identify the setting. The value of the key attribute must be unique for every setting and is read-only after the setting has been created. A setting may not be created without this attribute.	

Example

The following sample returns 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"
        }]
}
```

To create a new setting group:

```
POST /cloud-web/api/v2/settings

{
    "displayName":"My Setting Group",
    "name":"my-setting-group",
    "key":"name" //specifies that each setting in this group must have a "name" attribute
with a unique value
}
```

> **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 specifies 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" is 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.

Real-Time Reporting

The real-time reporting settings group is available under the following URI: `/api/v2/settings/reporting`. It contains following settings described below:

Setting	Description	Default Value	Permitted Values
defaultServiceLevelInterval	Specifies the maximum time (in seconds) it should take an agent to answer a call. The percentage of answered calls that were answered under this threshold can be retrieved via the statistics API. This settings has the following rule of evaluation: if setting not set explicitly, the default value from statistics.yaml is returned. If corresponding statistic/property not found - the serviceLevel threshold property is considered to be not set.	`TimeRangeRight` property of `statisticDefinitionEx` of ServiceLevel statistic defined for QUEUE object type. It is set to `60` (seconds) in `statistics.yaml` file shipped within IP. If set, this setting affects `ServiceLevel` statistic defined for all queues and skills in contact center.	Any positive integer. **Note:** changing this setting will cause stat server to calculate a new statistic, resetting statistics.
defaultTargetServiceLevelPercentage	this setting allows to store/retrieve value to be used in UI as default target service level. Does not affects reporting functionality on server side.	If not explicitly set, default value is `80` (not configurable).	Any integer between 1 and 100 (inclusive).

Supported operations:

Operation	Description
PUT	updates the value of option
GET	returns the list of settings

Note: POST and DELETE are not supported.

GET Example:

```
GET .../api/v2/settings/reporting
returns:
    {
       "statusCode": 0,
       "settings":
       [
           {
               "name": "defaultTargetServiceLevelPercentage",
               "value": "80"
           },
           {
               "name": "defaultServiceLevelInterval",
               "value": "60"
           }
       ],
       "key": "name"
    }
```

PUT Example for `defaultServiceLevelInterval`

```
PUT .../api/v2/settings/reporting
with body:
{
           "name": "defaultServiceLevelInterval",
           "value": "115"
}
```

PUT sample for defaultServiceLevelPercentage

```
PUT .../api/v2/settings/reporting
with body:
{
           "name": "defaultTargetServiceLevelPercentage",
           "value": "99"
}
```

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	Yes
operationName	String	The unique operation name that is used to set this state (for example, OutToLunch).	GET, POST, PUT	Yes
displayName	String	The name for the state.	GET, POST, PUT	Yes
state	Enum	The actual T-Server state (Ready/NotReady).	GET, POST, PUT	Yes
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/	GET, POST, PUT	N

Attribute	Type	Description	Access	Required
		WalkAway. This should be enforced by the API.		
reason	String	The reason for the agent's state (if specified, it must be unique as it is used as a reason code).	GET, POST, PUT	N

Examples

Each contact center initially has 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!"
}
```

User-Defined Setting Groups

Operations

The following operations are available for user-defined groups via the **/settings/{group-name}** URI:

Operation	Description	Permissions
GET	Retrieves an array of settings in this group	Contact Center Admin, Supervisor, Agent
POST	Creates a new setting in this group. Must include the "key" attribute specified when the group was created	Contact Center Admin
PUT	Updates a specific setting. Must include the "key" attribute specified when the group was created in order to identify the setting to update.	Contact Center Admin
DELETE	Removes a setting. Must include the "key" attribute to identify the setting.	Contact Center Admin

Attributes

The attributes for each setting group vary. There is no limitation as to the number of attributes defined or the values they contain -- beyond that the values must contain valid json. One important thing to note is that if you have an attribute which holds a json object, you will not be able to modify the individual fields in the object. To modify a specific field, the whole object must be passed via **PUT** overwriting the existing value (see examples below).

Storage

User-defined settings groups created using this API are only stored in Cassandra and are not synchronized to Configuration Server. Configuration Server-defined settings groups will continue to be imported.

Examples

We will use disposition codes as an example of a custom setting group. Disposition codes are used by agents to specify how a given call was completed. Note that this illustrates a sample configuration that might be used by a client to configure disposition codes. Disposition code configuration is not included in the API.

For the purposes of this example let's assume that a group called "dispositions" has been created. The "key" attribute is "name".

1. Create a disposition code with an attribute that contains a complex structure (possibleValues)

```
POST /settings/dispositions
{
        "name":"department" //key attribute
        "displayName":"Department"
        "possibleValues":[
             {
                 "name":"tech_support"
                 "displayName":"Tech Support"
                 "possibleValues":[
                      {
                           "displayName":"Computers"
                           "name":"computers"
                      },
                      {
                           "displayName":"Network"
                           "name":"network"
                      }
                  ]
             },
             {
                 "displayName":"Sales"
                 "name":"sales"
             }
        ]
}
```

2. Update the displayName "Computers" to "Computers!!!"

```
PUT /settings/dispositions
{
        "name":"department" //must include key attribute to identify setting
        "possibleValues":[
             {
                 "name":"tech_support"
                 "displayName":"Tech Support"
                 "possibleValues":[
                      {
                           "displayName":"Computers!!!"
```

```
                         "name":"computers"
                    },
                    {
                         "displayName":"Network"
                         "name":"network"
                    }
                ]
            },
            {
                "displayName":"Sales"
                "name":"sales"
            }
        ]
}
```

Note we must re-send the entire structure with the updated `displayName`.

3. Update displayName "Department" to "Department!"

```
PUT /settings/dispositions
{
        "name":"department", //key attribute
        "displayName":"Department!"
}
```

For top level settings it's enough to include the attribute (for example, `displayName`) and its new value along with the key to identify the setting.

4. Create another disposition...

```
POST /settings/dispositions
{
        "name":"helplevel", //key attribute
        "displayName":"Help Level"
        "possibleValues":[{"displayName":"Fixed!", "name":"fixed"}, {"displayName":"Not
Fixed", "name":"not_fixed"}]
}
```

...and GET everything:

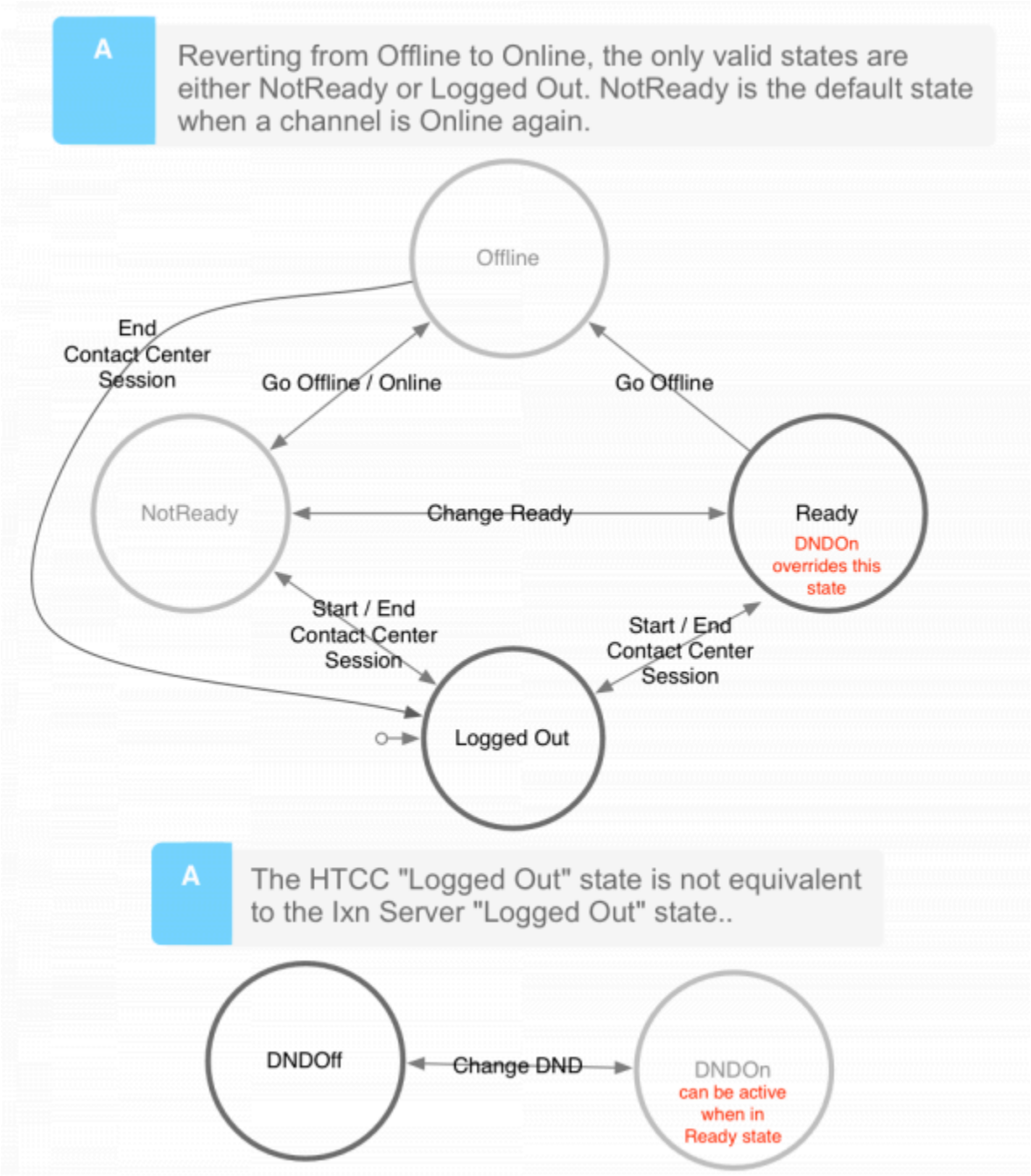
```
GET /settings/dispositions
{
"statusCode":0,
"key":"name",
"settings":[
    {
        "name":"department"
        "displayName":"Department!"
        "possibleValues":[
             {
                 "name":"tech_support"
                 "displayName":"Tech Support"
                 "possibleValues":[
                      {
                          "displayName":"Computers!!!",
                          "name":"computers"
                      },
                      {
```

```
                            "displayName":"Network"
                             "name":"network"
                        }
                    ]
            },
            {
                "displayName":"Sales"
                "name":"sales"
            }]
    },
    {
        "name":"helplevel",
        "displayName":"Help Level"
        "possibleValues":[{"displayName":"Fixed!", "name":"fixed"}, {"displayName":"Not
Fixed", "name":"not_fixed"}]
    }]
}
```

Channels API

A **channel** 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.

Web Services Agent States



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

Operation Name	HTTP Operation	Description	Permissions
Overall User State	GET	Returns all of an Agent's	Agent

Operation Name	HTTP Operation	Description	Permissions
		channel and device states.	

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 are 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 are 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 queries 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=***

Output

Success

```
{
        devices: [{
                ...
                id: "12345",
                userState: {
                        state: "NotReady",
                        reason: "Lunch",
                        displayName: "Out to Lunch",
                        workMode: “AuxWork”
                }
                ...
        }],
        channels: [{
                channel: "email",
                userState: {
                        state: "NotReady",
                        reason: "Lunch",
                        displayName: "Out to Lunch"
                },
                ...
        },{
                channel: "chat",
                userState: {
                        state: "NotReady",
                        reasonCode: "Lunch",
                        displayName: "Out to Lunch",
                },
```

```
                ...
        }],
        ...
}
```

Failure

```
{
        "statusCode": <any integer value above 0>,
        "statusMessage":details
}
```

Get All Devices

Returns all of a user's device states.

Input Parameters

GET on **/me/devices**

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

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

Get a Particular Device

Returns a user's specific device state.

Input Parameters

GET on **/me/devices/{id}**

Output

Success

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

Failure

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

Get All Multimedia Channels

Returns all of a user's channel states.

Input Parameters

GET on **/me/channels**

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

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

Get a Particular Multimedia Channel

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

Input Parameters

GET on **/me/channels/{channelId}**

Output

Success

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

Failed

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

Set Agent States Across all Channels

Use the following requests 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 is not ready to 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

Updates Agent States for a Channel

Method	POST		
URL	By channel: /me/**channels/{channelId}**		
Name	**Type**	**Mandatory**	**Description**
Parameters			
operationName	string	Yes	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 ready to receive 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 (for example, 'chat', 'email') into a "NotReady" state, meaning that the user will not 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

An **interaction** represents the activity on a particular channel, and thus is very tightly coupled with Channels. Activity in this case records the name of the action, and all relevant data (such as the reply text of an email or 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

Synchronous

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

HTTP Operation	Description	Permissions
GET	Returns the list of all active interactions for this Contact Center.	• 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 being handled by the agent.	{"statusCode":0,"interactions":[same message format as CometD notifications for interactions]}	Agent
GET	N/A	Returns a list of interaction URIs that are 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 the 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	Accepts a new interaction. This operation can be performed after a new interaction notification arrives. Accepting an interaction allows Agents to start working on it.	Agent

Operation Name	HTTP Operation	Description	Permissions
Reject	POST	Rejects a new interaction. This operation can be performed after a new interaction notification arrives. Rejecting an interaction allows an Agent the opportunity to work on a different interaction.	Agent
Cancel	POST	Cancels a new interaction. This operation can be performed after a client successfully completes either a Reply or ReplyAll operation. It is used when the client no longer wants to continue work on an incomplete interaction.	Agent
Complete	POST	Stops the interaction processing. This operation can be performed after an agent has finished working on an interaction.	Agent
PlaceInQueue	POST	Places the interaction into a queue. This operation can be performed on an interaction so that another person can continue processing it.	Agent
UpdateUserData	POST	Updates the interaction properties. This operation can be performed on an interaction that an agent is currently working on.	Agent
AttachUserData	POST	Updates the interaction properties. This operation can be performed on an interaction that an agent is currently working on.	Agent
DeleteUserData	POST	Updates the interaction properties. This operation can be performed on an	Agent

Operation Name	HTTP Operation	Description	Permissions
		interaction that an agent is currently working on.	
PlaceInWorkbin	POST	Places the interaction into a specified workbin. The agent can place the interaction in a workbin to continue the processing later.	Agent
GetWorkbinContent	POST	Returns the contents of a given workbin.	Agent
SingleStepTransfer	POST	Transfers interaction ownership from one agent to another agent.	Agent

Accept

An agent (User) can either accept or rejecting a new interaction invitation. Accepting the invitation associates the interaction with the agent. This association or "ownership" continues until that agent performs a "Complete", "Transfer", or "Re-direct" action on the interaction. A successful request results in an empty `HTTP 200` response. In most cases, the true response arrives on the CometD topic (channel) **/v2/me/interactions/**.

Input Parameters

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

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

Output

Success

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

Failed

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

Reject

An agent (User) can either accept or reject a new interaction invitation. A successful request results 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
}
```

Cancel

This operation can be performed after an agent (User) successfully creates a "Reply" or "ReplyAll" operation. This is typically called when the agent (User) no longer wants to continue work on an incomplete interaction.

HTTP Request

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

```
{
"operationName":"Cancel"
}
```

HTTP Response

Success

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

Failed

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

Complete

The agent can perform this operation after he or she accepts the interaction invitation. This operation ends the agent's ownership of the interaction. The agent (User) should perform this operation if he or she has finished working on the interaction

Input Parameters

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

```
{
"operationName":"Complete"
}
```

Output

Success

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

Failed

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

PlaceInQueue

The agent (User) can perform this operation after accepting the interaction invitation. This ends the agent's ownership of the interaction.

The agent should use this operation if another agent will handle further processing of the interaction.

Input Parameters

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

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

```
}
```

Output

Success

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

Failed

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

UpdateUserData

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

Input Parameters

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

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

Output

Success

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

Failed

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

AttachUserData

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

Input Parameters

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

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

Output

Success

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

Failed

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

DeleteUserData

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

Input Parameters

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

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

Output

Success

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

Failed

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

PlaceInWorkbin

The agent can use this operation to 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 returns the contents of a workbin. The content comes 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 contains the same information as found in the new notification messages.

SingleStepTransfer

This operation transfers interaction ownership from one agent to another agent. The functionality may vary among different media types. For further details, see the media type sub-topics.

Input

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

HTTP Response

Success

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

Failed

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

Operations

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

Operation Name	HTTP Operation	Description	Type	Permissions
Accept	POST	Accepts the interaction. The agent can perform this operation after a new interaction invitation arrives. Accepting an interaction allows the agent to start working on it.	General Interaction	Agent
Reject	POST	Rejects the interaction. The agent can perform this operation after a new interaction invitation arrives. Rejecting an interaction allows an agent to work on a differen interaction.	General Interaction	Agent
Reply	POST	Creates an outbound e-mail reply. The agent can perform this operation after he or she accepts a new inbound e-mail interaction.	E-mail Specific	Agent
ReplyAll	POST	Creates an outbound e-mail reply to all senders. The agent can perform this operation after he or she accepts a new inbound e-mail interation.	E-mail Specific	Agent
Cancel	POST	Cancels the outbound reply e-mail. The agent	E-mail Specific	Agent

Operation Name	HTTP Operation	Description	Type	Permissions
		can perform this operation after he or she creates a new outbound reply e-mail.		
Send	POST	Sends an outbound e-mail. The agent can perform this operation after he or she creates a new outbound reply e-mail.	E-mail Specific	Agent
Transfer	POST	Transfers an outbound e-mail interaction. The agent can perform this operation while he or she is working on an interaction. The interaction is transferred to another agent or to a group of agents associated with a skill.	General Interaction	Agent
Complete	POST	This operation can be performed after an agent has finished working on an interaction.	General Interaction	Agent
PlaceInQueue	POST	Places the interaction in a specific queue. The agent should use this operation if another agent should handle further processing of the interaction.	General Interaction	Agent

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 the Universal Contact Server. The operation is asynchronous, meaning that a 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`

- `FromnAddress` 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 the Interaction Server and the Universal Contact Server. The operation is asynchronous, meaning that a CometD notification will be sent upon successful completion of the 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"]
}
```

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
}
```

Create

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":"Create",
"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 crated (e.g. OutboundNew)
}
```

HTTP Response

Success

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

Failed

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

Attachments API

A Web Services attachment is the same as an e-mail attachment. In Platform SDK 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	{"statusCode":0,"documentId":"00009a8T8Y13002S"}	

HTTP Operation	Description	Output	Permissions
	Attachment and associates it with an Interaction. **Note:** The file should be bound to the 'attachment' variable.		

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).	{"statusCode":0}	Agent

See also:

- Interactions on E-mail Channel Notifications
- Interaction Notifications

Interactions On Chat Channel

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
InviteUser	POST	This operation allows the agent to initiate a conference with another agent on a specific interaction, allowing the two agents to work on the same interaction simultaneously. To leave the conference, agents must make a LeaveConference request. This functionality is limited to conferencing. Monitoring and coaching functionality is not available.	Chat Specific	Agent
EndChatSession	POST	This operation ends a chat session, but keeps the interaction alive for the agent to do 'after-call'	Chat Specific	Agent

Operation Name	HTTP Operation	Description	Type	Permissions
		work (for example, update attached data). The interaction can be completed by the Complete call. **Note:** Calling Complete on an interaction with an active chat session will end the chat session AND complete the interaction.		
Send	POST	This operation can be performed whenever an agent wants to send a message to other chat participants.	Chat Specific	Agent
SendStartTypingNotification	POST	This operation can be performed to send notifications to other chat participants that the agent has started typing (for example, 'agent typing').	Chat Specific	Agent
SendStopTypingNotification	POST	This operation can be performed to send notifications to other chat participants that the agent has stopped typing.	Chat Specific	Agent
SingleStepTransfer	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
Complete	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	Chat Specific	Agent

Operation Name	HTTP Operation	Description	Type	Permissions
		get the chat history. Only agents who are participating in the chat can execute this operation.		
LeaveConference	POST	This operation allows the agent to leave a currently active conference. When agents leave a conference, the last remaining agent retains ownership of the interaction.	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.

InviteUser

This operation allows the agent to initiate a conference with another agent on a specific interaction, allowing the two agents to work on the same interaction simultaneously. To leave the conference, agents must make a LeaveConference request. This functionality is limited to conferencing. Monitoring and coaching functionality is not available. Further details can be found within the specific media topic.

HTTP Request

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

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

HTTP Response

Success

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

Failure

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

EndChatSession

This operation ends a chat session, but keeps the interaction alive for the agent to do 'after-call' work (for example, update attached data). The interaction can be completed by the Complete call. **Note:** Calling Complete on an interaction with an active chat session will end the chat session AND complete the interaction.

HTTP Request

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

```
{
```

```
"operationName":"EndChatSession"
}
```

HTTP Response

Success

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

Failure

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

Send

This operation can be performed whenever an agent wants to send a message to other chat participants.

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.

SendStartTypingNotification

This operation can be performed to send notifications to other chat participants that the agent has started typing (for example, 'agent typing').

HTTP Request

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

```
{
"operationName":"SendStartTypingNotification"
}
```

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.

SendStopTypingNotification

This operation can be performed to send notifications to other chat participants that the agent has stopped typing.

HTTP Request

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

```
{
"operationName":"SendStopTypingNotification"
}
```

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
}
```

LeaveConference

This operation allows the agent to leave a currently active conference. When agents leave a conference, the last remaining agent retains ownership of the interaction.

HTTP Request

```
{
"operationName":"LeaveConference",
```

```
"agentId":a unique Agent identifier
}
```

HTTP Response

Success

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

Failed

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

See also:

- Interaction Notifications

Interaction Notifications

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

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

Email Interactions

When an agent sets his or her status for the email channel to Ready, the following notifications may occur:

Email Invite

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

```
[
    {
        'data': {
            'interaction': {
                'userData': {
                    'FirstName': 'qwerty',
                    'Header_Content-Type': 'multipart/
mixed;boundary="===============0346841814=="',
                    'RTargetTypeSelected': '2',
                    'RTargetObjectSelected': '?: 2>1',
                    '_ContainsAttachment': 'false',
                    'CBR-actual_volume': '',
                    'RVQID': '',
                    'To': 'support@ci-vm106',
                    'Header_Date': 'Mon,
                    28Oct201314: 08: 44-0700',
                    '_AttachmentsSize': '0',
                    '_AutoReplyCount': 0,
                    'RTargetObjSelDBID': '',
                    'Header_MIME-Version': '1.0',
                    'Mailbox': 'support',
                    'CBR-Interaction_cost': '',
                    'CBR-contract_DBIDs': '',
                    'ContactId': '0000Ha8S2A7X000N',
                    'RTargetAgentSelected': '1',
                    'CBR-IT-path_DBIDs': '',
                    'RTargetAgentGroup': '?: 2>1',
                    'RTargetRuleSelected': '',
                    'FromPersonal': '',
                    'RTargetPlaceSelected': 'Agent1',
                    '_AttachmentFileNames': '',
                    'RTenant': 'Environment',
```

```
                    'RRequestedSkills': None,
                    'Origination_Source': 'Email',
                    'RRequestedSkillCombination': '',
                    'RVQDBID': '',
                    'RStrategyDBID': '134',
                    'CustomerSegment': 'default',
                    'PegAG?: 2>1': 1,
                    'ServiceType': 'default',
                    'FromAddress': 'Customer1@ci-vm106',
                    'ServiceObjective': 0,
                    'Header_Message-ID': '<6h34v7hbfot8h0h.281020131408@ci-vm106>',
                    'RTargetRequested': '?: 2>1',
                    'EmailAddress': 'Customer1@ci-vm106',
                    'RStrategyName': 'SimpleEmailInStrategy',
                    'Subject': 'Rendering_response_category'
                },
                'currentQueue': 'SimpleEmailInQueue',
                'subType': 'InboundNew',
                'inQueues': {
                    '__STOP__': ''
                },
                'capabilities': [
                    'Accept',
                    'Reject'
                ],
                'content': {
                    'interactionId': '0000Ka8WP8G94H9A'
                },
                'outQueues': {
                    'SimpleEmailOutQueue': ''
                },
                'state': 'Invited',
                'type': 'Inbound',
                'id': '0000Ka8WP8G94H9A',
                'channel': 'email'
            },
            'messageType': 'InteractionStateMessage'
        },
        'channel': '/v2/me/interactions'
    }
]
```

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.

```
[
    {
        'data': {
            'userData': {
                'Header_Content-Type': 'multipart/
mixed;boundary="===============1840393583=="',
                'RTargetTypeSelected': '2',
                'RTargetObjectSelected': '?: 2>1',
                '_ContainsAttachment': 'false',
                'CBR-actual_volume': '',
                'RVQID': '',
                'To': 'support@ci-vm106',
                'Header_Date': 'Fri, 28Jun201313: 30: 34-0700',
                '_AttachmentsSize': '0',
```

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

Email Accepted

```
[
    {
```

```
        'data': {
            'interaction': {
                'userData': {
                    'FirstName': 'qwerty',
                    'Header_Content-Type': 'multipart/
mixed;boundary="==========0346841814=="',
                    'RTargetTypeSelected': '2',
                    'RTargetObjectSelected': '?: 2>1',
                    '_ContainsAttachment': 'false',
                    'CBR-actual_volume': '',
                    'RVQID': '',
                    'To': 'support@ci-vm106',
                    'Header_Date': 'Mon,
                    28Oct201314: 08: 44-0700',
                    '_AttachmentsSize': '0',
                    '_AutoReplyCount': 0,
                    'RTargetObjSelDBID': '',
                    'Header_MIME-Version': '1.0',
                    'Mailbox': 'support',
                    'CBR-Interaction_cost': '',
                    'CBR-contract_DBIDs': '',
                    'ContactId': '0000Ha8S2A7X000N',
                    'RTargetAgentSelected': '1',
                    'CBR-IT-path_DBIDs': '',
                    'RTargetAgentGroup': '?: 2>1',
                    'RTargetRuleSelected': '',
                    'FromPersonal': '',
                    'RTargetPlaceSelected': 'Agent1',
                    '_AttachmentFileNames': '',
                    'RTenant': 'Environment',
                    'RRequestedSkills': None,
                    'Origination_Source': 'Email',
                    'RRequestedSkillCombination': '',
                    'RVQDBID': '',
                    'RStrategyDBID': '134',
                    'CustomerSegment': 'default',
                    'PegAG?: 2>1': 1,
                    'ServiceType': 'default',
                    'FromAddress': 'Customer1@ci-vm106',
                    'ServiceObjective': 0,
                    'Header_Message-ID': '<6h34v7hbfot8h0h.281020131408@ci-vm106>',
                    'RTargetRequested': '?: 2>1',
                    'EmailAddress': 'Customer1@ci-vm106',
                    'RStrategyName': 'SimpleEmailInStrategy',
                    'Subject': 'Rendering_response_category'
                },
                'currentQueue': 'SimpleEmailInQueue',
                'startDate': 1382994529967L,
                'inQueues': {
                    '__STOP__': ''
                },
                'capabilities': [
                    'StopProcessing',
                    'UpdateUserData',
                    'AttachUserData',
                    'DeleteUserData',
                    'PlaceInWorkbin',
                    'SingleStepTransfer',
                    'Reply',
                    'ReplyAll'
                ],
                'content': {
                    'text': 'Thisisemailbody.\r\nTesting, hopefullythiswillwork',
```

```
                    'interactionId': '0000Ka8WP8G94H9A',
                    'fromAddress': 'Customer1@ci-vm106',
                    'toAddress': 'support@ci-vm106',
                    'subject': 'Rendering_response_category'
                },
                'outQueues': {
                    'SimpleEmailOutQueue': ''
                },
                'state': 'Accepted',
                'subType': 'InboundNew',
                'threadId': '0000Ka8WP8G94H99',
                'type': 'Inbound',
                'id': '0000Ka8WP8G94H9A',
                'channel': 'email'
            },
            'messageType': 'InteractionStateMessage'
        },
        'channel': '/v2/me/interactions'
    }
]
```

Chat Interactions

When an agent sets his or her status for the chat channel to Ready, the following notifications may occur:

Chat Invited

```
[
    {
        'data': {
            'userData': {
                'IdentifyCreateContact': '3',
                'RTargetTypeSelected': '2',
                'RTargetObjectSelected': '?: 2>1',
                'RVQID': '',
                'ChatServerAppName': 'es_chat',
                'ChatServerHost': 'hpe-voicevm-76.genesyslab.com',
                'RTargetObjSelDBID': '',
                'ChatServerPort': '7160',
                'CBR-Interaction_cost': '',
                'CBR-contract_DBIDs': '',
                'RTargetAgentSelected': 'Agent1',
                'CBR-IT-path_DBIDs': '',
                'RTargetRuleSelected': '',
                'RTargetPlaceSelected': 'Agent1',
                'CBR-actual_volume': '',
                'RTenant': 'Environment',
                'ChatServerDBID': '155',
                'RRequestedSkills': None,
                'RRequestedSkillCombination': '',
                'RVQDBID': '',
                'RStrategyDBID': '125',
                'CustomerSegment': 'default',
                'PegAG?: 2>1': 1,
                'ServiceType': 'default',
                'ServiceObjective': 0,
```

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

Chat Revoked

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

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

Chat Accepted

```
[
    {
    'data': {
        'interaction': {
            'userData': {
                'IdentifyCreateContact': '3',
                'RTargetTypeSelected': '2',
                'RTargetObjectSelected': '?: 2>1',
                'RVQID': '',
                'ChatServerAppName': 'esv_cht',
                'ChatServerHost': 'hpe-voicevm-55.genesyslab.com',
                'RTargetObjSelDBID': '',
                'ChatServerPort': '7160',
                'CBR-Interaction_cost': '',
                'CBR-contract_DBIDs': '',
                'RTargetAgentSelected': 'Agent1',
                'CBR-IT-path_DBIDs': '',
                'RTargetRuleSelected': '',
                'RTargetPlaceSelected': 'Agent1',
                'CBR-actual_volume': '',
                'RTenant': 'Environment',
                'ChatServerDBID': '145',
                'RRequestedSkills': None,
                'RRequestedSkillCombination': '',
                'RVQDBID': '',
                'RStrategyDBID': '110',
                'CustomerSegment': 'default',
                'PegAG?: 2>1': 1,
                'ServiceType': 'default',
                'ServiceObjective': 0,
```

```
                'RTargetRequested': '?: 2>1',
                'RTargetAgentGroup': '?: 2>1',
                'RStrategyName': 'SimpleChatInStrategy',
                'Subject': 'Customersupport'
            },
            'currentQueue': 'SimpleChatInQueue',
            'startDate': 1382984933000L,
            'inQueues': {
                'SimpleChatTranscriptQueue': u''
            },
            'capabilities': [
                'StopProcessing',
                'UpdateUserData',
                'AttachUserData',
                'DeleteUserData',
                'PlaceInWorkbin',
                'SingleStepTransfer',
                'LeaveChat',
                'EndChatSession',
                'SingleStepConference',
                'RequestChatHistory',
                'Send'
            ],
            'content': {
                'latestEventId': 3,
                'sessionId': '00002a94345B0MXX',
                'events': [
                    {
                        'userId': '0091526EACE5E036',
                        'messageType': 'PARTY_JOINED',
                        'userType': 'Client',
                        'visibility': 'All',
                        'nickName': 'test_agent_notificationL',
                        'id': 1
                    },
                    {
                        'userId': '0091526EACF0E038',
                        'messageType': 'PARTY_JOINED',
                        'userType': 'Agent',
                        'visibility': 'All',
                        'nickName': 'TestName',
                        'id': 2
                    },
                    {
                        'messageType': 'NOTICE',
                        'userId': '0091526EACF0E038',
                        'visibility': 'All',
                        'noticeType': 'TypingStarted',
                        'id': 3
                    }
                ]
            },
            'subType': 'InboundNew',
            'state': 'Accepted',
            'threadId': '00002a94345B0MXY',
            'type': 'Inbound',
            'id': '00002a94345B0MXX',
            'channel': 'chat'
        },
        'messageType': 'InteractionStateMessage'
    }
]
```

Chat Notifications


```
[
   {
        'data': {
            'interaction': {
                'userData': {
                    'IdentifyCreateContact': '3',
                    'RTargetTypeSelected': '2',
                    'RTargetObjectSelected': '?: 2>1',
                    'RVQID': '',
                    'ChatServerAppName': 'esv_cht',
                    'ChatServerHost': 'hpe-voicevm-55.genesyslab.com',
                    'RTargetObjSelDBID': '',
                    'ChatServerPort': '7160',
                    'CBR-Interaction_cost': '',
                    'CBR-contract_DBIDs': '',
                    'RTargetAgentSelected': 'Agent1',
                    'CBR-IT-path_DBIDs': '',
                    'RTargetRuleSelected': '',
                    'RTargetPlaceSelected': 'Agent1',
                    'CBR-actual_volume': '',
                    'RTenant': 'Environment',
                    'ChatServerDBID': '145',
                    'RRequestedSkills': None,
                    'RRequestedSkillCombination': '',
                    'RVQDBID': '',
                    'RStrategyDBID': '110',
                    'CustomerSegment': 'default',
                    'PegAG?: 2>1': 1,
                    'ServiceType': 'default',
                    'ServiceObjective': 0,
                    'RTargetRequested': '?: 2>1',
                    'RTargetAgentGroup': '?: 2>1',
                    'RStrategyName': 'SimpleChatInStrategy',
                    'Subject': 'Customersupport'
                },
                'currentQueue': 'SimpleChatInQueue',
                'startDate': 1382984933000L,
                'inQueues': {
                    'SimpleChatTranscriptQueue': ''
                },
                'content': {
                    'latestEventId': 5,
                    'sessionId': '00002a94345B0MXX',
                    'events': [
                        {
                            'messageType': 'NOTICE',
                            'userId': '0091526EACF0E038',
                            'visibility': 'All',
                            'noticeType': 'TypingStopped',
                            'id': 4
                        },
                        {
                            'messageType': 'NOTICE',
                            'userId': '0091526EACF0E038',
                            'visibility': 'All',
                            'noticeType': 'TypingStarted',
                            'id': 3
                        },
                        {
                            'userId': '0091526EACE5E036',
```

```
                            'messageType': 'PARTY_JOINED',
                            'userType': 'Client',
                            'visibility': 'All',
                            'nickName': 'test_agent_notificationL',
                            'id': 1
                        },
                        {
                            'userId': '0091526EACF0E038',
                            'messageType': 'PARTY_JOINED',
                            'userType': 'Agent',
                            'visibility': 'All',
                            'nickName': 'TestName',
                            'id': 2
                        },
                        {
                            'userId': '0091526EACF0E038',
                            'messageType': 'PARTY_LEFT',
                            'visibility': 'All',
                            'id': 5
                        }
                    ]
                },
                'subType': 'InboundNew',
                'state': 'PropertiesChanged',
                'threadId': '00002a94345B0MXY',
                'type': 'Inbound',
                'id': '00002a94345B0MXX',
                'channel': 'chat'
            },
            'messageType': 'InteractionStateMessage'
        },
        'channel': '/v2/me/interactions'
    }
]
```

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.	• Contact Center Admin	• fields=*: Will return all fields for every

Operation	Description	Permissions	Arguments
		• Agent	user instead of URI. • 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.	• 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.	• Contact Center Admin • Agent	
DELETE	Deletes the user specified by the ID.	• 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 you need detailed information, use the **?fields=*** request parameter. Note 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

Attributes

Attribute	Type	Description	Access Level
model	String	Optional. The device model.	GET, POST
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.	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	A list of operations currently available on the device. Possible values for users with role ROLE_AGENT: DoNotDisturbOn, DoNotDisturbOff, ForwardCallsOn,	GET

Attribute	Type	Description	Access Level
		ForwardCallsOff In addition, users who have the role ROLE_SUPERVISOR may see one or more of the following: ListenIn, BargeIn, Coach, CancelSupervisorMonitoring The exact combination depends on the particular contact center configuration as well as the current `supervisorMonitoringState` 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", "BargeIn", 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
supervisorMonitoringScope	String	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	GET, POST

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

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).	• 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.	• 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"
      },
      ...
   ]
}
```

Contacts API

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 at which this contact can be reached.	GET for all PUT, POST, DELETE for contacts of type "Custom" only	Custom, Queue, 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

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.	• Contact Center Admin • Agent
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

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

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
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	If the participant number contains `+country_code`, this will contain a list of structures, each of which will be: "country"—country infromation of the phone number in the JSON format: `"country":{"name":"United States","code":"US","callingCode":"1"}` "location"—region of the phone number. "localPhoneNumber"—phone number in domestic format. "E164digits"—phone number in e.164 format.	GET
mute	Off	Shows whether the call is muted.	GET

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.	• 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.	• Contact Center Admin • Agent (only for objects owned by this agent)

Call Recording API

The Call Recording API adds the ability to record agents' phone conversations and to manage call recordings.

See the following topics for details about the Call Recording API:

- Agent Operations

Agent Operations

The agent operations of the Call Recording API are asynchronous operations. Each operation has a corresponding notification which provides the state of the operation. Call recording operations are performed via a POST to the **/me/calls/{id}** URI with the `operationName` attribute set as described in the table below.

Operation name	Required Attributes	Description
StartCallRecording	n/a	Starts call recording. Recording stops when the call is completed or "StopCallRecording" is called either on the call or on the device.
StopCallRecording	n/a	Stop recording the specified phone call.
PauseCallRecording	n/a	Temporarily stops recording the specified phone call.
ResumeCallRecording	n/a	Resumes recording the specified phone call.

Notifications

The client application can subscribe to the standard call notifications topic **/notifications/me/calls** to receive recording event notifications. In contact centers where call recording is enabled, additional call recording state events will be sent when call recording state changes. The following attributes will be present in a call recording state change notification:

Attribute Name	Value type	Description
notificationType	String	Call recording state notifications will always be of type "CallRecordingStateChange."
call	Call resource	The call resource whose state has changed (the actual state will be specified in the call object -- see below).

API Extensions

The attribute `recordingState` will be added to the **call** resource. Its possible values are:

- *Recording*: The call is currently being recorded
- *Paused*: Recording is currently paused.
- *Stopped*: Recording has been stopped.

In addition, the call's capabilities will be extended with call recording operations. The capabilities list will contain the names of call recording operations that are available in the current recording state.

Examples

Start call recording:

Request:

```
POST http://localhost:8080/api/v2/me/calls/0071022ec8ac8056
{
  "operationName": "StartCallRecording"
}
```

Response:

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

Event:

```
{
  "data": {
    "messageType": "CallStateChangeMessage",
    "call": {
      "id": "0071022ec8ac8056",
      "state": "Established",
      "callUuid": "00RV9H7S6O8V3BSHAG7GK2LAES00002M",
      "deviceUri": "http://127.0.0.1:8080/api/v2/devices/55b8023d-573d-48d3-b4ac-
e29ba3c5861d",
      "uri": "http://127.0.0.1:8080/api/v2/me/calls/0071022ec8ac8056",
      "participants": [
        "5000"
      ],
      "participantsInfo": [
        {
          "country": {
            "name": "",
            "code": "",
            "callingCode": ""
          },
          "formattedPhoneNumber": "5000",
          "location": null,
          "E164digits": "5000",
          "isValidNumber": false
        }
      ],
      "ani": "5000",
      "dnis": "5001",
      "callType": "Internal",
      "capabilities": [
        "DeleteUserDataPair",
        "SingleStepTransfer",
        "InitiateConference",
        "InitiateTransfer",
        "UpdateUserData",
        "Hold",
        "Hangup",
        "DeleteUserData",
```

```
        "SingleStepConference",
        "SendDtmf",
        "AttachUserData",
        "PauseCallRecording",
        "StopCallRecording"
      ],
      "userData": {
        "GSIP_RECORD": "ON",
        "IW_CaseUid": "8dfca5ac-ed84-4f9a-d902-8ef3c2faad81",
        "IW_BundleUid": "ba431089-f52b-4ce2-0658-f8bdfdfe3034",
        "GSIP_REC_FN":
"00RV9H7S6O8V3BSHAG7GK2LAES00002M_5001_5000_5001_2013-08-09_21-11-22_hpe-
voicevm-84.genesyslab.com__%3Ccont_center_id%3E"
      },
      "duration": "106",
      "mute": "Off",
      "recordingState": "Recording"
    },
    "phoneNumber": "5001",
    "notificationType": "CallRecordingStateChange"
  },
  "channel": "/v2/me/calls"
}
```


Pause

Request:

```
POST http://localhost:8080/api/v2/me/calls/0071022ec8ac8056
{
  "operationName": "PauseCallRecording"
}
```

Response:

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

Event:


```
{
  "data": {
    "messageType": "CallStateChangeMessage",
    "call": {
      "id": "0071022ec8ac8056",
      "state": "Established",
      "callUuid": "00RV9H7S6O8V3BSHAG7GK2LAES00002M",
      "deviceUri": "http://127.0.0.1:8080/api/v2/devices/55b8023d-573d-48d3-b4ac-
e29ba3c5861d",
      "uri": "http://127.0.0.1:8080/api/v2/me/calls/0071022ec8ac8056",
      "participants": [
        "5000"
      ],
      "participantsInfo": [
        {
          "country": {
            "name": "",
            "code": "",
            "callingCode": ""
          },
```

```
          "formattedPhoneNumber": "5000",
          "location": null,
          "E164digits": "5000",
          "isValidNumber": false
        }
      ],
      "ani": "5000",
      "dnis": "5001",
      "callType": "Internal",
      "capabilities": [
        "DeleteUserDataPair",
        "SingleStepTransfer",
        "InitiateConference",
        "InitiateTransfer",
        "UpdateUserData",
        "Hold",
        "Hangup",
        "DeleteUserData",
        "SingleStepConference",
        "SendDtmf",
        "AttachUserData",
        "ResumeCallRecording",
        "StopCallRecording"
      ],
      "userData": {
        "GSIP_RECORD": "PAUSED",
        "IW_CaseUid": "8dfca5ac-ed84-4f9a-d902-8ef3c2faad81",
        "IW_BundleUid": "ba431089-f52b-4ce2-0658-f8bdfdfe3034",
        "GSIP_REC_FN":
"00RV9H7S6O8V3BSHAG7GK2LAES00002M_5001_5000_5001_2013-08-09_21-11-22_hpe-
voicevm-84.genesyslab.com__%3Ccont_center_id%3E"
      },
      "duration": "141",
      "mute": "Off",
      "recordingState": "Paused"
    },
    "phoneNumber": "5001",
    "notificationType": "CallRecordingStateChange"
  },
  "channel": "/v2/me/calls"
}
```


Resume

Request:

```
POST http://localhost:8080/api/v2/me/calls/0071022ec8ac8056
{
  "operationName": "ResumeCallRecording"
}
```

Response:

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

Event:


```
{
  "data": {
```

```
    "messageType": "CallStateChangeMessage",
    "call": {
      "id": "0071022ec8ac8056",
      "state": "Established",
      "callUuid": "00RV9H7S6O8V3BSHAG7GK2LAES00002M",
      "deviceUri": "http://127.0.0.1:8080/api/v2/devices/55b8023d-573d-48d3-b4ac-
e29ba3c5861d",
      "uri": "http://127.0.0.1:8080/api/v2/me/calls/0071022ec8ac8056",
      "participants": [
        "5000"
      ],
      "participantsInfo": [
        {
          "country": {
            "name": "",
            "code": "",
            "callingCode": ""
          },
          "formattedPhoneNumber": "5000",
          "location": null,
          "E164digits": "5000",
          "isValidNumber": false
        }
      ],
      "ani": "5000",
      "dnis": "5001",
      "callType": "Internal",
      "capabilities": [
        "DeleteUserDataPair",
        "SingleStepTransfer",
        "InitiateConference",
        "InitiateTransfer",
        "UpdateUserData",
        "Hold",
        "Hangup",
        "DeleteUserData",
        "SingleStepConference",
        "SendDtmf",
        "AttachUserData",
        "PauseCallRecording",
        "StopCallRecording"
      ],
      "userData": {
        "GSIP_RECORD": "ON",
        "IW_CaseUid": "8dfca5ac-ed84-4f9a-d902-8ef3c2faad81",
        "IW_BundleUid": "ba431089-f52b-4ce2-0658-f8bdfdfe3034",
        "GSIP_REC_FN":
"00RV9H7S6O8V3BSHAG7GK2LAES00002M_5001_5000_5001_2013-08-09_21-11-22_hpe-
voicevm-84.genesyslab.com__%3Ccont_center_id%3E"
      },
      "duration": "588",
      "mute": "Off",
      "recordingState": "Recording"
    },
    "phoneNumber": "5001",
    "notificationType": "CallRecordingStateChange"
  },
  "channel": "/v2/me/calls"
}
```

Stop

Request:

```
POST http://localhost:8080/api/v2/me/calls/0071022ec8ac8056
{
  "operationName": "StopCallRecording"
}
```

Response:

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

Event:


```
{
  "data": {
    "messageType": "CallStateChangeMessage",
    "call": {
      "id": "0071022ec8ac8056",
      "state": "Established",
      "callUuid": "00RV9H7S6O8V3BSHAG7GK2LAES00002M",
      "deviceUri": "http://127.0.0.1:8080/api/v2/devices/55b8023d-573d-48d3-b4ac-
e29ba3c5861d",
      "uri": "http://127.0.0.1:8080/api/v2/me/calls/0071022ec8ac8056",
      "participants": [
        "5000"
      ],
      "participantsInfo": [
        {
          "country": {
            "name": "",
            "code": "",
            "callingCode": ""
          },
          "formattedPhoneNumber": "5000",
          "location": null,
          "E164digits": "5000",
          "isValidNumber": false
        }
      ],
      "ani": "5000",
      "dnis": "5001",
      "callType": "Internal",
      "capabilities": [
        "DeleteUserDataPair",
        "SingleStepTransfer",
        "InitiateConference",
        "InitiateTransfer",
        "UpdateUserData",
        "Hold",
        "Hangup",
        "DeleteUserData",
        "SingleStepConference",
        "SendDtmf",
        "AttachUserData",
        "StartCallRecording"
      ],
      "userData": {
```

```
        "GSIP_RECORD": "OFF",
        "IW_CaseUid": "8dfca5ac-ed84-4f9a-d902-8ef3c2faad81",
        "IW_BundleUid": "ba431089-f52b-4ce2-0658-f8bdfdfe3034",
        "GSIP_REC_FN":
"00RV9H7S6O8V3BSHAG7GK2LAES00002M_5001_5000_5001_2013-08-09_21-11-22_hpe-
voicevm-84.genesyslab.com__%3Ccont_center_id%3E"
      },
      "duration": "613",
      "mute": "Off",
      "recordingState": "Stopped"
    },
    "phoneNumber": "5001",
    "notificationType": "CallRecordingStateChange"
  },
  "channel": "/v2/me/calls"
}
```

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	oldPassword—the user's old password, encoded using base64. newPassword—the user's new password, encoded using base64.	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.	/me /users/{id} (Supervisors and Admins only)

Request parameters

The following request parameters are available for the "ChangePassword" operation.

Parameter	Type	Description	Access Level
firstLogin	String	Allow the user to change his or her own password, when trying to login for the first time after the user was created the user's password was changed by an administrator. Valid values: **true|false** Default value: **false**	POST

Examples

The following changes the password of a user.

Request:

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

Response:

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

The following changes the password when a user logs in for the first time.

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	**place**—optional attribute that specifies the place name in Configuration Server. **loginCode**—optional attribute that specifies the agent login to use in the operation. **queue**—optional attribute that specifies the queue number. **channels**—string array that specifies a set of channels to use for the operation (for example, "voice").	This operation will perform the following steps: 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: 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	For each device: 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 the 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 the 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 cosult 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".
InitiateTransfer	**destination** — the destination party number for a transfer 'destination': {'phoneNumber': 'XXX'}	Initiates a transfer to the specified destination.	**/me/calls/{id}** — an interaction of type "call".
CompleteTransfer	**consultCallUri** – an optional parameter specifying the URI of a second call used to complete the transfer	Completes a transfer	**/me/calls/{id}** — an interaction of type "call".

Device/Channel Management

Operation	Attributes	Description	Valid URIs
DoNotDisturbOn		Sets "Do Not Disturb" on.	**/me**—set "Do Not Disturb" for all of a user's devices and channels. **/devices/{id}**—set "Do Not Disturb" on a specific voice device. **/channels/{id}**—set "Do Not Disturb" on a specific channel.
DoNotDisturbOff		Sets "Do Not Disturb" off.	**/me**—turn off "Do Not Disturb" for all of a user's devices and channels. **/devices/{id}**—turn off "Do Not Disturb" on a specific voice device. **/channels/{id}**—turn off "Do Not Disturb" on a specific channel.
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 nofication 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: *PasswordChangeSuccessful*: A password change operation has been executed successfully for the current user. User should use the new password upon next login. *PasswordChangeError*: 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.

The following notification is available via the **/v2/users** topic:

Attribute	Description	Applies to notification type
notificationType	The type of notification that has been received. Possible values: *PasswordChangeSuccessful*: A password change operation has been executed successfully for the referenced user. User should use the new password upon next login. *PasswordChangeError*: 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 has 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. It 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—Dual-tone multi-frequency (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 occured.
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"
}
```

Queues API

Attributes

Name	Type	Description	Required
id	String	Queue ID.	N
name	String	Queue name.	Y
description	String	Queue description.	N
channel	String	A name of the channel this queue belongs to.	Y
phoneNumber	String	Phone number this queue is assigned to.	Y*
internal	Boolean	A flag showing if this queue is an internal queue or it has an external phone number assigned.	Y*

***Note:** External phone numbers can be only assigned to external queues. For example, if you specify the "phoneNumber" attribute when creating a queue, you are not allowed to set the "internal" flag to "true". Conversely, if you set the "internal" flag to "true", you are not allowed to set the "phoneNumber" attribute - the queue number will be generated automatically in the following format: "0XXXX", where XXXX is any number from 4000 to 9999.

Subresouces

Attribute	Type	Description
settings	Object	Settings object for this queue.
routingTemplate	Object	The entire routing template object this queue is associated with.

Operations

List

This operation retrieves all available queues for a contact center.

Method

GET

Request mapping

.../api/v2/queues?fields={fields}&[resolveUris={resolveUris}]

Permissions

Contact Center Admin

Request parameters

Name	Required	Description
fields	Y	A comma-separated collection of object fields (or a "*" for all fields) to be retrieved.
resolveUris	N	A comma-separated collection of object URIs (or a "*" for all URIs) to be resolved.

Example

Request:

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

Response:

```
{
        "statusCode":0,
        "queues":[{
                "id":"...",
                "name":"...",
                "channel":"voice",
                ...
        },{
                "id":"...",
                "name":"...",
                "channel":"voice",
                ...
        },{

        ...

        {
                "id":"...",
                "name":"...",
                "channel":"voice",
                ...
        }]
}
```

List URI's

This operation retrieves URIs for all available queues for a contact center.

Method

GET

Request mapping

.../api/v2/queues

Permissions

Contact Center Admin

Example

Request:

```
GET .../api/v2/queues
```

Response:

```
{
        "statusCode":0,
        "uris":[
                "http://.../api/v2/queues/<queue1_id>",
                ...
                "http://.../api/v2/queues/<queueN_id>"
        ]
}
```

Read

Reads an existing queue from the current contact center.

Method

GET

Request mapping

.../api/v2/queues/{id}?[fields={fields}]&[resolveUris={resolveUris}]

Permissions

Contact Center Admin

Request parameters

Name	Required	Description
id	Y	The ID of the queue.
fields	N	A comma-separated collection of object fields (or a "*" for all fields) to be retrieved.
resolveUris	N	A comma-separated collection of object URIs (or a "*" for all URI's) to be resolved.

Example

Request:

```
GET .../api/v2/queues/<queue_id>?resolveUris=true
```

Response:

```
{
        "statusCode":0,
        "queue":{
                "id":"...",
                "name":"...",
                "description":"...",
                "channel":"voice",
                "internal":true,
                "phoneNumberDetails": {...},
                "phoneNumber":"03445"
        }
}
```

Call Recording Settings

Request mapping

.../api/v2/queues/{id}/settings/call-recording

Permissions

- Contact Center Admin

Request parameters

Name	Required	Descriptin
id	Yes	ID of a queue.

Read

Retrieves call recording settings for a queue.

Method

GET

Example

Request:

```
GET .../api/v2/queues/<queue_id>/settings/call-recording
```

Response:

```
{
        "statusCode":0,
        "key":"callRecording",
        "settings":{
                "callRecording":{
                        "percentage":80
                }
        }
}
```

Update

Updates call recording settings for a queue.

Method

PUT

Example

Request:

```
PUT .../api/v2/queues/<queue_id>/settings/call-recording

{
        "percentage":75
}
```

Response:

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

Statistics API

REST API for Gathering Statistics

The name of the statistic is used for statistic definition in the `statistics.yaml` file. See the list of statistics to be monitored.

Query for Most Recent Statistic Values

The following request can be used to retrieve the most recent statistic values. For example:

HTTP Operation: **GET**

URL: http://example.com/api/v1/stats/{objectId}

Here is an example showing the results of a query with a specific object ID:

```
{
       "objectId": "{objectId}",
       "statusCode":0,
       "statistics": [
           {
               "statistic": "numAnsweredCalls",
               "value": 222,
               "time": 1345144080379
           },
           {
               "statistic": "numCalls",
               "value": 500,
               "time": 1345143997464
           },
           {
               "statistic": "numDroppedCalls",
               "value": 200,
               "time": 1345144064936
           }
       ]
   }
```

Query for Range of Statistic Values

The following request can be used to retrieve the range of values for a given object and statistic:

HTTP Operation: **GET**

URL: http://example.com/api/v1/stats/{objectId}/{statisticName}

There are some optional request parameters that can be added as follows:

Parameter	Description
numValues	The default is '1' if this parameter is not provided. This can be set to a larger number in order to retrieve multiple statistic values. For example, if set to "100", it would return the last 100 values, if there are that many.
from	This can be used to set the start of time range that you are interested in.
to	This can be used to set the end of time range that you are interested in.

Example requests:

Request:	/stats/{objectId}/numCalls?numValues=10
Description:	This will return the 10 most recent values for the **numCalls** statistic

Request:	/stats/{objectId}/numCalls?from=1343779200000&to=13464575
Description:	This will return all statistic values for the statistic **numCalls** for the month of August 2012

Request:	/stats/{objectId}/numCalls?from=1343779200000&to=13464575
Description:	This will return the 3 most recent (last) statistic values for the statistic **numCalls** within the month of August 2012

The format response to the **GET** request is the same regardless of the parameters supplied. For example:

```
{
    "statusCode":0,
    "objectId": "888d13ad-e8aa-4532-a99a-eb8c6590e65a",
    "statistics": [
        {
            "statistic": "numCalls",
            "values": [
                {
                    "value": 300,
                    "time": 1344866399143
                },
                {
                    "value": 200,
                    "time": 1344866389143
                }
            ]
```

```
        }
    ]
}
```

Getting Service level statistic

To get the service level statistic the following entries **must** be present in `statistics.yaml` file:

```
---
#service level
name: ServiceLevel
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: TotalNumberInTimeRangePercentage
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallAnswered
    subject: DNAction
    timeRangeLeft: 0
    timeRangeRight: 60
---
name: ServiceLevel
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_Q
statisticDefinitionEx:
    category: TotalNumberInTimeRangePercentage
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallAnswered
    subject: DNAction
    timeRangeLeft: 0
    timeRangeRight: 60
```

They are already in the `statistics.yaml` file shipped. To get the list of the 100 most recent service level statistic values, use range query:

```
GET .../api/v1/stats/{objectId}/ServiceLevel?numValues=100
```

where {objectId} should be substituted with id of skill or queue.

Skills should be provisioned using our provisioning UI/API or, if configured manually in Configuration Manager or Genesys Administrator, must satisfy restrictions specified in the Web Services Configuration Options.

List of statistics to be monitored

Entry Format

Two modes of statistics definitions are supported, "simple" and "extended". In "simple" mode, the statistic type is defined in Configuration Manager and the `statistics.yaml` entry defines the additional statistics options and the statistic name in the API. In "extended" mode, the full definition of the statistic is provided in `statistics.yaml` entry. In this case there is no need to use Configuration Manager for statistic definition. The switch between modes is defined basing on presence of **statisticDefinitionEx** property. If the specified mode is in "extended" mode, **statisticDefinitionName** is ignored if present.

Simple Mode Entry Format

[+] Simple Mode Entry Format

Example:

```
name: test
statisticDefinitionName: TestAgentStat
objectType: QUEUE
timeProfile: OneDay
notificationMode: PERIODICAL
notificationFrequency: 10
timeRange: Range0-60
timeRange2: Range0-5
```

The properties meaning are:

- name - the name of statistic as it appears in API request
- statisticDefinitionName - the name of statistic type (definition) as it's in CME (see Stat Server User's Guide)
- objectType - the type of objects statistic is applied. Possible values: AGENT|QUEUE|SKILL_Q|SKILL_AG:
  - AGENT - statistic is monitored for agents
  - QUEUE - statistic is monitored for queues
  - SKILL_AG - statistic is monitored for agentGroup, associated with skill
  - SKILL_Q - statistic is monitored for virtual queue, associated with skill
- notificationMode - PERIODICAL
- notificationFrequency - the frequency of notification (in seconds)
- timeProfile - the name of timeProfile in Configuration Manager used for statistic aggregation (see Stat Server User's Guide)
- timeRange/timeRange2 - the timeranges names which are defined in Configuration Manager and are

used for calculating statistic (see Stat Server User's Guide)

Extended mode entry format

[+] Extended Mode Entry Format

To use this mode replace the **statisticDefinitionName** string with **statisticDefinitionEx** containing full definition of statistic type. Here is the sample format:

```
name: testServiceLevel
statisticDefinitionEx:
    category: TotalNumberInTimeRangePercentage
    mainMask: CallAnswered
    subject: DNAction
    intervalType: GrowingWindow
    dynamicTimeProfile: "0:00"
    timeRangeLeft: 0
    timeRangeRight: 60
objectType: QUEUE
notificationMode: PERIODICAL
notificationFrequency: 10
```

Fields **name**, **objectType**, **notificationMode**, **notificationFrequency** have the same meaning as for simple mode. The **statisticDefinitionEx** defines the statistic to be collected. Note the indentation here. No tabs can be used (See YAML 1.1 Spec).

The supported properties are:

Option	Type	Mandatory	Default Value	Note
category	String	Y		The statistic's category
mainMask	comma-delimited list	Y		The statistic's main mask
relativeMask	comma-delimited list	N	Empty list	The statistic's relative mask
subject	String	Y		The statistic's subject
intervalType	GrowingWindow OR SinceLogin OR SlidingSelection OR SlidingWindow	N		The statistic's interval type
dynamicTimeProfile	String	N		The time profile (interval) value to be used
timeRangeLeft	Integer	N	null	The "from/left/start/lower" value of TimeRange interval
timeRangeRight	Integer	N	null	The "to/right/end/upper" value of

Option	Type	Mandatory	Default Value	Note
				TimeRange interval
timeRangeLeft2	Integer	N	null	The "from/left/ start/lower" value of TimeRange2 interval
timeRangeRight2	Integer	N	null	The "to/right/end/ upper" value of TimeRange2 interval
dynamicFilter	String	N	null	filter to be used
distinguishByConnId	Boolean	N	null	look for DCID in StatServerUserGuide

Note that those properties are used for defining the statistics. Not all combinations are supported by StatServer. See Stat Server user guide for more info.

Queue Statistics

[+] Total_Answered

The number of customer interactions that entered a service type and were accepted, answered, or pulled by agent (s)/agent group (s) within the reporting time interval.

Definition in Configuration Server:

```
name: Total_Answered
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallAnswered
    subject: DNAction
```

[+] Total_Abandoned

The number of interactions that were terminated by the customer while waiting on the service type during the reporting time interval. This stat type excludes interactions that were distributed to an agent and then abandoned before the agent could answer (Call Abandoned While Ringing).

Definition in Configuration Server:

```
name: Total_Abandoned
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
```

```
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallAbandoned
    subject: DNAction
```

[+] Current_In_Queue

The number of customer interactions that are currently waiting in queue.

Definition in Configuration Server:

```
name: Current_In_Queue
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: CallWait
    subject: DNAction
```

No time profile.

[+] CurrMaxCallWaitingTime

The maximum waiting time for customer interactions currently waiting on a service Definition in Configuration Server:

```
name: CurrMaxCallWaitingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: CurrentMaxTime
    mainMask: CallWait
    relativeMask: CallWait
    subject: DNAction
```

No time profile.

[+] ServiceLevel

Ratio of call answered within given interval (specified by first time range) compared to (total number answered + total abandoned after delay specified by second time range).

```
name: ServiceLevel
statisticDefinitionEx:
    category: ServiceFactor1
    subject: DNAction
    intervalType: GrowingWindow
    dynamicTimeProfile: "0:00+1:00"
    timeRangeLeft: 0
    timeRangeRight: 120
    timeRangeLeft2: 0
    timeRangeRight2: 10
objectType: QUEUE
notificationMode: PERIODICAL
```

```
notificationFrequency: 10
```

[+] AverageWaitingTime

Average wait time for customer interactions what were entered and distributed/abandoned on a service type during reporting time interval Note: this will calculate wait time for all calls, including ones that were abandoned.

Definition in Configuration Server:

```
name: AverageWaitingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallWait
    relativeMask: CallWait
    subject: DNAction
```

Skills Statistics

[+] CurrentNotReadyAgents

Number of not ready agents with skill

```
name: CurrentNotReadyAgents
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: NotReadyForNextCall
    subject: AgentStatus
```

Time profile - not used.

[+] CurrentReadyAgents

Number of ready agents with skill

```
name: CurrentReadyAgents
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: WaitForNextCall
    subject: AgentStatus
```

Time profile - not used.

[+] CurrNumberInCall

Number of agents with skill in call

```
name: CurrNumberInCall
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, CallRinging,
CallDialing
    subject: AgentStatus
```

Time profile - not used.

[+] LongestIdleTime

The longest time an agent is currently waiting to receive a call (evaluated maximum of current idle time for all agents in group which are currently waiting to receive a call, may be 0 if no idle agents are present in group at the moment).

```
name: LongestIdleTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentMaxTime
    mainMask: WaitForNextCall
    subject: AgentStatus
```

Time profile - not used.

[+] AverageHandlingTime

The average amount of time during the reporting interval that agent/agent groups spent on customer interactions including After Call Work (ACW)/ wrap-up status.

```
name: AverageHandlingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, AfterCallWork
    relativeMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
    subject: DNStatus
```

[+] ServiceLevel

Ratio of call answered within given interval (specified by first time range) compared to (total number answered + total abandoned after delay specified by second time range).

```
name: ServiceLevel
statisticDefinitionEx:
    category: ServiceFactor1
    subject: DNAction
    intervalType: GrowingWindow
    dynamicTimeProfile: "0:00+1:00"
    timeRangeLeft: 0
    timeRangeRight: 120
    timeRangeLeft2: 0
    timeRangeRight2: 10
objectType: SKILL_Q
notificationMode: PERIODICAL
notificationFrequency: 10
```

Agent Statistics

[+] Other agent statistics

The definitions of other agent statistics are self-describing.

```
#agent
name: AverageHandlingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, AfterCallWork
    relativeMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
    subject: DNStatus
---
name: Productivity
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: AverageNumberPerRelativeHour
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInbound,CallOutbound,CallInternal,CallConsult,CallUnknown
    relativeMask: '*,~LoggedOut,~NotMonitored'
    subject: AgentStatus
---
name: InboundCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInbound
    subject: DNAction
---
name: InternalCalls
notificationFrequency: 10
```

```
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInternal
    subject: DNAction
---
name: OutboundCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallOutbound
    subject: DNAction
---
name: ConsultCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallConsult
    subject: DNAction
---
name: ReadyDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: WaitForNextCall
    subject: AgentStatus
---
name: WrapDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: AfterCallWork
    relativeMask: AfterCallWork
    subject: AgentStatus
---
name: TalkDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
```

```
    subject: DNAction
---
name: HoldDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask:
CallOnHoldInbound,CallOnHoldOutbound,CallOnHoldInternal,CallOnHoldConsult,CallOnHoldUnknown
    subject: DNAction
```

Resources

See Enabling Reporting in the *Workspace Web Edition & Web Services Deployment Guide*.

See Configuring Virtual Queues & Virtual Agent Groups in the *Workspace Web Edition & Web Services Deployment Guide*.

This list is the minimal list - it contains only statistics which are required for current version of UI.

[+] Sample statistics.yaml file

Internal statistics used for tracing the state of agent.

```
#internal stats
name: CurrentTargetState
statisticDefinitionEx:
    category: CurrentTargetState
    mainMask: '*'
    subject: DNStatus
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
objectType: AGENT
notificationMode: IMMEDIATE
notificationFrequency: 0
---
name: CurrentAgentState
notificationFrequency: 0
notificationMode: IMMEDIATE
objectType: AGENT
statisticDefinitionEx:
    category: CurrentState
    mainMask: '*'
    subject: DNAction
---
#queue
name: Total_Answered
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
```

```
    mainMask: CallAnswered
    subject: DNAction
---
name: Total_Abandoned
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallAbandoned
    subject: DNAction
---
name: Current_In_Queue
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: CallWait
    subject: DNAction
---
name: CurrMaxCallWaitingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: CurrentMaxTime
    mainMask: CallWait
    relativeMask: CallWait
    subject: DNAction
---
name: AverageWaitingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: QUEUE
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallWait
    relativeMask: CallWait
    subject: DNAction
---
#Skill AG
name: CurrentNotReadyAgents
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: AfterCallWork, NotReadyForNextCall
    subject: AgentStatus
---
name: CurrentReadyAgents
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: WaitForNextCall
    subject: AgentStatus
```

```
---
name: CurrNumberInCall
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, CallRinging,
CallDialing
    subject: AgentStatus
---
name: CurrentNumberLoggedInAgents
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentNumber
    mainMask: '*,~LoggedOut,~NotMonitored'
    subject: AgentStatus
---
name: LongestIdleTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: CurrentMaxTime
    mainMask: WaitForNextCall
    subject: AgentStatus
---
name: AverageHandlingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: SKILL_AG
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, AfterCallWork
    relativeMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
    subject: DNStatus
---
#agent
name: AverageHandlingTime
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: AverageTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound, AfterCallWork
    relativeMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
    subject: DNStatus
---
name: Productivity
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: AverageNumberPerRelativeHour
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInbound,CallOutbound,CallInternal,CallConsult,CallUnknown
```

```
    relativeMask: '*,~LoggedOut,~NotMonitored'
    subject: AgentStatus
---
name: InboundCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInbound
    subject: DNAction
---
name: InternalCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallInternal
    subject: DNAction
---
name: OutboundCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallOutbound
    subject: DNAction
---
name: ConsultCalls
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalNumber
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallConsult
    subject: DNAction
---
name: ReadyDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: WaitForNextCall
    subject: AgentStatus
---
name: WrapDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
```

```
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: AfterCallWork
    relativeMask: AfterCallWork
    subject: AgentStatus
---
name: TalkDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask: CallUnknown, CallConsult, CallInternal, CallOutbound, CallInbound
    subject: DNAction
---
name: HoldDuration
notificationFrequency: 10
notificationMode: PERIODICAL
objectType: AGENT
statisticDefinitionEx:
    category: TotalAdjustedTime
    dynamicTimeProfile: "0:00"
    intervalType: GrowingWindow
    mainMask:
CallOnHoldInbound,CallOnHoldOutbound,CallOnHoldInternal,CallOnHoldConsult,CallOnHoldUnknown
    subject: DNAction
---
#service level
name: ServiceLevel
statisticDefinitionEx:
    category: ServiceFactor1
    subject: DNAction
    intervalType: GrowingWindow
    dynamicTimeProfile: "0:00+1:00"
    timeRangeLeft: 0
    timeRangeRight: 120
    timeRangeLeft2: 0
    timeRangeRight2: 10
objectType: QUEUE
notificationMode: PERIODICAL
notificationFrequency: 10
---
name: ServiceLevel
statisticDefinitionEx:
    category: ServiceFactor1
    subject: DNAction
    intervalType: GrowingWindow
    dynamicTimeProfile: "0:00+1:00"
    timeRangeLeft: 0
    timeRangeRight: 120
    timeRangeLeft2: 0
    timeRangeRight2: 10
objectType: SKILL_Q
notificationMode: PERIODICAL
notificationFrequency: 10
```

Workbins

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": [
  {
      'workbinId': 'RmFjZWJvb2sgV29ya2JpbiBJblByb2dyZXNzL1ByaXZhdGVWaWV3',
      'active': 1,
      'workbinName': 'FacebookWorkbinInProgress',
      'viewName': 'FacebookWorkbinInProgress/PrivateView',
      'workbinType': 1
  },
  {
      'workbinId': 'VHdpdHRlciBXb3JrYmluIERyYWZ0L1ByaXZhdGVWaWV3',
      'active': 1,
      'workbinName': 'TwitterWorkbinDraft',
      'viewName': 'TwitterWorkbinDraft/PrivateView',
      'workbinType': 1
  },
  {
      'workbinId': 'VHdpdHRlciBXb3JrYmluIEluUHJvZ3Jlc3MvUHJpdmF0ZVZpZXc=',
      'active': 1,
      'workbinName': 'TwitterWorkbinInProgress',
      'viewName': 'TwitterWorkbinInProgress/PrivateView',
      'workbinType': 1
  },
  {
      'workbinId': 'VGVzdFdvcmtiaW4vUHJpdmF0ZVZpZXc=',
      'active': 1,
      'workbinName': 'TestWorkbin',
      'viewName': 'TestWorkbin/PrivateView',
      'workbinType': 1
  },
  {
      'workbinId': 'RmFjZWJvb2sgV29ya2JpbiBEcmFmdC9Qcml2YXRlVmlldw==',
```

```
      'active': 1,
      'workbinName': 'FacebookWorkbinDraft',
      'viewName': 'FacebookWorkbinDraft/PrivateView',
      '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": "<r4mo20o0o6j1niq.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": "00001a8QWK4A003P",
          "RStrategyDBID": "188",
          "FromPersonal": "",
          "InteractionState": 0
      }
 ]
}
```

Failure

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

For details on the statusCode value, refer to the All Methods section of the Return Values page.

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, refer to the All Methods section of the Return Values page.

PullInteraction

This request pulls an interaction from a workbin and assigns 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, refer to the All Methods section of the Return Values page.

SubscribeWorkbinEvents

This request subscribes 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, refer to the All Methods section of the Return Values page.

CometD Notifications

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

Message:

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

UnsubscribeWorkbinEvents

This request unsubscribes 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, refer to the All Methods section of the Return Values page.