



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Genesys Mobile Services API Reference

Service API

5/5/2025

---

## Contents

- **1 Service API**
  - 1.1 Overview
  - 1.2 Create Service
  - 1.3 Update Conversation Expiration
  - 1.4 Service Specific Request
  - 1.5 Query All Keys
  - 1.6 Query One Key
  - 1.7 Create or Update Service Storage
  - 1.8 Delete One Key Storage
  - 1.9 Delete Service

# Service API

## Overview

This API is used by customer-facing applications to manage different types of contact center related services. For example the app-to-connect-basic service provides the necessary contact center access information so the end user and associated application can initiate an interaction with the contact center.

### Important

Parameters that begin with an underscore (\_) are passed to ORS. Anything else is considered as user data, and is saved in storage. The stored data can be retrieved using the `_data_id` parameter passed in scxml.

## Create Service

This API query creates and initiates a service that you previously configured in the Admin UI in the Configured Services interface.

### Important

Before you start using a service by making REST-queries to its execution name, you must create it with this API query.

## Multipart or Urlencoded

### Multipart or Urlencoded Operation

Method	POST		
URL	/genesys/1/service/{service}		
Parameter	Type	Mandatory	Description
URI Parameters			
{service}	string	yes	The name of the service

Method	POST		
			that is to be created and initiated.
<b>Body:</b> The body can be either a MultiPart form or x-www-form-urlencoded form consisting of different items representing the key/value pairs associated with the given service type. In the case of MultiPart, the values can be strings or files, but with urlencoded, the values can be only strings.			

## Response

HTTP code	200
HTTP message	OK
Body	<p>A JSON object with the following: <code>{"id": "\${service_id}, {service_specific_data}"}</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>\${service_id}</code> is the identifier assigned to the created service instance.</li> <li>• <code>\${service_specific_data}</code> is service specific data that can be returned when the service is created.</li> </ul>

If a matching services does not find a match, it will return the following status code.

HTTP code	404
HTTP message	Not Found

## Example

The following example starts a request-interaction service with the end user's phone number and application data: current user's location, preferred language, and end user's picture.

## Operation

```
Request URL:
http://localhost:8080/genesys/1/service/request-interaction
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:13028
Content-Type:multipart/form-data;
boundary=---WebKitFormBoundaryy16qocbN6tmPORZL
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/GMS-web/resources/servicetest.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/535.7 (KHTML, like Gecko)
Chrome/16.0.912.77 Safari/535.7
Request Payload
```

```
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="_phone_number"
6504669999
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="_provide_code"
true
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="language"
french
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="current_location_latitude"
48.8583
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="current_location_longitude"
2.2944
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="FileKey"; filename="MyPic.png"
Content-Type: image/png
-----WebKitFormBoundaryy16qocbN6tmPORZL--
```

### Result

The above service will be started with an id of 39a98e24-b03b-4191-b756-1efe8f3b16b8.

```
HTTP 200 OK
{
  "_id": "39a98e24-b03b-4191-b756-1efe8f3b16b8",
  "_access_number": "8003449999",
  "_access_code": "7684"
}
```

## JSON

### JSON Body - Operation

Since 8.5.104, the Create Service query supports JSON inputs. You can now specify key/value pairs as a JSON object. Because ORS allows simple key/value parameters, GMS cannot submit complex JSON-structured values (such as maps and arrays) as ORS request parameters and convert complex values to strings.

For example:

```
{
  "address": "{street=mystreet, city=mycity}", // for map
  "phones": "[9002@SIP_Switch, 5555666]" // for array
}
```

### Important

The JSON API does not support form-encoded binaries.

<b>Method</b>	POST		
<b>URL</b>	/genesys/1/service/{service}		
<b>Parameter</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
<b>URI Parameters</b>			
<b>{service}</b>	string	yes	The name of the service to create and initiate.
<b>Header</b>			
Content-type: application/json;charset=UTF-8			
<b>Body:</b> A simple JSON structure representing key/value pairs.			
<ul style="list-style-type: none"><li>• Keys that begin with "_" are ignored for storage.</li><li>• No binary values.</li><li>• Complex JSON-structured values (map and arrays) are converted to strings.</li></ul>			
For example:			
<pre>{   "lastname" : "mylastname",   "address": {     "street": "mystreet",     "city": "mycity"   },   "phones": [     "9002@SIP_Switch",     "5555666",     "7775666"   ],   "_key": "ignored" }</pre>			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK

If the JSON format is incorrect, the query returns a 500 error message.

<b>HTTP code</b>	500
<b>HTTP message</b>	Server Error
Body	{"message": "Could not read JSON: Unexpected character ('\"' (code 34)): [...]"}.

## Example

```
POST http://localhost:8080/genesys/1/service/request-interaction
Content-Type: application/json
gms_user: default
```

```
{
  "_phone_number": "6504669999",
  "_provide_code": "true",
  "language": "french",
  "current_location_latitude": "48.8583",
  "current_location_longitude": "2.2944"
}
```

Status Code: 200 OK

## Update Conversation Expiration

## Service Specific Request

This API allows the application to perform a specific request against a given service.

### Important

This is only to be used for services that support such a request, otherwise it will be rejected.

## Operation

Method	POST		
URL	/genesys/1/service/{service_id}/{request}		
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The id of the service that is to be requested.
{request}	string	yes	This is the name of the request that is to be performed.
<b>Body:</b> The body can be either a MultiPart form or x-www-form-urlencoded form consisting of different items representing the key/value pairs associated with the given service request. In the case of MultiPart, the values can be strings or files but with urlencoded, the values can be only strings.			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK
<b>Body</b>	This will contain the appropriate output data (JSON data) that is defined by the given service request definition.

## Example

See the [Chat Interaction APIs](#) for an example.

## Query All Keys

This API queries all of the keys in the storage area that has already been created for the service.

Note: Introduced in 8.5.000.12.

## Operation

<b>Method</b>	GET		
<b>URL</b>	/genesys/1/service/{id}/storage		
<b>Parameter</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
<b>URI Parameters</b>			
{id}	string	yes	The id of the service.
<b>Body:</b> None			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK

## Example

The following example queries all of the keys associated with service efef8eb61-1f24-593d-90da-0034aca34b55.

### Operation

Request URL (method GET)  
`http://localhost:8080/genesys/1/service/efef8eb61-1f24-593d-90da-0034aca34b55/storage`

### Result

---



```
{"Key2": "Value7", "Key1": "Value6", "Key3": "Value8"}
```

## Query One Key

This API queries one of the keys in a storage area that has already been created for the service.

Note: Introduced in 8.5.000.12.

### Operation

Method	GET		
URL	/genesys/1/service/{id}/storage/{key}		
Parameter	Type	Mandatory	Description
URI Parameters			
{id}	string	yes	The id of the service.
{key}	string	yes	The key of the specifically requested value.
<b>Body:</b> None			

### Response

If the key exists, returns 200 OK and the following JSON format value: **{"key4": "value4"}** (not the key value itself).

HTTP code	200
HTTP message	OK

Returns 404: Not Found if the key is not found in the user data.

HTTP code	404
HTTP message	Not Found

### Example

The following example queries the value of Key1 from the data associated with id efef8eb61-1f24-593d-90da-0034aca34b55.

#### Operation

Request URL (method GET)  
http://localhost:8080/genesys/1/service/efef8eb61-1f24-593d-90da-0034aca34b55/storage/Key1

Request Method: GET

#### Result

Value1

## Create or Update Service Storage

This API allows the creation of a new storage area or an update of the existing storage for a specific service. The TTL of the user data storage is the same TTL as the service.

### Multipart or Urlencoded

#### Multipart or Urlencoded

##### Operation

Method	POST		
URL	/genesys/1/service/{id}/storage		
Parameter	Type	Mandatory	Description
URI Parameters			
{id}	string	yes	The id of the service.
<b>Body:</b> A MultiPart form or a URL encoded form consisting of different items representing the key/value pairs to store.			

##### Response

HTTP code	200
HTTP message	OK

#### Example

The following example stores:

Key1, Key2, Key3, and FileKey

#### Operation

Request URL:  
http://localhost:8080/genesys/1/service  
/efef8eb61-1f24-593d-90da-0034aca34b55/storage

Request Method:POST  
Status Code:200 OK  
Request Headersview source  
Accept:/\*/\*  
Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3  
Accept-Encoding:gzip,deflate,sdch

```
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:13028
Content-Type:multipart/form-data;
boundary=----WebKitFormBoundaryy16qocbN6tmPORZL
Host:localhost:8080
Origin:http://localhost:8080
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/535.7 (KHTML, like Gecko)
Chrome/16.0.912.77 Safari/535.7
Request Payload
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="Key1"
Value1
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="Key2"
Value2
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="Key3"
Value3
-----WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="FileKey"; filename="MyPic.png"
Content-Type: image/png
-----WebKitFormBoundaryy16qocbN6tmPORZL--
```

### Result

The above data is now stored.

HTTP 200 OK

## JSON

### JSON Body

Starting in 8.5.104, you can update storage data with JSON arrays.

Because ORS allows simple key/value parameters, GMS cannot submit complex JSON-structured values (such as maps and arrays) as ORS request parameters and convert complex values to strings.

For example:

```
{
  "address":{"street=mystreet, city=mycity}", // for map
  "phones":["9002@SIP_Switch, 5555666]" // for array
}
```

### Important

This query cannot handle binary values. To perform a request with an image, use the multipart/form API query.

## Operation

Method	POST		
URL	/genesys/1/service/{id}/storage		
Parameter	Type	Mandatory	Description
URI Parameters			
{id}	string	yes	The id of the service.
Header Parameters	'Content-type: application/json;charset=UTF-8		
<b>Body:</b> The body is a JSON structure representing the key/value pairs.			
<ul style="list-style-type: none"><li>• Keys that begin with "_" are ignored for storage.</li><li>• No binary values.</li><li>• Complex JSON-structured values (map and arrays) are converted to strings.</li></ul>			
For example:			
<pre>{   "lastname" : "mylastname",   "address": {     "street": "mystreet",     "city": "mycity"   },   "phones": [     "9002@SIP_Switch",     "5555666",     "7775666"   ],   "_key": "ignored" }</pre>			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK
<b>HTTP code</b>	500
<b>HTTP message</b>	Server Error
Body	{"message": "Could not read JSON: Unexpected character ('\"' (code 34)): [...]"} 

## Query Binary (One Binary Key)

This API queries one of the binary keys in a storage area that has already been created for the service.

Note: Introduced in 8.5.002.02

## Operation

<b>Method</b>	GET		
<b>URL</b>	/genesys/1/service/{id}/storage/binary/{key}		
<b>Parameter</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
<b>URI Parameters</b>			
<b>{id}</b>	string	yes	The id of the service.
<b>{key}</b>	string	yes	The key of the specifically requested value.
<b>Body:</b> None.			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK
<b>HTTP code</b>	404
<b>HTTP message</b>	Not Found, if the binary key does not exists in user storage.

## Example

The following example queries the value of myBinaryKey from the data associated with id efef8eb61-1f24-593d-90da-0034aca34b55.

## Operation

Request URL (GET Method)  
http://localhost:8080/genesys/1/service  
/efef8eb61-1f24-593d-90da-0034aca34b55/storage/binary/myBinaryKey

## Result

Binary stream content and its content type.

## Delete One Key Storage

This API deletes one of the keys (either binary or non-binary) in a storage area that has already been created for the service.

Note: Introduced in 8.5.002.02

### Operation

Method	DELETE		
URL	/genesys/1/service/{id}/storage/{key}		
Parameter	Type	Mandatory	Description
URI Parameters			
{id}	string	yes	The id of the service.
{key}	string	yes	The key to be deleted.
<b>Body:</b> None.			

### Response

HTTP code	200
HTTP message	OK

### Example

The following example deletes the value of Key1 from the data associated with id efef8eb61-1f24-593d-90da-0034aca34b55.

### Operation

Request URL:  
`http://localhost:8080/genesys/1/service/efef8eb61-1f24-593d-90da-0034aca34b55/storage/Key1`

Request Method: DELETE

### Result

OK

## Delete All Keys in Storage

This API deletes all keys (binary or non-binary) in a storage area that has already been created for the service.

Note: Introduced in 8.5.002.02

## Operation

<b>Method</b>	DELETE		
<b>URL</b>	/genesys/1/service/{id}/storage		
<b>Parameter</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
<b>URI Parameters</b>			
<b>{id}</b>	string	yes	The id of the service.
<b>Body:</b> None.			

## Response

<b>HTTP code</b>	200
<b>HTTP message</b>	OK

## Example

The following example deletes all of the keys from the data associated with id efef8eb61-1f24-593d-90da-0034aca34b55.

## Operation

Request URL:  
http://localhost:8080/genesys/1/service  
/efef8eb61-1f24-593d-90da-0034aca34b55/storage

Request Method: DELETE

## Result

OK

## Delete Service

This API deletes the service that was created and the storage area associated with it.

## Operation

<b>Method</b>	DELETE		
<b>URL</b>	/genesys/1/service/{id}		
<b>Parameter</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
<b>URI Parameters</b>			



Method	DELETE		
{id}	string	yes	The id of the service.
<b>Body:</b> None.			

## Response

HTTP code	200
HTTP message	OK

## Example

The following example deletes the service id efef8eb61-1f24-593d-90da-0034aca34b55 and all the keys from the data associated with it.

## Operation

Request URL:  
`http://localhost:8080/genesys/1/service/efef8eb61-1f24-593d-90da-0034aca34b55`

Request Method: DELETE

## Result

OK