Genesys Interaction Recording API

Genesys Interaction Recording 8.5.2

12/30/2021

Table of Contents

Genesys Interaction Recording API Reference

Welcome to the Genesys Interaction Recording Application Programming Interface (API) Reference. The purpose of this API reference is to explain:

- The resources and methods available to developers.
- The initialization and maintenance of the session between the Screen Recording Service and Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).
- The API provided by Interaction Recording Web Services.

The Genesys Screen Recording Service (SRS) API associates the desktop with a specific agent. As a result, voice recording automatically includes screen recording. That is, SRS is designed to run in the background on an Agent's PC to capture on-screen activity while the Agent handles voice interactions with the customer. SRS provides a communication channel over HTTP(S) for Agent Desktop applications that can initiate and manage the session between SRS and Interaction Recording Web Services (Web Services). For information on how to use the Screen Recording Service API, see Using the Screen Recording API.

> **Important**
>
> The Screen Recording Service API does not provide recording control functionality and therefore you cannot trigger recording using this API. To generate recording, you must have an associated interaction. The events related to recording control (start, stop) are generated by Interaction Recording Web Services (Web Services) and are passed to the Screen Recording Service using a different channel.

Each category presents information about relevant operations, related resources, request parameters, and return values.

This guide describes the following:

- Screen Recording Service API
- Interaction Recording Web Services API
- API Responses

Screen Recording Service API

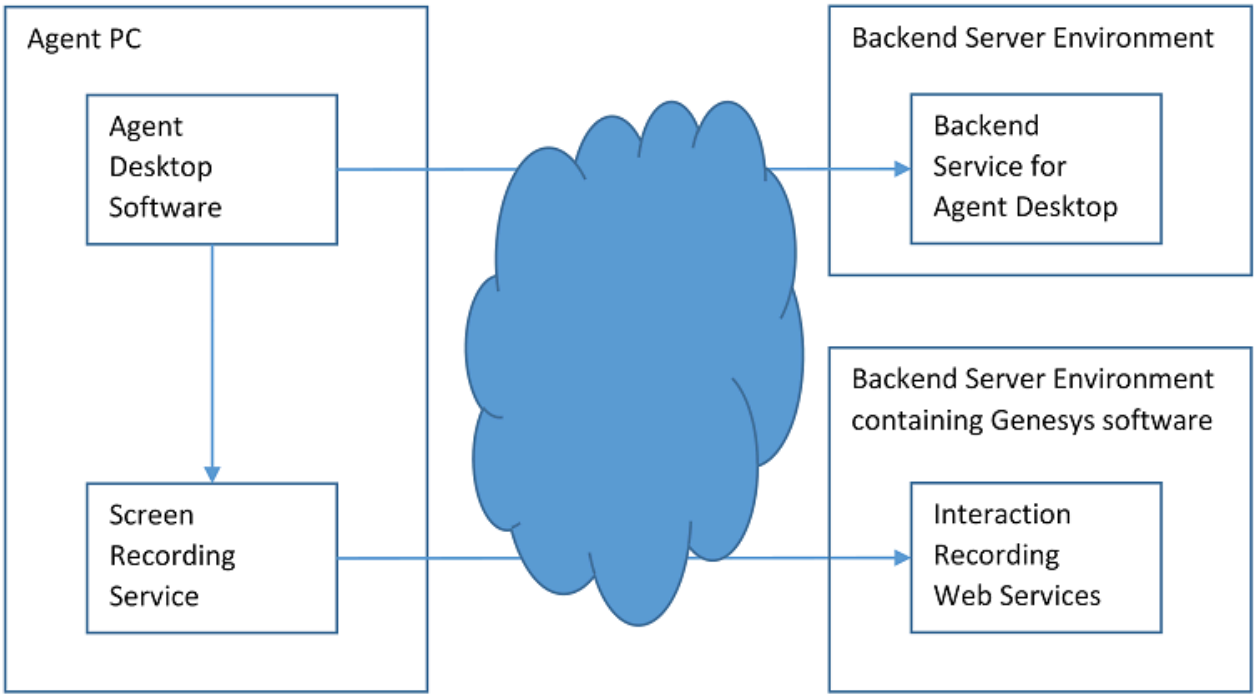
The Screen Recording Service API provides information on how to integrate screen recording functionality into a custom agent desktop. It contains the following sections:

Using the Screen Recording API

The Screen Recording feature within Genesys Interaction Recording (GIR) allows customers to capture the entire agent screen, including multiple monitors, for both voice and non-voice interactions delivered to the agent desktop. Typically, an agent uses either Workspace Desktop Edition (WDE) or Workspace Web Edition (WWE), which are already integrated with the Screen Recording Service (SRS), as their desktop. However, some customers may have their own custom agent desktop application. This section provides information on how to integrate screen recording functionality into a custom agent desktop.

Components

The following is a simplified component diagram.

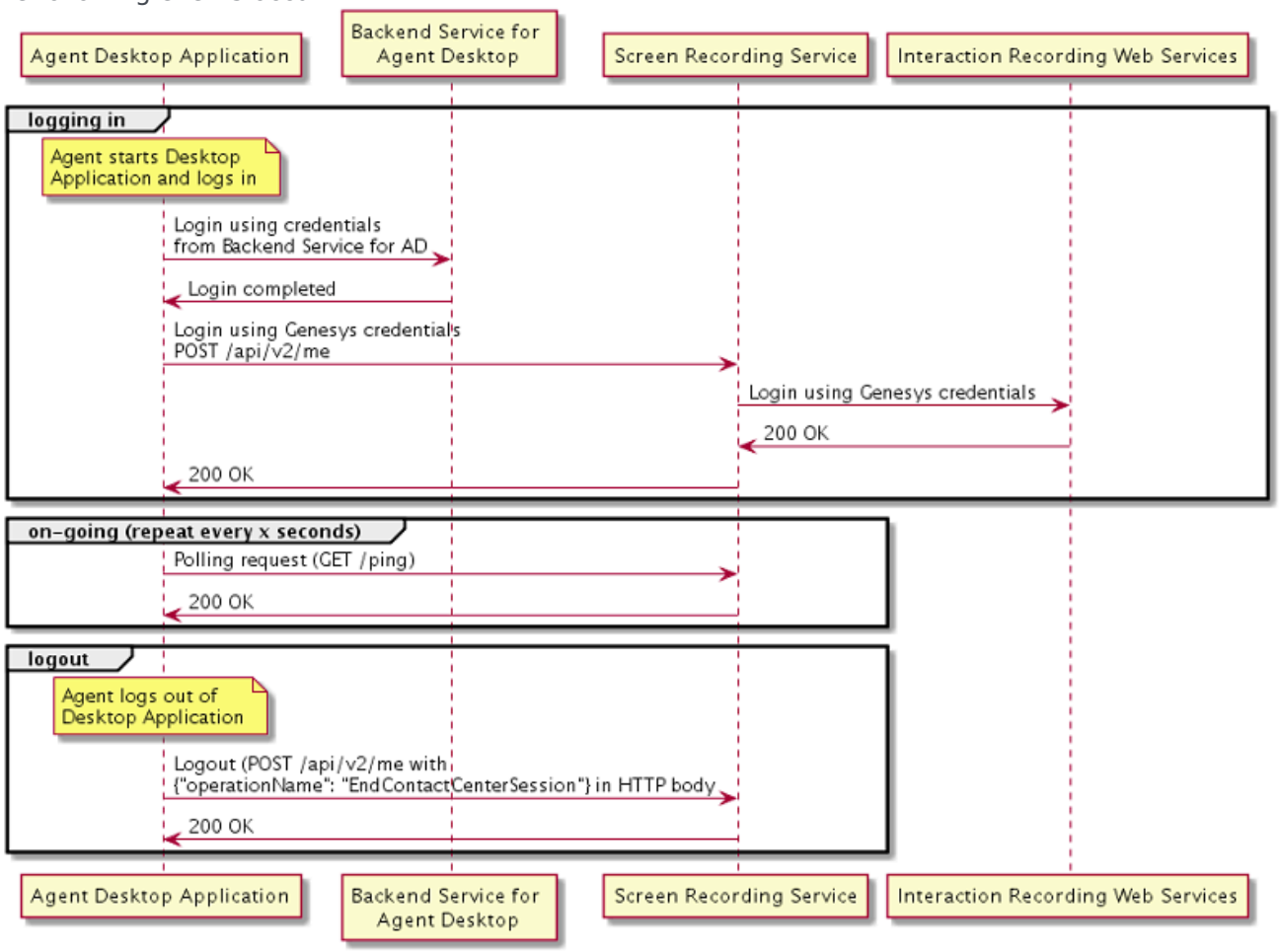


- **Agent Desktop Software**: The client application running on the agent's workstation. The agent desktop provides users access to interaction information as the interaction is delivered to the agent. This application also provides access to the functions, processes and related applications that are needed to successfully handle a customer interaction.
- **Screen Recording Service (SRS)**: A Windows service that is installed on the agent's machine. SRS receives instructions from the agent desktop and Interaction Recording Web Services (RWS) to control when to start and stop the screen recording on the agent's machine.

- **Interaction Recording Web Services (RWS)**: A software service running in the Genesys environment to control SRS and to store and manage recording files.
- **Backend Service for Agent Desktop**: The software services that the agent desktop application interacts with. Depending on how the software is deployed, it may not be in the same environment as RWS.

Call Flow Diagram

As shown in the figure, the agent desktop application must make appropriate API calls to SRS when the following events occur:



- **Agent login**: When an agent logs into the agent desktop, the Login API must be invoked.
- **Polling**: As the agent is using the agent desktop, the Polling API must be invoked periodically so that the timeout-and-clean-up mechanism is not triggered.
- **Agent logout**: When the agent logs out of the agent desktop, the Logout API must be invoked.

Cross-Origin Resource Sharing (CORS)

If the agent desktop is implemented as a web application running on the browser, this web application makes an HTTP or HTTPS request to the SRS running on the same machine as the browser. Make sure your web application is making requests to SRS as per the W3C CORS standard. Note that based on the CORS specification, if the agent desktop is communicating with its backend services using HTTP, then the request to SRS must also use HTTP. If the agent desktop is communicating with its backend services using HTTPS, then the request to SRS must also use HTTPS.

SRS supports the CORS pre-flight OPTIONS requests.

Cross-Site Request Forgery (CSRF) headers

To prevent Cross-Site Request Forgery (CSRF) attack, the following pair of headers can optionally be used to protect a user from unwittingly terminating a session established with SRS:

```
X-Support-CSRFP
X-CSRF-Token
```

At the time a login request is sent to SRS, if the HTTP request contains the "`X-Support-CSRFP: true`" header, then the HTTP response will contain the "`X-CSRF-Token: <token>`" header with a token that is used only for this session.

Within the same HTTP session, subsequent logout requests must contain the "`X-CSRF-Token: <token>`" request header with the same token initially obtained in the login response header.

A request may be rejected if the HTTP header is not included in the POST request.

Hot Seating vs. Default Place

When an agent logs into the agent desktop, depending on the contact center configuration settings, the agent may be asked to choose the "`place`" where he is located for the login session. This allows calls to be correctly routed to the phone numbers associated with the place at which the agent is located for this login session. This is known as hot seating.

If the agent is not prompted to enter the "`place`" at the time of login, then the default place configured for this agent is always used.

Client Login API

The login API is described in the Login Request with Configuration section.

Hot Seating

When the contact center is using hot seating, the login request payload must contain at least the

following parameters.

Parameter	Description
`server`	The server name and port number, and the protocol (http vs. https) that SRS uses to access RWS. This parameter contains a URL, for example, "https://myserver.com:443". The servers that the URL accesses must be in the same data center as the Genesys software that is working with the backend service for the agent desktop.
`place`	The name of the place that the agent selected at the time of login. This parameter must match the name of the `place` object configured in the Genesys configuration or when using Agent Setup.
`devices`	The names of the DN and the switch objects defined in the Genesys configuration.

> **Important**
>
> If the "`place`" contains only one DN, then the "`devices`" attribute is not required.
> If the "`place`" contains multiple DNs, then both the "`place`" and "`devices`" attributes are required. Every item in the devices list (that is, the "DN" object belonging to the "`switch`" object) must belong to the specified place.

Example Request Payload

```
[
  {
    "name": "server",
    "value": "https://rws.dc1.example.com"
  },
  {
    "name": "place",
    "value": "alicePlace"
  },
  {
    "name": "devices",
    "value": [
      {
        "dn": "DN1",
        "switch": "SwitchA"
      },
      {
        "dn": "DN2",
        "switch": "SwitchB"
      }
    ]
  }
]
```

> **Important**
>
> If an agent switches place in the middle of an active session, the agent desktop application must send a login API request to SRS again to monitor events from the new place and to control the recording behavior.

Default Place

If a contact center does not use hot seating, the agent will always use the assigned default place. In this case, the login request should contain only the "server" parameter.

Parameter	Description
server	The server name and port number, and the protocol (http vs. https) that SRS uses to access the RWS. This parameter contains a URL, for example, "https://myserver.com:443". The servers that the URL accesses must be in the same data center as the Genesys software that is working with the backend service for the agent desktop.

Example Request Payload

```
[
  {
     "name": "server",
     "value": "https://rws.dc1.example.com"
  }
]
```

Disaster Recovery

When SRS invokes the login API to access RWS, the "server" parameter indicates where SRS should send requests to RWS. The "peer_server" parameter should be used to specify the address of the backup RWS in the backup data center. In case of a disaster such as the entire data center becoming unavailable, the agent desktop application may alternatively attempt disaster recovery by sending a login request to SRS with the address of the recovery data center where the backup RWS resides in the "server" parameter.

Disaster recovery can be used for both hot seating and default place configurations.

Client Polling API

The polling API is documented in the Client Polling API section.

SRS has a timeout mechanism to terminate screen recordings if it does not receive a polling API call from the agent desktop application. Therefore, while an agent remains logged in to the agent desktop, the agent desktop application must send the polling API periodically to prevent SRS from

timing out. The default timeout value in SRS is 60 seconds, so the agent desktop application must invoke the polling API every 30 seconds.

Client Logout API

The logout API is documented in the Logout Request section.

When an agent logs out of the agent desktop, the agent desktop application must send the logout API to SRS. This API uses the same HTTP endpoint as the login API but with a different request payload, as shown below.

```
{"operationName": "EndContactCenterSession"}
```

Client Login API

Overview

Allows an Agent Desktop application to pass the agent's credentials. These credentials are used to start and end a Screen Recording Service session with Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).

If an error is sent in response, Interaction Recording Web Services (Web Services) attempts to log the agent in before abandoning the operation.

The Screen Recording Service supports:

- CORS requests and appends a correct set of CORS headers if it receives requests with the expected Origin header. If you're using Interaction Recording Web Services, see Cross-Origin Resource Sharing in this guide. If you're using Web Services, see Cross-Origin Resource Sharing in the Web Services API Reference.
- CSRF protection using X-Support-CSRFP and X-CSRF-Token headers.

Important

- The third party client application must communicate with the SR Service (SRS) through the local machine (`127.0.0.1:<SRS port>`), and not through the Interaction Recording Web Services (RWS) machine when making API calls. The third party application should only inform SRS that a user is logging in and provide the username and password. SRS will interact with RWS to know when to start/stop a screen recording associated with this specific user.
- Single Sign-on integration is not supported. Consequently, even if a user has previously authenticated to an identity provider, the user will have to supply credentials to the SR Service again for validation through RWS with the Configuration Server.

Pre-flight Request

Request URL	/api/v2/me
HTTP Method	OPTIONS
Mandatory Header	Origin: server-url

Response

200 OK

Example

Request

```
OPTIONS api/v2/me/
```

Response

```
200 OK
```

Login Request

Request URL	/api/v2/me
HTTP Method	GET
Mandatory Header	Authorization: Basic xxxxx= Origin: server-url
Optional Header	X-Support-CSRFP: True

Response

200 OK

Example

Request

```
GET /api/v2/me
```

Response

```
200 OK
```

Login Request with Configuration

> **Important**
> The SR Service supports Hot Seating. Please refer to the appropriate documentation to determine if the client / agent desktop being used provides support for Hot Seating integration with the SR Service.

Request URL	/api/v2/me
HTTP Method	POST
Mandatory Header	Authorization: Basic xxxxx= Origin: server-url
Optional Header	X-Support-CSRFP: True

Response

200 OK

Example

Request

```
POST /api/v2/me
```

Response

```
200 OK
```

Request Payload

```
[
  {
     "name": "server",
     "value": "https://rws.dc1.example.com"
  },
  {
     "name":"peer_server",
     "value":"https://rws.dc2.example.com"
  },
  {
     "name": "place",
     "value": "Place"
  },
  {
     "name": "devices",
     "value": [
```

```
        {
           "dn": "DN1",
           "switch": "Switch"
        },
        {
           "dn": "DN2",
           "switch": "Switch"
        }
     ]
  }
]
```

Name	Description	Example
"server"	The Interaction Recording Web Services server address and port.	{"name":"server","value":"https://web.services.s
"peer_server"	The Interaction Recording Web Services server address and port in the backup data center. **Note:** This parameter is not applicable for single data center deployments.	{"name":"peer_server","value":"https://dc2.web.
"place"	The place used by the agent.	{"name":"place","value":"Place"}
"devices"	The list of devices and switches used by the agent.	{"name":"devices", "value":[{ "dn": "DN1", "switch": "Switch" },{ "dn": "DN2", "switch": "Switch" }]}

> **Important**
> All of the parameters are optional.

Version Information

Request URL	/version
HTTP Method	GET

Response

200 OK

Example

Request

```
GET /version
```

Response

```
200 OK
{"version": "8.5.zyy.xx"}
```

Logout Request

Request URL	/api/v2/me
HTTP Method	POST
Mandatory Header	Authorization: Basic xxxxx= Origin: server-url Content-Type: application/json
Optional Header	Cookie: <session cookie> The Cookie Header can replace the mandatory Authorization header.
Optional Header	X-CSRF-Token: <token> This is mandatory if "X-Support-CSRFP" was used to establish the session.

Response

200 OK
403 Forbidden

Example

Request

```
POST api/v2/me/
```

Response

```
200 OK
```

Request payload

```
{"operationName": "EndContactCenterSession"}
```

Client Polling API

Overview

Allows the Agent Desktop to determine if the Screen Recording Service is available.

If an error is sent in response, the Agent Desktop attempts to poll the client before abandoning the operation.

The Screen Recording Service supports Cross-Origin Resource Sharing requests and appends a correct set of CORS headers if it receives requests with the expected Origin header. If you're using Interaction Recording Web Services, see Cross-Origin Resource Sharing in this guide for more information. If you're using Web Services, see Cross-Origin Resource Sharing in the Web Services API Reference for more information.

Every `/ping` request starts a timer in the Screen Recording Service. If the next `/ping` request is not received within the default 60 second timeout, the client disconnects from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and no longer accepts a Start/Stop Screen Recording request from the Web Service. If another `/ping` request is received after the timeout occurs, the client responds with a 401 triggering a login request from Interaction Recording Web Services (Web Services) to the client. When a new valid login request is received, SRS will establish a connection with the Web Service.

Pre-flight Request

Request URL	/ping
HTTP Method	OPTIONS
Mandatory Header	Origin: server-url

Response

200 OK

Example

Request

```
OPTIONS /ping
```

Response

```
200 OK
```

Ping Request

Request URL	/ping
HTTP Method	GET
Mandatory Headers	Authorization: Basic xxxxx= Origin: server-url
Optional Header	Cookie: <session cookie> The Cookie Header can replace the mandatory Authorization header.

Response

200 OK
401 Unauthorized

Example

Request

```
GET /ping
```

Response

```
200 OK
```

Interaction Recording Web Services API

Interaction Recording Web Services provides access to the following APIs:

- Search, Playback, and Delete API — Use this API to search for, play back, and delete recordings stored in the Genesys Interaction Recording (GIR) system.
- Settings API — You'll typically only need this API if you plan to enable recording privacy settings in the Genesys Interaction Recording API.
- Insertion API — Use this API to update the GIR database with records of existing call recordings already stored at a WebDAV storage location, so that they are accessible from the Genesys Interaction Recording user interfaces.
- Call Recording API — Use this API to control call recording functionality in your agent application.
- Recording Label API — Use this API to create and administer label definitions and label recordings.
- Recording Non-Deletion API — Use this API to tag and untag recordings for non-deletion.

The following topics are important for you to review as you'll need this information while using the Interaction Recording Web Services APIs.

Configuring Roles for Recording Users

Interaction Recording Web Services does not use standard Genesys access controls. Instead, it uses its own role-based security depending on settings in the **Annex** tab. The htcc/roles key must be defined in the **Options** of the **Person** object that you use to connect to the API. For example:

```
[htcc]
roles=supervisor
```

Role	Description
agent	Provides agent access. Agents are contact center employees who handle calls, hold chats sessions, and answer emails.
supervisor	Provides supervisor access. A resource whose primary role in the business is direct management of agents and may occasionally engage in the interaction-handling process during coaching or in emergency situations.
admin	Provides administrator access. An employee in the contact center who can create and edit other users, create reason codes, and assign skills to supervisors.
apiuser	Provides the same level of access as an administrator. Use this permission to designate an API user of system account that must be used by

Role	Description
	other server applications and this user does not represent an actual person.

> **Tip**
>
> You can link roles together as comma-separated values. For example:
>
> ```
> [htcc]
> roles=agent,supervisor,admin
> ```

Cross-Site Request Forgery protection

Interaction Recording Web Services provides protection against Cross-Site Request Forgery (CSRF) attacks by requiring a token in a custom header for all requests that modify data: PUT, POST, DELETE. Interaction Recording Web Services generates and stores this token along with the HTTP session. The token shares the life cycle of the HTTP session.

See CSRF protection for details about how to enable this security feature.

To get the CSRF token and the expected header name from Interaction Recording Web Services, just send a GET request — for example, **/api/v2/me**. The expected header name and token value are returned in two custom headers on the HTTP response: `X-CSRF-HEADER` and `X-CSRF-TOKEN`.

```
X-CSRF-HEADER: X-CSRF-TOKEN
X-CSRF-TOKEN: 4a92be65-ec55-4aa2-b9df-9518fd870f2f
```

You must cache the values of these headers because you'll need to use them on subsequent API requests that use PUT, POST, and DELETE so that Interaction Recording Web Services doesn't think the request is coming from a third party. For instance, when you attempt to perform the Insert Recording operation, you need include an HTTP header of X-CSRF-TOKEN with the corresponding value:

```
POST https://htcc-demo.genhtcc.com/internal-api/contact-centers/
57c0b771-b57c-4ea8-8655-7ef6d3c58ccc/recordings HTTP/1.1
Authorization: Basic <credentials>
X-CSRF-TOKEN: 4a92be65-ec55-4aa2-b9df-9518fd870f2f
Accept: application/json, application/xml, text/json, text/x-json, text/javascript,
text/xml
User-Agent: RestSharp/105.2.3.0
Content-Type: application/json
Host: htcc-demo.genhtcc.com
Cookie: JSESSIONID=sngukrzemiyxchpu5isbufmm;
AWSELB=854B09E30CD5CEDDEDA518240935B76DEAC5D82EC5038C4B8F22CD5165FF21C65BC292BAD05CEEB
17D7500F4A489957FB3A5C23BD09BC31CAF09526FCBEFD7CE491CD7E5B3
Content-Length: 88
Accept-Encoding: gzip, deflate
{
   ...
   request body
```

```
   ...
}
```

If you don't have that header in place, Interaction Recording Web Services returns an HTTP 403 error with a response in the Content of "Missing or invalid Csrf token".

Cookie support

In addition to the CSRF feature, Interaction Recording Web Services also requires your application to support cookies, specifically for the cookie that it returns. Without a cookie store, Interaction Recording Web Services returns the same HTTP 403 error with a message of "Missing or invalid Csrf token", even if the X-CSRF-TOKEN is specified in the HTTP Header. This is because it can't confirm that the X-CSRF-TOKEN you specify lines up with the cookie that the token is supposed to be tied to.

Read on for some sample requests and examples of how to implement CSRF protection:

Authorized request returning token headers

Request

```
GET /api/v2/me

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/35.0.1916.153 Safari/537.36
Authorization: Basic cGF2ZWxkQHJlZHdpbmdzLmNvbTpwYXNzd29yZA==
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=hac082exio454jcqk6ieqm4j
```

Response

```
200 - OK
Date: Mon, 23 Jun 2014 02:00:15 GMT
X-CSRF-HEADER: X-CSRF-TOKEN
Set-Cookie: JSESSIONID=1h49t997p4mgc1e108bz0cjntr;Path=/
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-CSRF-TOKEN: e2fcfafd-c600-4156-88ae-ca56babd24e1
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Transfer-Encoding: chunked
```

POST request including CSRF token

Request

```
POST /api/v2/me

Origin: chrome-extension://hgmloofddffdnphfgcellkdfbfbjeloo
X-CSRF-TOKEN: e2fcfafd-c600-4156-88ae-ca56babd24e1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/35.0.1916.153 Safari/537.36
Content-Type: application/json
Accept: */*
```

```
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=1h49t997p4mgc1e108bz0cjntr

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

Response

```
200 - OK
Date: Mon, 23 Jun 2014 02:02:51 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(8.1.14.v20131031)

{ "statusCode": 0 }
```

JavaScript example

```
<html>
        <head>

                <script type="text/javascript" src="./org/cometd.js"></script>
            <script type="text/javascript" src="./org/cometd/ReloadExtension.js"></script>
                <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/
jquery.min.js"></script>
                <script src="./jquery.cometd.js"></script>

                <script>

                ////////////////////////////////////////////////////////////////////////
                // Initialization
                ////////////////////////////////////////////////////////////////////////
                var baseUri = 'http://127.0.0.1:8080';
                var username = 'paveld@redwings.com';
                var password = 'password';

                var csrfHeaderName;
                var csrfToken;
                var cometd;

                $.ajaxSetup({
             beforeSend: function(xhr) {
                    if (csrfHeaderName && csrfToken) {
                       xhr.setRequestHeader(csrfHeaderName, csrfToken);
                    }
             }
         });

                $(document).ready(function() {
                        $('#getMeButton').click(getMe);
                        $('#startCometdButton').click(connectCometD);
                        $('#startSessionButton').click(startContactCenterSession);
                        $('#readyButton').click(ready);
                        $('#stopCometdButton').click(disconnectCometD);
                        $('#endSessionButton').click(endContactCenterSession);

                        cometd = $.cometd;
```

```
                    cometd.addListener('/meta/handshake', onHandshake);
                cometd.addListener('/meta/connect', onConnect);
                    cometd.addListener('/meta/disconnect', onDisconnect)

                    $(window).unload(function() {
                            cometd.disconnect();
                    });
            });

            ////////////////////////////////////////////////////////////////////////////
            // HTTP Helpers
            ////////////////////////////////////////////////////////////////////////////
            var get = function(params)
            {
                    var request = {
                url: baseUri + params.uri,
                type: 'GET',
                            crossDomain: true,
                            xhrFields: {
                                withCredentials: true
                            },
                success: function (data, textStatus, response) {
                        console.log(response.getAllResponseHeaders());

                        if (response.getResponseHeader('X-CSRF-HEADER') &&
response.getResponseHeader('X-CSRF-TOKEN')) {
                            csrfHeaderName = response.getResponseHeader('X-CSRF-HEADER');
                            csrfToken = response.getResponseHeader('X-CSRF-TOKEN');

                            console.log('csrfHeaderName: ' + csrfHeaderName);
                            console.log('csrfToken: ' + csrfToken);
                    }

                    if (params.callback) {
                                params.callback(data);
                        }
                },
                error: function (result) {
                    console.log(result);

                    if (params.error) {
                                            params.error(result);
                                    }
                }
            };

            if (params.includeCredentials) {
                    request.beforeSend = function (xhr) {
                                    xhr.setRequestHeader('Authorization', 'Basic ' +
window.btoa(username + ':' + password));
                            };
                    }

            $.ajax(request);
            };

            var post = function(params)
            {
                    var data = JSON.stringify(params.json, undefined, 2);

            var request = {
                url: baseUri + params.uri,
                type: 'POST',
```

```
                data: data,
    headers: {
        'Content-Type' : 'application/json'
    },
                crossDomain: true,
                xhrFields: {
                    withCredentials: true
                },
    handleAs: 'json',
    success: function(data) {
            if (params.callback) {
                    params.callback(data);
            }
    },
    error: function (req, err, exception) {
        console.log('Error! (' + req.status + ') : ' + err + ' ' + exception);
        if (params.error) {
                                params.error(result);
                        }
    }
       };

$.ajax(request);
}

////////////////////////////////////////////////////////////////////////////
// API Functions
////////////////////////////////////////////////////////////////////////////
var getMe = function() {
        get({
                uri: '/api/v2/me',
                includeCredentials: true
        });
};

var startContactCenterSession = function() {
        post({
                uri: '/api/v2/me',
                json: {
                        operationName: 'StartContactCenterSession',
                        channels: ['voice']
                }
        });
};

var ready = function() {
        post({
                uri: '/api/v2/me',
                json: {
                        operationName: 'Ready'
                }
        });
};

var endContactCenterSession = function() {
        post({
                uri: '/api/v2/me',
                json: {
                        operationName: 'EndContactCenterSession'
                },
                callback: onEndContactCenterSessionComplete
        });
};
```

```
		////////////////////////////////////////////////////////////////////////////
		// Callbacks
		////////////////////////////////////////////////////////////////////////////
		var onEndContactCenterSessionComplete = function() {
			csrfHeaderName = null;
			csrfToken = null;
		}

		////////////////////////////////////////////////////////////////////////////
		// CometD
		////////////////////////////////////////////////////////////////////////////

		var connected = false;
		var subscription;

		var onConnect = function(message) {
			if (cometd.isDisconnected()) {
				return;
			}

			var wasConnected = connected;
			connected = message.successful;
			if (!wasConnected && connected) {
				console.log('Cometd connected.');
			} else if (wasConnected && !connected) {
				console.log('Cometd disconnected...');
			}
		};

		var onDisconnect = function(message) {
			if (message.successful) {
				connected = false;
				console.log('Cometd disconnected.');
			}
		};

		var onMessage = function(message) {
			console.log('Cmetd message received:\n' + JSON.stringify(message,
null, 2));
		};

		var onHandshake = function(handshake) {
		if (handshake.successful === true) {
				if (subscription) {
					console.log('unsubscribing: ' + subscription);
					cometd.unsubscribe(subscription);
				}

				console.log('Subscribing to channels...');
				subscription = cometd.subscribe('/v2/me/*', onMessage);
		}
		};

		var connectCometD = function() {

			var reqHeaders = {};
			reqHeaders[csrfHeaderName] = csrfToken;

			cometd.unregisterTransport('websocket');
			cometd.unregisterTransport('callback-polling');
			cometd.configure({
			    url: baseUri + '/api/v2/notifications',
```

```
                            logLevel: "info",
                            requestHeaders: reqHeaders
                        });

                        cometd.handshake();
                };

                var disconnectCometD = function() {
                        cometd.disconnect();
                };

                </script>
        </head>

        <body>
                <button id='getMeButton'>Get Me</button>
                <br/>
                <button id='startCometdButton'>Start CometD</button>
                </br>
                <button id='startSessionButton'>Start Contact Center Session</button>
                <br/>
                <button id='readyButton'>Ready</button>
                <br/>
                <button id='stopCometdButton'>Stop CometD</button>
                </br>
                <button id='endSessionButton'>End Contact Center Session</button>
        </body>
</html>
```

Python example

```
import base64;
import httplib2;
import json;

GWS_BASE_URI = "http://127.0.0.1:8080/api/v2"
ADMIN_USERNAME = "mikeb@redwings.com"
ADMIN_PASSWORD = "password"

CONTACT_CENTER_USERS = [
    {
        "userName": "bobp@redwings.com",
        "firstName": "Bob",
        "lastName": "Probert",
        "password": "password",
        "phoneNumber": "5019",
        "role": "ROLE_AGENT"
    }
]

X_CSRF_HEADER = "x-csrf-header"
X_CSRF_TOKEN = "x-csrf-token"

jsessionid = None
csrfHeaderName = None
csrfTokenValue = None

http = httplib2.Http(".cache")

def create_request_headers():
    request_headers = dict()
    request_headers["Content-Type"] = "application/json"
```

```
    request_headers["Authorization"] = "Basic " + base64.b64encode(ADMIN_USERNAME + ":" +
ADMIN_PASSWORD)

    if jsessionid:
        request_headers["Cookie"] = jsessionid;
        print "Using JSESSIONID %s" % jsessionid;

    if csrfHeaderName and csrfTokenValue:
        print "Adding csrf header [%s] with value [%s]..." % (csrfHeaderName, csrfTokenValue)
        print
        request_headers[csrfHeaderName] = csrfTokenValue
    else:
        print "No csrf token, skipping..."
        print

    return request_headers

def post(uri, content):
    request_headers = create_request_headers()
    body = json.dumps(content, sort_keys=True, indent=4)

    print "POST %s (%s/%s)..." % (uri, ADMIN_USERNAME, ADMIN_PASSWORD)
    print body
    print

    response_headers, response_content = http.request(uri, "POST", body = body, headers =
request_headers)
    status = response_headers["status"]

    ugly_response = json.loads(response_content)
    pretty_response = json.dumps(ugly_response, sort_keys=True, indent=4)

    print "Response: %s" % (status)
    print "%s" % (pretty_response)
    print

    return response_headers, ugly_response

def get(uri):

    global csrfHeaderName
    global csrfTokenValue
    global jsessionid

    request_headers = create_request_headers()
    print "GET %s (%s/%s)..." % (uri, ADMIN_USERNAME, ADMIN_PASSWORD)
    print

    response_headers, response_content = http.request(uri, "GET", headers = request_headers)
    status = response_headers["status"]
    if response_headers["set-cookie"]:
        jsessionid = response_headers["set-cookie"]
        print "Set JSESSIONID %s..." % jsessionid

    ugly_response = json.loads(response_content)
    pretty_response = json.dumps(ugly_response, sort_keys=True, indent=4)

    print "Response: %s" % (status)
    print "%s" % (pretty_response)
    print

    if X_CSRF_HEADER in response_headers:
        csrfHeaderName = response_headers[X_CSRF_HEADER]
```

```
        print "Saved csrf header name [%s]" % csrfHeaderName

    if X_CSRF_TOKEN in response_headers:
        csrfTokenValue = response_headers[X_CSRF_TOKEN]
        print "Saved csrf token value [%s]" % csrfTokenValue
        print

    return response_headers, ugly_response

def check_response(response_headers, expected_code):
    if response_headers["status"] != expected_code:
        print "Request failed."
        exit(-1)

def create_user(user_info):
    user_name = user_info["userName"]
    print "Creating user [%s]..." % (user_name)

    uri = "%s/users" % (GWS_BASE_URI)

    user = {
        "userName": user_name,
        "password": user_info["password"],
        "firstName": user_info["firstName"],
        "lastName": user_info["lastName"],
        "roles": [user_info["role"]]
    }

    response_headers, response_content = post(uri, user)
    check_response(response_headers, "200")

    user_id = response_content["id"]
    print "User [%s] created. User id [%s]." % (user_name, user_id)

    return user_id

def assign_device_to_user(user_id, phone_number):
    print "Creating device [%s] and assigning to user [%s]..." % (phone_number, user_id)

    uri = "%s/users/%s/devices" % (GWS_BASE_URI, user_id)

    device = {
        "phoneNumber": phone_number
    }

    response_headers, response_content = post(uri, device)
    check_response(response_headers, "200")

    device_id = response_content["id"]
    print "Device [%s] created and assigned to user id [%s]." % (device_id, user_id)

def create_users_and_devices():
    for user_info in CONTACT_CENTER_USERS:
        user_id = create_user(user_info)
        assign_device_to_user(user_id, user_info["phoneNumber"])

def getToken():
    uri = "%s/diagnostics/version" % (GWS_BASE_URI)

    response_headers, response_content = get(uri)
    check_response(response_headers, "200")

if __name__ == "__main__":
```

```
getToken()
create_users_and_devices()
```

Cross-Origin Resource Sharing

Cross-Origin Resource Sharing (CORS) is a specification that enables open access across domain-boundaries.

Each contact center can define their own allow origin list through Interaction Recording Web Services access control settings.

Interaction Recording Web Services filters an incoming request by merging global `allowOrigins` and contact center access control settings by using an `Admin` account.

Operations

The following operations are available for this group:

Operation	Description	Permissions
GET	Retrieves an array of settings	Contact Center Admin
POST	Creates a new setting in this group. `allowedOrigins` is the only valid setting.	Contact Center Admin
PUT	Updates a setting.	Contact Center Admin
DELETE	Removes a setting.	Contact Center Admin

Settings

Attribute name	Description
allowedOrigins	An array of valid "origins" for this contact center. The CORS filter will use this list to validate incoming requests.

> **Tip**
> Wildcards are allowed in the context of a domain name for `allowedOrigins`, but `"*"` by itself is not permitted.

Examples

Retrieve access control settings

```
GET /settings/access-control {
   settings:[
{
  “name”:”allowedOrigins”,
  “value”: ["https://cloud.genhtcc.com", "https://*.genhtcc.com"]
}
]
```

Add "genesys.com" to the list of domains

```
PUT /settings/access-control {
   settings:
{
  "name":"allowedOrigins",
  "value": ["https://cloud.genhtcc.com", "https://*.genhtcc.com", "https://*.genesys.com"]
}
```

> **Important**
>
> When sending the above, the entire array must be sent

Return Values

Interaction Recording Web Services API requests 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.

Interaction Recording Web Services supports the following status codes:

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 could occur if an internal error occurred with Interaction Recording Web Services or with one of the servers working with Interaction Recording Web Services (for example: Cassandra or a Genesys Framework component).

Code	Description
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. More information is available in the Partial success section.
8	Change password demanded. Interaction Recording Web Services requested a password change for the user.
9	Processing incomplete.
10	Input validation error - the provided value is not within the range of valid values.
11	User requested to change read-only property.
12	Unable to retrieve resource error.
13	Unable to create resource error.
14	Unable to delete resource error.
15	Unable to update resource error.
16	Unable to assign resource error.
17	Unable to unassign resource error.
18	Resource already exists.
19	Resource already in use.
20	User is not authenticated. Any subsequent request should provide credentials.

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 for the contact center.

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).
path	The relative path of the resource
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"], "paths": ["uri3"]
}

{
        "statusCode":7 (partial success)
        "succeeded":[
                {
                        "id":<id1>,
                        "uri":"uri1",
                        "path":"path1",
                },
                {
                        "id":<id2>,
                        "uri":"uri2",
                        "path":"path2"
                }]
        "failed":[
                {
                        "statusCode":X,
                        "statusMessage":"msg",
                        "uri":"uri3",
```

```
                        "path": "path2"
                }
        ]
}
```

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",
                        "path":"path"
                },
                {
                        "id":<id>,
                        "uri":"uri2",
                        "path":"path2"
                }]

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

Delete:

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

{
        "statusCode":7 (partial success)
        "succeeded":[
                {
```

```
                        "id":<id1>,
                        "uri":"uri1",
                        "path":"path1"
                },
                {
                        "id":<id2>,
                        "uri":"uri2",
                        "path": "path2"
                }]

        "failed":[
                {
                        "statusCode":X,
                        "statusMessage":"...",
                        "uri":"uri3",
                        "path": "path2"
                }]
}
```

Search, Playback, and Delete API

When you use these APIs, you'll typically search and play back recordings as follows:

1. Request a list of call recordings based on your query parameters.
2. Receive a list of recordings in the response from Interaction Recording Web Services.
   Note: If your query generates many results, they are paginated and the response contains navigation links for the previous and next pages.
3. Play back one or more of these recordings using the URI provided by the response, or inspect the attributes associated with each recording.

You can use the Delete Recording API to remove a recording from the Genesys Interaction Recording system.

Recording resource

The recording resource includes details about the recording, such as start and stop times, along with the event history and media files that make up the recording.

Sample data


```
   {
  "id":"011AP643CSAPR4FKQGQE31TAES00029V",
  "callerPhoneNumber":"8522001",
  "dialedPhoneNumber":"+14160000001",
  "startTime":"2015-09-10T16:58:22.000+0000",
  "stopTime":"2015-09-10T16:58:34.000+0000",
  "mediaFiles":[
    {
       "mediaUri":"http://htcc1.genesyslab.com/api/v2/recordings/
011AP643CSAPR4FKQGQE31TAES00029V/play/4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
       "mediaPath":"/recordings/011AP643CSAPR4FKQGQE31TAES00029V/play/
4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
       "playPath":"/recordings/011AP643CSAPR4FKQGQE31TAES00029V/play/
4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
       "startTime":"2015-09-10T16:58:22.000+0000",
       "stopTime":"2015-09-10T16:58:34.000+0000",
       "callUUID":"011AP643CSAPR4FKQGQE31TAES00029V",

"mediaId":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22-006B015F-1000427D-00000001.mp3",
       "type":"audio/mp3",
       "duration":"12523",
       "tenant":"Environment",
       "ivrprofile":"CallRecProfile",
       "size":"199296",
       "parameters":{
          "dnis":"+14160000001",
          "connId":"006a0269722cd93f",
          "ani":"8522001",
```

```
        "dateTime":"2015-09-10T16:58:22Z",
        "rp.speechminer_uri":"http://135.225.52.210/interactionreceiver",
        "recordDN":"+14160000001",
        "agentId":"+14160000001",
        "rp.speechminer_auth":"rpSys:111111",
        "sipsAppName":"SIP_Server",
        "id":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22",
        "username":"screen1@genesys.com",
        "callUuid":"011AP643CSAPR4FKQGQE31TAES00029V"
     },
     "partitions":[
     ],
     "accessgroups":[
        "/"
     ]
  }
],
"eventHistory":[
  {
     "occurredAt":"2015-09-10T16:58:22.000+0000",
     "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
     "eventId":"2015-09-10 16:58:22.440_011AP643CSAPR4FKQGQE31TAES00029V",
     "event":"Data",
     "data":{
        "added":{
           "GSIP_RECORD":"ON",
           "GSIP_REC_FN":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22"
        }
     }
  },
  {
     "occurredAt":"2015-09-10T16:58:34.000+0000",
     "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
     "eventId":"2015-09-10 16:58:34.820_011AP643CSAPR4FKQGQE31TAES00029V",
     "event":"Data",
     "data":{
        "deleted":{
           "GSIP_RECORD":"ON"
        }
     }
  },
  {
     "occurredAt":"2015-09-10T16:58:22.000+0000",
     "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
     "contact":{
        "type":"User",
        "phoneNumber":"+14160000001",
        "userName":"screen1@genesys.com",
        "firstName":"Screen",
        "lastName":"GIR"
     },
     "event":"Joined"
  },
  {
     "occurredAt":"2015-09-10T16:58:21.000+0000",
     "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
     "contact":{
        "type":"External",
        "phoneNumber":"8522001"
     },
     "event":"Joined"
  },
  {
```

```
      "occurredAt":"2015-09-10T16:58:34.000+0000",
      "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
      "contact":{
         "type":"User",
         "phoneNumber":"+14160000001",
         "userName":"screen1@genesys.com",
         "firstName":"Screen",
         "lastName":"GIR"
      },
      "event":"Left"
   },
   {
      "occurredAt":"2015-09-10T16:58:34.000+0000",
      "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
      "contact":{
         "type":"External",
         "phoneNumber":"8522001"
      },
      "event":"Left"
   }
  ],
  "labels": [
    {
      "id": "646943bc-b9d2-48e7-9304-d9b4c92be2d2",
      "name": "Important",
      "type": "custom",
      "createTime": "2017-01-01T00:00:00.000+0000",
      "createUser": "Agent1"
    }
  ],
  "callType":"Inbound",
  "screenRecording": true,
  "region":"region1"
}
```


Resource details

Field	Description	Type
recording.id	The ID of the call recording.	string
recording.callerPhoneNumber	The phone number of the caller in the recorded call.	string
recording.dialedPhoneNumber	The phone number dialed to initiate the recorded call.	string
recording.startTime	The time when the call recording starts.	string
recording.stopTime	The time when the call recording stops.	string
recording.screenRecording	Indicates whether or not the call recording has one or more associated screen recordings.	boolean
recording.callType	The type of call that was recorded ('Unknown', 'Internal', 'Inbound', 'Outbound', 'Consult').	string
recording.nonDelete	Indicates whether or not the recording is protected from deletion.	boolean
recording.region	The region in which the call was recorded.	string
recording.mediaFiles	The collection of information for all the media files associated with the recording.	array
recording.labels	The collection of information for all the labels that have been	array

Field	Description	Type
	added to the call recording. **Note:** Labels are returned only when the value of the subresources parameter is * or labels.	
recording.eventHistory	The collection of information for all the events that occurred during the recorded call.	array
recording.mediaFiles[].mediaUri	This attribute is for internal Genesys use only.	string
recording.mediaFiles[].mediaPath	This attribute is for internal Genesys use only.	string
recording.mediaFiles[].playPath	The relative path to play back a recording file. If the recording is encrypted, it is decrypted before being played back to the user of the API. **Note:** The user who uses playPath to playback an encrypted recording must be granted permission to decrypt. The playPath path is **/recordings/{recordingId}/decrypt/<mediaUUID>.mp3** where **<recordingId>** is the ID of the recording requested, and **<mediaUUID>** is a type 4 UUID internally assigned by Interaction Recording Web Services. Using playPath to decrypt a recording segment is described in Get Recording Media by ID API.	string
recording.mediaFiles[].startTime	The start time for this media file.	string
recording.mediaFiles[].stopTime	The stop time for this media file.	string
recording.mediaFiles[].callUUID	The call UUID generated by the SIP Server for this media file. It may be different than the call UUID of other media files, in the case of multi-site transfer.	string
recording.mediaFiles[].mediaId	The unique ID for this media file generated by Genesys software.	string
recording.mediaFiles[].type	The MIME type of the media file.	string
recording.mediaFiles[].duration	The duration of the media file in milliseconds.	string
recording.mediaFiles[].size	The size of the media file in bytes.	string
recording.mediaFiles[].tenant	The tenant this media file belongs to.	string
recording.mediaFiles[].ivrprofile	The IVR Profile name that serviced this recording.	string
recording.mediaFiles[].parameters	This attribute is for internal Genesys use only.	object
recording.mediaFiles[].masks	A JSON array specifying the time stamps of the pause/resume periods if the recording is masked by a client application. Each object contains the time and type property.	array
recording.mediaFiles[].partitions	A JSON array of strings specifying the partition to which this media file belongs. For more information about partitions, see Access Control for Genesys Interaction Recording Users in the Genesys Interaction Recording Solution Guide.	array
recording.mediaFiles[].accessgroups	A JSON array of strings specifying the Agent Hierarchy group(s) to which this recording belongs. For more information about agent hierarchy and access groups, see Access Control for Genesys Interaction Recording Users in the Genesys Interaction Recording Solution Guide.	array
recording.mediaFiles[].pkcs7	This attribute is for internal Genesys use only.	string
recording.mediaFiles[].certAlias	This attribute is for internal Genesys use only.	array

Field	Description	Type
recording.mediaFiles[].masks[].time	The time of the masking event.	string
recording.mediaFiles[].masks[].type	The type of the masking event (paused, resume).	string
recording.eventHistory[].occurredAt	Specifies the time the event occurred.	string
recording.eventHistory[].calluuid	Specifies the Call UUID for the event.	string
recording.eventHistory[].event	The type of Event. It can be one of the following three values: • Joined • Left • Data	string
recording.eventHistory[].eventId	An eventId used to identify the data event.	string
recording.eventHistory[].contact	A JSON object representing the contact information of the caller who joined or left the recording; if the event type is "Joined" or "Left".	contact
recording.eventHistory[].contact.type	A JSON object representing the contact information of the caller who joined or left the recording, if the event type is Joined or Left it can have one of the following values: • User • External	string
recording.eventHistory[].contact.phoneNumber	The phone number for the caller.	string
recording.eventHistory[].contact.userName	The user name of the caller.	string
recording.eventHistory[].contact.firstName	The first name of the caller.	string
recording.eventHistory[].contact.lastName	The last name of the caller.	string
recording.eventHistory[].data	The key-value pairs to store additional information about the event if event type is "Data".	string
recording.eventHistory[].eventHistoryData[].added	A JSON object representing the key-value pairs associated to the added data event.	object
recording.eventHistory[].eventHistoryData[].updated	A JSON object representing the key-value pairs associated to the updated data event.	object
recording.eventHistory[].eventHistoryData[].deleted	A JSON object representing the key-value pairs associated to the deleted data event.	object
recording.labels[].name	The name of the label.	string
recording.labels[].id	The id of the label.	string
recording.labels[].type	The type of the label (Reserved or Custom).	string
recording.labels[].createTime	The time at which the label is created.	string
recording.labels[].createUser	The creator of the label.	string
recording.labels[].content	The key-value pairs to store additional information about the label.	object

Search Recordings API

This request retrieves a list of recordings that are stored in the Genesys Interaction Recording (GIR) system.

Request URL	/api/v2/recordings
HTTP Method	GET
Required Features	api-supervisor-recording

Prerequisites

To access the API, the user must be assigned a role of either "Administrator" or "Supervisor" in the Genesys Configuration system. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 Section 2 for reference.

In addition, if the media files are encrypted, the user must be assigned the proper privileges to decrypt the media file in order to use the playPath URI to decrypt and play back the recording. For details, see Access Control in the Genesys Interaction Recording Solution Guide.

Recording privacy settings

For privacy reasons, Interaction Recording Web Services can mask individual fields in the recording metadata returned by the Search Recordings operation. This masking prevents the fields from being exposed to users with the Supervisor role. Masking does not apply to users with the Administrator role.

You can specify which fields to mask by using two settings in the **recording** contact-center settings group.

> ### Important
> You shouldn't mask fields that contain links — for example, `mediaFiles[].mediaUri`, `mediaFiles[].mediaPath`, and `mediaFiles[].playPath`.

Name	Value
metadata.privacy.agent_fields	A comma-separated list of all the metadata fields to hide if the user does not have permission to view the agent metadata fields. The default value is blank.
metadata.privacy.customer_fields	A comma-separated list of all the metadata fields to hide if the user does not have permission to view the customer metadata fields. The default value is blank.

A user can GET, POST, PUT, and DELETE these settings at the contact center level with the **/api/v2/settings/recording** API using an admin account. For details about working with settings, see the

Settings API.

Parameters

> **Important**
>
> You must include at least **one** of the following parameters; otherwise, Interaction Recording Web Services returns an error.

Parameter	Value
callerPhoneNumber	Retrieves all recordings for any call containing the specified "ani" attribute (the number from which the call was made). When no wildcard is used, the query result contains the recordings for the calls where there is an exact match of stored value (alphanumeric-only) and request parameter. The request string can contain the * wildcard, which can substitute any number of any symbols in the request. Search is case sensitive and all non-alphanumeric characters except for * and ? are stripped away before the search is performed.
dialedPhoneNumber	Retrieves all recordings for any call containing the specified "dnis" attribute (the number dialed by the caller). When no wildcard is used, the query result contains the recordings for the calls where there is an exact match of stored value (alphanumeric-only) and request parameter. The request string can contain the * wildcard, which can substitute any number of any symbols in the request. Search is case sensitive an all non-alphanumeric characters except for * and ? are stripped away before the search is performed.
startTime	Retrieves all recordings that started at or after the specified time. This is specified as the number of milliseconds since epoch.
endTime	Retrieves all recordings that ended at or before the specified time. This is specified as the number of milliseconds since epoch. **Note:** The endTime parameter corresponds with the stopTime field in the recording resource that's returned by the request.
userName	Retrieves all recordings that involve a user with the first name, last name, or user name matching the specified query parameter. The query parameter is matched against the "contact" attribute for each item in the eventHistory array for each recording. The request performs an exact match without any modifiers, but you can also:

Parameter	Value
	• Use the * character to perform a wildcard match • When you place two words in the query parameter, it's an OR operation which means that the query returns all recordings involving a user with name matching either of the two words For example: .../api/v2/recordings?userName=bob alice☆tTime=0 • When you place two words in the query parameters with the word "AND" between them, it's an AND operation which causes the query to return all recordings involving users with names matching all of the words For example: .../api/v2/recordings?userName=bob AND alice☆tTime=0 • You should escape the following reserved characters with a leading backslash: + - = && || > < ! ( ) { } [ ] ^ " ~ * ? : \ / For example, (1+1)=2 would be \(1\+1\)\=2. Note: A space can also be a reserved character.
userData	Retrieves all recordings that have user data matching the specified query parameter. The query parameter is matched against the "data" attribute for each item in the eventHistory array for each recording. The match is against the "value" for each "data" attribute, not the "name". The request performs an exact match without any modifiers, but you can also: • Use the * character to perform a wildcard match • When you place two words in the query parameter, it's an OR operation which means that the query returns all recordings with user data values matching either of the two words For example: .../api/v2/recordings?userData=creditcard loan☆tTime=0 • When you place two words in the query parameters with the word "AND" between them, it's an AND operation which causes the query to return all recordings with user data values matching all of the words For example: .../api/v2/recordings?userData=cancel AND creditcard☆tTime=0 • You should escape the following reserved characters with a leading backslash: + - = && || > < ! ( ) { } [ ] ^ " ~ * ? : \ / For example, (1+1)=2 would be \(1\+1\)\=2. Note: A space can also be a reserved character.

Parameter	Value
includeLabels	Retrieves all recordings for any call that contains all of the labels in the comma-separated string of label names. The request string is a comma-separated list of label names. For example: `includeLabels=comment_good,rate_good`
excludeLabels	Retrieves all recordings for any call that does not contain any of the labels in the comma-separated string of label names. The request string is a comma-separated list of label names. For example: `excludeLabels=comment_bad,bad`

Note: You can search for all call recordings that have one or more associated screen recordings by specifying `includeLabels=__screenRecording`. Similarly, you can find all call recordings that do not have an associated screen recording by specifying `excludeLabels=__screenRecording`.

Mask settings and query parameters

Interaction Recording Web Services doesn't allow query parameters for the Genesys Interaction Recording API if the **metadata.privacy.agent_fields** or **metadata.privacy.customer_fields** are defined in the **recording** settings group. This prevents the user from using a search query parameter as a way to find records that are directly associated with a private field even if the result is masked. If a search parameter matches a field listed in one of the settings above, then Interaction Recording Web Services returns an HTTP response code of 403. The following query parameters may appear in the **metadata.privacy.agent_fields** or **metadata.privacy.customer_fields** settings:

- callerPhoneNumber
- dialedPhoneNumber
- userName
- userData

Pagination

You can use the following optional request parameters to control the pagination of results from Interaction Recording Web Services.

Parameter	Value
offset	Specifies the index of the first record to be returned. • Default value is 0.
limit	Specifies the number of records to be returned. • Maximum allowed value is 100.

Parameter	Value
	• Default value is 10.

Sample 1 - Simple query

Request

```
GET api/v2/recordings?startTime=1441065600000&offset=0&limit=100
```

Response


```
{
   "statusCode":0,
   "recordings":[
      {
         "id":"011AP643CSAPR4FKQGQE31TAES0002A0",
         "callerPhoneNumber":"+15559680001",
         "dialedPhoneNumber":"+15559680002",
         "startTime":"2015-09-10T18:36:58.000+0000",
         "stopTime":"2015-09-10T18:37:30.000+0000",
         "mediaFiles":[
            {
               "mediaUri":"http://abcdef01.com/api/v2/recordings/
011AP643CSAPR4FKQGQE31TAES0002A0/play/45bb2966-9911-45fd-8167-a6581d04af98.mp3",
               "mediaPath":"/recordings/011AP643CSAPR4FKQGQE31TAES0002A0/play/
45bb2966-9911-45fd-8167-a6581d04af98.mp3",
               "playPath":"/recordings/011AP643CSAPR4FKQGQE31TAES0002A0/play/
45bb2966-9911-45fd-8167-a6581d04af98.mp3",
               "startTime":"2015-09-10T18:36:58.000+0000",
               "stopTime":"2015-09-10T18:37:30.000+0000",
               "callUUID":"011AP643CSAPR4FKQGQE31TAES0002A0",

"mediaId":"011AP643CSAPR4FKQGQE31TAES0002A0_2015-09-10_18-36-57-006B015F-1000427F-00000001.mp3",
               "type":"audio/mp3",
               "duration":"33100",
               "tenant":"Environment",
               "ivrprofile":"CallRecProfile",
               "size":"528768",
               "parameters":{
                  "dnis":"+19059680002",
                  "connId":"006a0269722cd940",
                  "ani":"+19059680001",
                  "dateTime":"2015-09-10T18:36:57Z",
                  "rp.speechminer_uri":"http://000.001.02.003/interactionreceiver",
                  "agentId":"+15559680002",
                  "recordDN":"+15559680002",
                  "sipsAppName":"SIP_Server",
                  "rp.speechminer_auth":"rpSys:111111",
                  "id":"011AP643CSAPR4FKQGQE31TAES0002A0_2015-09-10_18-36-57",
                  "username":"agent2@genesys.com",
                  "callUuid":"011AP643CSAPR4FKQGQE31TAES0002A0"
               },
               "masks": [
                  {"time": "2015-09-10T18:37:22.034Z", "type", "paused"},
                  {"time": "2015-09-10T18:37:24.098Z", "type", "resume"}
               ],
               "partitions":[
```

```
        ],
        "accessgroups":[
           "/Newx"
        ]
     }
  ],
  "eventHistory":[
     {
        "occurredAt":"2015-09-10T18:36:58.000+0000",
        "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
        "eventId":"2015-09-10 18:36:58.347_011AP643CSAPR4FKQGQE31TAES0002A0",
        "event":"Data",
        "data":{
           "added":{
              "GSIP_RECORD":"ON",
              "GSIP_REC_FN":"011AP643CSAPR4FKQGQE31TAES0002A0_2015-09-10_18-36-57"
           }
        }
     },
     {
        "occurredAt":"2015-09-10T18:37:30.000+0000",
        "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
        "eventId":"2015-09-10 18:37:30.707_011AP643CSAPR4FKQGQE31TAES0002A0",
        "event":"Data",
        "data":{
           "deleted":{
              "GSIP_RECORD":"ON"
           }
        }
     },
     {
        "occurredAt":"2015-09-10T18:36:57.000+0000",
        "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
        "contact":{
           "type":"User",
           "phoneNumber":"+15559680001",
           "userName":"agent1@genesys.com",
           "firstName":"Agent1",
           "lastName":"Genesys"
        },
        "event":"Joined"
     },
     {
        "occurredAt":"2015-09-10T18:36:57.000+0000",
        "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
        "contact":{
           "type":"User",
           "phoneNumber":"+15559680002",
           "userName":"agent2@genesys.com",
           "firstName":"Agent2",
           "lastName":"Genesys"
        },
        "event":"Joined"
     },
     {
        "occurredAt":"2015-09-10T18:37:30.000+0000",
        "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
        "contact":{
           "type":"User",
           "phoneNumber":"+15559680001",
           "userName":"agent1@genesys.com",
           "firstName":"Agent1",
           "lastName":"Genesys"
```

```
               },
               "event":"Left"
            },
            {
               "occurredAt":"2015-09-10T18:37:30.000+0000",
               "calluuid":"011AP643CSAPR4FKQGQE31TAES0002A0",
               "contact":{
                  "type":"User",
                  "phoneNumber":"+15559680002",
                  "userName":"agent2@genesys.com",
                  "firstName":"Agent2",
                  "lastName":"Genesys"
               },
               "event":"Left"
            }
         ],
         "callType":"Internal",
         "screenRecording":true,
         "region":"region1"
      },
      {
         "id":"011AP643CSAPR4FKQGQE31TAES00029V",
         "callerPhoneNumber":"8522001",
         "dialedPhoneNumber":"+15550000001",
         "startTime":"2015-09-10T16:58:22.000+0000",
         "stopTime":"2015-09-10T16:58:34.000+0000",
         "mediaFiles":[
            {
               "mediaUri":"http://abcdef01.com/api/v2/recordings/
011AP643CSAPR4FKQGQE31TAES00029V/play/4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
               "mediaPath":"/recordings/011AP643CSAPR4FKQGQE31TAES00029V/play/
4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
               "playPath":"/recordings/011AP643CSAPR4FKQGQE31TAES00029V/play/
4f82ee48-4fe4-40e1-961d-309ba6e38698.mp3",
               "startTime":"2015-09-10T16:58:22.000+0000",
               "stopTime":"2015-09-10T16:58:34.000+0000",
               "callUUID":"011AP643CSAPR4FKQGQE31TAES00029V",

"mediaId":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22-006B015F-1000427D-00000001.mp3",
               "type":"audio/mp3",
               "duration":"12523",
               "tenant":"Environment",
               "ivrprofile":"CallRecProfile",
               "size":"199296",
               "parameters":{
                  "dnis":"+15550000001",
                  "connId":"006a0269722cd93f",
                  "ani":"8522001",
                  "dateTime":"2015-09-10T16:58:22Z",
                  "rp.speechminer_uri":"http://010.020.03.040/interactionreceiver",
                  "recordDN":"+14160000001",
                  "agentId":"+14160000001",
                  "rp.speechminer_auth":"rpSys:111111",
                  "sipsAppName":"SIP_Server",
                  "id":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22",
                  "username":"screen1@genesys.com",
                  "callUuid":"011AP643CSAPR4FKQGQE31TAES00029V"
               },
               "partitions":[
               ],
               "accessgroups":[
                  "/"
               ]
```

```
      }
   ],
   "eventHistory":[
      {
         "occurredAt":"2015-09-10T16:58:22.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "eventId":"2015-09-10 16:58:22.440_011AP643CSAPR4FKQGQE31TAES00029V",
         "event":"Data",
         "data":{
            "added":{
               "GSIP_RECORD":"ON",
               "GSIP_REC_FN":"011AP643CSAPR4FKQGQE31TAES00029V_2015-09-10_16-58-22"
            }
         }
      },
      {
         "occurredAt":"2015-09-10T16:58:34.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "eventId":"2015-09-10 16:58:34.820_011AP643CSAPR4FKQGQE31TAES00029V",
         "event":"Data",
         "data":{
            "deleted":{
               "GSIP_RECORD":"ON"
            }
         }
      },
      {
         "occurredAt":"2015-09-10T16:58:22.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "contact":{
            "type":"User",
            "phoneNumber":"+15550000001",
            "userName":"screen1@genesys.com",
            "firstName":"Screen",
            "lastName":"GIR"
         },
         "event":"Joined"
      },
      {
         "occurredAt":"2015-09-10T16:58:21.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "contact":{
            "type":"External",
            "phoneNumber":"8522001"
         },
         "event":"Joined"
      },
      {
         "occurredAt":"2015-09-10T16:58:34.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "contact":{
            "type":"User",
            "phoneNumber":"+15550000001",
            "userName":"screen1@genesys.com",
            "firstName":"Screen",
            "lastName":"GIR"
         },
         "event":"Left"
      },
      {
         "occurredAt":"2015-09-10T16:58:34.000+0000",
         "calluuid":"011AP643CSAPR4FKQGQE31TAES00029V",
         "contact":{
```

```
                  "type":"External",
                  "phoneNumber":"8522001"
               },
               "event":"Left"
            }
         ],
         "callType":"Inbound",
         "screenRecording":true,
         "region":"region1"
      }
   ],
   "totalCount": 2,
   "nextPath":"/recordings/?startTime=1441065600000&offset=100&limit=100",
   "nextUri":"http://abc6-de01.us.int.genesyslab.com/api/v2/
recordings?startTime=1441065600000&offset=100&limit=100"
}
```


Sample 2 - Masked results

The following example shows hows masked results are returned when privacy settings are enabled.

Request

```
GET api/v2/recordings/recordings/00MT035FS4APRCEK4SL131TAES0000CF
```

Response

Masked results for the recording resource:


```
{
 "id": "00MT035FS4APRCEK4SL131TAES0000CH",
 "callerPhoneNumber": "*******",
 "dialedPhoneNumber": "*******",
 "startTime": "2015-09-24T15:25:29.000+0000",
 "stopTime": "2015-09-24T15:27:01.000+0000",
 "mediaFiles": [...],
 "eventHistory":[...],
 "callType": "Inbound",
 "screenRecording":false,
 "region": "region1"
 }
```


Masked results for mediaFiles:


```
"parameters": {
            "dnis": "*******",
            "connId": "008902697711d191",
            "ani": "*******",
            "dateTime": "2015-09-24T15:25:44Z",
            "rp.speechminer_uri": "http://135.225.52.210/interactionreceiver",
            "recordDN": "+19059680001",
            "agentId": "********", (agent id)
            "rp.speechminer_auth": "rpSys:111111",
            "sipsAppName": "SIP_Server",
            "id": "011AP643CSAPR4FKQGQE31TAES0002VE_2015-09-24_15-25-44",
            "username": "**************",
            "callUuid": "011AP643CSAPR4FKQGQE31TAES0002VE"
         }
```

Masked results for eventHistory:

```
"eventHistory": [
  {
          "occurredAt": "2015-09-24T15:25:44.000+0000",
          "calluuid": "011AP643CSAPR4FKQGQE31TAES0002VE",
          "contact": {
            "type": "User",
            "phoneNumber": "*******",  (extension)
            "userName": "*******",   (username)
            "firstName": "*******",  (name)
            "lastName": "*******"  (name)
},
          "event": "Joined"
        } ...]
```

Masked results for attachedData:

```
"eventHistory": [
        {
          "occurredAt": "2014-11-04T22:34:24.000+0000",
          "calluuid": "00PHK76MF0A2LDLR4SL131TAES00007V",
          "eventId": "04-NOV-14 1_00PHK76MF0A2LDLR4SL131TAES00007V",
          "event": "Data",
          "data": {
            "added": {
              "GSIP_RECORD": "*******",
              "GRECORD_PARTITIONS": "genesys_markham",
              "BusinessCall": "1",
              "GSIP_REC_FN": "00JKBG7S2OA57CEKQGQE31TAES000037_2014-11-04_22-34-09",
              "OtherTrunkName": "External01",
              "sip-iscc-cofid": "location=SIP_Switch;cofid=287",
              "RVQDBID": "",
              "WrapUpTime": "0"
            },
            "updated": {
              "GSIP_REC_FN": "00PHK76MF0A2LDLR4SL131TAES00007V_2014-11-04_22-34-25"
            },
            "deleted": {
              "GSIP_RECORD": "*******",
              "sip-iscc-cofid": "location=SIP_Switch;cofid=287",
              "WrapUpTime": "0"
} }
```

Sample 3 – Label Filter Query

Request

```
GET /recordings?includeLabels=comment<resources=*
```

Response

```
{
  ...
  "mediaFiles": [...],
  "eventHistory": [...],
  "labels": [
    {
      "path": "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31",
```

```
      "name": "comment",
      "id": "8c9a634f-343d-49b3-8322-b571be520b31",
      "createTime": "2017-04-22T12:42:46.000+0000",
      "createUser": "supervisor",
      "content": {
        "time": "2016-04-22T12:42:44.000+0000",
        "text": "This is an awesome comment!"
      }
    }
  ],
  ...
}
```

Response

Field	Description
statusCode	Indicates whether the request was correctly handled. It returns 0 if the response was correctly handled by the server, and other values if it was not correctly handled by the server. See Return Values for details.
statusMessage	For a status that is not 0, this message provides a human readable reason for reason of failure.
nextPath	The relative path for the next page of results, if there are more recordings matching the query criteria beyond this result set. The Interaction Recording Web Services path prefix should be prepended to this to form the full URL to access the next page of results. This is provided as a convenience to quickly navigate between pages; it simply makes use of the offset and limit pagination parameters described above.
prevPath	The relative path for the previous page of results, if this was not the first page in the result set. The Interaction Recording Web Services path prefix should be prepended to this to form the full URL to access the previous page of results. This is provided as a convenience to quickly navigate between pages; it simply makes use of the offset and limit pagination parameters described above.
nextUri	This field is for internal Genesys use only.
prevUri	This field is for internal Genesys use only.
recordings	This is a JSON array containing the list of recordings that satisfy the query parameters. See recording resource for details about the fields in each recording.
totalCount	The total number of recordings that satisfy the query parameters.

Get Recording by ID API

This request retrieves a specific recording that is stored in the Genesys Interaction Recording (GIR) system.

Request URL	/api/v2/recordings/{recordingId}
HTTP Method	GET
Required Features	api-supervisor-recording

Prerequisites

To access the API, the user must be assigned a role of either **Administrator** or **Supervisor** in the Genesys Configuration system. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 section 2 for reference.

In addition, if the media files are encrypted, the user must be assigned the proper privileges to decrypt the media file to use the playPath URI to decrypt and play back the recording. For details, see |Access Control in the Genesys Interaction Recording Solution Guide.

Recording privacy settings

For privacy reasons, Interaction Recording Web Services can mask individual fields in the recording metadata returned by the Get Recording by ID operation. This masking prevents the fields from being exposed to users with the Supervisor role. Masking does not apply to users with the Administrator role.

You can specify which fields to mask by using two settings in the **recording** contact-center settings group.

> ### Important
> You shouldn't mask fields that contain links — for example, `mediaFiles[].mediaUri`, `mediaFiles[].mediaPath`, and `mediaFiles[].playPath`.

Name	Value
metadata.privacy.agent_fields	A comma-separated list of all the metadata fields to hide if the user does not have permission to view the agent metadata fields. The default value is blank.
metadata.privacy.customer_fields	A comma-separated list of all the metadata fields to hide if the user does not have permission to view the customer metadata fields. The default value is blank.

A user can GET, POST, PUT, and DELETE these settings at the contact center level with the **/api/v2/settings/recording** API using an admin account. For details about working with settings, see the Settings API.

Parameters

Parameter	Mandatory	Value
recordingId	Yes	The ID associated with the call recording that should be retrieved.
subresources	No	The comma-delimited string that contains the names of the sub-resources that should be retrieved along with the call recording. Currently, only labels are supported. Valid Values: • * - Returns all the sub-resources. • **labels** - Returns the labels sub-resource.

Response

If the operation completes successfully, the specified recording is returned.
See recording resource for details about the fields in each recording.

HTTP Status Code	Reason	Response Model
200	The specified call recording was retrieved successfully.	**statusCode**: Indicates whether the request was correctly handled. It returns 0 if the response was correctly handled by the server, and other values if it was not correctly handled by the server. See Return Values for details. **statusMessage** (string): For a status that is not 0, this message provides a human readable reason for reason of failure.
403	Forbidden to get the requested recording.	
404	The specified recording cannot be accessed.	
500	An error occurred during the retrieval of the call recording.	

Get Recording Media by ID API

This request decrypts and retrieves the call recording media file identified by **recordingId** and **mediaUUID**.

Request URL	/api/v2/recordings/{recordingId}/decrypt/{mediaUUID}.mp3
HTTP Method	GET
Required Features	api-supervisor-recording

Request URL	/api/v2/recordings/{recordingId}/decrypt/{mediaUUID}.mp3
	api-recordings-decryption-proxying

Prerequisites

- To access the API, the user must be assigned a role of **Administrator**, **Supervisor**, or **Agent** in the Genesys Configuration system. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 section 2 for reference.
- Enable the Local Decrypt URI Prefix. Refer to Local Decrypt URI Prefix for Call Recording and Screen Recording.

Parameters

The path **/recordings/{recordingId}/decrypt/{mediaUUID}.mp3** is sourced from the **recording.mediaFiles[].playPath** property in a successful response to **GET /recordings/{recordingId}**, described in the Get Recording by ID API section.

Parameter	Mandatory	Value
recordingId	Yes	The ID of the call recording to retrieve.
mediaUUID	Yes	The UUID of the media file to retrieve.

Response

HTTP Status Code	Reason	Response Model
200	The requested call recording media file was decrypted and retrieved successfully.	**statusCode**: Indicates whether the request was correctly handled. It returns 0 if the response was correctly handled by the server, and other values if it was not correctly handled by the server. See Return Values for details. **statusMessage** (string): For a status that is not 0, this message provides a human readable reason for reason of failure.
403	Forbidden to get the requested recording.	
404	The specified recording or media file cannot be accessed.	
500	An error occurred during the decryption and retrieval of the call recording media file.	

Delete Recording by ID API

This request deletes a specific call recording along with its associated screen recordings and all the media files.

Request URL	/api/v2/recordings/{recordingId}
HTTP Method	DELETE
Required Features	N/A

Prerequisites

To access the API, the user must be assigned a role of **Administrator** in the Genesys Configuration system. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 section 2 for reference.

Parameters

Parameter	Mandatory	Value
recordingId	Yes	The ID of the call recording to delete.

Response

HTTP Status Code	Reason	Response Model
200	The call recording along with its associated screen recordings, and all their media files, were deleted successfully.	**`statusCode`**: Indicates whether the request was correctly handled. It returns 0 if the response was correctly handled by the server, and other values if it was not correctly handled by the server. See Return Values for details. **`statusMessage`** (string): For a status that is not 0, this message provides a human readable reason for reason of failure.
403	Forbidden to delete the requested recording.	
404	The recording to delete cannot be found.	
500	An error occurred during the deletion of a call recording, its associated screen recordings, or the media files.	
503	The recording database is read-only and therefore a call recording and its associated screen recordings cannot be deleted.	

Insertion API

The Genesys Interaction Recording Insertion API allows users to insert records into the Genesys Interaction Recording system. You can use this API to update the GIR database with records of existing call recordings already stored at a WebDAV storage location, so that they are accessible from the Genesys Interaction Recording user interfaces.

Insert Recording

This request inserts one single record representing one recording into the GIR database.

Request URL	/internal-api/contact-centers/*{id}*/recordings
HTTP Method	POST
Required Features	api-supervisor-recording

Prerequisites

Contact Center ID

The HTTP request URI you use to insert a new recording contains a "contact-center-id" that represents the Business Unit for which the request is to be made. For more information about how to get the contact center ID for your deployment, see Configuring the Storage Credentials for Web Services in the Configuration for Voice Recordings section of the Genesys Interaction Recording Solution Guide.

Authentication

This API uses a different authentication model than the other Interaction Recording Web Services APIs. For those APIs, you make a request with a username and password that belong to a normal user of the Genesys environment who also exists in Configuration Server. To use this Genesys Interaction Recording Insertion API, the HTTP request must send a Basic Authentication header with the opsUsername and opsUserPassword defined in the application.yaml file.

Cross Site Request Forgery Protection Header

As with any POST request to Interaction Recording Web Services, you must add the Cross Site Request Forgery Protection (CSRF) headers to the requests. See Cross Site Request Forgery Protection for details about how to get the values for these HTTP headers.

Request Body

The request body contains information for one single recorded interaction. This API accepts only a body of the request with Content-Type "application/json", and the request body is a JSON object. The following is a description of the format of the JSON object.

Attribute	JSON Data Type	Description	Mandatory
id	String	This is the unique ID of the recording resource being created. If the ID does not already exist in the GIR database, a new record for this recording is created. If the ID already exists in the GIR database, it merges the existing record with information from the POST request. At this time, Genesys does not support updating an existing record by using an existing ID in the POST request, and it's the HTTP client's responsibility to ensure the ID is unique	Yes
callerPhoneNumber	String	The phone number for the caller who initiated the phone call.	Yes
dialedPhoneNumber	String	The phone number dialed to initiated the phone call.	Yes
region	String	The region in which the recording occurred.	Yes
mediaFiles	Array	A recording for a specific call may consist of one or more media files. See Media File Descriptor for details.	Yes
eventHistory	Array	A history of events relevant to the call recording that occurred during the lifetime of the call. See Event History Descriptor for details.	No
callType	String	Call type can be one of the following values: • Internal • Inbound • Outbound • Consult • Unknown	No

Media File Descriptor

The request body can contain one or more Media File Descriptors. Here is a description of the format of this structure:

Attribute	JSON Data Type	Description	Mandatory
callUUID	String	The call UUID associated with the given media file. It does not have to be unique. This value is not used by the GIR system and is simply be returned when the recording is queried.	Yes
startTime	String	The time at which the media file began recording. This must be in the ISO datetime format. For example: 2013-05-17T11:45:32.000-0800	Yes
stopTime	String	The time at which the media file finished recording. This must be in the ISO datetime format. For example: 2013-05-17T11:45:32.000-0800	Yes
mediaId	String	Unique identifier for the media file that is used by clients. It is the client's responsibility to ensure this ID is unique.	Yes*
type	String	The MIME type of the media file. For example: audio/mp3	No
duration	String	The duration of the media file, in milliseconds.	No
size	String	The size of the media file, in bytes.	No
tenant	String	The name of the tenant to which this recording belongs. It should match the name of a tenant on your Genesys Configuration Server.	No
ivrprofile	String	The name of the IVR Profile that serviced this recording.	No
parameters	JSON Object	This is some additional information related to the recording. You	Yes*

Attribute	JSON Data Type	Description	Mandatory
		should populate the parameters described in the following rows in order to realize full GIR functionality.	
parameters.username	String	If the media recording is recorded on an agent, set this parameter to the agent username as it exists in Configuration Server. Otherwise, don't include this parameter.	No
parameters.agentId	String	If parametres.username exists, set this paramter to the name of the agent login object in Configuration Server used on the switch for this recording. Otherwise, don't include this parameter.	No
parameters.recordDN	String	The DN the on which SIP Server pinned the recording. This can be the agent extension DN if parameters.username exists. If the recording is pinned on a Trunk DN or an IVR, parameters.username is not provided and parameters.recordDN is just a DN.	No
parameters.dnis	String	The number being dialed for this recording segment.	No
parameters.ani	String	The number from which the call was made for this recording segment.	No
parameters.callUuid	String	A unique ID for the call that generated this recording segment. It should match the callUUID for this media file.	Yes*
parameters.id	String	A unique ID used for external cross-referencing by SpeechMiner. You may use the CallUUID for this media file for this parameter.	Yes*

Attribute	JSON Data Type	Description	Mandatory
masks	JSON Array	A JSON array specifying the time stamps of the pause and resume periods if the recording is masked by a client application. Each object contains the time and type property. For example: `"masks": [ {"time": "2013-02-06T10:23:10.034Z", "type": "paused"}, {"time": "2013-02-06T10:32:14.034Z", "type": "resume"} ]`	No
pkcs7	String	If the media file is encrypted, this describes the PKCS7 envelop for the encryption performed on the media file. How the media file should be encrypted is out of scope for this API.	No
certAlias	JSON Array	An array of strings of the aliases for the certificates used to encrypt the media file.	No
partitions	JSON Array	A JSON array of strings specifying the partition to which this media file belongs. For more information about partitions, see Access Control for Genesys Interaction Recording Users in the GIR Interaction Recording Solution Guide.	No
accessgroups	JSON Array	A JSON array of strings specifying the Agent Hierarchy group(s) to which this recording belongs. For more information about agent hierarchy and access groups, Access Control for Genesys Interaction Recording Users in the GIR Interaction	No

Attribute	JSON Data Type	Description	Mandatory
		Recording Solution Guide.	
mediaDescriptor	JSON Object	A JSON object used to describe the media file.	Yes
mediaDescriptor.storage	String	The type of storage used for the media resource. The only supported value is: • webDAV	Yes
mediaDescriptor.storage_version	String	The version of the storage being used. It is for information use only.	No
mediaDescriptor.path	String	The storage-specific path to the location where the media is stored.	Yes

* - The API accepts a record without this field, but it is still considered mandatory for the GIR solution to function in a consistent way.

Event History Descriptor

The request body may contain one or more event descriptors. There are three types of events: "Joined", "Left" or "Data". Here is a description of the format of this structure:

Attribute	JSON Data Type	Description	Mandatory
occurredAt	String	The time at which the event happened. This must be in the ISO datetime format. For example: 2013-05-17T11:45:32.000-0800	Yes
event	String	The type of event. It must be one of the following values: • Left • Joined • Data	Yes
calluuid	String	The Call ID associated with this event.	No
contact	JSON Object	Contains information that describes the caller who joined or left the call.	Yes, if the event type is Left or Joined.

Attribute	JSON Data Type	Description	Mandatory
contact.type	String	The type of caller. It must be one of the following values: • User • External	Yes
contact.phoneNumber	String	The phone number for the caller.	Yes
contact.userName	String	The user name of the caller.	Yes, if the contact type is User.
contact.firstName	String	The first name of the caller.	Yes, if the contact type is User.*
contact.lastName	String	The last name of the caller.	Yes, if the contact type is User.*
eventId	String	An eventId used to identify the data event. This must be unique between all events across all the recording segments.	Yes, if the event type is Data.
data	JSON object	A JSON object representing the changes in data for an event of type Data. This free-form JSON object can be used to represent arbitrary data and operations (for example, added, deleted, updated, and so on).	Yes, if the event type is Data.

* - The API accepts a record without this field, but it is still considered mandatory for the GIR solution to function in a consistent way.

Post Insertion Operations

After inserting recording metadata into Interaction Recording Web Services (RWS), you must also post this metadata to the SpeechMiner Interaction Receiver to be able to access the recording through SpeechMiner. For more information, refer to Re-posting the Recordings to SpeechMiner.

Sample

Request

```
POST /internal-api/contact-centers/57c0b771-b57c-4ea8-8655-7ef6d3c58ccc/recordings
Content-Type: application/json

{
```

```
"id":"alex-test-id",
"callerPhoneNumber":"+14160000000",
"dialedPhoneNumber":"+14161111111",
"region":"Can-West",
"callType":"Inbound",
"mediaFiles":[
   {
      "callUUID":"123123",
      "startTime":"2015-06-15T23:59:00.000",
      "stopTime":"2015-06-15T23:59:59.000",
      "mediaId":"123123-FFAB-ABAD",
      "type":"audio/mp3",
      "duration":"59000",
      "size":"67324",
      "parameters":{
         "username":"agent.bob@example.com"
      },
      "masks":[
         {
            "time":"2015-06-15T23:59:10.034Z",
            "type":"paused"
         },
         {
            "time":"2013-06-15T23:59:20.034Z",
            "type":"resume"
         }
      ],
      "mediaDescriptor":{
         "storage":"webDAV",
         "path":"http://example.com/file1.mp3"
      }
   },
   {
      "callUUID":"234222",
      "startTime":"2015-06-15T23:58:00.000",
      "stopTime":"2015-06-15T23:58:59.000",
      "mediaId":"123123-FFAB-ABAC",
      "type":"audio/mp3",
      "mediaDescriptor":{
         "storage":"webDAV",
         "path":"http://example.com/file2.mp3"
      }
   }
],
"eventHistory":[
   {
      "occurredAt":"2015-06-15T23:58:30.000+0000",
      "calluuid":"234222",
      "eventId":"2015-09-10 13:31:00.887_011AP643CSAPR4FKQGQE31TAES00029B",
      "event":"Data",
      "data":{
         "added":{
            "drink":"cola",
            "food":"burger"
         }
      }
   },
   {
      "occurredAt":"2015-06-15T23:58:00.000",
      "calluuid":"234222",
      "event":"Joined",
      "contact":{
         "type":"User",
```

```
            "phoneNumber":"417001",
            "userName":"agent.bob@example.com",
            "firstName":"Bob",
            "lastName":"Smith"
         }
      }
   ]
}
```

Response

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

Field	Description
statusCode	Indicates whether the request was correctly handled. It returns 0 if the response was correctly handled by the server, and other values if it was not correctly handled by the server. See Return Values for details.
statusMessage	For a status that is not 0, this message provides a human readable reason for reason of failure. See Return Values for details.

Settings API

Interaction Recording Web Services includes the Settings API, which you can use to create custom settings groups and settings to suit the needs of your client application. You'll typically only need this API if you plan to enable recording privacy settings in the Genesys Interaction Recording API.

If you query for settings groups (**GET /api/v2/settings** or **GET /api/v2/settings?subresources=*.**, for example), Interaction Recording Web Services returns the Settings resource, which includes URIs that correspond to your available settings groups.

Settings resource

The settings resource includes properties that describe the settings group, which are returned from GET requests to **/api/v2/settings** or GET requests to **/api/v2/settings?subresources=***.

Sample data

```
{
   "settings":[
      {
         "name":"metadata.privacy.agent_fields",
         "value":"callerPhoneNumber, dialedPhoneNumber, dnis, ani, agentId, username,
phoneNumber, userName, firstName, lastName, GSIP_RECORD"
      },
      {
         "name":"metadata.privacy.customer_fields",
         "value":"callerPhoneNumber, dialedPhoneNumber, dnis, ani, agentId, username,
phoneNumber, userName, firstName, lastName, GSIP_RECORD"
      }
   ]
}
```

Resource details

Field	Description
name	A URI-compatible name for the settings group. This name is used as part of the URI to access the group: (for example, **GET /settings/my-settings-group**)
displayName	Name that describes the settings group.
key	The name of the key attribute for this group's settings. Whenever an individual setting needs to be modified, this key attribute is 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

Field	Description
	created without this attribute. If the key attribute is missing, then Interaction Recording Web Services uses "name" as the default identifying attribute.
uri	The URI to the settings group.
path	The path to the settings resource.

Custom settings

You can use custom settings for any purpose in your client application — storing user preferences is a common example. If you want to enable recording privacy settings in the Genesys Interaction Recording API, you'll need to use the Settings API to create settings that tell Interaction Recording Web Services which fields to mask in the data it returns for GetRecordings requests.

The key used in the setting group can be defined when you create the group, and the structure of the setting property values themselves (beyond the key property) can have any structure as long as it's valid JSON.

Attributes

The attributes for each setting group vary. There is no limitation 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.

Storage

The custom settings groups you create using the Settings API are only stored in Cassandra and not synchronized to Configuration Server. Settings groups you define in the Genesys configuration continue to be imported as before.

Check out the examples below to see how to work with custom settings:

Get available settings groups

Request

```
GET http://198.51.100.3:8090/api/v2/settings
```

Response

```
{
   "settings":[
      {
         "displayName":"metadata.privacy.agent_fields",
         "key":"name",
         "uri":"http://198.51.100.3:8090/api/v2/settings/metadata.privacy.agent_fields"
```

```
      },
      {
         "displayName":"metadata.privacy.customer_fields",
         "key":"name",
         "uri":"http://198.51.100.3:8090/api/v2/settings/metadata.privacy.customer_fields"
      }
   ],
   "statusCode":0
}
```

Create a new settings group

Request

```
POST /api/v2/settings
{
   "name": "client-settings",
   "displayName": "Client Settings",
   "key": "name"
}
```

Response

```
{
   "statusCode":0,
   "id":"client-settings",
   "path":"/settings/client-settings",
   "uri":"http://dev-ip9-187.gws.genesys.com:8090/api/v2/settings/client-settings"
}
```

Create a new setting

Request

```
POST /api/v2/settings/client-settings
{
    "name": "Zone",
    "value": "North"
}
```

Response

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

Update a setting

Request

```
PUT /api/v2/settings/client-settings
{
    "name": "Zone",
    "value": "South"
}
```

Response

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

Create a setting with a complex property value

Request

```
POST /api/v2/settings/client-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"
      }
   ]
}
```

Response

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

Update a setting with a complex property value

Request

```
PUT /api/v2/settings/client-settings
{
   "name":"department",
   "possibleValues":[
      {
         "name":"tech_support",
         "displayName":"Tech Support",
         "possibleValues":[
            {
               "displayName":"Computers!!!",
               "name":"computers"
            },
```

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

Response

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

Get settings in the group

Request

```
GET /api/v2/settings/client-settings
```

Response

```
{
   "statusCode":0,
   "settings":[
      {
         "possibleValues":[
            {
               "name":"tech_support",
               "displayName":"Tech Support",
               "possibleValues":[
                  {
                     "displayName":"Computers!!!",
                     "name":"computers"
                  },
                  {
                     "displayName":"Network",
                     "name":"network"
                  }
               ]
            },
            {
               "displayName":"Sales",
               "name":"sales"
            }
         ],
         "name":"department",
         "displayName":"Department"
      },
      {
         "name":"Zone",
         "value":"South"
      }
   ],
   "key":"name"
```

```
}
```

Delete a setting

Request

```
DELETE /api/v2/settings/client-settings
{
    "name": "Zone"
}
```

Response

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

Delete a settings group

Request

```
DELETE /api/v2/settings/client-settings
```

Response

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

Call Recording API

You can use the Call recording API section of the Web Services and Applications API to control call recording in your application.

Recording Label API

Overview

Use this API to create and manage label definitions, and to apply those label definitions to recordings. A label definition defines a label, which can then be applied to a recording. For example, a label definition could be created to mark a recording for further review.

Access this API and its Operations at the base path `/api/v2`. For example:
```
http://localhost:8080/api/v2/recording-label-definitions
```

Prerequisites

- Configure the Elasticsearch schema v2 to enable support for labels and non-deletion features. For existing deployments, follow the migration steps here: Migrating an Existing Elasticsearch Deployment to Schema V2. New deployments after GIR version 8.5.214.02 will have Elasticsearch schema v2 enabled by default.
- Configure the SpeechMiner Settings for Interaction Recording Web Services.

Roles and Permissions

- To access the below listed APIs, the user must be assigned to one of the required roles as mentioned in the specifications. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 Section 2 for reference.
- Specific permissions support your ability to create and administer label definitions, and to set and remove labels from recordings. See: Configuring Permissions for Recording Labels.

Creating a label definition

Request URL	/recording-label-definitions/	
HTTP Method	POST	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_ADD_LABEL_DEFINITIO

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**
name	String	Yes		N/A	The name of the label definition.	• Is case-insensitive. • Must only contain ASCII characters. • Must not include space. • Must be unique among other label definitions. • Must not start with two underscores("__") as it is reserved.
displayName	String	No		Defaults to the value of name.	The display name of the label definition.	Must be unique among other label definitions.
description	String	No		Empty string	The description of the label definition.	

Response

HTTP Status	
Status Codes	**Situations**
201 CREATED	The label definition was created successfully.
400 BAD REQUEST	Bad request received.
403 FORBIDDEN	Forbidden to create the label definition.
409 CONFLICT	The label definition was not created as one with the same name exists already.

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Situations**	**Notes**
statusCode	Integer	Yes	0	N/A	Label definition was created.	
			1 (RequiredParameterMissing)		The required payload attributes are missing.	
			2 (InvalidRequestParameter)		The constraints of payload attributes are not met.	
			3 (OperationForbidden)		The requesting user does not have the permissions required or is attempting to create with reserved name.	
			18 (ResourceAlreadyExists)		A label definition with the same name exists already.	
			20 (unauthenticated)		The requesting user is not properly authenticated or using a non-ops API as an ops user.	
statusMessage	String	No	• Not specified. • A message providing information.	N/A		The statusMessage will provide more information about a failure if statusCode is not 0.

labelDefinition	JSON object	No	JSON object containing the label definition. • path (string): The path to the resource. • name (string): The name of the label definition. • displayName (string, optional): The display name of the label definition. • description (string, optional): The description of the label definition.	N/A	The newly created label definition or existing one in case of an attempt to create an already existing one.	

Example Request

```
POST /recording-label-definitions/
{
        "name": "SomeLabelDefinition",
        "displayName": "New Label Definition",
        "description": "A new label definition"
 }
```

Example Responses

```
{
        "statusCode": 0,
        "labelDefinition": {
                "path": "/recording-label-definitions/2365adc7-67bb-448e-b32d-04731faa9231",
                "name": "SomeLabelDefinition",
                "displayName": "New Label Definition",
                "description": "A new label definition"
```

```
        }
}
{
        "statusCode": 2,
        "statusMessage": "Invalid value specified."
}
{
        "statusCode": 3,
        "statusMessage": "This operation is not allowed."
}
{
        "statusCode": 18,
        "statusMessage": "Resource exists already."
        "labelDefinition": {
                "path": "/recording-label-definitions/a6a6f1d9-788a-47b4-bdeb-80b3d598cfa6",
                "name": "ExistingLabelDefinition",
                "displayName": "Existing Label Definition."
                "description": "An existing label definition"
        }
}
```

Updating a label definition

Request URL	/recording-label-definitions/<labelDefinitionId>	
HTTP Method	PUT	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_ADD_LABEL_DEFINITIO

Note: You can use the path attribute from the response returned by the GET or POST operations for label definition.

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**
name	String	Yes		N/A	The name of the label definition.	Name cannot be updated—must be specified to the same value as the existing name.
displayName	String	No		Default to the value of name if not specified.	The display name of the label definition.	Must be unique among other label definitions.

description	String	No		Empty string	The description of the label definition.	

Response

HTTP Status	
Status Codes	**Situations**
200 OK	The label definition was updated successfully.
400 BAD REQUEST	Bad request received.
403 FORBIDDEN	Forbidden to update the label definition.
404 NOT FOUND	Cannot find the label definition to update.

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Situations**	**Notes**
statusCode	Integer	Yes	0	N/A	Label definition was updated.	
			1 (RequiredParameterMissing)		The required payload attributes are missing.	
			2 (InvalidRequestParameter)		The constraints of payload attributes are not met.	
			3 (OperationForbidden)		The requesting user does not have the permissions required or is attempting to update with different name.	
			6 (ResourceNotFound)		The specified label definition cannot be found.	

			20 (Unauthenticated)		The requesting user is not properly authenticated or using a non-ops API as an ops user.	
statusMessage	String	No	• Not specified. • A message providing information.	N/A		The statusMessage will provide more information about a failure if statusCode is not 0.
labelDefinition	JSON object	Yes	• name (string): The name of the label definition. • displayName (string, optional): The display name of the label definition. • description (string, optional): The description of the label definition. • path (string, optional): The path of the label definition. • Must be unique among other label	N/A		

			definitions.			

Example Request

```
PUT /recording-label-definitions/{labelDefinitionId}
{
        "name": "SomeLabelDefinition",
        "displayName": "New Label Definition",
        "description": "A new label definition"
 }
```

Example Responses

```
{
        "statusCode": 0,
        "labelDefinition": {
                "path": "/recording-label-definitions/2365adc7-67bb-448e-b32d-04731faa9231",
                "name": "SomeLabelDefinition",
                "displayName": "Updated Label Definition",
                "description": "An updated label definition"
        }
}

{
        "statusCode": 2,
        "statusMessage": "Invalid value specified."
}

{
        "statusCode": 3,
        "statusMessage": "This operation is not allowed."
}

{
        "statusCode": 6,
        "statusMessage": "Custom label definition cannot be found."
}

{
        "statusCode": 20,
        "statusMessage": "Access denied."
}
```

Deleting a label definition

Request URL	/recording-label-definitions/{labelDefinitionId}
HTTP Method	DELETE

Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_DELETE_LABEL_DEFIN

Note: You can use the path attribute from the response returned by the GET or POST operations for label definition.

Response

HTTP Status		
Status Codes	**Situations**	**Notes**
200 OK	The label definition was deleted successfully.	
403 FORBIDDEN	Forbidden to delete the label definition.	
404 NOT FOUND	Cannot find the label definition to delete.	

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Situations**	**Notes**
statusCode	Integer	Yes	0	N/A	The label definition was deleted.	
			3 (OperationForbidden)		The requesting user does not have the permissions required.	
			6 (ResourceNotFound)		The specified label definition cannot be found.	
			20 (Unauthorized)		The requesting user is ops.	
statusMessage	String	No	• Not specified.	N/A		The statusMessage will provide more

			• A message providing information.			information about a failure if statusCode is not 0.

Example Responses

```
{
        "statusCode": 0
}

{
        "statusCode": 3,
        "statusMessage": "This operation is not allowed."
}

{
        "statusCode": 5,
        "statusMessage": "Access denied."
}

{
        "statusCode": 6,
        "statusMessage": "Custom label definition cannot be found."
}
```

Retrieving label definitions

Request URL	/recording-label-definitions	
HTTP Method	GET	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	N/A

Note: Returns all the label definitions.

Parameters						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**
type	Comma-separated string of label definition types.	No	• Reserved • Custom	N/A	The type of label definition to return.	If the type is not specified, the Reserved and Custom

						label definitions are returned. If type is empty ("?type="), the Reserved and Custom label definitions are returned. If type contains a valid value, only the type of label definition that was specified is returned.
fields	Comma-separated string of fields.	No	• name • displayName • description • *	N/A	The fields to be returned from the label definition. Path is always returned. Use "*" to return all fields.	If fields is not specified, the path and name will be returned. If fields is empty("?fields="), the path is returned. If fields contains a valid value, the path and the specified fields are returned.

Response

HTTP Status	Fields	Possible Values	Situations
200 OK	statusCode	0 (Ok)	Retrieval of all label definitions succeeded.
		7 (PartialResponse)	Retrieval of some label definitions failed.
		12 (UnableToRetrieveResources)	Retrieval of all label definitions failed.
	labelDefinitions	An array of label definitions.	Always

Label definition response data

Fields	Description
path	The path of the label definition.
name	The name of the label definition.
type	The type of the label definition.

displayName	The display name of the label definition.
description	The description of the label definition.

Example

```
GET /recording-label-definitions
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/211e698a-279e-389b-840a-8c89c8e8b97c",
      "name": "__evaluated"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550a",
      "name": "comment"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550b",
      "name": "importantTag"
    }
  ]
}

GET /recording-label-definitions?fields=
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/211e698a-279e-389b-840a-8c89c8e8b97c"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550a"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550B"
    }
  ]
}

GET /recording-label-definitions?fields=*
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/211e698a-279e-389b-840a-8c89c8e8b97c",
      "name": "__evaluated",
      "type": "Reserved",
      "displayName": "Evaluated",
      "description": "A label indicating the interaction on which it is applied has been
evaluated."
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550a",
      "name": "comment",
      "type": "Custom",
      "displayName": "Comment",
      "description": "A label for recording comment."
    },
    {
```

```
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550B",
      "name": "importantTag",
      "type": "Custom",
      "displayName": "Important",
      "description": "A label to tag recordings that are important."
    }
  ]
}

GET /recording-label-definitions?fields=name,displayName
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/211e698a-279e-389b-840a-8c89c8e8b97c",
      "name": "__evaluated",
      "displayName": "Evaluated"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550a",
      "name": "comment",
      "displayName": "Comment"
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550B",
      "name": "importantTag",
      "displayName": "Important"
    }
  ]
}

GET /recording-label-definitions?fields=*&type=Custom
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550a",
      "name": "comment",
      "type": "Custom",
      "displayName": "Comment",
      "description": "A label for recording comment."
    },
    {
      "path": "/recording-label-definitions/feda71ce-ea61-4960-b39a-c0a3063d550B",
      "name": "importantTag",
      "type": "Custom",
      "displayName": "Important",
      "description": "A label to tag recordings that are important."
    }
  ]
}

GET /recording-label-definitions?type=Reserved
{
  "statusCode": 0,
  "labelDefinitions": [
    {
      "path": "/recording-label-definitions/211e698a-279e-389b-840a-8c89c8e8b97c",
      "name": "__evaluated"
    }
  ]
}
```

Adding a label to a recording

Request URL	/recordings/<recid>/labels	
HTTP Method	POST	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_ADD_LABEL

Payload				
Properties	**Type**	**Mandatory**	**Default Values**	**Description**
name	String	Yes	N/A	The name of the label definition to be added to the recording.
content	JSON Object	No	empty	The content associated with the label to be added to the recording. This is a free-form JSON element that accepts valid JSON types.

Response

HTTP Status	
Status Codes	**Situations**
201 CREATED	The label has been added to the recording successfully.
403 FORBIDDEN	• The recording requested can be found, but the label cannot be added to it for some reason (for example, region not matched, requesting user cannot access the recording). • The recording requested cannot be found. • There already exists a label instance that is identical to the one to add on the same recording. • The label definition name cannot be found.

Payload					
Fields	**Type**	**Mandatory**	**Possible Values**	**Situations**	**Notes**
statusCode	Integer	Yes	0 (Ok)	The label was added to the requested recording successfully.	
			13 (UnableToCreateResource)	Label failed to be added to the requested recording.	
			18 (ResourceAlreadyExists)	There already exists a label instance that is identical to the one to add on the same recording.	
statusMessage	String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.
id	String	No	ID of the created label.	HTTP status is 201.	
path	String	No	Path of the created label.	HTTP status is 201.	

Example Requests

```
POST /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels
{
   "name" :  "comment" ,
   "content" : {
     "time" :  "2017-01-01T15:07:17Z" ,
     "text" :  "This is an awesome comment!"
   }
}

POST /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels
{
   "name" :  "importantTag"
}

POST /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels
{
   "name" :  "comment" ,
   "content" : {
     "time" :  "2017-08-05T15:07:17Z" ,
     "rating" :  "excellent" ,
```

```
    "by" : {
      "userName" :  "supervisor_1" ,
      "firstName" :  "super" ,
      "lastName" :  "visor"
    }
  }
}
```

Example Responses

```
201 CREATED
{
  "statusCode": 0,
  "id": "8c9a634f-343d-49b3-8322-b571be520b31",
  "path": "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31"
}
```

```
403 FORBIDDEN
{
  "statusCode": 13,
  "statusMessage": "Cannot add label to recording."
}
```

Adding a label to multiple recordings

Request URL	/recording-labels	
HTTP Method	POST	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_ADD_LABEL

Payload					
Level-0 Properties	**Level-1 Properties**	**Type**	**Mandatory**	**Default Values**	**Description**
recordingIds	N/A	JSON Array	Yes	N/A	The list of recording IDs that the label needs to be added to.
label	name	String	Yes	N/A	The name of the label definition to be added to the

					recordings.
	content	JSON Object	No	empty	The content associated with the label to be added to the recordings. This is a free-form JSON element that accepts valid JSON types.

Response

HTTP Status	Situations
200 OK	The requested list of recordings is empty.
201 CREATED	The label has been added to all the requested recordings successfully and at least one label resource has been created.
207 MULTI-STATUS	The label failed to be added to some but not all of the requested recordings, but succeeded for the rest.
403 FORBIDDEN	The label failed to be added to all of the requested recordings.

Payload					
Fields	**Type**	**Mandatory**	**Possible Values**	**Situations**	**Notes**
statusCode	Integer	Yes	0 (Ok)	The label is added to all the requested recordings successfully.	
			7 (PartialResponse)	The label succeeded on some of the requested recordings, but failed on the rest.	
			13 (UnableToCreateResource)	The label failed to be added to all the requested recordings.	

statusMessage	String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.
succeeded	JSON Array	Yes	The recording id to created label URI mapping for the recordings to which label has been successfully added.	It will always be displayed in the response but can be empty.	
failed	JSON Array	Yes	A list of blocks that describes the failures.	It will always be displayed in the response but can be empty.	

Example Requests

```
POST /recording-labels
{
   "recordingIds" : [
     "bf697521-3ffc-46f1-b840-f7230e940df3" ,
     "76705af9-65d4-46a4-af84-e984ef09ec5d"
   ],
   "label" : {
     "name" :  "comment" ,
     "content" : {
       "time" :  "2017-08-05T15:07:17Z" ,
       "text" :  "This is an awesome comment!"
     }
   }
}

POST /recording-labels
{
   "recordingIds" : [
     "bf697521-3ffc-46f1-b840-f7230e940df3" ,
     "76705af9-65d4-46a4-af84-e984ef09ec5d"
   ],
   "label" : {
     "name" :  "importantTag"
   }
}

POST /recording-labels
{
   "recordingIds" : [
     "bf697521-3ffc-46f1-b840-f7230e940df3" ,
     "76705af9-65d4-46a4-af84-e984ef09ec5d"
   ],
   "label" : {
```

```
     "name" :  "comment" ,
     "content" : {
       "time" :  "2017-08-05T15:07:17Z" ,
       "rating" :  "excellent" ,
       "by" : {
         "userName" :  "supervisor_1" ,
         "firstName" :  "super" ,
         "lastName" :  "visor"
       }
     }
   }
}
```

Example Responses

```
200 OK
{
  "statusCode": 0,
  "succeeded": [],
  "failed": []
}

201 CREATED
{
  "statusCode": 0,
  "succeeded": [
    {
      "recordingId": "bf697521-3ffc-46f1-b840-f7230e940df3",
      "id": "8c9a634f-343d-49b3-8322-b571be520b31",
      "path": "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31"
    },
    {
      "recordingId": "76705af9-65d4-46a4-af84-e984ef09ec5d",
      "id": "1574da8d-ed8d-419d-b2e7-f00c1dabe0dc",
      "path": "/recordings/76705af9-65d4-46a4-af84-e984ef09ec5d/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc"
    }
  ],
  "failed": []
}

207 MULTI-STATUS
{
  "statusCode": 7,
  "statusMessage": "Add of label failed on some recordings.",
  "succeeded": [
    {
      "recordingId": "bf697521-3ffc-46f1-b840-f7230e940df3",
      "id": "8c9a634f-343d-49b3-8322-b571be520b31",
      "path": "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31"
    }
  ],
  "failed": [
    {
      "recordingId": "76705af9-65d4-46a4-af84-e984ef09ec5d",
      "statusCode": 13,
      "statusMessage": "Cannot add label."
    },
    {
      "recordingId": "d0464cf3-f81f-49b5-bc0e-bd166161d756",
```

```
      "statusCode": 13,
      "statusMessage": "Internal error."
    }
  ]
}

403 FORBIDDEN
{
  "statusCode": 13,
  "statusMessage": "Add of label failed on all the recordings.",
  "succeeded": [],
  "failed": [
    {
      "recordingId": "d0464cf3-f81f-49b5-bc0e-bd166161d756",
      "statusCode": 13,
      "statusMessage": "Cannot add label."
    }
  ]
}
```

Updating a label on a recording

Request URL	/recordings/<recid>/labels/<id>	
HTTP Method	PUT	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_ADD_LABEL

Payload				
Properties	**Type**	**Mandatory**	**Default Values**	**Description**
content	JSON Object	No	empty	The content associated with the label to be updated to the recording. This is a JSON object which can be customized depending on the application. The examples below show a possible use case of adding user comments on a recording.

Effect

When a label is updated:

- The content of the label is updated to the one that is specified in the payload.
- The field createTime is updated to the time when the update action was performed.
- The field createUser is updated to the username of the user who requested the update.

Response

HTTP Status	
Status Codes	**Situations**
200 OK	The requested label has been updated successfully.
403 FORBIDDEN	• The recording requested can be found but the label cannot be updated to it for whatever reason (for example, region not matched, requesting user cannot access the recording). • The recording requested cannot be found.
404 NOT FOUND	The requested label cannot be found.

Payload					
Fields	**Type**	**Mandatory**	**Possible Values**	**Situations**	**Notes**
statusCode	Integer	Yes	0 (Ok)	The label has been updated to the requested recording successfully.	
			6 (ResourceNotFound)	The label cannot be found.	
			15 (UnableToUpdateResource)	• The recording requested can be found but the label cannot be updated to it for whatever reason (for	

				example, region not matched, requesting user cannot access the recording). • The recording requested cannot be found.	
statusMessage	String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.

Example Requests

```
PUT /recordings/<labels>
{
   "content" : {
     "time" :  "2017-08-05T15:07:17Z" ,
     "text" :  "This is an awesome comment!"
   }
}

PUT /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/5dbdb466-5377-4351-bd1e-
c4066437d410
{
   "content" : {
     "time" :  "2017-08-05T15:07:17Z" ,
     "rating" :  "excellent"
     "by" : {
         "userName" :  "supervisor_1" ,
         "firstName" :  "super" ,
         "lastName" :  "visor"
     }
   }
}
```

Example Responses

```
200 OK
{
   "statusCode" : 0
}

403 FORBIDDEN
{
```

```
   "statusCode" : 15,
   "statusMessage" :  "Cannot update the label for recording."
}

404 NOT FOUND
{
   "statusCode" : 6,
   "statusMessage" :  "Requested label cannot be found."
}
```

Deleting a label from a recording

Request URL	/recordings/<recid>/labels/<id>	
HTTP Method	DELETE	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_DELETE_LABEL

Response

HTTP Status	
Status Codes	**Situations**
200 OK	• The label has been deleted from the requested recording successfully.
403 FORBIDDEN	• The recording requested can be found but the label cannot be deleted from it for some reason (for example, region not matched, requesting user cannot access the recording). • The recording requested cannot be found.

Payload					
Fields	**Type**	**Mand-atory**	**Possible Values**	**Situations**	**Notes**
statusCode	Integer	Yes	0 (Ok)	The label has been deleted from the requested recording	

				successfully. OR The label cannot be found within the requested recording.	
			14 (UnableToDeleteResource)	The label failed to be deleted from the requested recording.	
statusMessage	String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.

Example Responses

```
DELETE /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc
200 OK
{
   "statusCode" : 0
}

DELETE /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31
403 FORBIDDEN
{
   "statusCode" : 14,
   "statusMessage" :  "Label failed to be deleted from the recording."
}
```

Getting all labels from a recording

Request URL	/recordings/<recid>/labels
HTTP Method	GET
Required Roles	• Administrator • Supervisor • Agent
Required Permissions	N/A

Parameters						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**
Fields	comma-separated string of fields	No	• * • name • createTime • createUser • content	N/A	The fields to be returned back for the label definition If fields parameter is set, path and id are always returned regardless if they are specified or not If fields value is set to "*", all the fields will be returned in the response.	if fields not specified, path, id, and name will be returned; else if fields set to empty, path and id will be returned; else, path, id, and fields specified will be returned.

Response

HTTP Status	
Status Codes	**Situations**
200 OK	The retrieval of the labels succeeded or there are no labels associated with the recording.
403 FORBIDDEN	• The recording for which the labels are requested can be found but cannot be retrieved for some reason (for example, region not matched, requesting user cannot access the recording). • The recording for which the labels are requested cannot be found.

Payload						
Fields	**Subfields**	**Type**	**Mandatory**	**Possible Values**	**Situations**	**Notes**
statusCode		Integer	Yes	0 (Ok)	The retrieval of the labels succeeded.	
				12 (UnableToRetrieveResource)	The retrieval of the labels failed.	

statusMessage		String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.
labels		Array of labels	No	Each label will have the fields requested.	Always	
	id	String	No	The ID of the created label.	Always	
	path	String	No	The path of the created label.	Always	
	name	String	No	The name of the label definition.	Not specified or if fields parameter is received.	
	createTime	String	No	The time when the label was last updated.	If the fields parameter is received.	
	createUser	String	No	The username of the user who last updated the label on the recording.	If the fields parameter is received.	
	content	JSON object	No	The JSON object containing custom content, if populated.	If the fields parameter is received.	The content is a JSON object that can contain custom data.

Examples

```
GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df5/labels
200 OK
{
   "statusCode" : 0,
   "labels" : []
}

GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df4/labels
```

```
200 OK
{
   "statusCode" : 0,
   "labels" : [
     {
       "path" :  "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31" ,
       "name" :  "comment" ,
       "id" :  "8c9a634f-343d-49b3-8322-b571be520b31"
     }
   ]
}
GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels?fields=
200 OK
{
   "statusCode" : 0,
   "labels" : [
     {
       "path" :  "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31" ,

       "id" :  "8c9a634f-343d-49b3-8322-b571be520b31"
     }
   ]
}
GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels?fields=content
200 OK
{
   "statusCode" : 0,
   "labels" : [
     {
       "path" :  "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/
8c9a634f-343d-49b3-8322-b571be520b31",
       "id" :  "8c9a634f-343d-49b3-8322-b571be520b31" ,
       "content" : {
         "time" :  "2017-04-22T12:42:44.000+0000" ,
         "comment" :  "This is an awesome comment!"
       }
     }
   ]
}
GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels?fields=*
200 OK
{
   "statusCode" : 0,
   "labels" : [
     {
       "path" :  "/recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels
/8c9a634f-343d-49b3-8322-b571be520b31" ,
       "name" :  "comment" ,
       "id" :  "8c9a634f-343d-49b3-8322-b571be520b31" ,
       "createTime" :  "2017-04-22T12:42:46.000+0000" ,
       "createUser" :  "supervisor" ,
       "content" : {
         "time" :  "2017-04-22T12:42:44.000+0000" ,
         "comment" :  "This is an awesome comment!"
       }
     }
   ]
}

GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels
403 FORBIDDEN
```

```
{
   "statusCode" : 12,
   "statusMessage" :  "Recording cannot be found."
}

GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels?fields=*
403 FORBIDDEN
{
   "statusCode" : 12,
   "statusMessage" :  "Recording cannot be accessed."
}
```

Getting a specific label from a recording

Request URL	/recordings/<recid>/labels/<id>
HTTP Method	GET
Required Roles	• Administrator • Supervisor • Agent
Required Permissions	N/A

Response

HTTP Status	
Status Codes	**Situations**
200 OK	The retrieval of the label succeeded.
403 FORBIDDEN	• The recording for which the label is requested can be found but cannot be retrieved for some reason (for example, region not matched, requesting user cannot access the recording). • The recording for which the label is requested cannot be found.
404 NOT FOUND	• The label cannot be found by the ID specified.

Payload					
Fields	**Type**	**Mandatory**	**Possible Values**	**Situations**	**Notes**

statusCode	Integer	Yes	0 (Ok)	Retrieval of the label succeeded.	
			6 (ResourceNotFound)	The label cannot be found.	
			12 (UnableToRetrieveResource)	Retrieval of the label failed.	
statusMessage	String	No	• Not specified. • A message providing information.		The statusMessage will provide more information about a failure if statusCode is not 0.
label	JSON Object	No	The actual data of the label requested. A label contains the following information: • Path (string): the path of the label. • Name (string): the name of the label. • Type (string): the type of the label. • ID (string): the ID of the label. • createTime (string): the time the label was last created or updated. • createUser (string): the last user that created or updated the label.	It will only be displayed if the retrieval of the label succeeded.	

			• content (a JSON object): an object containing custom data, if populated.		

Examples

```
GET /recordings/bf697521-3ffc-46f1-b840-f7230e940da3/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc
200 OK
{
   "statusCode" : 0,
   "label" : {
     "path" :  "/recordings/bf697521-3ffc-46f1-b840-f7230e940da3/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc" ,
     "name" :  "comment" ,
     "id" :  "1574da8d-ed8d-419d-b2e7-f00c1dabe0dc" ,
     "createTime" :  "2017-04-20T04:11:02.000+0000" ,
     "createUser" :  "supervisor" ,
     "content" : {}
   }
}

GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc
403 FORBIDDEN
{
   "statusCode" : 12,
   "statusMessage" :  "Recording cannot be found."
}

GET /recordings/bf697521-3ffc-46f1-b840-f7230e940df3/labels/1574da8d-ed8d-419d-
b2e7-f00c1dabe0dc
404 NOT FOUND
{
   "statusCode" : 6,
   "statusMessage" :  "Label cannot be found for the recording."
}
```

Recording Non-Deletion API

Overview

When you protect recordings from deletion, they cannot be deleted during Media Lifecycle Management purge tasks, by SpeechMiner, API, or any method. Protecting and unprotecting from deletion also affects associated screen recordings.

The following API operations are accessible at the base path `/api/v2`
For example, `http://localhost:8080/api/v2/recordings/<recid>`

Prerequisites

- Configure the Elasticsearch schema v2 to enable support for labels and non-deletion features. For existing deployments, follow the migration steps here: Migrating an Existing Elasticsearch Deployment to Schema V2. New deployments after GIR version 8.5.214.02 will have Elasticsearch schema v2 enabled by default.
- Configure the SpeechMiner Settings for Interaction Recording Web Services.

Roles and Permissions

- To access the below listed APIs, the user must be assigned to one of the required roles as mentioned in the specifications. For details, see Configuring Roles for Recording Users. The Genesys Interaction Recording API uses Basic HTTP Authentication. See RFC 2617 Section 2 for reference.
- Specific permissions enable your ability to protect (and unprotect) recordings from deletion. See: Configuring Permissions for Recording Non-Deletion

Apply Non-Deletion to a Recording

Use this operation to prevent a voice recording (and any associated screens) from being deleted.

Request URL	/recordings/*<recid>*	
HTTP Method	POST	
Required Roles	• Administrator	• Supervisor • Agent

Required Permissions	N/A	RECORDING_PERMISSION_APPLY_NON_DELETE

Payload						
Attribute	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**
operationName	String	Yes	applyNonDelete	N/A	Mark a recording for non-deletion.	All the associated screen recordings at the time of the operation will also be updated.

Response

HTTP Status		
Status Codes	**Situations**	**Notes**
200 OK	The recording(s) has been marked for non-deletion successfully.	
400 BAD REQUEST	The operationName does not match any of the possible values described above.	
403 FORBIDDEN	• The requesting user does not have the roles required. • The requesting user does not have the recording permissions required.	
404 NOT FOUND	The recording for which to mark non-deletion cannot be found.	
500 INTERNAL SERVER ERROR	If an internal error occurs.	

Attribute	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Situations**	**Notes**
Payload						
statusCode	Integer	Yes	0 (Ok)	N/A	The recording has been marked for non-deletion.	

Attribute	JSON Data Type	Mandatory	Possible Values	Default Value	Situations	Notes
			2 (InvalidRequestParameter)		The operationName does not match the above-mentioned operations.	
			3 (OperationForbidden)		The requesting user does not have the recording permissions required.	
			4 (InternalError)		If internal error occurs.	
			5 (Unauthorized)		The requesting user does not have the roles required.	
			6 (ResourceNotFound)		The recording for which to mark non-deletion cannot be found.	
statusMessage	String	No	• Not specified. • A message providing information.	N/A		The statusMessage will provide more information about a failure if statusCode is not 0.

Example Request

```
# Mark voice recording d15a8bee-a720-4def-9674-0767e6fde196 and its screen recordings(if any)
for non-deletion
POST /recordings/d15a8bee-a720-4def-9674-0767e6fde196
{
     "operationName" :  "applyNonDelete"
}
```

Example Response

```
200 OK
{
     "statusCode" : 0
}

400 BAD REQUEST
{
     "statusCode" : 2,
     "statusMessage" :  "Parameter 'operationName' is invalid: The specified value is not
within valid range"
}

403 FORBIDDEN
{
     "statusCode" : 5,
     "statusMessage" :  "Insufficient user roles."
}

403 FORBIDDEN
{
     "statusCode" : 3,
     "statusMessage" :  "Insufficient recording permissions."
}

404 NOT FOUND
{
     "statusCode" : 6,
     "statusMessage" :  "Requested recording [f2197c79-3304-4427-9e73-48a5a8903484] cannot be
found."
}

500 INTERNAL SERVER ERROR
{
     "statusCode" : 4,
     "statusMessage" :  "Internal server error - please contact administrator."
}
```

Remove Non-Deletion from a Recording

Request URL	/recordings/<*recid*>	
HTTP Method	POST	
Required Roles	• Administrator	• Supervisor • Agent
Required Permissions	N/A	RECORDING_PERMISSION_UNAPPLY_NON_DELET

Payload						
Attributes	**JSON Data Type**	**Mandatory**	**Possible Values**	**Default Value**	**Description**	**Notes**

operationName	String	Yes	unapplyNonDeletion	N/A	Unmark a recording for non-deletion.	All the associated screen recordings at the time of the operation will also be updated.

Response

Status Codes	Situations	Notes
HTTP Status		
200 OK	The recording(s) has been unmarked for non-deletion successfully.	
400 BAD REQUEST	The operationName does not match any of the possible values described above.	
403 FORBIDDEN	• The requesting user does not have the roles required. • The requesting user does not have the recording permissions required.	
404 NOT FOUND	The recording for which to unmark non-deletion cannot be found.	
500 INTERNAL SERVER ERROR	An internal error occurs.	

Attribute	JSON Data Type	Mandatory	Possible Values	Default Value	Situations	Notes
Payload						
statusCode	Integer	Yes	0 (Ok)	N/A	The recording has been unmarked for non-deletion.	
			2 (InvalidRequestParameter)		The operationName does not match any of the possible	

Attribute	JSON Data Type	Mandatory	Possible Values	Default Value	Situations	Notes
					values described above.	
			3 (OperationForbidden)		The requesting user does not have the recording permissions required.	
			4 (InternalError)		If internal error occurs.	
			5 (Unauthorized)		The requesting user does not have the roles required.	
			6 (ResourceNotFound)		The recording for which to unmark non-deletion cannot be found.	
statusMessage	String	No	• Not specified. • A message providing information.	N/A		The statusMessage will provide more information about a failure if statusCode is not 0.

Example Request

Use this operation to remove a voice recording's protection from deletion.

```
# Unmark voice recording d15a8bee-a720-4def-9674-0767e6fde196 and its screen recordings(if
any) for non-deletion
POST /recordings/d15a8bee-a720-4def-9674-0767e6fde196
{
     "operationName" :  "unapplyNonDelete"
}
```

Example Response

```
200 OK
{
     "statusCode" : 0
}

400 BAD REQUEST
{
     "statusCode" : 2,
     "statusMessage" :  "Parameter 'operationName' is invalid: The specified value is not
within valid range"
}

403 FORBIDDEN
{
     "statusCode" : 5,
     "statusMessage" :  "Insufficient user roles."
}

403 FORBIDDEN
{
     "statusCode" : 3,
     "statusMessage" :  "Insufficient recording permissions."
}

404 NOT FOUND
{
     "statusCode" : 6,
     "statusMessage" :  "Requested recording [f2197c79-3304-4427-9e73-48a5a8903484] cannot be
found."
}

500 INTERNAL SERVER ERROR
{
     "statusCode" : 4,
     "statusMessage" :  "Internal server error - please contact administrator."
}
```

API Responses

Successful Results

A successful response to a request is marked by HTTP Status Code 200 (OK). In that case, your application may get additional information in the header and the body of the response. Refer to the Response section of your operation's page to get the detailed list of returned information. Questions about the returned content can be submitted as comments in this wiki. The following table lists the standard HTTP codes used for a successful response.

Code	Title	Description	Mandatory Headers
200	OK	Success!	Access-Control-Allow-Headers: Authorization,Origin Access-Control-Allow-Origins: server-url Access-Control-Allow-Methods: GET, POST, OPTIONS Access-Control-Allow-Credentials: true

Errors

For responses with HTTP status code 40x or 50x, the response body contains an application-specific description of the error instead of a representation of the requested resource. Errors consist of an application-specific error code and description, and are formatted in JSON as follows:

```
error = {
 "http_method" : "POST",
"title" : "bad parameter",
"description" : "bad parameter 'server mode' reason : server mode has not a correct value
'something'",
"code" : 4020,
"uri" : "http://localhost:8080/server/mode"
}
```

The following table lists the specific errors that can be returned. This list is not restrictive; additional error codes could be returned due to external web servers and layers involved:

Code	Title	Description
400	Bad Request	General error which can be one of the following reasons: • Missing required parameter.

Code	Title	Description
		• Parameter value of unexpected type. • Invalid object syntax. • Missing required attribute.
401	Unauthorized	Credentials are missing or incorrect, or the given user is not allowed to execute a given service (such as an administrative service method that changes the profile schema).
403	Forbidden	The operation is forbidden and the reason is specified in the error message.
404	Not Found	The specified URI is invalid, or the requested resource does not exist.
405	Method Not Allowed	Returned when an unsupported operation is requested. For instance, if a resource supports only PUT and GET operations, a POST request on this resource returns this error.
500	Internal Server Error	An unexpected error occurred (for instance, a runtime exception).
502	Bad Gateway	Returned when one or more of the backend systems required to fulfill the response are either unavailable or returned an error.
503	Service Unavailable	The application is unable to process the given request.