Web Services API Reference

Recording Access

12/13/2025

Recording Access

Contents

Recordings will be created in Web Services by an external Genesys component and will be accessible from the Web Services API. The **/recordings** URI is used to access recordings externally while **/ops/contact-centers/{id}/recordings** is used by the previously mentioned component. This section will discuss both the internal and the external API's. "OPS Admin" permission means that the functionality is available via the ops URI.

Operations

The following operations are supported for the **recording** resource:

Operation	Description	Permissions
GET	Returns a list of all recording URIs for the contact center . Note that GET `/recordings` without any *query* parameters will return an error. This operation is only be permitted with a query parameter as GET `/recordings` could result in millions of results. If GET with a query results in a data set of over 100 recordings, only the first 100 results are returned.	• OPS Admin • Contact Center Admin

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

Operation	Description	Permissions
GET	Returns a "recording" resource which corresponds to a single call with a unique id. The recording may consist of multiple media files.	• OPS Admin • Contact Center Admin
DELETE	Remove the specified recording	• OPS Admin

Region Affinity

Each recording has a region value associated with it. The region value is a mandatory parameter when creating the recording using a POST request . Each Web Services server has a region associated with it. When a Web Services server receives a request to read a call recording by Id/ play recordings, the region settings of the server and the region value of recording is compared. If they are not the same, the server returns `NOT_FOUND`. When querying, the server adds an additional filter to each query, the region of the server and the recording must match exactly.

Delete a Call Recording

Delete a call recording by using the DELETE operation on the following path:

```
/api/v2/ops/contact-centers/{contact-center-id}/recordings/{recording-id}[?subresources=*]
```

Parameters

Name	Accept Value	Description	Default
subresources	* Means true, otherwise false	Flag to to specify if the media resource is deleted from external storage. If not specified, only meta-data will be deleted.	false

Example

Delete only meta-data.

```
DELETE /api/v2/ops/contact-centers/a5539ef5-7dd5-4150-818f-ee1bcd447cbe/recordings/
bf697521-3ffc-46f1-b840-f7230e940df3

{
        "statusCode": 0
}
```

Delete both meta-data and media.

```
DELETE /api/v2/ops/contact-centers/a5539ef5-7dd5-4150-818f-ee1bcd447cbe/recordings/
bf697521-3ffc-46f1-b840-f7230e940df3?subresources=*

{
        "statusCode": 0
}
```

If some of the recording media file cannot be deleted, recording meta-data is deleted and a partial success with a description of the failed media deletion is returned.

```
DELETE /api/v2/ops/contact-centers/a5539ef5-7dd5-4150-818f-ee1bcd447cbe/recordings/
bf697521-3ffc-46f1-b840-f7230e940df3?subresources=*

{
        "statusCode": 7,
        "failed":
        [
                {
                        "id": "76705af9-65d4-46a4-af84-e984ef09ec5d",
                        "mediaDescriptor": {"path": "http://10.10.15.89/media1.mp3",
"storage": "webDAV"},
                        "'statusCode": 4,
                        "statusMessage": "Fail to send request deleting http://10.10.15.89/
media1.mp3'
                }
```

```
          ]
}
```

Attributes

The following attributes are available for the **recording** resource.

Attribute	Type	Description	Access Level	Valid value
id	String	Unique id of a resource as provided by the OPS admin. It is the responsibility of the request to guarantee the uniqueness of an id.	GET, POST	Any non-empty string with no leading or ending spaces
callerPhoneNumber	String	The source phone of a call	GET, POST	Any non-empty string with no leading or ending spaces
dialedPhoneNumber	String	The dialed phone number	GET, POST	Any non-empty string with no leading or ending spaces
region	String	The region of the recording	GET, POST	Any non-empty string with no leading or ending spaces
mediaFiles	Array of `MediaFileDescriptor`	A recording for a specific call may consist of multiple media files. This array contains a set of media file descriptors. The structure of is described in	GET, POST	Must not be empty
eventHistory	Array of `EventDescriptor`	A history of events relevant to call recording that occured during the lifetime of the call.	GET, POST	Must not be empty

Tip

Start time of a call is not posted. It is evaluated as the minumum start time in the media files.

Tip

If a POST operation updates an existing resource, the `mediaFiles` collections is updated as follows for each posted media file desicriptor:

- The media file descriptor is added to the collection if the media file descriptor is unique.
- The media file descriptor overwrites the existing record if a descriptor with specified media URI is already present in the collection.

Tip

If a POST operation updates an existing resource, the `eventHistory` collections is updated as follows for each posted event descriptor:

- The event descriptor is added to the collection if the event is not already present in the collection.
- The event descriptor overwrites the existing record if it is already present in the collection. A record is considered to describe the same event if the `occurredAt`, `event`, and `contact phoneNumber` are the same.

The following attributes are available for **MediaFileDescriptor:**

Attribute	Type	Description	Access Level	Valid Value	Mandatory
mediaUri	URI	The URI pointing to Web Services which can be used to fetching the resource. Contact center admin authorization credentials must be provided.	GET		Yes
mediaDescriptor	`MediaDescriptor`	Descriptor of media resource which Web Services will use to fetch.	POST	Valid mediaDescriptor	Yes
originalMediaDesc	`MediaDescriptor`	Descriptor of media resource	GET (OPS only)	The mediaDescriptor	Yes

Attribute	Type	Description	Access Level	Valid Value	Mandatory
		which Web Services will use to fetch the media file. Only returned when requested by OPS admin.		sent in the original POST	
callUUID	String	The CallUUID associated with the mediafile	GET, POST	Any non-empty string with no leading/ending spaces	Yes
startTime	DateTime	The time the media file began recording. It is reported by when uploading.	GET, POST	not null, valid date	Yes
stopTime	DateTime	The recording's stop time.	GET, POST	not null, valid date	Yes
mediaId	String	Unique identifier for the media file that is used by clients to refer to the same media file.	GET, POST	String	No
type	String	Type of the media file	GET, POST	String	No
duration	String	Duration of the media file	GET, POST	String	No
size	String	Size of the media file in bytes	GET, POST	String	No
tenant	String	Tenant name of the recording	GET, POST	String	No
ivrprofile	String	IVR Profile name that serviced the recording	GET, POST	String	No

parameters	JSON Map	Map of additional metadata information.	GET, POST	JSON Map
				Example:
				`{`
				`"record": "source",`
					No

Attribute	Type	Description	Access Level	Valid Value	Mandatory

				`"recorddn", "2222"`
				`}`	

masks	Object Array	List of time stampts of pause/resume periods if the recording is masked by client	GET, POST	JSON Object Array
				Example:
				`[`
				`{"time": "2013-02-06T10:23:10.034Z", "type", "paused"},`
				`{"time": "2013-02-06T10:32:14.034Z", "type", "resume}`
				`]`
					No

pkcs7	String	PKCS7 envelop if the media file is encrypted	GET, POST	String	No
certAlias	String Array	A list of alias of encryption certificates if the media file is encrypted.	GET, POST	String Array	No
partitions	String Array	A list of partition names for the media file	GET, POST	String Array.	No
accessgroups	String Array	Access groups identified for the agent in the recording.	GET, POST	String Array.	No

The following attributes are present in **mediaDescriptor**:

Attribute	Type	Description	Valid Values	Mandatory
storage	String	The type of storage used for media resource	• `awsS3`-amazonS3 storage • `webDAV`-WebDAV storage for premise deployment	yes

Attribute	Type	Description	Valid Values	Mandatory
storage_version	String	the version of storage used (internal)	any non-empty string	no
path	String	The path which is used to identify the resource inside the storage	• For awsS3, the path to resource inside the bucket • For webDAV, the path of the WebDAV resource	yes
auth	Map - nested structure	The authentication info which can be used for storage acccess	Storage-specific	Storage-specific. • awsS3- ignored . • webDAV- not required.
data	Map - nested structure	Storage-specific- data with any additional info about the resource	Storage-specific- For awsS3 it must contain **bucket** property with the name of **bucket** where the resource is.	Storage-specific • For awsS3, yes. • For webDAV, no

Warning

Resource identity rules are storage-specific.

- awsS3- storage bucket and path information is used to determine if two media descriptors define the same resource. If two media descriptors have the same buckets and the same path, the resources are considered to be the same.
- WebDAV- path is used. If two media descriptors have same path, the resources are considered to be the same.

The following attributes are available for **EventDescriptor**:

Warning

All properties are mandatory.

Attribute	Type	Description	Access Level	Valid values
occurredAt	DateTime	The time at which this event occurred.	GET, POST	not null, valid date
contact	Contact	A contact resource (see further) describing some entity that joined/left the call.	GET, POST	not null
event	String: Joined or Left	Describes whether the contact joined or left the call.	GET, POST	Joined, Left

The Contact structure has the following attributes:

Attribute	Type	Description	Acces Level	Mandatory
phoneNumber	String	Phone number of the contact.	GET, POST	Yes. Can never be null or empty but can be `UNKNOWN`. Must be a non-empty string without leading or ending spaces.
type	External OR User	Describes the type of contact. Can be an external caller or a contact center agent.	GET, POST	Yes.
userName	String	Username of the contact	GET, POST	Yes. If type of contact is `user`, value must be a non-empty string without leading or ending spaces.
firstName	String	First name of the contact	GET, POST	Yes. If the type of contact is `user`, value must be a non-empty string without leading or ending spaces.
lastName	String	Last name of the contact	GET, POST	Yes. If the type of contact is `user`, value must be a non-empty string

Attribute	Type	Description	Acces Level	Mandatory
				without leading or ending spaces.

Query Parameters

The contact center administrator is able to provide query parameters to the **GET /recordings** operation to specify a set of criteria for the returned recordings. The available parameters are described below. This operation will return the response containing the array of matching call recordings . This array can be empty if no matching recording found.

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified `ani` attribute. The exact match of stored number and request parameter is used. The stored number and request parameter are alphanumeric only, but the request string can contain ***** wildcards which can substitute any number of symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified `dnis` attribute. The exact match of stored number and request parameter is used. The stored number and request parameter are alphanumeric only, but the request string can contain ***** wildcards which can substitute any number of symbols in the request. Search is case-sensitive.
startTime	Retrieves all recordings which started after the specified time
endTime	Retrieves all recordings which ended before the specified time
userName	Retrieves all recordings in `eventHistory->contacts` which the specified userName/firstName/Lastname is present. You can use ***** wildcards to specify only part of the username/firstname/lastname. If more than one word is used (divided by spaces), the records containing any of provided terms as username, firstname or lastname will be included. You can retrieve records containing all terms by using the `AND` keyword. For example, `?userName= Alice AND Amber` will search for recordings with `events->contact-> username/firstName/lastName` containing Alice and Amber. Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap.

Parameter Name	Description
	These matcher are supported: • Exactly Mach the entire value like (for example: `tom`) • wild card value (for example: `tom*` will find records with `tomas`) • Combination, if query strings are separated by space (for example: `tom jerry` -> will look for recordings which contain `tom` or `jerry`)

> **Important**
>
> Multiple query parameters can be applied as part of one request. When this is the case, they form an AND condition for the operation. For example, `GET /recordings?startTime=A&endTime=B` will retrieve all recordings between A and B.

This operation will return the response containing the array of matching call recordings . This array can be empty if no matching recording found.

Result Pagination

Query results can be returned in multiple pages. Addtional parameters are provided to customize pagination.

Parameters	Description	Default
offset	Index of the first record to be returned.	0
limit	Size of each page. Cannot be larger than 100.	100

Examples

The following example creates a call recording using Amazon S3 Storage:

```
POST recordings/
       {
   "id":"00PV5Q27MG8AB8VNE49362LAES000014",
   "callerPhoneNumber":"+16504668888",
   "dialedPhoneNumber":"+14155551234",
   "region":"test",
   "mediaFiles":[
      {
         "startTime":"2013-05-17T11:45:32.000-0800",
         "stopTime":"2013-05-17T14:15:36.000-0800",
         "callUUID":"callUUID",
         "mediaDescriptor":{
            "storage":"awsS3",
            "path":"FTtests/sample1.mp3",
            "data":{
               "bucket":"stage-recordings"
            }
         },
         "mediaId": "00PV5Q27MG8AB8VNE49362LAES000013_029FO32T0G8LDFTEK9I4K8DAES0KTDVE.wav",
         "type": "audio/wav",
         "duration": "8000",
         "tenant": "Environment",
         "ivrprofile": "DefaultProfile",
         "size": "8544",
         "parameters": {"record": "source", "recorddn": "2222"},
         "masks": [{"time": "2013-02-06T10:23:10.034Z", "type": "paused"}],
         "pkcs7": "SJOSSWKGUEKSHSVHSU",
         "certAlias": ["cert1", "cert2"],
         "partitions": ["part1", "part2"],
         "accessgroup": ["US", "US/CA"]
      }
   ],
   "eventHistory":[
      {
         "occurredAt":"2013-05-18T22:25:56.000+0000",
         "contact":{
            "type":"User",
            "phoneNumber":"+14103585834",
            "userName":"dima.p@genesys.com",
            "firstName":"Dima",
            "lastName":"Pokh"
         },
         "event":"Left"
      }
   ]
}
```

The following example creates a call recording using WebDAV Storage:

```
POST recordings/
       {
   "id":"00PV5Q27MG8AB8VNE49362LAES000014",
   "callerPhoneNumber":"+16504668888",
   "dialedPhoneNumber":"+14155551234",
   "region":"test",
   "mediaFiles":[
      {
         "startTime":"2013-05-17T11:45:32.000-0800",
         "stopTime":"2013-05-17T14:15:36.000-0800",
         "callUUID":"callUUID",
         "mediaDescriptor":{
            "storage":"webDAV",
            "path":"http://10.10.15.83:8080/sample1.mp3"
         },
         "mediaId": "00PV5Q27MG8AB8VNE49362LAES000013_029FO32T0G8LDFTEK9I4K8DAES0KTDVE.wav",
         "type": "audio/wav",
         "duration": "8000",
         "tenant": "Environment",
         "ivrprofile": "DefaultProfile",
         "size": "8544",
         "parameters": {"record": "source", "recorddn": "2222"},
         "masks": [{"time": "2013-02-06T10:23:10.034Z", "type": "paused"}],
         "pkcs7": "SJOSSWKGUEKSHSVHSU",
         "certAlias": ["cert1", "cert2"],
         "partitions": ["part1", "part2"],
         "accessgroup": ["US", "US/CA"]
      }
   ],
   "eventHistory":[
      {
         "occurredAt":"2013-05-18T22:25:56.000+0000",
         "contact":{
            "type":"User",
            "phoneNumber":"+14103585834",
            "userName":"dima.p@genesys.com",
            "firstName":"Dima",
            "lastName":"Pokh"
         },
         "event":"Left"
      }
   ]
}
```

The following example retrieves a call recording descriptor using a simple query as the contact center administrator. Please note the substitution of mediaURI in place of mediaDescriptor:

```
GET  /recordings/recordings?startTime=0
```

```
    {
       "statusCode": 0,
       "recordings":
       [
           {
               "id": "00PV5Q27MG8AB8VNE49362LAES000013",
               "callerPhoneNumber": "+16504668888",
               "dialedPhoneNumber": "+14155551234",
               "startTime": "2013-05-17T19:45:32.000+0000",
               "stopTime": "2013-05-17T22:15:36.000+0000",
               "mediaFiles":
               [
                   {
                       "startTime": "2013-05-17T19:45:32.000+0000",
                       "stopTime": "2013-05-17T22:15:36.000+0000",
                       "callUUID": "callUUID",
                       "mediaUri": "http://172.21.80.29:8080/cloud-web/api/v2/recordings/
00PV5Q27MG8AB8VNE49362LAES000013/play/0ce11ff0-a6cf-43fa-aaa1-baad43ac98b4.mp3",
                       "mediaId":
"00PV5Q27MG8AB8VNE49362LAES000013_029FO32T0G8LDFTEK9I4K8DAES0KTDVE.wav",
                       "type": "audio/wav",
                       "duration": "8000",
                       "tenant": "Environment",
                       "ivrprofile": "DefaultProfile",
                       "size": "8544",
                       "parameters": {"record": "source", "recorddn": "2222"},
                       "masks": [{"time": "2013-02-06T10:23:10.034Z", "type": "paused"}],
                       "pkcs7": "SJOSSWKGUEKSHSVHSU",
                       "certAlias": ["cert1", "cert2"],
                      "partitions": ["part1", "part2"],
                      "accessgroup": ["US", "US/CA"]
                   }
               ],
               "eventHistory":
               [
                   {
                       "occurredAt": "2013-05-18T22:25:56.000+0000",
                       "contact":
                       {
                           "type": "User",
                           "phoneNumber": "+14103585834",
                           "userName": "admin@genesys",
                           "firstName": "Slava",
                           "lastName": "Sayko"
                       },
                       "event": "Left"
                   }
               ],
               "region": "development"
           }
       ]
    }
```

Example of result in multiple pages:

```
GET  /recordings/recordings?startTime=0&offset=40&limit=20
    {
       "statusCode": 0,
       "recordings":
       [
           {
               "id": "00PV5Q27MG8AB8VNE49362LAES000013",
               ...
           },
           ...
           {
               "id": "00PV5Q27MG8AB8VNE49362LAES000013",
               ...
           }
       ]
       "prevUri": "http://10.10.15.89/api/v2/recordings/
recordings?startTime=0&offset=20&limit=20",
       "nextUri": "http://10.10.15.89/api/v2/recordings/
recordings?startTime=0&offset=60&limit=20"
    }
```