



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.

API Reference

Update Expiration Time

12/15/2025

Contents

- 1 Update Expiration Time
 - 1.1 Description
 - 1.2 Operation
 - 1.3 Response
 - 1.4 Example

Update Expiration Time

PUT /genesys/1/cs/services/{service_id}

Description

You can make a conversation expire by adding a Time-to-live (TTL) or expiration time to a Service by updating your resource with this query. When you update the TTL or service expiration time, you overwrite previous updates and any expiration time **set in your configuration**. You can also use `ttl` and `expiration_time` attributes in **Service Start Events** and **Service End Events**.

Important

Updating the conversation expiration time has a performance impact and uses as much CPU than three event queries.

Operation

Important

You need a WRITE role to use this request.

Method	PUT		
URL	/genesys/1/cs/services/{service_id}		
Parameter	Type	Mandatory	Description
URI Parameters			
{service_id}	string	yes	The id of the service to request.
expiration_time	string	no	Time at which the service should expire, using the ISO 8601 representation: [YYYY] - [MM] - [DD]T[HH]:[mm]:[ss].[SSS]Z
ttl	integer	no	Time-to-live of the full conversation in seconds from this event time.

Update Expiration Time

Method	PUT		
			This parameter takes precedence over <code>expiration_time</code> parameter which is discarded if you pass both parameters.

Response

The following table shows the correct response for a successful request. In case of error, you will see standard **HTTP response error codes**.

HTTP code	200
HTTP message	OK
Body	Empty

Example

```
PUT http://localhost:8010/genesys/1/cs/services/115-cc038ec2-c284-41e5-8560-b006743e1935
{
  "ttl": 86400
}
```

Response: 200 OK