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

Context Services 8.1.x

1/6/2022

Table of Contents

Context Services API Reference **5**
List of Operations **6**
HTTP Response Codes and Errors **10**
Unauthorized Strings **14**
Interaction Operations **17**
Query Interactions 18
Interaction Resources 21
Callback Interaction 22
Chat Interaction 29
E-mail Interactions 33
Phone Call Interaction 41
Server Operations **45**
Query Server Status 46
Set Server Mode 48
Service Operations **50**
Associate Service 52
Complete Service 56
Complete State 60
Complete Task 65
Delete Record From Service Extension 69
Delete Record From State Extension 71
Delete Record From Task Extension 73
Perform State Transition 75
Query Service by ID 81
Query Services 87
Query State by ID 93
Query States 98
Query Task by ID 101
Query Tasks 105
Start Service 109
Start State 114
Start Task 119
Update Service Extension 123
Update Record In Service Extension 125
Update Record In State Extension 127

- Update Record In Task Extension 129
- Update State Extension 131
- Update Task Extension 133
- Service Resources 135
  - End Event List 136
  - Extension 144
  - State Transition Event 147
  - Service 151
  - State 154
  - Start Event List 157
  - Task 165

Profile Operations **168**

- Bulk Profile Import 169
- Create Customer Profile 175
- Delete Customer Profile 178
- Delete Record From Profile Extension 180
- Identify Customer 182
- Insert Extension Records 186
- Merge Customer Profile 188
- Query Customer Profile 191
- Update Customer Profile 194
- Update Record In Profile Extension 197
- Profile Resources 199
  - Customer Profile 200
  - Profile Attribute Schema 202
  - Profile Extension 206
  - Profile Extension Schema 208

List of Schema Operations **210**

- Create Identification Key 211
- Create Profile Extension Schema 214
- Create Task Extension Schema 217
- Create State Extension Schema 220
- Create Service Extension Schema 223
- Change Metadata Cache 226
- Delete Metadata Identification Keys 228
- Delete Metadata Profile Extensions 230
- Delete Metadata Services Extensions 232

Delete Metadata States Extensions 234
Delete Metadata Tasks Extensions 236
Get Identification Keys 238
Get Metadata 240
Query Business Attribute Schema 243
Query Metadata Roles 246
Query Profile Extension Schema 249
Query Profile Schema 252
Query Service Extension Schema 255
Query State Extension Schema 258
Query Task Extension Schema 261
Schema Resources 264
Attribute Schema 265
Extension Schema 268
Identification Key 272

Context Services API Reference

Description

The API Reference covers all the representations and methods available through the Context Services. Each subcategory of this page presents a set of operations, and the related resources.

List of Operations

	Purpose: Entry point of the API Reference.

Contents of the API Reference

Each operation category includes a subcategory which contain the related resources.

Operations

- Interaction Operations
  - GET /customers/${customer id}/interactions
  - GET /services/${service id}/interactions
  - GET /interactions/${interaction id}
- Profile Operations
  - POST /profiles
  - GET /profiles
  - POST /profiles/${customer id}/extensions
  - GET /profiles/${customer id}
  - PUT /profiles/${customer id}
  - Since 8.0.200.00:
    - PUT /profiles/${customer id}/extensions/${ext name}/by/unique
    - PUT /profiles/${customer id}/extensions/${ext name}/delete/by/unique
  - Since 8.1.300.xx:
    - PUT profiles/${customer_id}/merge/${src_id}/
- Service Operations
  - Since: 8.0.100.00
    - POST /services/start
    - POST /customers/${customer id}/services/${service id}
    - POST /services/${service id}/end
    - GET /services/${service id}
    - GET /services/anonymous/${contact key}/active

        - GET /services/anonymous/${contact key}/completed
        - GET /services/anonymous/${contact key}
        - GET /customers/${customer id}/services/active
        - GET /customers/${customer id}/services/completed
        - GET /customers/${customer id}/services
    - Since: 8.0.200.00
        - PUT /services/${service id}/extensions/${ext name}
        - PUT /services/${service id}/extensions/${ext name}/by/unique
        - PUT /services/${service id}/extensions/${ext name}/delete/by/unique
- State Operations
    - Since: 8.0.100.00
        - POST /services/${service id}/states/${state id}/end
        - POST /services/${service id}/states/transition
        - GET /services/${service id}/states
        - GET /services/${service id}/states/${state id}
        - POST /services/${service id}/states/start
    - Since: 8.0.200.00
        - PUT /services/${service id}/states/${state id}/extensions/${ext name}
        - PUT /services/${service id}/state/${state id}extensions/${ext name}/by/unique
        - PUT /services/${service id}/states/${state id}/extensions/${ext name}/delete/by/unique
- Task Operations
    - Since: 8.0.100.00
        - POST /services/${service id}/tasks/${task id}/end
        - GET /services/${service id}/tasks
        - GET /services/${service id}/tasks/${task id}
        - POST /services/${service id}/tasks/start
    - Since: 8.0.200.00
        - PUT /services/${service id}/tasks/${task id}/extensions/${extension name}
        - PUT /services/${service id}/task/${task id}extensions/${ext name}/by/unique
        - PUT /services/${service id}/task/${task id}/extensions/${ext name}/delete/by/unique
- Server Operations
    - GET /server/status
    - POST /server/mode
- Schema Operations

- GET /metadata/profiles/
- POST /metadata/profiles/extensions
- GET /metadata/profiles/extensions
- GET /metadata/identification-keys
- POST /metadata/identification-keys
- Since 8.0.2:
  - POST /metadata/services/extensions
  - GET /metadata/services/extensions
  - POST /metadata/states/extensions
  - GET /metadata/states/extensions
  - POST /metadata/tasks/extensions
  - GET /metadata/tasks/extensions
- Available since 8.1.3
  - DELETE /metadata/profiles/extensions/${extension-name}
  - DELETE /metadata/services/extensions/${extension-name}}
  - DELETE /metadata/states/extensions/${extension-name}
  - DELETE /metadata/tasks/extensions/${extension-name}
  - DELETE /metadata/identification-keys/${id_key-name}

Resources

- Interaction Resources
  - Chat Interaction
  - Callback Interaction
  - E-mail Interactions
  - Phone Call Interaction
- Profile Resources
  - Customer Profile
  - Extension
  - Identification Key
- Service Resources
  - End Event List
  - Extension
  - Start Event List
  - State Transition Event

    - Service
    - State
    - Task
- Schema Resources
    - Attribute Schema
    - Extension Schema

HTTP Response Codes and Errors

Purpose: Describes the HTTP responses that your application can receive from UCS.

Introduction

The Universal Contact Server returns HTTP status codes and messages for every operation, in the requested format. Status codes match standard HTTP codes, but messages can differ and provide additional details included in the header of the response. Additional result and error codes may be returned due to external web servers and layers involved in your operations.

Read Log Lines

Log lines are compliant with the following syntax:

```
14:33:39.485[1] Trc[2] 29998 [let-3851106] 2010-04-02[3]    14:33:39[3]    127.0.0.1[4]
8080[5]    GET[6]    /cv/server/status[7]    -    200[8]    -    0    2[9]
http://localhost:8080[10]    Apache-HttpClient/4.0-beta2 (java 1.5)[11]    -
```

- [1] = timestamp
- [2] = loglevel
- [3] = date and time when the request was received
- [4] = ip address
- [5] = server port
- [6] = http method
- [7] = requested url
- [8] = http response code
- [9] = execution time in ms
- [10] = called host/port
- [11] = user agent making the call

Successful Result

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 by Context Services for a successful response.

Successful results

Code	Title	Description
200	OK	Success!
201	Created	The request has been fulfilled and resulted in a new resource being created.
204	No Content	The operation was successful and returned with no content.

Errors

For responses with HTTP status code 4xx or 5xx, the response body contains an application-specific description of the error instead of a representation of the requested resource. Context Services 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 operations can encounter. This list is not restrictive; additional error codes could be returned due to external web servers and layers involved:

Errors

Code	Title	Description
303	See Other	General error which can occur if your application refers to a URI which moved, due to API changes.
400	Bad Request	General error which can be one of the following reasons: • Missing required parameter. • Parameter value of unexpected type. • Invalid object syntax<ref name="javascript">If you used JavaScript objects in your operation's input.</ref>. • Missing required attribute<ref

Code	Title	Description
		name="javascript"/>.
401	Not Authorized	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<ref name="r400">Not available for now.</ref>	Forbidden	The operation is forbidden and the reason is specified in the error message. This error is returned in the following cases: • Attempt to query customer profile or service information when the server is in Maintenance Mode. • Attempt to alter the schema while the server is in Production Mode. • Attempt to add customer profiles prior to the definition of a profile schema. • Attempt to change the server mode times out, due to maintenance operations or schema modifications to complete.
404	Not Found	The specified URI is invalid, or the requested resource (such as a customer, service, state, task, extension, or identification key) 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.
415	Unsupported Media Type	In the header of your request, Content-Type is not set to a valid value. Most operations of Context Services support only "application/json". Check the operation description in the API.
500	Internal Server Error	An unexpected error occurred in UCS (for instance, a runtime exception). The error message suggests to forward logs to Genesys Customer Support.

Code	Title	Description
502<ref>Context Services is not responsible for this message. This error may occur due to external proxies or cache involved in your operation.</ref>	Bad Gateway	Returned when one or more of the backend systems required to fulfill the response (the CV database or UCS, for example) are either unavailable or returned an error.
503	Service Unavailable	UCS is unable to process the given request. Example situations include: • Requests on a whole, or for a specific client/tenant/role, are being throttled. • The server is switching from production to maintenance mode, or vice versa, and does not accept requests for the moment.

Unauthorized Strings

	Specifies the list of strings that your application cannot use to create resources and schemas.

Returned Error

Context Services operations enable your application to create resources, schemas, and extensions. However, the keywords identified on this page should not be used to create or fill in attribute values. If your application does use one of the reserved keywords in PUT or POST operations, then a HTTP 400 Bad Request is returned with a "bad parameter" message in the following format:

```
bad parameter '${parameter-name}' reason: '${unauthorized-keyword}' keyword is not authorized.
```

For instance, if a PUT operation used the 'Select' keyword as a field value for the parameter 'name', your application receives the following message:

```
bad parameter 'name' reason : 'Select' keyword is not authorized.
```

Keywords Reserved to Context Services

- extension
- none
- profile
- service
- state
- task

Keywords Reserved to Database

- alter
- by
- comment
- constraint
- create

- cursor
- database
- delete
- from
- having
- identity
- index
- inner
- insert
- is
- join
- left
- null
- order
- outer
- prepare
- primary
- procedure
- return
- right
- select
- set
- size
- table
- truncate
- union
- update
- when
- where

Type Keywords

- bigint
- binary

- bit
- blob
- boolean
- char
- clob
- currency
- date
- datetime
- decimal
- double
- float
- int
- integer
- long
- longvarbinary
- money
- nchar
- number
- numeric
- real
- smalldatetime
- smallint
- smallmoney
- string
- time
- timestamp
- tinyint
- varbinary
- varchar
- varchar2

Interaction Operations

	Purpose: Groups operations which handle multimedia interactions.

Interaction Resources

- Callback Interaction
- Chat Interaction
- E-mail Interaction
- Phone Call Interaction

Interaction Operations

- GET /customers/${customer id}/interactions
- GET /services/${service id}/interactions
- GET /interactions/${interaction id}

Query Interactions

	• GET /customers/${customer_id}/interactions • GET /services/${service_id}/interactions • GET /interactions/${interaction_id}
	Available since 8.0.100.000

Description

Retrieves multimedia interactions and their data.

Operation

Queries Multimedia Interactions

ID	CV.WS.IXN.2		
Method	GET		
URL	1. By Customer: /customers/**${customer_id}**/interactions 2. By Service: /services/**${service_id}**/interactions 3. By Interaction ID: /interactions/**${interaction_id}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes if query is by customer.	Customer ID, required for URI 1.
${service_id}	integer	yes if query is by service.	Service ID, required for URI 2.
${interaction_id}	integer	yes if query is by interaction.	Service ID, required for URI 3.

The output sample shows a "structured_text" parameter which provides a URL to the structured text content (which is not returned as part of the query): the user application performs a separate GET to the given URL.

Similarly, the following URLs are available for GETting interaction content:

- /interactions/${interaction_id}/plain_text

- /interactions/${interaction_id}/structured_text
- /interactions/${interaction_id}/binary_content

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	An array of one or more Interaction Resources.

Example

Operation

```
GET http://ucshost:8080/interactions/00001a57JGQ00BW8
```

Result


```
{
  "external_id" : "00AF4A7951D60009",
  "established_date" : "2009-08-05T09:33:15.000Z",
  "text" : "http://ucshost:8080/interactions/00001a57JGQ00BW8/plain_text",
   "media_type" : "chat",
   "udata" : {
     "RTargetTypeSelected" : "2",
     "RRequestedSkillCombination" : "",
     "ChatServerHost" : "SUITE80",
     "CustomerSegment" : "default",
     "RTargetObjectSelected" : "Agent Group Everybody",
     "RTargetRuleSelected" : "",
     "RTargetPlaceSelected" : "Place_1002",
     "RTargetAgentGroup" : "Agent Group Everybody",
     "RTargetObjSelDBID" : "123",
     "ServiceType" : "default",
     "_smsSrcNumber" : "+33298143456",
     "RTargetAgentSelected" : "1002",
     "RTenant" : "defaultTenant",
     "RVQID" : "",
     "CBR-contract_DBIDs" : "",
     "CBR-Interaction_cost" : "",
     "ChatServerPort" : "23874",
     "_smsDestNumber" : "+33298143322",
     "RTargetRequested" : "Agent Group Everybody",
     "CBR-IT-path_DBIDs" : "",
     "RStrategyDBID" : "480",
     "CBR-actual_volume" : "",
     "RStrategyName" : "sms-session-inbound"
```

```
 },
 "binary_content" : "http://ucshost:8080/interactions/00001a57JGQ00BW8/binary_content",
 "structured_text" : "http://ucshost:8080/interactions/00001a57JGQ00BW8/structured_text",
 "status" : 3,
 "subject" : "SMS chat",
 "timeshift" : 1,
 "type" : "Inbound",
 "interaction_id" : "00001a57JGQ00BW8",
 "allow_children" : true,
 "end_date" : "2009-08-05T09:34:04.140Z",
 "owner_id" : 124,
 "sub_type" : "InboundNew",
 "released_date" : "2009-08-05T09:33:59.000Z",
 "entity_type" : 2,
 "creator_app_id" : 175,
 "thread_id" : "00001a57JGQ00BW9",
 "start_date" : "2009-08-05T09:33:10.000Z",
 "structured_text_mime" : "text/xml",
 "customer_id" : "00001a57JGQ00BVU"
}
```

Interaction Resources

Purpose

Groups the Interaction resources used in Interaction Operations.

Callback Interaction

Purpose

Provides context information relevant to a given multimedia callback interaction. A customer can ask for a callback if he or she is not available for the moment. In that case, based on the application's inputs, the system schedules the callback interaction at a time which meets the user's expectations.

Resource

Field	Type	Mandatory	Description
Callback-related information			
callback_status	integer	no	Status for the callback. • 0-UNKNOWN • 1-RECEIVED • 2-TIMER • 3-QUEUED • 4-DISTRIBUTED • 5-SUBMITTED_TO_AGT • 6-REJECTED_BY_AGT • 7-CANCELED_BY_AGT • 8-PROCESSED_BY_AGT • 9-FINAL • 10-NOT_QUEUED • 11-ERROR • 12-REPLACED • 13-AGT_TRANSFER • 14-ENDTIME_EXPIRED
desired_response_type	string	yes	Possible types: • Voice • VoIP • Email

Field	Type	Mandatory	Description
start_time	date/time	yes	Time for starting the callback.
end_time	date/time	no	Limit time for executing the callback.
customer_number	string	yes	Contact information for the customer. Depends on the media type (i.e. phone number, IP address for VoIP, or email address).
attempts	integer	yes	The number of callback attempts that have been made.
dn	string	no	The DN to which the CallBack server sent the request.
location	string	no	Name of the switch to which the DN belongs.
callback_server_id	string	no	Identifies the CallBack server that handled the request.
call_result	integer	no	One of the following results: • 0 - OK • 1 - TRANSFERRED • 2 - CONFERENCED • 3 - GENERAL_ERROR • 4 - SYSTEM_ERROR • 5 - REMOTE_RELEASE • 6 - BUSY • 7 - NO_ANSWER • 8 - SIT_DETECTED • 9 - ANSWERING_MACHINE • 10- ALL_TRUNKS_BUSY • 11- SIT_INVALID_NUM • 12- SIT_VACANT • 13- SIT_OPER_INTERCEPT

Field	Type	Mandatory	Description
			• 14- SIT_UNKNOWN • 15- SIT_NO_CIRCUIT • 16- SIT_REORDER • 17- SIT_FAX_DETECTED • 18- QUEUE_FULL • 19- CLEARED • 20- OVERFLOWED • 21- ABANDONED • 22- REDIRECTED • 23- FORWARDED • 24- CONSULT • 25- PICKEDUP • 26- DROPPED • 27- DROPPED_NO_ANSWER • 28- UNKNOWN • 29- COVERED • 30- CONVERSE_ON • 31- BRIDGED • 32- SILENCE • 33- ANSWER • 34- NU_TONE • 35- NO_DIAL_TONE • 36- NO_PROGRESS • 37- NO_RING_BACK • 38- NO_ESTABLISHED • 39- PAGER_DETECTED • 40- WRONG_PARTY • 41- DIAL_ERROR • 42- CALL_DROPPED_ERROR

Field	Type	Mandatory	Description
			• 43- SWITCH_ERROR • 44- NO_FREE_PORT_ERROR • 45- TRANSFER_ERROR • 46- STALE • 47- AGT_CALLBACK_ERROR • 48- GRP_CALLBACK_ERROR • 49- DEAFENED • 50- HELD • 51- DO_NOT_CALL • 52- CANCEL_RECORD • 53- WRONG_NUMBER
callback_type	integer	no	Possible types: • 0-UNKNOWN • 1-ASAP • 2-SCHEDULED
Common interaction fields			
customer_id	string	yes	The ID of the customer.
interaction_id	string	yes	The Genesys GUID (or Call ID) for the interaction.
status	integer	yes	Possible statuses: • 0-NEW • 1-PENDING • 2-IN_PROCESS • 3-STOPPED
entity_type	integer	yes	Possible types: • 0-EMAIL_IN • 1-EMAIL_OUT • 2-CHAT

Field	Type	Mandatory	Description
			• 3-PHONE_CALL • 5-CALLBACK • 6-COBROWSE • 7-INTERACTION (open media)
type	string	yes	The type of the interaction (such as inbound, outbound, etc.)
sub_type	string	no	The application-specific subtype which further classifies the attribute "type".
media_type	string	yes	The media type of the interaction: • voice • email • chat • callback • and so on.
parent_id	string	no	The ID of the parent interaction, if applicable.
thread_id	string	no	The ID of the thread to which this interaction should be appended, or the ID of the interaction at the root of the thread.
external_id	string	no	The reference identifier for an external system.
owner_id	integer	no	The DB ID of the Person who owns the interaction. (The Person is an object defined in the Genesys Configuration Server.)
creator_app_id	integer	yes	The DB ID of the application that created the interaction.
start_date	date/time	yes	The UTC time of the interaction's creation, using the ISO 8601 representation<ref

Field	Type	Mandatory	Description
			name="iso8601">http://en.wikipedia.o wiki/ISO_8601</ref>: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
end_date	date/time	no	The UTC time of the interaction's end, using the ISO 8601 representation<ref name="iso8601" />: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
timeshift	integer	no	The number of minutes to add to or to substract from the specified UTC<ref name="iso8601" /> time in order to get the local time of the interaction's creation.
category_id	string	no	The category assigned to this interaction by the Classification Server or the desktop's manual override.
allow_children	Boolean	no	True if child interactions can be added. Default value is `true`.
subject	string	no	An arbitrary textual subject for the interaction.
stopped_reason	string	no	The application-specified reason for which the interaction ended.
text	text	no	Plain text for the interaction.
structured_text	url	no	The URL of the interaction's structured text.
structured_text_mime	string	yes if structured_text	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the structured text, if specified (such as text/html). Mandatory if the parameter "structured_text" is specified.

Field	Type	Mandatory	Description
binary_content	url	no	The URL of the the interaction's binary content..
binary_content_mime	string	yes if binary_content	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the binary content, if specified (such as text/html).
binary_content_size	integer	yes if binary_content	The size of the binary content in bytes. Mandatory if the parameter "binary_content" is specified.
udata	hash	no	A hash of key value pairs for the user data attached to the interaction.

References

<references />

Chat Interaction

Purpose

Describes chat data for a multimedia interaction.

Resource

Name	Type	Mandatory	Description
Chat-related information			
established_date	date/time<ref name="iso8601">UTC Time, using the ISO 8601 representation: [YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z</ref>	no	The date/time when the chat session was established with a contact center resource.
released_date	date/time<ref name="iso8601"/>	no	The date/time when the chat session was released from the contact center resource.
Common Interaction fields			
customer_id	string	yes	The ID of the customer.
interaction_id	string	yes	The Genesys GUID (or Call ID) for the interaction.
status	integer	yes	Possible statuses: • 0-NEW • 1-PENDING • 2-IN_PROCESS • 3-STOPPED
entity_type	integer	yes	Possible types: • 0-EMAIL_IN • 1-EMAIL_OUT • 2-CHAT • 3-PHONE_CALL

Name	Type	Mandatory	Description
			• 5-CALLBACK • 6-COBROWSE • 7-INTERACTION (open media)
type	string	yes	The type of the interaction (such as inbound, outbound, etc.)
sub_type	string	no	The application-specific subtype which further classifies attribute "type".
media_type	string	yes	The media type of the interaction: • voice • email • chat • callback • and so on.
parent_id	string	no	The ID of the parent interaction, if applicable.
thread_id	string	no	The ID of the thread to which this interaction should be appended, or the ID of the interaction at the root of the thread.
external_id	string	no	The reference identifier for an external system.
owner_id	integer	no	The DB ID of the Person who owns the interaction. (The Person is an object defined in the Genesys Configuration Server.)
creator_app_id	integer	yes	The DB ID of the application that created the interaction.
start_date	date/time<ref name="iso8601"/>	yes	The UTC time of the interaction's creation.
end_date	date/time<ref name="iso8601"/>	no	The UTC time of the interaction's end.
timeshift	integer	no	The number of minutes

Name	Type	Mandatory	Description
			to add to or to substract from the specified UTC<ref name="iso8601" /> time in order to get the local time of the interaction's creation.
category_id	string	no	The category assigned to this interaction by the Classification Server or the desktop's manual override.
allow_children	boolean	no	True if child interactions can be added. Default value is `true`.
subject	string	no	An arbitrary textual subject for the interaction.
stopped_reason	string	no	The application-specified reason for which the interaction ended.
text	text	no	Plain text for the interaction.
structured_text	url	no	The URL of the interaction's structured text.
structured_text_mime	string	yes if structured_text	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the structured text, if specified (such as text/html). Mandatory if the parameter "structured_text" is specified.
binary_content	url	no	The URL of the the interaction's binary content..
binary_content_mime	string	yes if binary_content	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the binary content, if specified (such as text/html).
binary_content_size	integer	yes if binary_content	The size of the binary content in bytes.

Name	Type	Mandatory	Description
			Mandatory if the parameter "binary_content" is specified.
udata	hash	no	A hash of key value pairs for the user data attached to the interaction.

<references />

E-mail Interactions

Purpose

Provides information for e-mail address and interaction management.

E-mail Address

E-Mail Address Fields

Field	Type	Required	Description
address	string	yes	The e-mail address.
personal	string	no	The personal part of the e-mail address, for instance, a displayable such as a localized name.

Inbound E-mail

Inbound e-mails are e-mail interactions which are received by the application or the Genesys Solution.

E-mail In

Field	Type	Required	Description
Inbound-specific fields			
from	E-mail Address	no	The sender's e-mail address.
reply_to	E-mail Address	no	The address used to reply.
to	E-mail Address[]	no	The list of recipient addresses, as an array of e-mail addresses.
cc	E-mail Address[]	no	The list of addresses for copied recipients, as an array of e-mail addresses.
sent_date	date/time	no	The UTC<ref name="iso8601">Wikipedia ISO 8601</ref> date/

Field	Type	Required	Description
			time at which the e-mail was sent.
Common Interaction fields			
customer_id	string	yes	The ID of the customer.
interaction_id	string	yes	The Genesys GUID (or Call ID) for the interaction.
status	integer	yes	Possible statuses: • 0-NEW • 1-PENDING • 2-IN_PROCESS • 3-STOPPED
entity_type	integer	yes	Possible types: • 0-EMAIL_IN • 1-EMAIL_OUT • 2-CHAT • 3-PHONE_CALL • 5-CALLBACK • 6-COBROWSE • 7-INTERACTION (open media)
type	string	yes	The type of the interaction (such as inbound, outbound, etc.)
sub_type	string	no	The application-specific subtype which further classifies the attribute "type".
media_type	string	yes	The media type of the interaction: • voice • email • chat • callback • and so on.
parent_id	string	no	The ID of the parent interaction, if

Field	Type	Required	Description
			applicable.
thread_id	string	no	The ID of the thread to which this interaction should be appended, or the ID of the interaction at the root of the thread.
external_id	string	no	The reference identifier for an external system.
owner_id	integer	no	The DB ID of the Person who owns the interaction. (The Person is an object defined in the Genesys Configuration Server.)
creator_app_id	integer	yes	The DB ID of the application that created the interaction.
start_date	date/time	yes	The UTC time of the interaction's creation, using the ISO 8601 representation<ref name="iso8601">http://en.wikipedia.o wiki/ISO_8601</ref>: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
end_date	date/time	no	The UTC time of the interaction's end, using the ISO 8601 representation<ref name="iso8601" />: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
timeshift	integer	no	The number of minutes to add to or to substract from the specified UTC<ref name="iso8601" /> time in order to get the local time of the interaction's creation.
category_id	string	no	The category assigned to this interaction by the Classification Server or the desktop's manual override.
allow_children	Boolean	no	True if child interactions can be added. Default value is `true`.

Field	Type	Required	Description
subject	string	no	An arbitrary textual subject for the interaction.
stopped_reason	string	no	The application-specified reason for which the interaction ended.
text	text	no	Plain text for the interaction.
structured_text	url	no	The URL of the interaction's structured text.
structured_text_mime	string	yes if structured_text	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the structured text, if specified (such as text/html). Mandatory if the parameter "structured_text" is specified.
binary_content	url	no	The URL of the the interaction's binary content..
binary_content_mime	string	yes if binary_content	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the binary content, if specified (e.g. text/html).
binary_content_size	integer	yes if binary_content	The size of the binary content in bytes. Mandatory if the parameter "binary_content" is specified.
udata	hash	no	A hash of key value pairs for the user data attached to the interaction.

<references />

Outbound E-mail

Outbound e-mails are e-mail interactions which are sent from the application or from the Genesys

Solution.

E-mail Out

Field	Type	Mandatory	Description
Outbound-Specific fields			
from	E-mail Address	no	The sender's e-mail address.
reply_to	E-mail Address	no	The e-mail address used to send a reply.
to	E-mail Address[]	no	The list of recipient addresses, as an array of e-mail addresses.
cc	E-mail Address[]	no	The list of addresses for copied recipients, as an array of e-mail addresses.
bcc	E-mail Address[]	no	The list of addresses for blind-copied recipients, as an array of e-mail addresses.
sent_date	date/time	no	The UTC<ref>http://en.wikipedia.org/wiki/ISO_8601</ref> date/time at which the e-mail was sent.
reference_id	integer	no	Contains the external identifier (message ID) of the parent interaction.
reviewer_id	integer	no	The application-specified ID of the Person who reviewed the e-mail.
standard_response_id	string	no	The ID of the standard response item from which this e-mail was generated.
Common Interaction fields			
customer_id	string	yes	The ID of the customer.
interaction_id	string	yes	The Genesys GUID (or Call ID) for the interaction.
status	integer	yes	Possible statuses: • 0-NEW • 1-PENDING • 2-IN_PROCESS

Field	Type	Mandatory	Description
			• 3-STOPPED
entity_type	integer	yes	Possible types: • 0-EMAIL_IN • 1-EMAIL_OUT • 2-CHAT • 3-PHONE_CALL • 5-CALLBACK • 6-COBROWSE • 7-INTERACTION (open media)
type	string	yes	The type of the interaction (such as inbound, outbound, etc.)
sub_type	string	no	The application-specific subtype which further classifies the attribute "type".
media_type	string	yes	The media type of the interaction: • voice • email • chat • callback • and so on.
parent_id	string	no	The ID of the parent interaction, if applicable.
thread_id	string	no	The ID of the thread to which this interaction should be appended, or the ID of the interaction at the root of the thread.
external_id	string	no	The reference identifier for an external system.
owner_id	integer	no	The DB ID of the Person who owns the interaction. (The Person is an object defined in

Field	Type	Mandatory	Description
			the Genesys Configuration Server.)
creator_app_id	integer	yes	The DB ID of the application that created the interaction.
start_date	date/time	yes	The UTC time of the interaction's creation, using the ISO 8601 representation<ref name="iso8601">http://en.wikipedia.o wiki/ISO_8601</ref>: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
end_date	date/time	no	The UTC time of the interaction's end, using the ISO 8601 representation<ref name="iso8601" />: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
timeshift	integer	no	The number of minutes to add to or to substract from the specified UTC<ref name="iso8601" /> time in order to get the local time of the interaction's creation.
category_id	string	no	The category assigned to this interaction by the Classification Server or the desktop's manual override.
allow_children	Boolean	no	True if child interactions can be added. Default value is `true`.
subject	string	no	An arbitrary textual subject for the interaction.
stopped_reason	string	no	The application-specified reason for which the interaction ended.
text	text	no	Plain text for the interaction.
structured_text	url	no	The URL of the interaction's structured text.
structured_text_mime	string	yes if structured_text	The mime type<ref

Field	Type	Mandatory	Description
			name="mime">http://www.iana.org/assignments/media-types</ref> for the structured text, if specified (such as text/html). Mandatory if the parameter "structured_text" is specified.
binary_content	url	no	The URL of the the interaction's binary content..
binary_content_mime	string	yes if binary_content	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the binary content, if specified (e.g. text/html).
binary_content_size	integer	yes if binary_content	The size of the binary content in bytes. Mandatory if the parameter "binary_content" is specified.
udata	hash	no	A hash of key value pairs for the user data attached to the interaction.

<references />

Phone Call Interaction

	Purpose: Describes phone call fields.

Description

Provides fields for a phone call interaction.

Resource

Field	Type	Mandatory	Description
duration	integer	no	The duration of the call in seconds.
outcome	string	no	The application-specified outcome of the call.
phone_number	string	no	The phone number called.
tconnection_id	string	no	The connection ID assigned to the call by the Genesys T-Server.
customer_id	string	yes	The ID of the customer.
interaction_id	string	yes	The Genesys GUID (or Call ID) for the interaction.
status	integer	yes	Possible statuses: • 0-NEW • 1-PENDING • 2-IN_PROCESS • 3-STOPPED
entity_type	integer	yes	Possible types: • 0-EMAIL_IN • 1-EMAIL_OUT • 2-CHAT

Field	Type	Mandatory	Description
			• 3-PHONE_CALL • 5-CALLBACK • 6-COBROWSE • 7-INTERACTION (open media)
type	string	yes	The type of the interaction (such as inbound, outbound, and so on).
sub_type	string	no	The application-specific subtype which further classifies attribute "type".
media_type	string	yes	The media type of the interaction, including: • voice • email • chat • callback
parent_id	string	no	The ID of the parent interaction, if applicable.
thread_id	string	no	The ID of the thread to which this interaction should be appended, or the ID of the interaction at the root of the thread.
external_id	string	no	The reference identifier for an external system.
owner_id	integer	no	The DB ID of the Person who owns the interaction. (The Person is an object defined in the Genesys Configuration Server.)
creator_app_id	integer	yes	The DB ID of the application that created the interaction.
start_date	date/time	yes	The UTC time of the interaction's creation, using the ISO 8601 representation<ref name="iso8601">http://en.wikipedia.o

Field	Type	Mandatory	Description
			wiki/ISO_8601</ref>: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
end_date	date/time	no	The UTC time of the interaction's end, using the ISO 8601 representation<ref name="iso8601" />: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`
timeshift	integer	no	The number of minutes to add to or substract from the specified UTC<ref name="iso8601" /> time in order to get the local time of the interaction's creation.
category_id	string	no	The category assigned to this interaction by the Classification Server or the desktop's manual override.
allow_children	Boolean	no	Default value of `true` indicates that child interactions can be added.
subject	string	no	An arbitrary textual subject for the interaction.
stopped_reason	string	no	The application-specified reason for which the interaction ended.
text	text	no	Plain text for the interaction.
structured_text	url	no	The URL of the interaction's structured text.
structured_text_mime	string	yes if structured_text	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the structured text, if specified (text/html). Mandatory if the parameter "structured_text" is specified.
binary_content	url	no	The URL of the the interaction's binary

Field	Type	Mandatory	Description
			content.
binary_content_mime	string	yes if binary_content	The mime type<ref name="mime">http://www.iana.org/assignments/media-types</ref> for the binary content, if specified (text/html).
binary_content_size	integer	yes if binary_content	The size of the binary content in bytes. Mandatory if the parameter "binary_content" is specified.
udata	list	no	A list of key-value pairs for the user data attached to the interaction.

<references />

Server Operations

	Purpose: Lists the operations that your application can perform on UCS.

Server Operations

- GET /server/status
- POST /server/mode

Query Server Status

	GET /server/status
	Available since 8.0.100.00

Description

Queries the status of the Universal Contact Server (UCS). The status that is returned includes in particular the Server Mode (production or maintenance), but it also includes additional information and valuable counters to analyse the activity of the UCS. Some counters, such as "request_today", "client_errors_today", and "server_errors_today", are not persistent. They are automatically reset if the UCS is restarted.

Operation

Query Server Status

ID	CV.WS.STAT
Method	GET
URL	/server/status

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200	
HTTP message	OK	
Name	**Type**	**Description**
Body		
version	string	The version of the UCS.
system_time	date/time<ref name="format">The date/time format is "yyyy-MM-ddTHH:mm:ss.SSSZ".</ref>	The system time

HTTP code	200	
mode	token • production • maintenance	The Server Mode.
max_memory	long	The maximum amount of memory the JVM will attempt to use, in KB.
allocated_memory	integer	The amount of memory currently in use by the JVM, in KB.
free_memory	integer	The total memory available to the JVM, in KB.
started	date/time<ref name="format"/>	Start date of the UCS.
last_request	date/time<ref name="format"/>	The time of the last web service request.

<references />

Example

Operation

```
GET /server/status
```

Result

```
"max_memory" : 518464,
"server_errors_today" : 0,
"client_errors_today" : 0,
"request_today" : 0,
"system_time" : "2010-02-24T15:56:46.369Z",
"used_memory" : 7947,
"allocated_memory" : 63936,
"started" : "2010-02-24T14:35:28.668Z",
"free_memory" : 55989,
"last_request" : "2010-02-24T14:35:27.964Z",
"mode" : "production",
"version" : "8.0.100.11"
```

Set Server Mode

	POST /server/mode

Description

Manages the UCS Mode for the Context Services. Two modes are available: Maintenance and Production.

- If the server switches from maintenance to production mode, the system waits a configured amount of time (default 1 minute) for the completion of all schema management requests (see Schema Operations).
- If the server switches from production to maintenance mode, the system waits for outstanding profile or service-related functions to complete.
  While the system is attempting to switch modes, all web service requests are denied with `HTTP Status Code 503 (Service Unavailable)`.

Read the Server Mode page of the developer's guide for further details about maintenance and production mode.

Operation

ID	CV.WS.MODE		
Method	POST		
URL	/server/mode		
Parameter	**Type**	**Mandatory**	**Description**
mode	token • maintenance • production	**yes**	The mode to set (case-insensitive).

Response

The UCS Server status is returned as a JSON object with the following fields: The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Header	Location /server/status
Body	{"server changed":"${mode}"} where: • ${mode} is the string for the new UCS mode (production or maintenance).

Example

Operation

```
 POST /server/mode
{"mode":"production"}
```

Result

```
{"server changed":"production"}
```

Service Operations

	Purpose: Groups the operations that your application can create for both services, states, tasks, and interaction resources.

- Service Operations
  - Since: 8.0.100.00
    - POST /services/start
    - POST /customers/${customer id}/services/${service id}
    - POST /services/${service id}/end
    - GET /services/${service id}
    - GET /services/anonymous/${contact key}/active
    - GET /services/anonymous/${contact key}/completed
    - GET /services/anonymous/${contact key}
    - GET /customers/${customer id}/services/active
    - GET /customers/${customer id}/services/completed
    - GET /customers/${customer id}/services
  - Since: 8.0.200.00
    - PUT /services/${service id}/extensions/${ext name}
    - PUT /services/${service id}/extensions/${ext name}/by/unique
    - PUT /services/${service id}/extensions/${ext name}/delete/by/unique
- State Operations
  - Since: 8.0.100.00
    - POST /services/${service id}/states/${state id}/end
    - POST /services/${service id}/states/transition
    - GET /services/${service id}/states
    - GET /services/${service id}/states/${state id}
    - POST /services/${service id}/states/start
  - Since: 8.0.200.00
    - PUT /services/${service id}/states/${state id}/extensions/${ext name}
    - PUT /services/${service id}/state/${state id}extensions/${ext name}/by/unique
    - PUT /services/${service id}/states/${state id}/extensions/${ext name}/delete/by/unique
- Task Operations

- Since: 8.0.100.00
  - POST /services/${service id}/tasks/${task id}/end
  - GET /services/${service id}/tasks
  - GET /services/${service id}/tasks/${task id}
  - POST /services/${service id}/tasks/start
- Since: 8.0.200.00
  - PUT /services/${service id}/tasks/${task id}/extensions/${extension name}
  - PUT /services/${service id}/task/${task id}extensions/${ext name}/by/unique
  - PUT /services/${service id}/task/${task id}/extensions/${ext name}/delete/by/unique

Associate Service

	POST /customers/${customer_id}/services/${service_id}

Description

Your application can use this operation after the service was created with the Start Service operation to associate an Anonymous Service with a customer, or to associate the service with a different customer. This operation overrides prior associations. For example:

1. Associate 'service 1' with 'customer 1'
2. Associate 'service 1' with 'customer 2'

Result: 'service 1' no longer appears in the active or completed service history of 'customer 1'. See Query Services for further information. This operation can also update some fields of the service resource in the body of the request. In 8.0.2, your application can specify values for extensions (single-valued and multi-valued). If your application already recorded a value or a set of values for a given extension, the previous record is replaced with the new one.

Operation

Prerequisites The service is started, see Start Service.

Associate a service with a customer

ID	CV.WS.SRV.2		
Method	POST		
URL	/customers/**${customer_id}**/services/**${service_id}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The ID of the customer.
${service_id}	integer	yes	The ID of the service.
Optional Body: Service Start Event<ref> This operation's body can include some fields of the Service Start Event for update (for instance, the interaction ID or the resource ID).</ref>			
service_type	long or string	no	The unique ID associated with the type of service, typically the DB ID of a value in the Service Type Business Attribute.

ID	CV.WS.SRV.2		
			Refer to Configuration Options for more details on *Business Attribute* mapping.
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (e.g. Genesys DB ID) for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (e.g. GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration

ID	CV.WS.SRV.2		
			platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given service/state/task, for instance, e-mail, voice, chat, etc.
est_duration	integer	no	The estimated service/state/task duration in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension n> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extensions, where *<extension n>* corresponds to the unique name of the service extension. Your application can specify as many extensions as needed as long as you defined their schemas with the Create Service Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further

details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP Title	OK

Example

Operation
The following code sample associates the customer with a service, and specifies records for the predefined "Feedback" single-valued extension, and for the "Satisfaction" multi-valued extension.

```
POST /customers/0004Va58A92T0018/services/1500
{
 "timestamp": "2009-05-12T12:05:12.145Z",
 "interaction_id": "123ABCAADFJ1259ACF",
 "media_type": 1,
 "resource_id": 5005,
 "resource_type": 2,
 "disposition": 10,
 "Feedback":
  {
          "FeedbackType":"survey",
          "rating":7,
          "notes":"warm welcome at frontdesk, thanks for the nice trip"
  },
 "Satisfaction": [
  {
          "rating":2,
          "pertinence":8,
          "useful":true,
          "place":"Terranova mexico resort"
  },
  {
          "rating":8,
          "pertinence":4,
          "useful":false,
          "place":"Fancy resort Paris"
  }
 ]
}
```

Result

```
200 OK
```

Complete Service

	POST /services/${service_id}/end

Description

Terminates the given service. Your application can update service information in the body of this operation. In 8.0.2, your application can specify values for extensions (single-valued and multi-valued). If your application already recorded a value or a set of values for a given extension, the previous record is replaced with the new one.

Operation

Complete a Service

ID	CV.WS.SRV.3		
Method	POST		
URL	/services/**${service_id}**/end		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameter			
${service_id}	integer	yes	In the request URI, the unique ID of the service.
Body: Service End Event<ref>The body of this operation corresponds to the Service End Event resource.</ref>			
session_id	string	no	The ID of the related session (for instance, the orchestration session or any other business session). Limited to 32 characters.
interaction_id	string	no	The ID of the interaction at the service event's creation. Limited to 50 characters.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business">Refer to Configuration Options

ID	CV.WS.SRV.3		
			for more details on *Business Attribute* mapping.</ref> of application which issued the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The Genesys DB ID for the application which issues the service event, such as a GVP VoiceXML application, Orchestration SCXML application, etc.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource providing the service (e.g. GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • The Genesys DB ID of a specific GVP or orchestration platform • The DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given service, such as e-mail, voice, chat, etc.
disposition	long or string	no	The unique ID for the business disposition<ref name="business"/> assigned to the given service. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization's business dispositions.
disposition_desc	string	no	The text description which provides additional context on

ID	CV.WS.SRV.3		
			the business disposition. Limited to 64 characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server adds it when the service event is processed.
<extension n> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extensions, where *<extension n>* corresponds to the unique name of the service extension. Your application can specify as many extensions as needed as long as you defined their schemas with the Create Service Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP Title	No Content

Example

Operation

```
POST /services/36981723/end
{
  "timestamp": "2009-05-12T12:05:12.145Z",
  "interaction_id": "123ABCAADFJ1259ACF",
  "media_type": 1,
  "resource_id": 5005,
  "resource_type": 2,
  "disposition": 10,
  "disposition_desc": "SUCCESS"
  "Feedback":
   {
       "FeedbackType":"survey",
       "rating":7,
       "notes":"warm welcome at frontdesk, thanks for the nice trip"
   },
   "Satisfaction": [
   {
       "rating":2,
       "pertinence":8,
       "usefull":true,
       "place":"Terranova mexico resort"
   },
   {
       "rating":8,
       "pertinence":4,
       "usefull":false,
       "place":"Fancy resort Paris"
   }
 ]
}
```

Result

```
204 No Content
```

Complete State

	POST /services/${service_id}/states/${state_id}/end
	Available since: 8.0.100.00

Description

This operation terminates the state and updates its information with the body content.

Operation

Completes the state

ID	CV.WS.SRV.6		
Method	POST		
URL	/services/**${service_id}**/states/**${state_id}**/end		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The unique 64-bit ID of the related service.
${state_id}	integer	yes	The 32-bit ID of the state to complete.
Body: State End Event<ref>This body can contain fields from the State End Event resource.</ref>			
session_id	string	no	The ID of the related session (for instance, the orchestration session or any other business session). Limited to 32 characters.
interaction_id	string	no	The ID of the interaction at the service event's creation. Limited to 50 characters.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"> Refer to Configuration Options for more details on *Business*

ID	CV.WS.SRV.6		
			Attribute mapping.</ref> of application which issued the state event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The Genesys DB ID for the application which issues the state event, such as a GVP VoiceXML application, Orchestration SCXML application, etc.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource providing the related service (e.g. GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the related service. For instance: • The Genesys DB ID of a specific GVP or orchestration platform • The DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given state, such as e-mail, voice, chat, etc.
disposition	long or string	no	The unique ID for the business disposition<ref name="business"/> assigned to the given state. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization’s business dispositions.
disposition_desc	string	no	The text description which provides

ID	CV.WS.SRV.6		
			additional context on the business disposition. Limited to 64 characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation: `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server adds it when the service event is processed.
<extension name>	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK

Example

Operation
The following sample completes a state and updates the associated single-valued extension

"Feedback". It assumes that you are using integers for values.

```
POST /services/6739/states/5362/end
{
 "interaction_id":"00001a57JGQ00BVS",
 "disposition": 10,
 "disposition_desc": "SUCCESS",
  "application_type":400,
  "application_id":40,
  "resource_type":200,
  "resource_id":20,
  "media_type":2,
  "Feedback":
   {
    "FeedbackType":"survey",
    "rating":7,
    "notes":"warm welcome at frontdesk, thanks for the nice trip"
   }
 }
```

Result

```
200 OK
```

Operation
The following sample completes a state and updates the associated single-valued extension "Feedback". It assumes that you configured business values, which allow string values.

```
POST /services/6739/states/5362/end
{
  "interaction_id":"00001a57JGQ00BVS",
  "disposition": 10,
  "disposition_desc": "SUCCESS",
  "application_type":"customer_online_survey",
  "application_id":40,
  "resource_type":"html",
  "resource_id":20,
  "media_type":"webform",
  "Feedback":
   {
    "FeedbackType":"survey",
    "rating":7,
    "notes":"warm welcome at frontdesk, thanks for the nice trip"
   }
 }
```

Result

```
200 OK
```

Operation
The following sample completes a state and updates the associated single-valued extension "Feedback" and multi-valued extension "Satisfaction".

```
POST /services/6739/states/5362/end
{
 "interaction_id":"00001a57JGQ00BVS",
 "disposition": 10,
 "disposition_desc": "SUCCESS",
 "application_type":"customer_online_survey",
```

```
  "application_id":40,
  "resource_type":"html",
  "resource_id":20,
  "media_type":"webform",
  "Feedback":
  {
    "FeedbackType":"survey",
    "rating":7,
    "notes":"warm welcome at frontdesk, thanks for the nice trip"
   },
  "Satisfaction": [
   {
    "rating":2,
    "pertinence":8,
    "usefull":true,
    "place":"Terranova mexico resort"
   },
   {
     "rating":8,
     "pertinence":4,
     "usefull":false,
     "place":"Fancy resort Paris"
   }
   ]
}
```

Result

```
200 OK
```

Complete Task

	POST /services/${service_id}/tasks/${task_id}/end
	Available since: 8.0.100.00

Description

This operation completes the task of a given type for the given service. The type is one of the enumerated values recorded for the corresponding business attribute. See Business Attributes in Context Services for further details.

- UCS assigns auto-incrementing identifiers to tasks, similar to the case of services and tasks.
- States and tasks are assigned 32-bit identifiers.

Operation

ID	CV.WS.SRV.9		
Method	POST		
URL	/services/**${service_id}**/tasks/**${task_id}**/end		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The unique 64-bit ID of the service.
${state_id}	integer	no	The unique 32-bit ID of the state.
${task_id}	integer	yes	The unique 32-bit ID of the task.
Body: Task End Event<ref>This body contains fields from the Task End Event resource.</ref>			
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer,

ID	CV.WS.SRV.9		
			and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (e.g. Genesys DB ID) for the application issuing the event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (e.g. GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given task, for instance, e-mail, voice, chat, etc.
disposition	long or string	no	The unique ID for the business disposition<ref name="business">Refer to Configuration Options for more details on Business Attribute mapping.</ref> assigned to the given service/state/task. For

ID	CV.WS.SRV.9		
			instance, this can be the DB ID of a Business Attribute used to enumerate a given organization's business dispositions.
disposition_desc	string	no	The text description which provides additional context on the business disposition. Limited to 64 characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server adds it when the service event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension. Your application can add as many task extensions as needed, as long as you created a corresponding Extension Schema with the Create Task Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation

```
POST /services/95/tasks/3/end
 {
 "interaction_id":0000CSS0FXZUP6UR,
 "state_id":24,
 "disposition": 10,
 "disposition_desc": "SUCCESS",
 "Survey":
  {
   "url":"http://ourServer/storage/userAnswers",
   "question1":7,
   "question2":true,
   "question3":"will be better with cable tv and on-demand video"
  },
  "Proposal": [
          {
           "car type":"cabriolet",
           "price":25 000,
           "seats":2,
           "comments":"200 cv, hardtop"
          },
          {
           "car type":"S.U.V.",
           "price":70 000,
           "seats":8,
           "comments":"4wd, leather seats"
          }
  ]
}
```

Result

```
 204 No Content
```

Delete Record From Service Extension

	PUT /services/${service_id}/extensions/${ext_name}/delete/by/unique
	Available since: 8.0.200.00

Description

Deletes a single record in a multi-valued extension. In the operation's body, you must specify the unique attributes that identify the record in the given extension. See Unique Attributes for further information.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.22		
Method	PUT		
URL	/services/**${service_id}**/extensions/**${ext_name}**/delete/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	long	yes	The service ID.
${ext_name}	string	yes	The name of the Service Extension to modify.
Body<ref>Unique attributes of the Service Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime • currency • boolean	yes	Unique attributes (names and values) which identify the record to delete.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation deletes the relatedOffers record identified by the attribute "offer_name" of service 8389.

```
PUT /services/8389/extensions/relatedOffers/by/unique
{
   "offer_name":"3 times payment GOLD"
}
```

Result

```
204 No Content
```

Delete Record From State Extension

	PUT /services/${service_id}/states/${state_id}/extensions/${ex by/unique
	Available since: 8.0.200.00

Description

Deletes a single record in a multi-valued extension for a given state. In the operation's body, you must specify the unique attributes which identify the record to delete. See Unique Attributes for further information.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.23		
Method	PUT		
URL	/services/**${service_id}**/states/**${state_id}**/extensions/**${ext_name}**/delete/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	long	yes	The service ID.
${state_id}	integer	yes	The state ID.
${ext_name}	string	yes	The name of the State Extension to modify.
Body<ref>"Unique" record attributes of the State Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime • currency	yes	Unique attributes (names and values) which identify the record to delete.

ID	CV.WS.SRV.23		
	• boolean		

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation deletes the Satisfaction record of state 1 in service 8389. The concerned record is identified by the "place" attribute.

```
PUT /services/8389/states/1/extensions/Satisfaction/by/unique
{
   "place":"Tons of Hill, Paris"
  }
```

Result

```
204 No Content
```

Delete Record From Task Extension

	PUT /services/${service_id}/task/${task_id}/extensions/${ext_n by/unique
	Available since: 8.0.200.00

Description

Deletes a single record in a multi-valued extension for a given task. In the body of the operation, you must specify the unique attributes which identify the record to delete. See Unique Attributes for further information.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.24		
Method	PUT		
URL	/services/**${service_id}**/task/**${task_id}**/extensions/**${ext_name}**/delete/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	long	yes	The service ID.
${task_id}	integer	yes	The task ID.
${ext_name}	string	yes	The name of the Task Extension to modify.
Body<ref>Unique attributes of the Task Extension record.</ref>			
<attribute n>	• string • integer • long • double • date • datetime • currency	yes	Attribute names and values of the task extension, marked as unique in the Extension Schema, and used to identify the record to delete.

ID	CV.WS.SRV.24		
	• boolean		

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation removes the "car type" Proposal record of task 1 in service 8389.

```
PUT /services/8389/tasks/1/extensions/Proposal/by/unique
{
    "car type":"cabriolet"
}
```

Result

```
204 No Content
```

Perform State Transition

	POST /services/${service_id}/states/transition
	Available since: 8.0.100.00

Description

Performs a state transition for a given service. In many cases, your application is responsible for completing a given state and starting a new state, both in the context of a single interaction. In that context, your application can use this operation (instead of two separate Start State and Complete State operations) to reduce both the data duplication and the overall UCS services workload.

Operation

ID	CV.WS.SRV.7		
Method	POST		
URL	/services/${service_id}/states/transition		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
service_id	integer	yes	The unique ID of the related service.
Body: State Transition Event<ref>This body contains fieds from the State Transition Event.</ref>			
from	hash	yes	The hash of the following fields describing the state from which the service is transitioning: • **state_id (mandatory)** : The 32-bit ID of the state to complete. • disposition (optional): The unique ID for the business disposition assigned to the given state. For instance, this can be the DB ID of a

ID	CV.WS.SRV.7		
			Business Attribute used to enumerate a given organization's business dispositions<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>. • disposition_desc (optional): The text description which provides additional context on the business disposition. Limited to 64 characters.
to	hash	yes	The hash of the following fields, which describe the state into which the service is transitioning: • **state_type (mandatory)**: The unique ID associated with the type of service, typically the DB ID of a value in the Service Type Business Attribute<ref name="business"/>. • est_duration (optional): The estimated service duration in seconds.
The following fields apply both to the state to complete and the state to start:			
session_id	string	no	The ID of the related session. For example, the orchestration session or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer,

ID	CV.WS.SRV.7		
			and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (Genesys DB ID) for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (such as GVP, Agent Desktop, or Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given state. For example: e-mail, voice, chat, etc.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref>

ID	CV.WS.SRV.7		
			representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: /service/**${service_id}**/states/**${state_id}** where: • ${service_id} is the service ID. • ${state_id} is the ID of the created state.
Body	{"state_id": **${state_id}**} where: ${state_id} is the created state ID.

Example

Operation

```
POST /services/735692/states/transition
```

```
 {
   "timestamp": "2009-05-07T12:05:20.157",
   "session_id": "11000ABC-80236C1A-1010",
   "interaction_id": "123ABC908ABFFD8080",
   "from": {
     "state_id": 1001,
     "disposition": 1,
     "disposition_desc": "SUCCESS",
     "Feedback":
       { "FeedbackType":"survey","rating":7,
         "notes":"warm welcome at frontdesk, thanks for the nice trip"
       },
    "Satisfaction": [
    {
      "rating":2,
      "pertinence":8,
      "usefull":true,
      "place":"Terranova mexico resort"
     },
     {
      "rating":8,
      "pertinence":4,
      "usefull":false,
      "place":"Fancy resort Paris"
     }
    ]
  },
  "to": {
    "state_type": 8,
    "est_duration": 500,
    "Sponsoring": { "Rank":"first","expire":7,
      "notes":"give customer free meal" }
   }
  }

POST /services/735692/states/transition
{
  "timestamp": "2009-05-07T12:05:20.157",
  "session_id": "11000ABC-80236C1A-1010",
  "interaction_id": "123ABC908ABFFD8080",
  "from": {
    "state_id": 1001,
    "disposition": 1,
    "disposition_desc": "SUCCESS",
    "Feedback": { "FeedbackType":"survey","rating":7,
           "notes":"warm welcome at frontdesk, thanks for the nice trip" },
    "Sponsoring":  { "Rank":"first","expire":7,
           "notes":"give customer free meal" }
  },
  "to": {
    "state_type": 8,
    "est_duration": 500,
    "Satisfaction": [
     {
      "rating":7,
      "pertinence":3,
      "usefull":false,
      "place":"Australian beach resort complex"
     },
     {
      "rating":9,
      "pertinence":8,
      "usefull":true,
```

```
      "place":"Caribbean beach complex"
     }
    ]
  }
}
```

Result

```
 201 Created
 {"state_id":15158}
```

Query Service by ID

	GET /services/${service_id}

Description

Enables your application to query a service if its ID is known. By default, no tasks or states are returned. The example below shows how your application can use optional parameters to customize its request to specify which nested tasks and states of the service should be part of the response.

Operation

ID	CV.WS.SRV.14		
Method	GET		
URL	/services/**${service_id}**		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	**yes**	The service ID.
active_states	bool	no	• "true" to include information on this service's active states in the results • "false" (default)
completed_states	bool	no	• "true" to include information on this service's completed states in the results • "false" (default)
active_tasks	bool	no	• "true" to include information on this service's active tasks in the results • "false" (default)

ID	CV.WS.SRV.14		
completed_tasks	bool	no	• "true" to include information on this service's completed tasks in the results • "false" (default)
extensions Supported since 8.0.2	string	no	Extension name to retrieve with the results. Your application can specify several names separated by commas, as long as you created corresponding Extension Schema with the Create Service Extension Schema operation. For example: *extensions=Feedback,SpecialOffer*.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200		
HTTP message	OK		
Field Name	**Type**	**Mandatory**	**Description**
Body: Service<ref>This response includes the representation of the Service associated with the service ID.</ref>			
service_type	long or string	yes	The unique ID associated with the service type, typically the DB ID of a value in the Service Type Business Attribute. Refer to Configuration Options for more details about *Business Attribute* mapping.
service_id	integer	yes	The service's unique ID.<ref name="toplevel">Only top-level objects include this field. For instance, if your application

HTTP code	200		
			retrieves a service which includes nested states and tasks, only the top-level service object contains this ID.</ref>
started	Service Start Event	yes	Start event (see Start Service). This event does not include the "est_duration" field, if "est_duration" is specified at the service level of the response.
completed	Service End Event	no	The related end event if the service is completed (see Complete Service). This event does not include the "disposition" and "disposition_desc" fields if they are specified at the service level of the response.
customer_id	string	no	The customer's unique ID<ref name="toplevel"/>. If the service has no customer ID, the service is Anonymous Servic.
est_duration	integer	no	The estimated time for completing the service, in seconds.
duration	integer	no	The total duration in milliseconds. Only relevant for completed services.
disposition	integer	no	ID of the Business disposition ID. Only relevant for completed services.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
active_states	State[]	no<ref name="option">The field is returned if the corresponding option was set to true in the request.</ref>	The array of the started states which are not completed.

HTTP code	200		
completed_states	State[]	no<ref name="option"/>	The array of completed states through which the service has made transitions.
active_tasks	Task[]	no<ref name="option"/>	The array of the started tasks which are not completed.
completed_tasks	Task[]	no<ref name="option"/>	The array of the tasks which are completed as part of the service delivery.
interactions	Interaction[]	no<ref name="option"/>	The array of the interactions which are associated with this service.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extension matching the parameter passed in the operation.

<references />

Example

Operation The following operation retrieves the given service (ID=2000) with its active states and tasks to resume the activity.

```
 GET /services/
3005?active_states=trueCleted_states=true~tive_tasks=trueCleted_tasks=true&extensions=ClientInfo,relatedOffers
```

Result

```
{
  "completed_tasks" : [ {
    "service_id" : 3005,
    "duration" : 82937,
    "disposition_desc" : "normal ending",
    "task_id" : 2001,
    "task_type" : 55,
    "est_duration" : 540,
    "started" : {
      "timestamp" : "2010-06-03T08:49:53.053Z",
      "interaction_id" : "587"
    },
    "disposition" : 5,
    "completed" : {
      "timestamp" : "2010-06-03T08:51:15.990Z",
      "interaction_id" : "587"
    }
  } ],
  "service_id" : 3005,
  "active_states" : [ {
    "service_id" : 3005,
```

```
    "state_id" : 4000,
    "state_type" : 100,
    "est_duration" : 60,
    "started" : {
      "timestamp" : "2010-06-03T08:48:18.257Z",
      "application_type" : 400,
      "resource_id" : 20,
      "media_type" : 2,
      "resource_type" : 200,
      "application_id" : 40,
      "interaction_id" : "51"
    }
  } ],
  "ClientInfo" : {
    "userAgent" : "Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115
Firefox/3.6 (.NET CLR 3.5.30729)",
    "clientIp" : "192.168.1.1",
    "contentType" : "Content-Type : application/json;charset=UTF-8"
  },
  "active_tasks" : [ {
    "service_id" : 3005,
    "state_id" : 4001,
    "task_id" : 2000,
    "task_type" : 55,
    "est_duration" : 540,
    "started" : {
      "timestamp" : "2010-06-03T08:49:45.943Z",
      "interaction_id" : "587"
    }
  } ],
  "service_type" : 100,
  "est_duration" : 300,
  "started" : {
    "timestamp" : "2010-09-07T07:58:16.313Z",
    "application_type" : 400,
    "resource_id" : 10,
    "media_type" : 2,
    "resource_type" : 200,
    "application_id" : 40,
    "interaction_id" : "56"
  },
  "completed_states" : [ {
    "service_id" : 3005,
    "state_id" : 4001,
    "duration" : 182907,
    "disposition_desc" : "normal ending",
    "state_type" : 200,
    "est_duration" : 300,
    "started" : {
      "timestamp" : "2010-06-03T08:48:51.473Z",
      "application_type" : 400,
      "resource_id" : 6000,
      "media_type" : 3,
      "resource_type" : 100,
      "application_id" : 40,
      "interaction_id" : "8001"
    },
    "disposition" : 5,
    "completed" : {
      "timestamp" : "2010-06-03T08:51:54.380Z",
      "interaction_id" : "1587"
    }
  } ],
```

```
  "relatedOffers" : [ {
    "offer_name" : "VIP credit card black ed.",
    "type" : "9",
    "comments" : "proposed to all client"
  }, {
    "offer_name" : "3 times payment GOLD",
    "type" : "4",
    "comments" : "limited offer"
  }, {
    "offer_name" : "life insurance",
    "type" : "3",
    "comments" : "healt check to be done before approval"
  } ],
  "contact_key" : "bob"
}
```

- The service=2000 does not contain 'completed' values. It is an active service.
- There was no active states for this service.
- There is a single active task.

Query Services

- **GET /services/anonymous/${contact_key}**
- **GET /customers/${customer_id}/services**

Description

The operations on this page allow users to query for active and/or completed services, either for a specific customer or for anonymous services matching a contact key. The latter case primarily addresses service resumption use cases for services in which the customer is not yet identified. For example, consider the following:

- The customer calls in from phone number 408-832-7799.
- The application starts a service, noting contact key "408-832-7799" for further reference.
- The application is not able to identify the caller based on the phone number.
- IVR identifies that the customer is calling to set up a new wireless account.
- The customer is placed in an agent queue. After 10 minutes, the call is dropped.
- The customer calls back in from the same number. The application still cannot identify the customer based on that number.
- The application looks up whether there are any outstanding services with identification token "408-832-7799"--one is returned.
- The application asks the customer whether he or she is calling back in regards to the previous request or for a new wireless request. Upon confirmation, the customer is placed back in the agent queue.

Operations

Query Anonymous Services

This operation queries anonymous services only.

ID	CV.WS.SRV.10
Method	GET
URI	1. Active: /services/anonymous/**${contact_key}**/active 2. Completed: /services/anonymous/**${contact_key}**/completed

ID	CV.WS.SRV.10		
	3. All: /services/anonymous/**${contact_key}**		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${contact_key}	string	yes	The "contact key" is supplied at the service creation if the service is started before the customer's identification. This key can be used to look up the service with this method, or for further association once the customer is identified (see Associate Service).
started_from	date/time<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601 representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`</ref>	no	Filters the services started on or after the given date/time.
started_to	date/time	no	Filters the services started prior to the given date/time.
completed_from	date/time<ref name="ISO 8601"/>	no	Filters the services that were completed on or after the given date/time. Only applicable when querying service history.
completed_to	date/time<ref name="ISO 8601"/>	no	Filters the services completed prior to the given date/time. Only applicable when querying service history.
service_type	integer	no	Filters the services by the application-defined service type<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>. your application can specify several types separated by a comma, as follows: *service_types=1,3,45*
active_states	bool	no	• "true" to include

ID	CV.WS.SRV.10		
			information on the service's active states in the results. • "false" by default.
active_tasks	bool	no	• "true" to include information on the service's active tasks in the results. • "false" (default).
completed_states	bool	no	• "true" to include information on the service's completed states in the results. • "false" (default).
completed_tasks	bool	no	• "true" to include information on the service's completed tasks in the results. • "false" (default).
extensions Supported since 8.0.2	string	no	Name or list of names for the service extension to return with the matching service resources. You can specify as multiple extension names separated by commas, as follows: *extensions=Purchase,Feedback*. By default, no extensions are returned.

<references />

Query Customer Services

This operation queries services which are not anonymous, that is, services which are associated with a customer id.

ID	CV.WS.SRV.10		
Method	GET		
URIs	1. Active Services: /customers/${customer_id}/services/active 2. Completed Services: /customers/${customer_id}/services/completed 3. All Services: /customers/${customer_id}/services		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The Customer ID, to get associated active or completed services. Note that a customer can be associated with a service either at the service creation or later in the process (see Associate Service).
started_from	date/time<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601 representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`</ref>	no	Filters the services started on or after the given date/time.
started_to	date/time<ref name="ISO 8601"/>	no	Filters the services started prior to the given date/time.
completed_from	date/time<ref name="ISO 8601"/>	no	Filters the services that were completed on or after the given date/time. Only applicable when querying service history.
completed_to	date/time<ref name="ISO 8601"/>	no	Filters the services completed prior to the given date/time. Only applicable when querying service history.
service_type	long or string	no	Filters the services by the application-defined service type<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>.
active_states	bool	no	• "true" to include information on the

ID	CV.WS.SRV.10		
			service's active states in the results. • "false" by default.
active_tasks	bool	no	• "true" to include information on the service's active tasks in the results. • "false" (default).
completed_states	bool	no	• "true" to include information on the service's completed states in the results. • "false" (default).
completed_tasks	bool	no	• "true" to include information on the service's completed tasks in the results. • "false" (default).
extensions Supported since 8.0.2	string	no	Name or list of names for the service extension to return with the matching service resources. You can specify several extension names separated by commas, as follows: *extensions=Purchase,Feedback*. By default, no extensions are returned.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	Service[].

Example

Operation

```
GET /customers/ABC1234/services/active?active_states=true
```

Response

```
  [ // returned in an array
   { "customer_id": "ABC1234",
    "service_id": 4692834,
    "est_duration": 86400,
    "started": {
     "timestamp": "2009-05-07T12:05:20.157",
     // additional [[Start Event List#Service_Start_Event|Start Event]] fields
   },
   "active_states":
    [// included given specification of "results" attribute
     { // array of one or more State objects
      "state_id": 5005,
      "state_type": 8, // service delivery
      "started": {
        "timestamp": "2009-05-07T12:08:53.298",
        // additional [[Start Event List#Service_Start_Event|Start Event]] fields
       }
     }
    ]
  }]
```

Note that, in the above example, the inline State objects do not include active/completed tasks.

Query State by ID

	GET /services/${service_id}/states/${state_id}
	Available since: 8.0.100.00

Description

This operation allows you to retrieve a single state for a given service. By default, no task information in returned with the state representation. Optional URI parameters enable to include the state's nested tasks in the response.

Operation

ID	CV.WS.SRV.15		
Method	GET		
URL	/services/**${service_id}**/states/**${state_id}**		
Field name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The service ID.
${state_id}	integer	yes	The state ID.
active_tasks	bool	no	• *true* to include information on the service's active tasks in the results. • *false* (default)
completed_tasks	bool	no	• *true* to include information on the service's completed tasks in the results. • *false* (default)
extensions Supported since 8.0.2	string	no	Names (separated by commas) of the state extensions to return with the matching state resources. You can

ID	CV.WS.SRV.15		
			specify as many extension names as needed, as long as you created them previously with the Create State Extension Schema. By default, no extension is returned.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200		
HTTP message	OK		
Field Name	**Type**	**Mandatory**	**Description**
Body: State<ref>This response includes the State representation with the state ID.</ref>			
state_id	integer	yes	The 32-bit integer ID of the state, assigned by the UCS when the state is started.
state_type	long or string	yes	The unique ID associated with the state type. For instance, this ID can be the DB ID of a value in the Service Type Business Attribute. Enumerated values could be one of the following: 1. Customer identification 2. Service identification 3. Assign service agent 4. Waiting for service agent 5. Offering another service while waiting for agent 6. Offering callback 7. Callback pending

HTTP code	200		
			8. Delivering service 9. Waiting for customer input 10. Offering another service while delivering service Refer to Configuration Options for more details on *Business Attribute* mapping.
service_id	integer	yes if not nested in service resource	The service's unique ID if the state is not part of a top-level service resource which includes a service ID.<ref name="toplevel">Only top-level representations include this field. For instance, if your application retrieves a service which includes nested states and tasks, only the top-level service resource contains this ID.</ref>
started	State Start Event	yes	Related start event. This event should not contain the "est_duration" field, if it already exists at the state level.
completed	State End Event	no	The related end event if the state is completed. This event should not repeat the "disposition" and "disposition_desc" fields if they already exist at the state level.
customer_id	string	no	The customer's unique ID<ref name="toplevel"/>.
est_duration	integer	no	The estimated time for completing the state, in seconds.
duration	integer	no	The total duration in milliseconds. Only relevant for completed states.

HTTP code	200		
disposition	long or string	no	ID of the Business Disposition ID. Only relevant for completed states.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
active_tasks	Task[]	no<ref name="option">This field is returned if the corresponding option is set to true in the URI parameters.</ref>	The array of tasks that are started but not yet completed.
completed_tasks	Task[]	no<ref name="option"/>	The array of Tasks that are completed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension matching the parameter passed in the operation.

<references />

Example

Operation

```
GET /services/2/states/1?active_tasks=trueCleted_tasks=true&extensions=Feedback,Satisfaction
```

Result

```
200 OK
[ // array of one or more state object
{
  "service_id" : 1,
  "state_id" : 0,
  "state_type" : 100,
  "est_duration" : 60,
  "started" : {
    "timestamp" : "2010-05-18T15:23:34.447Z",
    "application_type" : 400,
    "resource_id" : 20,
    "media_type" : 2,
    "resource_type" : 200,
    "application_id" : 40,
    "interaction_id" : "51"
  },
 "Feedback":
   {"FeedbackType":"survey",
   "rating":7,
   "notes":"warm welcome at frontdesk, thanks for the nice trip"}
  },
```

```
 "Satisfaction": [
  {
   "rating":2,
   "pertinence":8,
   "usefull":true,
   "place":"Terranova mexico resort"
  },
  {
   "rating":8,
   "pertinence":4,
   "usefull":false,
   "place":"Fancy resort Paris"
  }
  ],

  "active_tasks": [
    // included given specification of "results" attribute
    { // array of one or more Task objects
      "task_id": 25080,
      "task_type": 5, // application-defined task type
      "est_duration": 300,
      "started": {
        "timestamp": "2009-05-07T12:08:53.298",
      }
    }
  ]
  "completed_tasks": [
    { "task_id": 24027,
      "task_type": 2, // e.g. task for processing quote
      "disposition": 10, // e.g. quote completed
      "disposition_desc": "<text describing quote result>",
      "started": {
        "timestamp": "2009-05-07T12:02:23.715",
        "interaction_id": "123ABC908ABFFD8080"
      },
      "completed": {
        "timestamp": "2009-05-07T12:06:23.715",
        "interaction_id": "157C9A208AFD523D01"
      }
    }
  ]
},
{
  "service_id" : 1,
  "state_id" : 1,
  "state_type" : 100,
  "est_duration" : 60,
  "started" : {
    "timestamp" : "2010-05-18T15:23:41.977Z",
    "application_type" : 400,
    "resource_id" : 20,
    "media_type" : 2,
    "resource_type" : 200,
    "application_id" : 40,
    "interaction_id" : "51"
  }
}
]
```

Query States

	GET /services/${service_id}/states
	Available since: 8.0.100.00

Description

This operation queries the states for a given service. By default, all the states are returned, but your application can filter the results to retrieve only active or completed states, based on the specified URI. Additional URI parameters enable to include task information in the response.

Operation

Query states

ID	CV.WS.SRV.12		
Method	GET		
URL	1. All (default): /services/**${service_id}**/states 2. Active States: /services/**${service_id}**/states/active 3. Completed States: /services/**${service_id}**/states/completed		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	string	yes	The service ID generated from the Start Service operation.
state_types	integer	no	Filters specific service state types. The possible enumerated values are available in the filter field of the State resources. You can specify several state_type values separated by commas.
active_tasks	bool	no	• *true* to include information the service's active

ID	CV.WS.SRV.12		
			tasks in the results. • *false* (default)
completed_tasks	bool	no	• *true* to include information on the service's completed tasks in the results. • *false* (default)
extensions Supported since 8.0.2	string	no	Names separated by commas of the state extensions to return with the matching state resources. By default, no extensions are returned.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	State[]

Example

Operation

```
GET /services/ABC1235XZY/states/active~tive_tasks=trueCleted_tasks=true&extensions=Feedback
```

Result The following output shows a possible response for the previous request:

```
200 OK
[ // array of one or more state object
  {
    "service_id" : 1,
    "state_id" : 0,
    "state_type" : 100,
    "est_duration" : 60,
    "started" : {
```

```
    "timestamp" : "2010-05-18T15:23:34.447Z",
    "application_type" : 400,
    "resource_id" : 20,
    "media_type" : 2,
    "resource_type" : 200,
    "application_id" : 40,
    "interaction_id" : "51"
    "Feedback":
     { "FeedbackType":"survey",
       "rating":7,
       "notes":"warm welcome at frontdesk, thanks for the nice trip"}
     },

    "active_tasks": [
     // included given specification of "results" attribute
     { // array of one or more Task objects
       "task_id": 25080,
       "task_type": 5, // application-defined task type
       "est_duration": 300,
       "started": {
       "timestamp": "2009-05-07T12:08:53.298",
      }
     }
    ]
    "completed_tasks": [
     { "task_id": 24027,
       "task_type": 2, // e.g. task for processing quote
       "disposition": 10, // e.g. quote completed
       "disposition_desc": "<text describing quote result>",
       "started": {
         "timestamp": "2009-05-07T12:02:23.715",
         "interaction_id": "123ABC908ABFFD8080"
        },
       "completed": {
         "timestamp": "2009-05-07T12:06:23.715",
         "interaction_id": "157C9A208AFD523D01"
       }
     }
    ]
 },
 {
   "service_id" : 1,
   "state_id" : 1,
   "state_type" : 100,
   "est_duration" : 60,
   "started" : {
   "timestamp" : "2010-05-18T15:23:41.977Z",
   "application_type" : 400,
    "resource_id" : 20,
    "media_type" : 2,
    "resource_type" : 200,
   "application_id" : 40,
   "interaction_id" : "51"
  }
}
]
```

Query Task by ID

	GET /services/${service_id}/tasks/${task_id}
	Available since: 8.0.100.00

Description

Queries a task by its ID for the given service.

Operation

ID	CV.WS.SRV.16		
Method	GET		
URL	/services/**${service_id}**/tasks/**${task_id}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The service ID.
${task_id}	integer	yes	The task ID.
extensions	string	no	Names (separated by commas) of the task extensions to return with the matching task resources. You can specify as many extension names as needed as long as you previously created the corresponding schema with Create Task Extension Schema. By default, no extension is returned.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200		
HTTP message	OK		
Field Name	**Type**	**Mandatory**	**Description**
Body: Task<ref>This response includes the Task representation associated with the task ID.</ref>			
state_id	integer	no	The 32-bit integer ID for the state service associated with the task.
task_id	integer	yes	The 32-bit integer ID of the task, assigned by the UCS when the task is started.
task_type	long or string	yes	The unique ID associated with the type of service<ref name="business">Refer to Configuration Options for more details on *Business Attribute* mapping.</ref>. For instance, this ID can be the DB ID of a value in the Service Type Business Attribute.
service_id	integer	yes if not specified in top-level resources<ref name="toplevel">Only top-level representations include this field. For instance, if your application retrieves a service which includes nested states and tasks, only the top-level service representations contains this ID.</ref>	The service's unique ID if the task is not part of a top-level state or service resource which includes a service ID.
started	Task Start Event	yes	Related start event. This start event does not contain the "est_duration" field, if this field is already specified at the task level.
completed	Task End Event	no	The related end event if this task is completed. This end event does not include the "disposition" and "disposition_desc" fields if they are already specified at the task

HTTP code	200		
			level.
customer_id	string	no	The customer's unique ID<ref name="toplevel"/>.
est_duration	integer	no	The estimated time for completing the item, in seconds.
duration	integer	no	The total duration in milliseconds. Only relevant for completed tasks.
disposition	long or string	no	ID of the Business Disposition ID<ref name="business"/>. Only relevant for completed tasks.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension if specified in parameter of the operation.

<references />

Example

Operation

```
GET /services/942/tasks/12?extensions=Survey,Proposal
```

Result

```
{
 "interaction_id":42,
 "est_duration":460,
 "state_id":24,
 "task_type":55,
 "Survey":
  {
   "url":"http://ourServer/storage/userAnswers",
   "question1":7,
   "question2":true,
   "question3":"will be better with cable tv and on-demand video"
  },
  "Proposal": [
   {
    "car type":"cabriolet",
    "price":25 000,
```

```
    "seats":2,
    "comments":"200 cv, hardtop"
   },
   {
    "car type":"S.U.V.",
    "price":70 000,
    "seats":8,
    "comments":"4wd, leather seats"
   }
   ]
}
```

Query Tasks

	GET /services/${service_id}/tasks
	Available since: 8.0.100.00

Description

This operation queries the tasks for a given service. By default, all the tasks are returned, but your application can filter the results to retrieve only active or completed tasks, based on the specified URI. Your application can also filter the returned tasks with additional URI parameters to select one or more task type and a given state.

Operation

ID	CV.WS.SRV.13
Method	GET
URL	1. All (default): /services/**${service_id}**/tasks 2. Active Tasks: /services/**${service_id}**/tasks/active 3. Completed Tasks: /services/**${service_id}**/tasks/completed

Field Names	Type	Mandatory	Description
URI Parameters			
${service_id}	string	yes	The service ID generated from the service creation.
state_id	integer	no	Used to filter tasks by state ID. To filter both on the state id and the task_type, your application can also use this field identically to task_type. For example: /services/${service_id}/tasks/completed?task_types=1,3&state_id=700
task_types	long or string	no	Filters specific service task types separated by commas.

ID	CV.WS.SRV.13		
extensions	string	no	Names, separated by commas, of the task extensions to return with the matching task resources. You can specify as many extension names as needed, as long as you previously created the corresponding schema with Create Task Extension Schema. By default, no extension is returned.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	Task[].

Example

Retrieve completed tasks

Operation Retrieve completed tasks for service 6001:

```
GET /services/15/tasks/completed?task_types=STT-DefineCustomerNeed%2CSTT-
FillCustomerInfo&state_id=5&extensions=Survey
```

Result

```
[ {
 "service_id" : 1,
 "state_id" : 2,
 "task_id" : 0,
 "task_type" : 55,
 "est_duration" : 55,
 "started" : {
   "timestamp" : "2010-05-20T15:41:52.380Z",
   "interaction_id" : "65"
 },
```

```
 "Survey":
  {"url":"http://ourServer/storage/userAnswers",
  "question1":7, "question2":true,
  "question3":"will be better with cable tv and on-demand video"
  },
 "Proposal": [
   {
    "car type":"cabriolet",
    "price":25 000,
    "seats":2,
    "comments":"200 cv, hardtop"
   },
   {
    "car type":"S.U.V.",
    "price":70 000,
    "seats":8,
    "comments":"4wd, leather seats"
   }
  ]
}, {
 "service_id" : 1,
 "state_id" : 1,
 "task_id" : 1,
 "task_type" : 12,
 "est_duration" : 900,
 "started" : {
   "timestamp" : "2010-05-20T15:42:57.207Z",
   "interaction_id" : "954"
 },
} ]
```

Retrieve active tasks

Operation Retrieve active tasks for service 6001:

```
 GET /services/6001/tasks/active
```

Result

```
[ {
  "service_id" : 6001,
  "task_id" : 15001,
  "task_type" : 1433,
  "started" : {
    "timestamp" : "2010-04-25T16:07:12.133Z",
    "application_type" : 1426,
    "media_type" : 1010,
    "resource_type" : 1423
  }
} ]
```

Operation Retrieve all tasks for service 6001:

```
  GET /services/6001/tasks
```

Result

```
[ {
  "service_id" : 6001,
  "duration" : 0,
```

```
  "task_id" : 15000,
  "task_type" : 1433,
  "started" : {
    "timestamp" : "2010-04-25T16:07:12.133Z",
    "application_type" : 1426,
    "media_type" : 1010,
    "resource_type" : 1423
  },
  "completed" : {
    "timestamp" : "2010-04-25T16:07:12.133Z",
    "application_type" : 1426,
    "media_type" : 1010,
    "resource_type" : 1423
  }
},{
  "service_id" : 6001,
  "task_id" : 15001,
  "task_type" : 1433,
  "started" : {
    "timestamp" : "2010-04-25T16:07:12.133Z",
    "application_type" : 1426,
    "media_type" : 1010,
    "resource_type" : 1423
  }
} ]
```

Start Service

	POST /services/start
	Available since: 8.0.200.00

Description

Starts a service based on:

- The service type passed in the body.
- The contact information available (customer ID or contact key for later lookup).

If your application does not assign a customer ID, your service is anonymous. In that case, your application must assign a contact key. For further information, read the following pages:

- Documentation:CS:Developer:Anonymous Service
- Service Start Event

Your application is not responsible for the service ID creation. UCS automatically assigns 64-bit integer IDs to the created services.

Operation

Start Service

ID	CV.WS.SRV.1		
Method	POST		
URL	/services/start		
Field Name	**Type**	**Mandatory**	**Description**
Body: Service Start Event<ref>The body of this operation is mandatory and makes use of the Service Start Event.</ref>			
customer_id	string	yes if no *contact_key*	The unique ID of the customer associated with the given service. Services with no *customer_id* are anonymous. In this case, the event should include a *contact_key*.
contact_key	string	yes if no *customer_id*	Key for later lookup of

ID	CV.WS.SRV.1		
			the service. This *contact_key* is mandatory if no customer ID is specified. For example, the application might store a PIN or ANI as the contact key when the service is first started. Later, if the customer is identified then the contact key is used to lookup the existing service record and associate it with the customer.
service_type	long or string	yes	The unique ID associated with the type of service, typically the DB ID of a value in the Service Type Business Attribute<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>.
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business" />application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (Genesys DB ID) for the application issuing the service event, such as a GVP VoiceXML

ID	CV.WS.SRV.1		
			application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (such as GVP, Agent Desktop, or Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given service, for instance, e-mail, voice, chat, etc.
est_duration	integer	no	The estimated service duration in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension n> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extensions, where *<extension n>* corresponds to the unique name of the service extension. Your

ID	CV.WS.SRV.1		
			application can specify as many extensions as needed as long as they were previously defined with the Create Service Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: **${base_uri}**/service/**${service_id}** where: • ${base_uri} is the URI of the created service. • ${service_id} is the service ID.
Body	{"service_id": **${service_id}**} where: • ${service_id} is the service ID.

Example

Operation

```
POST /services/start
{
  "timestamp": "2009-05-12T12:05:12.145Z",
  "interaction_id": "123ABCAADFJ1259ACF",
  "application_type":400,
  "application_id":40,
  "est_duration":60,
  "contact_key":"42",
  "service_type":100,
  "media_type": 1,
  "resource_id": 5005,
```

```
  "resource_type": 2,
  "disposition": 10,
  "coupon": {
          "coupon_name": "DISCOUNTCODE15"
  },
  "satisfaction": {
    "score": 85,
    "agentID": 2025
  },
  "relatedOffers": [
          {
           "offer_name":"VIP credit card black ed.",
           "type":9,
           "comments":"proposed to all client"
          },
          {
           "offer_name":"3 times payment GOLD",
           "type":4,
           "comments":"limited offer"
          }
          ,{
           "offer_name":"life insurance",
           "type":3,
           "comments":"healt check to be done before approval"
          }
  ]
}
```

Result

```
HTTP 201 Created
location: /services/12349857
{"service_id": 12349857}
```

Start State

	POST /services/${service_id}/states/start
	Available since: 8.0.100.00

Description

Starts a service state of a given type for a given service. UCS assigns auto-incrementing identifiers to the states, just like to services and tasks. States and tasks are assigned 32-bit identifiers.

Operation

ID	CV.WS.SRV.5		
Method	POST		
URL	/services/**${service_id}**/states/start		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The service ID is specified in the request URI.
Body: State Start Event<ref>This body contains fields from the State Start Event resource.</ref>			
state_type	long or string	yes	The unique ID associated with the state type. Typically, the DB ID of a value in the Service Type Business Attribute<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>. Enumerated values could be one of the following: 1. Customer identification 2. Service identification 3. Assign service agent 4. Waiting for service

ID	CV.WS.SRV.5		
			agent 5. Offering another service while waiting for agent 6. Offering callback 7. Callback pending 8. Delivering service 9. Waiting for customer input 10. Offering another service while delivering service
previous_state_id	integer	no	The ID of the previous state.
session_id	string	no	The ID of the related session; for instance, the orchestration session or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (Genesys DB ID) for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of

ID	CV.WS.SRV.5		
			resource which provides the service (such as GVP, Agent Desktop, or Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given state; for instance, e-mail, voice, or chat.
est_duration	integer	no	The estimated state duration in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: **${base_uri}**/service/**${service_id}**/states/**${state_id}** where: • ${base_uri} is the URI of the created service. • ${service_id} is the service ID. • ${state_id} is the ID of the created state.
Body	{"state_id": **${state_id}**} where: • ${state_id} is the stateID.

Example

Operation

```
POST /services/12345/states/start
{
 "interaction_id":"51",
 "application_type":400,
 "application_id":40,
 "resource_type":200,
 "resource_id":20,
 "media_type":2,
 "est_duration":60,
 "state_type":100,
 "Feedback":
  {
   "FeedbackType":"survey",
   "rating":7,
   "notes":"warm welcome at frontdesk, thanks for the nice trip"
  },
 "Satisfaction": [
  {
   "rating":2,
   "pertinence":8,
   "usefull":true,
   "place":"Terranova mexico resort"
  },
  {
```

```
   "rating":8,
   "pertinence":4,
   "usefull":false,
   "place":"Fancy resort Paris"
  }
  ]
}
```

Result

```
 {"state_id": 1258276}
```

Start Task

	POST /services/${service_id}/tasks/start
	Available since: 8.0.100.00

Description

This operation starts a task of a given type for the given service. The type is one of the enumerated values recorded for the corresponding business attribute. See Business Attributes in Context Services for further details. UCS assigns auto-incrementing identifiers to states, services, and tasks. For states and tasks, the assigned identifiers are 32-bit integers; for services the identifier is a 64-bit integer.

Operation

ID	CV.WS.SRV.8		
Method	POST		
URL	/services/**${service_id}**/tasks/start		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The unique 64-bit ID of the service.
Body: Task Start Event<ref>This body contains fields from the Task Start Event resources.</ref>			
state_id	integer	no	The 32-bit integer ID of the state. See State.
task_type	long or string	yes	The unique ID associated with the type of task, typically the DB ID of a value in a Business Attribute representing customer service tasks for the given organization<ref name="business"> Refer to Configuration Options for more details on Business Attribute mapping.</ref>.
session_id	string	no	The ID of the related session, for instance, the orchestration

ID	CV.WS.SRV.8		
			session or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (Genesys DB ID) for the application issuing the event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (such as GVP, Agent Desktop, or Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given task; for instance, e-mail, voice, or chat.
est_duration	integer	no	The estimated task

ID	CV.WS.SRV.8		
			duration in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension. Your application can add as many task extensions as needed, as long as you created corresponding Extension Schema with the Create Task Extension Schema operation.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: **${base_uri}**/service/**${service_id}**/tasks/**${task_id}** where: • ${base_uri} is the URI of the created service. • ${service_id} is the service ID. • ${task_id} is the created ID for the new task.
Body	{"task_id": **${task_id}**}

HTTP code	201
	where: • **${task_id}** is the task ID.

Example

The following example prerequisites are the creation of an Extension Schema for the "Survey" single-valued extension and one for the "Proposal" multi-valued extension. **Operation**

```
POST /services/21456878/tasks/start
{
 "interaction_id":42,
 "est_duration":460,
 "state_id":24,
 "task_type":customer info,
 "Survey":
  {
   "url":"http://ourServer/storage/userAnswers",
   "question1":7,
   "question2":true,
   "question3":"will be better with cable tv and on-demand video"
  },
  "Proposal": [
          {
           "car type":"cabriolet",
           "price":25 000,
           "seats":2,
           "comments":"200 cv, hardtop"
          },
          {
           "car type":"S.U.V.",
           "price":70 000,
           "seats":8,
           "comments":"4wd, leather seats"
          }
  ]
}
```

Result

```
 {"task_id": 15928}
```

Update Service Extension

	PUT /services/${service_id}/extensions/${ext_name}

Description

This operation replaces the extension value with a new extension value. The former value of the extension is lost. This operation supports the update of multi-valued extensions.

Operation

Updates a service's extension value

ID	CV.WS.SRV.4		
Method	PUT		
URL	/services/**${service_id}**/extensions/**${ext_name}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The ID of the service.
${ext_name}	string	yes	The name of the extension
Body			
<attribute n> or <attribute n>[]	value	yes	Attributes for single-valued extensions, or array of attributes for multi-valued extensions (previously defined through the Create Service Extension Schema operation).

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK

Example

Operation
The following operation updates the single-valued extension "score".

```
PUT /services/8389/extensions/score
{
   "score": 85,
   "agentID": 2025
}
```

Operation
The following operation updates the multi-valued extension "relatedOffers".

```
PUT /services/8389/extensions/relatedOffers
[
   {
     "offer_name":"VIP credit card black ed.",
     "type":9,
     "comments":"proposed to all client"
   },
   {
     "offer_name":"3 times payment GOLD",
     "type":4,
     "comments":"limited offer"
   }
  ,{
     "offer_name":"life insurance",
     "type":3,
     "comments":"health check to be done before approval"
   }
]
```

Result

```
200 OK
```

Update Record In Service Extension

	PUT /services/${service_id}/extensions/${ext_name}/by/unique
	Available since: 8.0.200.00

Description

Updates the content of a single record in a multi-valued extension. The attributes which are part of the 'unique' list specified at the Extension Schema creation are passed in the body and are used to find the correct record to update. This update cannot change the values of the attributes which are part of the 'unique' list specified at the Extension Schema creation. 'Unique' attributes identify which record should be updated.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.19		
Method	PUT		
URL	/services/**${service_id}**/extensions/**${ext_name}**/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	string	yes	The service ID.
${ext_name}	string	yes	The name of the Service Extension to update.
Body<ref>Record attributes for the Service Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime • currency	yes	Individual attributes defined in the Service Extension Schema. <attribute n> corresponds to the attribute name defined in the schema. • If the attribute is unique, its value is used to select the record to update. It is not updated.

ID	CV.WS.SRV.19		
	• boolean		• If the attribute is not unique, its value is updated.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation updates the "3 times payment GOLD" record of the extension *relatedOffers* in the service 8389.

```
PUT /services/8389/extensions/relatedOffers/by/unique
{
   "offer_name":"3 times payment GOLD",
   "type":4,
   "comments":"extented offer time"
}
```

Result

```
204 No Content
```

Update Record In State Extension

	PUT /services/${service_id}/state/${state_id}extensions/${ext_unique
	Available since: 8.0.200.00

Description

Updates the content of a single record in a multi-valued extension for a given state of a given service. The attributes which are part of the 'unique' list specified at the Extension Schema creation are passed in the body and are used to find the correct record to update. This update cannot change the values of the attributes which are part of the 'unique' list specified at the Extension Schema creation. 'Unique' attributes identify which record should be updated.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.20		
Method	PUT		
URL	/services/**${service_id}**/states/**${state_id}**/extensions/**${ext_name}**/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	long	yes	The service ID.
${state_id}	integer	yes	The state ID.
${ext_name}	string	yes	The name of the State Extension to update.
Body<ref>Record attributes for the State Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime	yes	Individual attributes defined in the State Extension Schema. <attribute n> corresponds to the attribute name defined in the schema. • If the attribute is unique, its value is

ID	CV.WS.SRV.20		
	• currency • boolean		used to select the record to update. It is not updated. • If the attribute is not unique, its value is updated.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation updates the 'Tons of Hill, Paris' record of extension relatedOffers in service 8389.

```
PUT /services/8389/states/1/extensions/Satisfaction/by/unique
{
   "rating":2,
   "pertinence":8,
   "usefull":true,
   "place":"Tons of Hill, Paris"
  }
```

Result

```
204 No Content
```

Update Record In Task Extension

	PUT /services/${service_id}/task/${task_id}extensions/${ext_na unique
	Available since: 8.0.200.00

Description

Updates the content of a single record in a multi-valued extension for a given state of a given service. The attributes which are part of the 'unique' list specified at the Extension Schema creation are in the body and are used to find the correct record to update. This update cannot change the values of the attributes which are part of the 'unique' list specified at the Extension Schema creation. 'Unique' attributes identify which record should be updated.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.SRV.21		
Method	PUT		
URL	/services/**${service_id}**/tasks/**${state_id}**/extensions/**${ext_name}**/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	long	yes	The service ID.
${task_id}	integer	yes	The state ID.
${ext_name}	string	yes	The name of the Task Extension to update.
Body<ref>Record attributes for the Task Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime	yes	Individual attributes defined in the Task Extension Schema. <attribute n> corresponds to the attribute name defined in the schema.

ID	CV.WS.SRV.21		
	• currency • boolean		

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation updates the "cabriolet" record of the *Proposal* extension in the task 1 of service 8389.

```
PUT /services/8389/tasks/1/extensions/Proposal/by/unique
{
    "car type":"cabriolet",
    "price":25 000,
    "seats":3,
    "comments":"250 cv, hardtop"
}
```

Result

```
204 No Content
```

Update State Extension

	PUT /services/${service_id}/states/${state_id}/extensions/${ex
	Available since: 8.0.200.00

Description

This operation replaces the extension value with a new extension value. The former value of the extension is lost.

Operation

Updates a state's extension value

ID	CV.WS.SRV.17		
Method	PUT		
URL	/services/**${service_id}**/states/**${state_id}**/extensions/**${ext_name}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The ID of the service.
${state_id}	integer	yes	The ID of the state.
${ext_name}	string	yes	The name of the extension.
Body			
<attribute n>	value	yes	Values for one or more extension attributes (previously defined through the Create State Extension Schema operation).

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	OK

Example

Operation: Update a multi-valued extension

```
PUT /services/8389/states/1/extensions/Satisfaction
[
 {
 "rating":2,
 "pertinence":8,
 "usefull":true,
 "place":"Terranova mexico resort"
},
{
 "rating":8,
 "pertinence":4,
 "usefull":false,
 "place":"Fancy resort Paris"
}
]
```

Result

```
200 OK
```

Operation: Update a single-valued extension

```
PUT /services/8389/states/1/extensions/Feedback
{
   "FeedbackType":"survey",
   "rating":7,
   "notes":"warm welcome at frontdesk, thanks for the nice trip"
}
```

Result

```
200 OK
```

Update Task Extension

	PUT /services/${service_id}/tasks/${task_id}/extensions/${exte
	Available since: 8.0.200.00

Description

This operation replaces the extension value with a new extension value. The former value of the extension is lost.

Operation

Updates a state's extension value

ID	CV.WS.SRV.18		
Method	PUT		
URL	/services/**${service_id}**/tasks/**${task_id}**/extensions/**${extension_name}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${service_id}	integer	yes	The ID of the service.
${task_id}	integer	yes	The ID of the task.
${extension_name}	string	yes	The name of the extension
Body			
<attribute n>	• string • integer • long • double • date • datetime • currency • boolean	yes	Values for one or more extension attributes (previously defined through the Create Task Extension Schema operation).

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	OK

Example

Operation
Update a multi-valued extension.

```
PUT /services/8389/tasks/42/extensions/Proposal
[
  {
   "car type":"cabriolet",
  "price":25 000,
  "seats":2,
  "comments":"200 cv, hardtop"
},
{
 "car type":"S.U.V.",
 "price":70 000,
 "seats":8,
 "comments":"4wd, leather seats"
}
]
```

Result

```
200 OK
```

Operation
Update a single-valued extension.

```
PUT /services/8389/tasks/42/extensions/Survey
{
  "url":"http://ourServer/storage/userAnswers",
  "question1":7,
  "question2":true,
  "question3":"will be better with cable tv and on-demand video"
 }
```

Result

```
200 OK
```

Service Resources

Purpose

Groups the Service resources.

End Event List

	Purpose: Lists the End Event resources used to describe the end of a service, state, or task.
	Available since: 8.0.100.00

Service End Event

Field	Type	Mandatory	Description
service_id	integer	yes	The unique 64-bit ID of the given service.
session_id	string	no	The ID of the related session (for instance, the orchestration session or any other business session). Limited to 32 characters.
interaction_id	string	no	The ID of the interaction at the service event's creation. Limited to 50 characters.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business">fckLRRefer to Configuration Options for more details on *Business Attribute* mapping.</ref> of application which issued the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The Genesys DB ID for the application which issues the service event, such as a GVP VoiceXML application, Orchestration SCXML application, and so on.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business" /> of

Field	Type	Mandatory	Description
			resource providing the service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • The Genesys DB ID of a specific GVP or orchestration platform • The DB ID of a given agent, according to the context
media_type	long or string	no	The media type<ref name="business" /> applicable to the given service, such as e-mail, voice, chat, and so on.
disposition	long or string	no	The unique ID for the business disposition<ref name="business" /> assigned to the given service. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization's business dispositions.
disposition_desc	string	no	A text description that provides additional information about the business disposition. Limited to 64 characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server

Field	Type	Mandatory	Description
			adds it when the service event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extension. Your application can add as many service extensions as needed, as long as you have created a corresponding Extension Schema for each one by using the Create Service Extension Schema operation.

<references />

Related Operations

- Complete Service

State End Event

Field	Type	Mandatory	Description
service_id	integer	yes	The unique 64-bit ID of the given service.
state_id	integer	yes	The 32-bit ID of the given state.
session_id	string	no	The ID of the related session (for instance, the orchestration session or any other business session). Limited to 32 characters.
interaction_id	string	no	The ID of the interaction at the service event's creation. Limited to 50 characters.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business">fckLRRefer to Configuration Options for more details on *Business Attribute* mapping.</ref> of application which issued the state event. May be

Field	Type	Mandatory	Description
			used to group related applications, potentially across resource types.
application_id	integer	no	The Genesys DB ID for the application which issues the state event, such as a GVP VoiceXML application, Orchestration SCXML application, and so on.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business" /> of resource providing the related service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the related service. For instance: • The Genesys DB ID of a specific GVP or orchestration platform • The DB ID of a given agent, according to the context
media_type	long or string	no	The media type<ref name="business" /> applicable to the given state, such as e-mail, voice, chat, and so on.
disposition	long or string	no	The unique ID for the business disposition<ref name="business" /> assigned to the given state. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization’s business dispositions.
disposition_desc	string	no	A text description that provides additional information about the business disposition. Limited to 64

Field	Type	Mandatory	Description
			characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server adds it when the service event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you have created a corresponding Extension Schema for each one by using the Create State Extension Schema operation.

<references />

Related Operations

- Complete State

Task End Event

Field	Type	Mandatory	Description
service_id	integer	yes	The unique 64-bit ID of the given service.
state_id	integer	no	The ID of the related state. If the task is optional, you have no obligation to associate it with a state. For instance, you can associate this task with a higher service level.

Field	Type	Mandatory	Description
task_id	integer	yes	The ID of the given task.
session_id	string	no	The ID of the related session (for instance, the orchestration session or any other business session). Limited to 32 characters.
interaction_id	string	no	The ID of the interaction at the service event's creation. Limited to 50 characters.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business">fckLRRefer to Configuration Options for more details on *Business Attribute* mapping.</ref> of application which issued the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The Genesys DB ID for the application which issues the service event, such as a GVP VoiceXML application, Orchestration SCXML application, and so on.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business" /> of resource providing the service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • The Genesys DB ID of a specific GVP or orchestration platform • The DB ID of a given agent, according to

Field	Type	Mandatory	Description
			the context
media_type	long or string	no	The media type<ref name="business" /> applicable to the given task, such as e-mail, voice, chat, and so on.
disposition	long or string	no	The unique ID for the business disposition<ref name="business" /> assigned to the given task. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization's business dispositions.
disposition_desc	string	no	A text description that provides additional information about the business disposition. Limited to 64 characters.
timestamp	date/time	no	The UTC time at which the service event was raised, with a precision of milliseconds, using the ISO 8601<ref name="iso8601">See http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server adds it when the service event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension. Your application can add as many Task extensions as needed, as long as you have created a corresponding Extension Schema for each one by using the Create Task Extension Schema operation.

<references />

Related Operations

- Complete Task

Extension

	Purpose: Describes an extension. • For Customers Available since: 8.0.100.00 • For services, states, or tasks Available since: 8.0.200.00

Description

Each extension fulfills the contract of an Extension Schema that your application created in anterior operations. Your application can create schemas for the following extensions:

- Available since: 8.0.100.00
  - Profile extensions, created through the Create Profile Extension Schema operation, which provides extensions for profiles.
- Available since: 8.0.200.00
  - Service extensions, created through the Create Service Extension Schema, which provides extensions at the service level.
  - State Extension, created through the Create State Extension Schema operation, which provides extensions at the service level.
  - Task extension, created through the Create Task Extension Schema operation, which provides extensions at the task level.

Once the extension schema is defined, your application can create and add extension records when dealing with standard profile, services, states, and task operations. For instance, if you start a task, you can specify extension records in your operation body as shown in the example section.

Profile Extension

Profile Extension

Field	Type	Mandatory	Description
customer_id	string	yes if not nested in a profile	The unique ID of the given customer. This field is mandatory if the extension is **not** nested inside a Customer Profile.

Field	Type	Mandatory	Description
name	string	yes	The unique, case-insensitive name of the extension. • Starts with a letter to which you can append letters, numbers, or underscores. • Restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on index creation.)
<attribute n>	• string • integer • long • double • date • datetime • currency • boolean	yes	Individual attributes, defined in the Extension Schema. <attribute n> corresponds to the attribute name defined in the schema.

Service/State/Task Extension

Service/State/Task Extension

Field	Type	Mandatory	Description
service_id	integer	yes	The unique ID of the service associated with the extension.
state_id	integer	yes if state extension	The unique ID of the state associated with the extension. For task and service extensions, you do not need to specify this id.
task_id	integer	yes if task extension	The unique ID of the task associated with the extension. For state and service extensions, you do not need to specify

Field	Type	Mandatory	Description
			this id.
name	string<ref name="string">Do not use Unauthorized Strings as attribute values.</ref>	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on creation of index).
<attribute n>	• boolean • string<ref name= "string" /> • integer • long • double • date • datetime • currency	yes	Individual attributes, as defined in the extension schema. <attribute n> corresponds to the attribute name defined in the schema. Optional attributes do not need to be specified.

<references />

State Transition Event

	Purpose: State Transition event used to describe the transition of a state.
	Available since: 8.0.100.00

Description

Combines state event resources to describe the completion of a given state and the start of a new state in a single request, rather than issuing separate start and end requests.

Resource

When your application deals with this resource, parts of the *from* and *to* fields are shared. See below.

State Transition Event

Field	Type	Mandatory	Description
service_id	integer	yes	The unique ID of the related service.
from	hash	yes	The hash of the following fields describing the state from which the service is transitioning: • *state_id (mandatory)*: The 32-bit ID of the state to complete. • disposition (optional): The unique ID for the business disposition assigned to the given state. For instance, this can be the DB ID of a Business Attribute used to enumerate a given organization's business dispositions<ref

Field	Type	Mandatory	Description
			name="business">. Refer to Configuration Options for more details on Business Attribute mapping.</ref>. • disposition_desc (optional): The text description which provides additional context on the business disposition. Limited to 64 characters.
to	hash	yes	The hash of the following fields, which describe the state into which the service is transitioning: • **state_type (mandatory)**: The unique ID associated with the type of service, typically the DB ID of a value in the Service Type Business Attribute<ref name="business"/>. • est_duration (optional): The estimated service duration in seconds.
The following fields apply both to the state to complete and the state to start:			
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref

Field	Type	Mandatory	Description
			name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (e.g. Genesys DB ID) for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (for example, GVP, Agent Desktop, or Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given service/state/task, for instance, e-mail, voice, chat, etc.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`.

Field	Type	Mandatory	Description
			If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you have created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Related Operations

- Perform State Transition

Service

	Purpose: Describes the Service resource.
	Available since: 8.0.100.00

Description

Provides a consolidated view of the service information.

Services are customer commitments defined by the business application (IVR, Orchestration, Agent, etc.) which interacts with the customer. Each service potentially spans multiple interactions over a variety of media channels. Each service can be divided into States which include a list of Tasks. The service is completed once the states and their tasks are completed. Your application is responsible for state transitions and task management. The service resources (creation, start, or end) are managed through Service Operations, which update the service information through service events (Start Event and End Events).

Resource

This table lists the resource fields that your application can deal with through operations. According to the operation, your application may not use the totality of the resource.

Field	Type	Mandatory	Description
service_type	long or string	yes	The unique ID associated with the type of service, typically, the DB ID of a value in the Service Type Business Attribute. Refer to Configuration Options for more details about *Business Attribute* mapping.
service_id	integer	yes	The service's unique ID.<ref name="toplevel">Only top-level objects include this field. For instance, if your application retrieves a service which includes nested states and tasks, only the top-level service

Field	Type	Mandatory	Description
			object contains this ID.</ref>
started	Service Start Event	yes	Start event (see Start Service). This event should not include the "est_duration" field, if it already exists at the service level.
completed	Service End Event	no	The related end event if the service is completed (see Complete Service). This field should not include the "disposition" and "disposition_desc" fields if they already exist at the service level.
customer_id	string	no	The customer's unique ID<ref name="toplevel"/>. If the service has no customer ID, the service is Anonymous Service.
est_duration	integer	no	The estimated time for completing the service, in seconds.
duration	integer	no	The total duration in milliseconds. Only relevant for completed services.
disposition	long or string	no	ID of the Business disposition ID. Only relevant for completed services.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
active_states	State[]	no	The array of the started states which are not completed.
completed_states	State[]	no	The array of completed states through which the service has made transitions.
active_tasks	Task[]	no	The array of the started tasks which are not complete.

Field	Type	Mandatory	Description
completed_tasks	Task[]	no	The array of the tasks which are complete as part of the service delivery.
interactions	Interaction[]	no	The array of the interactions which are associated with this service.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extension. Your application can add as many service extensions as needed, as long as you have created the corresponding Extension Schema with the Create Service Extension Schema operation.

<references />

Related Operations

- Query Services
- Query Service by ID

State

	Purpose: Describes a State Resource.

Description

Describes one of the service states.

Throughout their lifecycle, services go through states such as:

- customer identification
- agent assignment
- service delivery

The exact sequence of states depends on the steps which your customer service application implements (for example, IVR, orchestration, or agent application). Your application is responsible for starting and completing states, but also for managing state transitions. Each State can also contain nested Task resources (zero or more) that your application should manage as well.

Resource

This table lists the resource fields that your application can deal with through operations. According to the operation, your application may not use the totality of the resource.

Field	Type	Mandatory	Description
state_id	integer	yes	The 32-bit integer ID of the state, assigned by the UCS when the state is started.
state_type	long or string	yes	The unique ID associated with the state type. For instance, this ID can be the DB ID of a value in the Service Type Business Attribute. Enumerated values could be one of the following: 1. Customer identification

Field	Type	Mandatory	Description
			2. Service identification 3. Assign service agent 4. Waiting for service agent 5. Offering another service while waiting for agent 6. Offering callback 7. Callback pending 8. Delivering service 9. Waiting for customer input 10. Offering another service while delivering service Refer to Configuration Options for more details on *Business Attribute* mapping.
service_id	integer	yes/no	The service's unique ID if the state is not part of a top-level service resource which includes a service ID.<ref name="toplevel">Only top-level representations include this field. For instance, if your application retrieves a service which includes nested states and tasks, only the top-level service resource contains this ID.</ref>
started	State Start Event	yes	Related start event. This field should not repeat the "est_duration" field, if it already exists.
completed	State End Event	no	The related end event if the state is completed. This field should not repeat the "disposition" and "disposition_desc" fields if they already exist.

Field	Type	Mandatory	Description
est_duration	integer	no	The estimated time for completing the state, in seconds.
duration	integer	no	The total duration, in milliseconds. Only relevant for completed states.
disposition	long or string	no	ID of the Business disposition ID. Only relevant for completed states.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
active_tasks	Task[]	no	The array of tasks that are started but not yet complete.
completed_tasks	Task[]	no	The array of Task that are complete.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you have created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Start Event List

	Purpose: Lists the Start Event resources used to describe the start of a service, state, or task.
	Available since: 8.0.100.00

Service Start Event

Description This resource describes information useful to manage the start or the creation of a service.

Field	Type	Mandatory	Description
customer_id	string	yes if no *contact_key*	The unique ID of the customer associated with the given service. Services with no customer_id are anonymous. In this case, the event should include a contact_key.
contact_key	string	yes if no *customer_id*	Key for later lookup of the service. This contact_key is mandatory if no customer ID is specified. For example, the application might store a PIN, or the ANI, as the contact key when the service is first started. Later, if the customer is identified, the contact key is used to lookup the existing service record and associate it with the customer.
service_type	long or string	yes	The unique ID associated with the type of service, typically the DB ID of a value in the Service Type Business Attribute<ref name="business">. Refer to Configuration Options for more details on Business Attribute mapping.</ref>.

Field	Type	Mandatory	Description
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business" />application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID, for example, Genesys DB ID, for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/>

Field	Type	Mandatory	Description
			applicable to the given service, for instance, e-mail, voice, chat, etc.
est_duration	integer	no	The estimated service duration, in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Service extension. Your application can add as many service extensions as needed, as long as you have already created corresponding Extension Schema with the Create Service Extension Schema operation.

<references />

Related Operations

- Start Service
- Associate Service

State Start Event

Description This resource describes the start of the service state.

Field	Type	Mandatory	Description
service_id	integer	yes	The unique 64-bit ID of the service.
state_type	long or string	yes	The unique ID associated with the

Field	Type	Mandatory	Description
			state type, typically, the DB ID of a value in the Service Type Business Attribute<ref name="business">. Refer to Configuration Options for more details on Business Attribute mapping.</ref>. Enumerated values may be from the following: 1. Customer identification 2. Service identification 3. Assign service agent 4. Waiting for service agent 5. Offering another service while waiting for agent 6. Offering callback 7. Callback pending 8. Delivering service 9. Waiting for customer input 10. Offering another service while delivering service
previous_state_id	integer	no	The ID of the previous state.
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the

Field	Type	Mandatory	Description
			service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID, for example, Genesys DB ID, for the application issuing the service event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service, for instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.
media_type	long or string	no	The media type<ref name="business"/> applicable to the given state, for instance, e-mail, voice, chat, etc.
est_duration	integer	no	The estimated state duration in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does

Field	Type	Mandatory	Description
			not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	State extension. Your application can add as many state extensions as needed, as long as you have already created corresponding Extension Schema with the Create State Extension Schema operation.

<references />

Related Operations

- Complete State
- Perform State Transition
- Query State by ID
- Query States
- Start State

Task Start Event

Description: Describes the start of a Task.

Field	Type	Mandatory	Description
service_id	integer	yes	The unique 64-bit ID of the service.
state_id	integer	no	The 32-bit integer ID of the state. See State.
task_type	long or string	yes	The unique ID associated with the type of task, typically the DB ID of a value in a Business Attribute representing customer service tasks for the given organization<ref name="business"> Refer to Configuration Options for more details on Business

Field	Type	Mandatory	Description
			Attribute mapping.</ref>.
session_id	string	no	The ID of the related session, for instance, the orchestration session, or any other business session.
interaction_id	string	no	The ID of the related Genesys interaction. This ID can be used by other Genesys reporting products such as Stat Server, URS, Composer, and GVP.
application_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of application issuing the service event. May be used to group related applications, potentially across resource types.
application_id	integer	no	The unique ID (e.g. Genesys DB ID) for the application issuing the event, such as a GVP VoiceXML application or an Orchestration SCXML application.
resource_type	long or string	no	The unique ID associated with the type or class<ref name="business"/> of resource which provides the service (for example, GVP, Agent Desktop, Orchestration).
resource_id	integer	no	The unique DB ID for the specific resource which provides the service. For instance: • the Genesys DB ID of a specific GVP or orchestration platform. • the DB ID of a given agent, according to the context.

Field	Type	Mandatory	Description
media_type	long or string	no	The media type<ref name="business"/> applicable to the given task, for instance, e-mail, voice, chat, etc.
est_duration	integer	no	The estimated task duration, in seconds.
timestamp	date/time	no	The UTC time at which the event was raised, with a precision of milliseconds, using the ISO 8601<ref name="ISO 8601">http://en.wikipedia.org/wiki/ISO_8601</ref> representation : `[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. If the application does not specify this timestamp, the server does it when the event is processed.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension. Your application can add as many task extensions as needed, as long as you have created corresponding Extension Schema with the Create Task Extension Schema operation.

Task

	Purpose: Represents a Task.
	Available since: 8.0.100.00

Description

Describes one task.

Tasks allow your application to divide a State into a collection of tasks, each with its own disposition. Zero or more tasks can be executed as part of the Service Delivery state. Your application is responsible for managing these tasks. If the state is optional, your application can associate the task with a service instead of a state.

Resource

This table lists the resource fields that your application can deal with through operations. According to the operation, your application may not use the totality of the resource.

Field	Type	Mandatory	Description
state_id	integer	no	The 32-bit integer ID for the state associated with the task.
task_id	integer	no	The 32-bit integer ID of the task, assigned by the UCS when the task is started.
service_id	integer	no	The service's unique ID if the task is not part of a top-level state or service resource which includes a service ID.<ref name="toplevel">Only top-level representations include this field. For instance, if your application retrieves a service which includes nested states and tasks, only the top-level service

Field	Type	Mandatory	Description
			representations contain this ID.</ref>
started	Task Start Event	yes	Related start event. This event should not include the "est_duration" field, if it already exists at the task level.
completed	Task End Event	no	The related end event if this task is completed. This event should not repeat the "disposition" and "disposition_desc" fields if they already exist at the task level.
customer_id	string	no	The customer's unique ID<ref name="toplevel" />.
est_duration	integer	no	The estimated time for completing the item, in seconds.
duration	integer	no	The total duration, in milliseconds. Only relevant for completed tasks.
disposition	long or string	no	ID or name of the Business Disposition ID associated with the task. See Business Attributes for further details. Only relevant for completed tasks.
disposition_desc	string	no	The reason for assigning the business disposition with the service element. Limited to 256 characters.
<extension name> Supported since 8.0.2	Extension (single-valued) or Extension[] (multi-valued)	no	Task extension. Your application can add as many task extensions as needed, as long as you have created corresponding Extension Schema with the Create Task Extension Schema operation.

<references />

Related Operations

- Complete Task
- Query Task by ID
- Query Tasks
- Start Task

Profile Operations

- POST /profiles
- GET /profiles
- POST /profiles/${customer id}/extensions
- GET /profiles/${customer id}
- PUT /profiles/${customer id}
- Since 8.0.200.00:
  - PUT /profiles/${customer id}/extensions/${ext name}/by/unique
  - PUT /profiles/${customer id}/extensions/${ext name}/delete/by/unique
- Since 8.1.000.10:
  - POST /profiles/import
- Since 8.1.200.xx:
  - DELETE /profiles/${customer_id}
- Since 8.1.300.xx:
  - PUT profiles/${customer_id}/merge/${src_id}/

Bulk Profile Import

	POST /profiles/import **Available since:** 8.1.000.10

Description

Imports several customer profiles in a single operation. Your application should use this operation if you need to import a large number of profiles into the UCS database.

- The operation supports the import of `.csv` files if the file is already uploaded on UCS' local file system.
- The operation manages:
  - Database consistency (data type conversion according to metadata).
  - Propagation of identification keys.
  - Backward compatibility with the APIs.

You cannot submit the profiles as a JSON array.

The operation's description of the supported format must be compliant with RFC4180:

- The first record must contain the column name for each field.
- There is one record per line and each line terminates with a line feed.
- Commas separate fields by default; if you wish to use another separator, set up the delimiter field of your operation.
- Spaces are considered as part of a field and will not be ignored (trimmed).

This operation returns the list of contacts which were not imported, with descriptions of the errors or ambiguities which prevented the server from adding them to the database.

Roles

If role-based access control is enabled in your environment, your application will need to be assigned the following roles in order to successfully perform a bulk import:

- UCS.Customer.executeBulkImport (Perform Bulk Import)
- UCS.Customer.createProfile (Create Customer Profile)
- UCS.Customer.createProfileExtension (Create Profile Extension)

Your application must be assigned the `UCS.Customer.createProfileExtension` role only if profile extensions will be created by the bulk import.

Operation

Prerequisites:

- The Customer Profile Schema exists.
- The UCS<ref name="ucs">Universal Contact Server</ref> is in maintenance mode.

Your operation's header must comply with the following requirements:

- Content-Type is set to 'multipart/form-data'.
- Request is composed of a first part of type 'application/json', named 'description' and containing the JSON body as described below.

ID	CV.WS.PROF.IMPORT.1		
Method	POST		
URL	/profiles/import		
Parameter	**Type**	**Mandatory**	**Description**
Body			
format	"JSON" **Not supported** "CSV"	yes	Defines the format of the submitted data.
profiles	string	no	File path, for instance 'C:/profiles.csv'. Your application is responsible for uploading the file on the server's local filesystem.
id-key	string	no	A valid identification key. If specified, the server checks to see whether the profile already exists prior to the insertion. If the profile does exist in the database, the record containing the duplicate profile is skipped.
delimiter	char or "," by default	no	Defines the character used to separate the submitted profiles.
batch-size	integer	no	The number of records

ID	CV.WS.PROF.IMPORT.1		
	64 by default		to insert or update before a database COMMIT is issued (default to 64).
max-errors	integer	no	The number of errors to allow before ending the bulk load. An error can indicate that a given record was not parsed correctly, or that an ambiguity exists with an existing customer record (see response below).
encoding	string	no	Encoding character to use for file import. Default is UTF-8<ref> UTF-8 without BOM, see related bug http://bugs.sun.com/view_bug.do;jsessionid=fdff2c415891f7
start-from-index	integer	no	The record number from which to start importing.
create-log	boolean	no	This flag turns extended import results logging on or off. The log data is written to an output file that has the same name as the input file, with an extension of `.log` appended. The default value is `false`.

Response

The Context Management Service API answers every request with an HTTP code. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

HTTP code	200
HTTP message	OK
Body	{"errors": [{"index": '''$index_id1''', "reason": '''$reason_1'''}, {"index": '''$index_id2''', "reason": '''$reason_2'''}], "ambiguous": [{"index": '''$index_id2''', "reason":

HTTP code	200
	'''$reason_3'''}] } where: • **${index_idn}** is the index of the imported record in the source array. • "Ambiguous" lists the source data items which were ignored because of *Unresolved identity*: The given identification key is not uniquely identifying the customer; for instance, attempting to use "Name" to identify the customer, but there are more than one customer with the same name. • "Ambiguous" lists the source data items which were invalid for some reason. The reason description is one of the errors which can occur when inserting a profile (see Create Customer Profile).

Examples

Raw HTTP

Operation

```
POST /cv/profiles/import HTTP/1.1
Keep-Alive: 10
Content-Length: 320
Content-Type: multipart/form-data; boundary=zaHf8xb3LfVOXZ0o2KUyqWD4Zlfumh0R9uyMN
Host: localhost:9090
Connection: Keep-Alive
Expect: 100-Continue

--zaHf8xb3LfVOXZ0o2KUyqWD4Zlfumh0R9uyMN
Content-Disposition: form-data; name="description"
Content-Type: application/json; charset=UTF-8
Content-Transfer-Encoding: 8bit

{"delimiter":",","profiles":"D:/successContentProfiles.csv","format":"csv","max-errors":20}
--zaHf8xb3LfVOXZ0o2KUyqWD4Zlfumh0R9uyMN--
```

Here is a sample `.csv` file containing 10 records:

```
FirstName, LastName, PhoneNumber[2], PhoneNumber[1], EmailAddress, Address.AddressType,
Address.PostCode,Address.Address, Address.County, Address.City, Address.Country
 kertuleordi,gusch,1-355-530-4087,1-204-489-2084,kertuleordi@gusch.net,7,4703,"25,
et malesuada vulputate Proin",Pomona,Germany,Fullerton
 danye,hna,1-551-483-3810,1-455-658-9039,danye@hna.net,6,5101,"blabl \nDonec",Kailua,East
Timor,
```

```
Duluth
 almady,vanah,1-215-196-0375,1-632-596-1889,almady@vanah.net,0,4602,facilisis fermentum
lacus.,Irving,Cambodia,Jackson
 julaudricelsa,mosessaglowsket,1-363-017-4320,
1-600-591-2531,julaudricelsa@mosessaglowsket.net,
4,2644,id bibendum,Fort Lauderdale,Sudan,Orem
 clemara,gonebali,1-334-657-4832,1-226-703-2238,clemara@gonebali.net,6,6552,
aliquet rutrum nonummy pellentesque ac,Chula Vista,Jamaica,Waterbury
 lonni,mi,1-021-445-4631,1-716-116-8811,lonni@mi.net,7, 102,Duis vulputate lacus.
 felis eu eu et sagittis ut,Allentown,Russian Federation,North Charleston
 louignaelgarcecio,jecklettorneimermandetett,1-615-348-3144,1-161-040-
1742,louignaelgarcecio@jecklettorneimermandetett.net,1,6533,
nisl Suspendisse imperdiet,Jacksonville,Italy, Baltimore
 jantasie,jachils,1-324-661-2865,1-220-082-1681,jantasie@jachils.net,5,0171,neque lacus.
Donec mauris Donec facilisis metus. est,Columbia,
Central African Rep,Thousand Oaks
 hizellaighie,wollosch,1-207-086-4460,1-160-603-4275,hizellaighie@wollosch.net,
4,9653,Vestibulum ipsum eu sit,Albuquerque,Romania,Philadelphia
```

HTML Example

Operation

Consider the following code sample embedding the JSON request, which enables you to select the `.csv` file:

```
<html>
 <body>
 Insert JSON request in textbox below and select your CSV file.
 <form method="post" action="http://localhost:8080/cv/profiles/import">
 <br/>'application/json' import parameters:<br/>
 <TEXTAREA name="description" rows="10" cols="80">
{ profiles:"D:/myProfiles.csv", format:"csv", delimiter:",", batch-size:64, max-errors:20
}</TEXTAREA>
 <input type="submit" value="Send" />
 </form>
 </body>
 </html>
```

Result The service might return the following information:

```
{
  "errors": [
   {
    "index": 3,
    "reason": "Bad parameter 'Importing profile' reason : number of profile attributes
cannot be different than declared"
   },
   {
    "index": 4,
    "reason": "Bad parameter 'Invalid value' reason : \"no-integer\" is an invalid value for
type 'Integer'"
   }
  ]
  "ambiguous": [
   {
    "index": 12,
    "reason": "Some customer(s) are already matching this profile : idProfile1, idProfile2"
   },
```

```
   {
    "index": 14,
    "reason": "Some customer(s) are already matching this profile : idProfile15, idProfile232"
   }
  ]
}
```

Apache Example

The following code sample shows how to write an HTTP client for Apache which enables an import.

```
DefaultHttpClient client = new DefaultHttpClient();
HttpRequestRetryHandler retryHandler = new HttpRequestRetryHandler() {
 public boolean retryRequest(IOException exception, int executionCount, HttpContext context) {
 return false;
 }
};
client.setHttpRequestRetryHandler(retryHandler);



HttpPost httpPost = new HttpPost("http://localhost:9090/profiles/import");

JSONObject parameters = new JSONObject();
parameters.put("format", "csv");
parameters.put("profiles", "D:/myProfiles.csv");
parameters.put("delimiter", ",");
parameters.put("max-errors", 20);

MultipartEntity reqEntity = new MultipartEntity(HttpMultipartMode.STRICT);
reqEntity.addPart("description", new StringBody(parameters.toString(),
MediaType.APPLICATION_JSON.toString(), Charset.forName("UTF-8")));
httpPost.setEntity(reqEntity);

HttpResponse response = client.execute(httpPost);
HttpEntity resEntity = response.getEntity();

String res = EntityUtils.toString(resEntity, "UTF-8");

client.getConnectionManager().shutdown();
```

References

<references />

Create Customer Profile

	POST /profiles
	Available since: 8.0.100.00

Description

Creates a Customer Profile and returns the ID created by UCS. In 8.0.1, profiles are built on top of legacy UCS Contact Attributes. They are multi-valued and support the primary attributes. If an attribute has several values, the first value in the JSON array is the primary attribute and additional values are non-primary attributes. In 8.0.2, standard multi-valued extensions are available.

Operation

Prerequisites: The Customer Profile Schema exists.

ID	CV.WS.PROF.3		
Method	POST		
URL	/profiles		
Parameter	**Type**	**Mandatory**	**Description**
Body: Customer Profile<ref>See Customer Profile for more information about the Customer Profile.</ref>			
customer_id	string	yes	The unique ID of the given customer. Limited to 16 characters.
<attribute n>	• string • datetime	yes	Customer attributes, where **<attribute n>**is the attribute name<ref>See the configuration options for further details.</ref>. Attributes are not part of a separated array, see the example below. The date/time formats are ISO 8601<ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. Do not use Unauthorized Strings as attribute values.
<extension n>	Extension or Extension[]	no	Extensions, where

ID	CV.WS.PROF.3		
			<extension n> corresponds to the unique name of the profile extension resource. • For single-valued extensions, the extension's value is a single extension object (see Extension). • For multi-valued extensions, the extension's value is an array of zero or more Extensions.

<references />

Response

The Context Management Service API answers every request with an HTTP code. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: /profiles/**${customer_id}** where: • ${customer_id} is the created customer ID.
Body	{"customer_id": **${customer_id}**} where: • ${customer_id} is the customer ID.

Example

Operation The following operation would create the customer profile for "Bruce Banner," as well as three associated telephone records that are grouped in the "Phone" extension. The e-mail address

bruce.banner@marvelous.com is the primary e-mail attribute, while the other e-mail address is non-primary.

```
POST /profiles/
{
   "FirstName": "Bruce",
   "LastName": "Banner",
   "DOB": "1962-05-10",
   "EmailAddress": [
       "bruce.banner@marvelous.com",
       "b.banner@hulk.dom"
     ],
   "Phone": [
   {
      "PhoneType":0,
      "prefix":"+33",
      "PhoneNumber":"3145926535",
      "description":"family phone",
      "start_availabilty":"2009-12-18T18:30:00.000Z",
      "end_availabilty":"2009-12-18T21:40:00.000Z"
   },
   {
      "PhoneType":2,
      "prefix":"+33",
      "PhoneNumber":"6543210",
      "description":"businness calls only, no sales",
      "start_availabilty":"2009-12-18T09:30:00.000Z",
      "end_availabilty":"2009-12-18T17:45:00.000Z"
   },
   {
      "PhoneType":5,
      "prefix":"+33",
      "PhoneNumber":"951357456",
      "description":""
   }
  ]
}
```

Result

```
Content-Location        http://ucsserver.mycompany.com:8080/path/profiles/00038b5SCVPU0007
{"customer_id":"0004Va58A92T0017"}
```

Delete Customer Profile

	DELETE /profiles/${customer_id}
	Available since: 8.1.200.xx

Description

Deletes a customer profile and all it's extension records and identifications keys records. All associated services are also deleted. If those services records are used by states or tasks, they will also be deleted.

Operation

ID	CV.WS.PROF.8		
Method	DELETE		
URI	/profiles/${customer_id}		
Parameter	**Type**	**Mandatory**	**Description**
${customer_id}	string	yes	The unique ID of the given customer that is specified in the URI.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following tables show the correct response for a successful request, and the response if the customer cannot be deleted. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	Success

Response	
HTTP code	404
HTTP message	Not Found

Example

Operation The following operation deletes a particular customer:

```
DELETE /profiles/0004Va58A92T0018
```

Result

```
 200 Success
```

Delete Record From Profile Extension

	PUT /profiles/${customer_id}/extensions/${ext_name}/delete/by/unique
	Available since: 8.0.200.00

Description

Deletes a single record in a multi-valued extension. You must specify the list of 'unique' attributes which identify the record to delete. See Unique Attributes for further information.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.PROF.7		
Method	PUT		
URL	/profiles/${customer_id}/extensions/${ext_name}/delete/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The unique ID of the given customer. Limited to 16 characters.
${ext_name}	string	yes	The name of the Profile Extension to modify.
Body Unique attributes of the Profile Extension record.			
<attribute n>	• string • integer • long • double • date • datetime • currency • boolean	yes	<attribute n> corresponds to the attribute name defined 'unique' in the Extension Schema.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation updates the Phone record associated with the unique "PhoneNumber":"3145926535".

```
PUT /profiles/0000Sb5U97XE000Y/extensions/Phone/delete/by/unique
{"PhoneType":0, "prefix":"+33", "PhoneNumber":"3145926535",
"description":"family phone", "start_availability":"2009-12-18T18:30:00.000Z",
"end_availability":"2009-12-18T21:40:00.000Z"}
```

Result

```
204 No Content
```

Identify Customer

	GET /profiles
	Available since: 8.0.100.00

Description

Identifies a customer based on the Identification Key information submitted with the query. In addition to the identified customer ID, your application can retrieve profile and extension information based on the query fields. Query strings should be URL Encoded<ref>http://en.wikipedia.org/wiki/URL_encoding</ref>. If the "id_key" parameter is not specified, then UCS<ref name="ucs"></ref> uses the following algorithm:

1. Starting with the identification key with the lowest ID, UCS determines the number of attributes in the key which are covered by the data specified in the "expr" parameter.
2. If all attributes are covered, then use this key. Otherwise, record the number of sequential attributes covered, starting from the left. For example, if key 1 consists of attributes (a, b, c), and "expr" contains (a, b) then the score is 2.
3. If no key is selected after iterating through all the defined keys, choose the key with the highest score from Step 2. In the event of a tie, UCS selects the key with the smallest ID.

Operation

Prerequisites

- The identification keys exist.
- The UCS is in production mode.

Use the Create Identification Key operation to create Identification Keys.

ID	CV.WS.CUSTID.1		
Method	GET		
URL	/profiles		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
id_key	string	no	The key name used for identification.

ID	CV.WS.CUSTID.1		
			UCS can infer the key based on the attributes included in the "expr" parameter. However, explicitly specifying the identification key with your application improves performance.
include_profile	token	no	• "no" (default) to return the list of customers IDs without their profile attributes. • "yes" to return all the profile attributes of the identified customers.
extensions	string	no	The names of the extension to return with the customer profiles. Your application can specify multiple values separated with comas, such as: *extensions=contacts,purchases*
include_extensions	token	no	• "always" (default) to return the extensions specified with the field "extension" if one or more customers are identified. • "unique" to return the extensions identified in parameter "extensions" if a unique customer is identified.
<attribute name>	string	**yes**	One or more identification keys used to identify the customer, for instance: *last_name=Doe&first_name=Jane* Attribute names with no prefix are supposed to be profile attributes. To refer to the attributes of an extension,

ID	CV.WS.CUSTID.1		
			your application must use the extension name as a prefix for the attribute: *<ext-name>.<attribute-name>*

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	• One result: Customer Profile • Several results: Customer Profile[] • No result: []

Example

Operation

```
 GET /profiles/
contacts.phone_number=408-888-3214&extensions=contacts,purchases€clude_profile=yes
 €clude_extensions=unique
```

Result: Multiple profiles
If multiple customers called from the specified phone number, then the output contains profiles for each matched customer:

```
 200 OK
 [{"customer_id": "2DC255C02AF", "name": "Doe, John", "birthdate": "1976-05-10"},
 {"customer_id": "DDF295802AF", "name": "Doe, Jane", "birthdate": "1978-02-25"}]
```

Result: One profile
If a single customer is called from the specified phone number, then the complete profile for that customer is returned.

```
200 OK
{ "customer_id": "0004Va58A92T0017 ",
 "name": "Doe, John",
 "birthdate": "1976-05-10",
 "EmailAddress": [
  "john.doe@genesyslab.com",
```

```
  "jd34@hotmail.dom"],
 "contacts": [
  {"phone": "408-555-1234", "ext": "1234"},
  {"phone": "408-832-7712"} ],
 "preferences": { "newsletter": "y", "new_offers": "n", "email": "html" }
}
```

Result: No result
If no customers match the specified criteria, then the output is an empty array:

```
 []
```

<references />

Insert Extension Records

	POST /profiles/${customer_id}/extensions
	Available since: 8.0.100.00

Description

This operation inserts one or more extension records, which replace former records if they already exist.

Operation

ID	CV.WS.PROF.4		
Method	POST		
URL	/profiles/**${customer_id}**/extensions		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The unique ID of the given customer. Limited to 16 characters.
Body			
<extension n>	Extension or Extension[]	no	Profile extensions, where <extension n> corresponds to the unique name of the profile extension resource. • For single-valued extensions, the extension's value is a single Extension. • For multi-valued extensions, the extension's value is an an array of zero or more Extension.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	OK

Example

Operation

```
POST /profiles/0004Va58A92T0017/extensions
  {
"customer_id":"0004Va58A92T0017",
"Phone": [
 {"PhoneType":0,
 "prefix":"+33",
 "number":"3145926535",
 "description":"family phone",
 "start_availabilty":"2009-12-18T18:30:00.000Z",
 "end_availabilty":"2009-12-18T21:40:00.000Z"},
 {"PhoneType":2,
 "prefix":"+33",
 "number":"6543210",
 "description":"business calls only, no sales",
 "start_availabilty":"2009-12-18T09:30:00.000Z",
 "end_availabilty":"2009-12-18T17:45:00.000Z"},
 {"PhoneType":5,
 "prefix":"+33",
 "number":"951357456",
 "description":"",
 ] }
}
```

Result

```
 200 OK
```

UCS added the phone extension to the extensions of the specified customer.

Merge Customer Profile

	PUT /profiles/${customer_id}/merge/${src_id}
	Available since: 8.1.300.xx

Description

Merges two profiles together, transferring all relative data from one to the other (attributes, services, states and tasks and all of their extensions). This operation is not reversible and the source profile will be deleted.

Operation

ID	CV.WS.PROF.MERGE.1		
Method	PUT		
URI	/profiles/${customer_id}/merge/${src_id}		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The customer ID where the data is to be merged.
${src_id}	string	yes	The customer ID of the source customer data to be merged. This ID will be deleted once the merge is complete.
Body Parameters			
keep_uniq_from_src	boolean	no	If there are duplicate attributes, determines which attribute to keep: • true—Keeps the attribute from the source. • false—Keeps the attribute from the destination (default).<ref name=

ID	CV.WS.PROF.MERGE.1		
			"keep_uniq_from_src"> For single valued profile extensions, and profile extension with unique attributes, the default is to keep the values from the destination profile. If `keep_uniq_from_src` is set to `true`, the source values will be kept, and the destination values will be discarded. Multi-valued extension content is copied to destination profile.</ref>
keep_active_service_only	boolean	no	Determines whether to merge the active records: • true—Merge active service records only. • false—Merge active and completed records (default).

<references /> At the end of the operation the source profile will be deleted with all completed services if the `keep_active_service_only` is set to `true`. The default operation is to transfer all active and completed services to the destination profile. Duplicate core attributes will be discarded. Core attribute duplication is determined by a case sensitive comparison of their values. If mixed behavior is needed where some unique/single extension values must be kept from the source and others from the destination, it is recommended to update the destination profile before merging the profiles. Profile merging is transactional, either the whole operation is a success or nothing is changed. If a source profile has a lot of services assigned (hundreds)to it, the merge operation can take a while, set your client timeouts accordingly.

Example

Operation

```
/profiles/00001b8BBKDX000D/merge/00001b8BBKDX000H
Method: PUT
{
  "keep_active_service_only": false,
  "keep_uniq_from_src": false
```

```
}
```

Result

```
Status: Status (204) - The server has fulfilled the request but does not need to return an
entity-body, and might want to return updated meta-information
Reference: http://localhost:8222/profiles/00001b8BBKDX000D
Transfer-Encoding: null
Content-Type: application/octet-stream
```

Query Customer Profile

	GET /profiles/${customer_id}
	Available since 8.0.100.00

Description

Queries the Customer Profile based on the customer ID parameter. According to the query's fields, your application can retrieve profile and extension attributes in addition to the identified customer ID. The response always contains all attributes of the customer profile resource, and all attributes of the specified extensions, to avoid ambiguity in further updates (see Update Customer Profile). To get the schema of the profile and extension attributes, use the Query Profile Schema operation.

Operation

Prerequisites: A Customer Profile Schema (a list of Attribute Schema) is available.

ID	CV.WS.PROF.1		
Method	GET		
URL	/profiles/**${customer_id}**		
Field Name	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	Specified in the URI, the ID of the customer.
extensions	string	no	The list of extensions to return within the result. Your application can specify multiple extension names, separated with comas. For instance: *extensions=contacts,purchases*

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200		
HTTP message	OK		
Field Name	**Type**	**Mandatory**	**Description**
Body: Customer Profile<ref>The returned result contains parts of the profile or the complete profile according to the parameters of the request. See Customer Profile for further information on the resource.</ref>			
customer_id	string	yes	The unique ID of the given customer. Limited to 16 characters.
<attribute n>	• string • datetime	yes	Customer attributes, where **<attribute n>** is the attribute name<ref>See the configuration options for further details.</ref>. Attributes are not part of a separated array, see the example below. The date/time formats are ISO 8601<ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`.
<extension n>	Extension or Extension[]	no	By default, no extension is returned. The returned extensions are specified in the parameters of the request. For further details, see Extension.

<references />

Example

For example, the customer profile (id=0004Va58A92T0017) has three attributes: "name","birthdate" and "EmailAddress". This profile also has two defined extensions:

- contacts: one or more phone numbers (a multi-valued extension).
- preferences: a single-valued extension representing customer preferences for receiving a weekly newsletter, receiving notification of new offers, and the desired e-mail type.

Operation

```
GET /profiles/0004Va58A92T0017?extensions=contacts,preferences
```

Result

```
200 OK
{ "customer_id": "0004Va58A92T0017 ",
 "name": "Doe, John",
 "birthdate": "1976-05-10",
 "EmailAddress": [
  "john.doe@genesyslab.com",
  "jd34@hotmail.dom"],
 "contacts": [
  {"phone": "408-555-1234", "ext": "1234"},
  {"phone": "408-832-7712"} ],
 "preferences": { "newsletter": "y", "new_offers": "n", "email": "html" }
}
```

Update Customer Profile

	PUT /profiles/${customer_id}
	Available since 8.0.1

Description

Overrides the attributes of the customer profile: prior attribute values are lost and replaced with the attribute list specified in the body of the request. This operation uses the standard convention for HTTP PUT, which requires you to update the profile with the complete profile to avoid losing information. Your application should use this operation in conjunction with the Query Customer Profile operation:

1. Query the Customer Profile,
2. Modify the Customer Profile data,
3. Update the Customer Profile.

Extensions can be part of the profile used for the update:

- If your application does not include any extensions in the body, no modification occurs on extensions.
- If your application specifies extension values, extension information is overriden. Your application must specify the complete extension set to avoid losing data.

Operation

Prerequisites

- The profile schema exists (see Query Profile Schema).
- Your application retrieved the Customer Profile (see Query Customer Profile).
- The UCS<ref name="ucs">Universal Contact Server</ref> is in production mode.

ID	CV.WS.PROF.2		
Method	PUT		
URL	/profiles/**${customer_id}**		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The ID of the customer.

ID	CV.WS.PROF.2		
			Limited to 16 characters.
Body: The customer profile information <ref>The complete Customer Profile representation, compliant with the defined profile schema (see Query Profile Schema).</ref>			
customer_id	string	yes	The unique ID of the given customer.
<attribute n>	• string • datetime	yes	The complete set of customer attributes, where **<attribute n>**is the attribute name<ref>See the configuration options for further details.</ref>. The date/time formats are ISO 8601<ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`. Attributes will be overriden. This representation must contain all the profile attributes' values, including those that weren't modified.
<extension n>	Profile Extension or Profile Extension[]	no	Extensions, where <extension n> corresponds to the unique name of the profile extension resource. If you specify extensions, extensions are overriden. If you wish to modify an extension value, this representation must contain all the profile extensions, including those that weren't modified.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	OK

Example

The following example assumes that:

- FirstName, LastName, DOB are existing attributes.
- Address is an extension.

Operation

```
PUT http://ucsserver.mycompany.com:8080/path/profiles/00027a52JCGY000M
Content-Location /profiles/00027a52JCGY000M
Content-Type: application/json
  {
 "FirstName": "Bruce",
 "LastName": "Banner",
 "DOB": "1962-05-10",
 "EmailAddress": [
      "bruce.banner@marvelous.com",
      "b.banner@hulk.dom"
 ],
 "Address": { "Type":1, "Address":"21 JumpStreet", "City":"Hollywood",
              "County":"Santa Barbara", "PostCode":"555", "Country":"United States" }
}
```

Result
The above representation is now the profile of the customer whose id is 00027a52JCGY000M.

```
HTTP 200 OK
[]
```

Update Record In Profile Extension

	PUT /profiles/${customer_id}/extensions/${ext_name}/by/unique
	Available since: 8.0.200.00

Description

Updates the content of a single record in a multi-valued extension. The attributes which are part of the 'unique' list specified at the Extension Schema creation are passed in the body and used to find the correct record to update. This update cannot change the values of the attributes which are part of the 'unique' list of the Extension Schema; 'unique' attributes are used as identifiers for the given record. **Workaround:** If you wish to update a value for an attribute stamped as unique, you must delete the concerned record, then recreate this record with its correct values.

Operation

Prerequisites: The extension record exists.

ID	CV.WS.PROF.6		
Method	PUT		
URL	/profiles/**${customer_id}**/extensions/**${ext_name}**/by/unique		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
${customer_id}	string	yes	The unique ID of the given customer. Limited to 16 characters.
${ext_name}	string	yes	The name of the Profile Extension to update.
Body<ref>Record for the Profile Extension.</ref>			
<attribute n>	• string • integer • long • double • date • datetime	yes	Individual attributes defined in the Profile Extension Schema. <attribute n> corresponds to the attribute name defined in the schema. • If the attribute is unique, its value is

ID	CV.WS.PROF.6		
	• currency • boolean		used to select the record to update. It is not updated. • If the attribute is not unique, its value is updated.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	204
HTTP message	No Content

Example

Operation The following operation updates the Phone record associated with the unique "PhoneNumber":"3145926535".

```
PUT /profiles/0000Sb5U97XE000Y/extensions/Phone/by/unique
{
 "PhoneType":0,
 "prefix":"+33",
 "PhoneNumber":"3145926535",
 "description":"family phone",
 "start_availabilty":"2009-12-18T18:30:00.000Z",
 "end_availabilty":"2009-12-18T21:40:00.000Z"
}
```

Result

```
204 No Content with empty body
```

Profile Resources

Purpose

Groups the Profile resources.

Customer Profile

Description

Describes a customer profile. Available since: 8.0.100.00. Customer profiles are built on top of legacy UCS Contact Attributes. They are multi-valued and support the primary attributes. If an attribute has several values, the first one of the JSON array is the primary attribute, the others are non-primary. Each customer profile consists in a list of attributes defined via business attributes, see the configuration options for further details. These attributes share the same level with the customer_id field (provided by the UCS to identify the customer).

In addition to these attributes, your application can add extensions (whose schema is defined with Schema Operations). See the table and example below.

Resource

Field	Type	Mandatory	Description
customer_id	string	yes	The unique ID of the given customer. Limited to 16 characters.
<attribute n>	• string • datetime	yes	Customer attributes, where **<attribute n>**is the attribute name<ref>See the configuration options for further details.</ref>. Attributes are not part of a separated array, see the example below. The date/time formats are ISO 8601<ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`.
<extension n>	Extension or Extension[]	no	Profile extensions, where <extension n> corresponds to the unique name of the profile extension resource. • For single-valued extensions, the result is a single Extension.

Field	Type	Mandatory	Description
			• For multi-valued extensions, the field value is an an array of zero or more Extension.

<references />

Example

The following output sample presents the representation for the customer *John Doe*. It includes 2 attributes : FirstName and LastName. The EmailAddress is an extension which contains two contact records.

```
{
   "FirstName": "John",
   "LastName": "Doe",
   "EmailAddress": [
     "john.doe@genesyslab.com",
     "jd34@hotmail.dom"
   ]
}
```

Profile Attribute Schema

	Purpose: Describes a Profile Attribute schema.

Description

Describes a Profile Attribute. Each attribute field of the profile corresponds to a recorded schema which details the attribute name, type, and content. In 8.0.1 release, the configuration of a new attribute schema is performed in the Configuration Manager. Select in "Tenant" the item *specific configuration*, and under "Business Attributes" 'Contact Attributes'. Then, create a new Business Attribute Value. By default the new Attribute is of type string. To assign another type to the schema, in Annex, create a section "settings", and a key "type" with value "string" or "date".

Predefined Attributes Schema

Default Attributes visible in the Configuration Server are the following:

- First Name (string)
- Last Name (string)
- Phone Number (string)
- E-mail Address (string)
- Title (string)

Default 'internal' Attributes (not visible in the Configuration Server) are the following:

- LastCalledAgent_EmployeeID (string)
- LastCalledAgent_TimeStamp (date)
- PreferredAgent_EmployeeID (string)

Additional attributes are available for each active media defined in the Business Attribute "Media Type" of the Configuration Manager:

- LCA_EmpIID_<media> (string)
- LCA_TimeStamp_<media> (date)
- Pref_EmpIID_<media> (string)

Attributes of type date/time are formatted in UTC ISO 8601: "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
Example: "1981-03-17T02:00:00.000Z"

Resource

Field	Type	Mandatory	Description
name	string	yes	The name is unique for each attribute within the given profile. • Starts with a letter, and can be followed with letters, numbers, or underscores. • Is restricted to a maximum of 30 characters (Maximum RDBMS<ref name= "RDBMS">http://en.wikipedia.org/wiki/Relational_database_management_s shared limit) In profile attribute schemas, the name is **not** case-sensitive.
type	token	yes	The attribute type, which can be one of the following types: • Boolean • String • Integer • Long • Double • Date • DateTime • Currency Notes: 1. The constraints on these types (the minimum and maximum values supported for the integer type) vary with the RDBMS<ref name="RDBMS"

Field	Type	Mandatory	Description
			/>[1]</ref> 2. Customer profiles are built on top of UCS' Contact Attributes, and support only String and DateTime attribute types. However, all attribute types are supported for profile extensions.
default	numeric or string	no	Provides a default value for the attribute. For numeric attributes, you do not need quotation marks. Quotation marks are for string, date, and datetime attributes. By default, the date/time format is ISO 8601 UTC <ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`.
length	integer	no	Reserved to attributes of type `String`. Defines the maximum number of characters for the string. • Default length is `256`; • Maximum length is `4000`.
encrypt	Boolean	no	• `false` by default. • `true` if the server must encrypt the attribute value before saving it to the UCS database.
mandatory	Boolean	no	• `false` by default; • `true` if the attribute is mandatory when inserting a new `Profile Extension`

Field	Type	Mandatory	Description
			record.

<references />

Related Operation

- Query Profile Schema

Profile Extension

	Purpose: Presents an extension for a customer profile.

Description

Extends the customer profile with additional attribute resources. Note that for each extension resource, you must define first a profile extension schema. Extensions, like profiles, consist of a series of one or more Profile Attribute resources, which are defined with a predefined Profile Extension Schema.

Resource

Profile Extension

Field	Type	Mandatory	Description
customer_id	string	yes if not part of a profile.	The unique ID of the given customer. This field is mandatory if the extension is **not** nested inside a Customer Profile.
name	string	yes	The unique, case-insensitive name of the extension. • Starts with a letter to which you can append letters, numbers, or underscores. • Restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on index creation.)
<attribute n>	string integer long double date	yes	Individual attributes, defined in the Profile Extension Schema. <attribute n> corresponds to the

Field	Type	Mandatory	Description
	datetime currency		attribute name defined in the schema.

Profile Extension Schema

	Purpose: Describes a type of extension available for the customer profiles.

Description

Describes one of the Profile Extension resources. Before your application can start adding extensions to a given profile, your application must define this extension's schema, which includes its nested attributes.

Resource

Field	Type	Mandatory	Description
name	string	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed by letters, numbers, or underscores. The name is restricted to a maximum of 26 characters (Maximum RDBMS shared limit on the creation of index).
type	token	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with any given customer. • Classifier: the single-valued extension has no attribute.

Field	Type	Mandatory	Description
			Only single-valued profile extensions are supported for now.
attributes	array	no	The array of *zero or more* attribute objects. Empty when type="classifier".

Example

The following schema describes the Address extension.

```
"name":"Address",
"type":"single-valued",
"attributes": [
    {"name":"AddressType","type":"integer","default":0},
    {"name":"Address","type":"string","length":256},
    {"name":"City","type":"string","length":32},
    {"name":"County","type":"string","length":32},
    {"name":"PostCode","type":"string", "length":10},
    {"name":"Country","type":"string","length":32}
    ]
```

List of Schema Operations

- Schema Operations
  - GET /metadata/profiles/
  - POST /metadata/profiles/extensions
  - GET /metadata/profiles/extensions
  - GET /metadata/identification-keys
  - POST /metadata/identification-keys
  - Since 8.0.2:
    - POST /metadata/services/extensions
    - GET /metadata/services/extensions
    - POST /metadata/states/extensions
    - GET /metadata/states/extensions
    - POST /metadata/tasks/extensions
    - GET /metadata/tasks/extensions
  - Available since 8.1.0
    - GET /metadata/business-attributes
    - GET /metadata/business-attributes/${business-attribute-name}
    - GET /metadata/business-attributes?cv-attribute=${attribute-full-name}
  - Available since 8.1.0
    - GET ${content-type} /metadata
  - Available since 8.1.3
    - DELETE /metadata/profiles/extensions/${extension-name}
    - DELETE /metadata/services/extensions/${extension-name}}
    - DELETE /metadata/states/extensions/${extension-name}
    - DELETE /metadata/tasks/extensions/${extension-name}
    - DELETE /metadata/identification-keys/${id_key-name}

Create Identification Key

	POST /metadata/identification-keys
	Available since: 8.0.100.00

Description

Creates the key used to identify customers. Your application must define the Identification Key prior to any customer identification attempt. You can build identification keys as a combination of attributes:

- One or more attributes of the customer profile.
- One or more attributes of an extension.

> **Important**
> Your application cannot create Identification keys with a blend of profile attributes and extension attributes.

The following identification key representation identifies customers based on the `PhoneNumber` attribute.

```
{ "name": "idPhone",
 "attributes": ["PhoneNumber"]}
```

Let's imagine that the phone number is part of the **contactExt** extension; the corresponding object is:

```
{ "name": "idPhone",
 "source": "contactExt",
 "attributes": ["PhoneNumber "]}
```

This example assumes that one or more customers can share the same phone number, which could apply to members of a family for instance, and makes the identification key not unique across the calling customers. To make it unique, use the "unique" field of the identification key (see the Identification Key for further details).

Operation

Prerequisites: The server is in Maintenance mode.

Use the Set Server Mode operation to switch modes.

Create Identification Key

ID	CV.WS.PROF.META.7		
Method	POST		
URL	/metadata/identification-keys		
Parameter	**Type**	**Mandatory**	**Description**
Body: Includes the identification key to create This body contains the Identification Key resource.			
name	string	yes	The unique, case-insensitive name of the key. The name must start with a letter, and can be followed with letters, numbers, or underscores. The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on creation of index.)
source	string	no	Can take on one of the following values: • "profile" (default value): indicates that the attributes specified in parameter "attribute" belong to the core customer profile. • The unique name of the given Profile Extension, used to create the Identification Key.
attributes	string[]	yes	The array names of one or more attributes of the extension or of the profile.
unique	bool	no	• "false" by default. • "true" to indicate that the given attributes of the profile or extension are unique across all customers.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: metadata/identification-keys/**${key-name}** where: • ${key-name} is the name specified in the submitted identification key resource.
Body	{"name": "**${key-name}**"} where: • ${key-name} is the name of the created key.

Example

The following sample creates a key which identifies the customer based on the provided phone ID. After this key is created, the further profile creations are indexed based on the Customer Phone ID.
Operation

```
POST /metadata/identification-keys
{
  "name": "idPhone",
  "extension": "contacts",
  "attributes": ["phone_number"]
}
```

Result

```
 HTTP 201 Created
 {"name": "idPhone"}
```

Create Profile Extension Schema

	POST /metadata/profiles/extensions
	Available since: 8.0.100.00

Description

Creates a profile extension schema resource, which is mandatory to query and use profile extension schemas. Your application can create profile extension schemas when the UCS is in Production mode, but Genesys recommends to create extensions when the server is in maintenance mode.

- Read more about UCS Modes for Context Services.
- Read more about Extensions.

8.0.1 does not support multi-valued extensions.

Operation

Prerequisites:

- The server is in Maintenance mode.

Create Profile Extensions Schema

ID	CV.WS.PROF.META.4		
Method	POST		
URL	/metadata/profiles/extensions		
Parameter	**Type**	**Mandatory**	**Description**
Body: The schema to create.<ref>This is the Profile Extension Schema to create.</ref>			
name	string	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed with letters, numbers, or underscores. The name is restricted to a maximum of 26 characters (Maximum RDBMS shared limit on creation of index).

ID	CV.WS.PROF.META.4		
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with a given customer.
attributes	Attribute Schema[]	no	The array of *zero or more* attributes. Empty when type="classifier".
unique	string[]	no	Lists attributes which are unique in the scope of a given customer. The customer profile cannot include several extension records with identical values for these attributes. For example, let's consider a contact extension with the attributes "phone_num", "ext", "time_of_day". To ensure that a given customer does not have two contact extensions with the same phone number, set the following in the extension schema: unique = ["phone_num", "ext"]

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: **${base_uri}**/profiles/extensions/**${extension-name}** where • ${extension-name} is the name of the extension provided in the request's entity body. • ${base_uri} is the base URI of Context Services.
Body	{ "name": "**${extension-name}**"} where: • ${extension-name} is the name of the created extension.

Example

Operation

```
POST http://ucsserver.mycompany.com:8080/cms/profiles/extensions
{
 "name":"Address",
 "type":"single-valued",
 "attributes": [
     {"name":"AddressType","type":"integer","default":0},
     {"name":"Address","type":"string","length":256},
     {"name":"City","type":"string","length":32},
     {"name":"County","type":"string","length":32},
     {"name":"PostCode","type":"string", "length":10},
     {"name":"Country","type":"string","length":32}
     ]
 }
```

Result

```
201 Created
Content-Location        http://ucsserver.mycompany.com:8080/cms/profiles/extensions/Address
{"name":"Address"}
```

Create Task Extension Schema

	POST /metadata/tasks/extensions
	Available since: 8.0.200.00

Description

Creates an extension schema for task resources. Your application must create a schema which defines the task extension before this extension can be used. For instance, if your application needs to fill in feedback data in some task, the first step is to create the corresponding schema which details the lists of attributes that compose the new extension. Then, once the schema is created, your application can use the extension records in task resources which fulfill the extension schema. See Extension for further details. Your application can create the extension schema when the UCS is in Production mode, but Genesys recommends to create extensions when the server is in maintenance mode.

- Read more about UCS Modes for Context Services.
- Read more about Extensions.

Operation

Prerequisites: Universal Contact Server is in maintenance mode.

ID	CV.WS.SRV.META.7		
Method	POST		
URL	/metadata/tasks/extensions		
Name	**Type**	**Mandatory**	**Description**
Body: The Schema of the Task Extension <ref>This body contains a complete Extension Schema resource, which defines the content of the task extension.</ref>			
name	string	yes	The unique name of the extension. The name, which is **not** case-sensitive, must start with a letter to be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on

ID	CV.WS.SRV.META.7		
			creation of index.)
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with a given customer.
attributes	Attribute Schema[]	yes	The array of attributes that compose the extension.
unique	string[]	no	Lists attributes (separated by commas) which are unique in the scope of a given task. The resource cannot include several extension records with identical values for these attributes.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: /metadata/tasks/extensions/**${extension-name}** where: • ${extension-name} is the name of the created extension.
Body	{ "name": "**${extension-name}**"}

HTTP code	201
	where: • ${extension-name} is the name of the created extension.

Example

Operation

```
POST /metadata/tasks/extensions
[
  {
   "attributes": [
     {"name":"url","type":"string","length":"1024","mandatory":"true"},
     {"name":"question1","type":"integer","mandatory":"true"},
     {"name":"question2","type":"boolean"},
     {"name":"question3","type":"string","length":"256"}
     ],
   "name":"Survey",
   "type":"single-valued"
  }
]
```

Result

```
201 Created
{ "name": "Feedback"}
```

Create State Extension Schema

	POST /metadata/states/extensions
	Available since: 8.0.200.00

Description

Creates an extension schema for state resources. Your application must create a schema which defines the state extension before this extension can be used. For instance, if your application needs to fill in feedback data in some state, the first step is to create the corresponding schema which details the lists of attributes that compose this extension. See Extension for further details.

Operation

Genesys advises to use this operation in maintenance mode, rather than in production mode.

Create State Extension Schema

ID	CV.WS.SRV.META.5		
Method	POST		
URL	/metadata/states/extensions		
Name	**Type**	**Mandatory**	**Description**
Body: The Schema of the State Extension <ref>This body contains a complete Extension Schema resource, which defines the state extension to create.</ref>			
name	string	yes	The unique name of the extension. The name, which is **not** case-sensitive, must start with a letter to be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on creation of index.)
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a

ID	CV.WS.SRV.META.5		
			single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with a given customer.
attributes	Attribute Schema[]	yes	The array of attribute schemas.
unique	string[]	no	Lists attributes (separated by commas) which are unique in the scope of the given state. The resource cannot include several extension records with identical values for these attributes.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: /metadata/states/extensions/**${extension-name}** where: • ${extension-name} is the name of the created extension.
Body	{ "name": "**${extension-name}**"} where: • ${extension-name} is the name of the created extension.

Example

Operation

```
POST /metadata/states/extensions
{
 "attributes":
   [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
   ],
 "name":"Feedback",
 "type":"single-valued"
}
```

Result

```
201 Created
{ "name": "Feedback"}
```

Create Service Extension Schema

	POST /metadata/services/extensions
	Available since: 8.0.200.00

Description

Creates an extension schema for service resources. Your application must create a schema which defines the state extension before this extension can be used. For instance, if your application needs to fill in feedback data in some state, the first step is to create the corresponding schema which details the lists of attributes that compose this extension. See Extension for further details.

- Read more about UCS Modes for Context Services.
- Read more about Extensions.

Operation

Genesys advises to use this operation in maintenance mode, rather than in production mode.

Create Service Extension Schema

ID	CV.WS.SRV.META.1		
Method	POST		
URL	/metadata/services/extensions		
Name	**Type**	**Mandatory**	**Description**
Body: The Schema of the Service Extension <ref>This body contains a complete Extension Schema resource, which defines the service extension to create.</ref>			
name	string	yes	The unique name of the extension. The name, which is **not** case-sensitive, must start with a letter to be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on creation of index.)

ID	CV.WS.SRV.META.1		
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with a given customer.
attributes	Attribute Schema[]	yes	The array of attribute schemas.
unique	string[]	no	Lists attributes (separated by commas) which are unique in the scope of a given service. The resource cannot include several extension records with identical values for these attributes.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: /metadata/services/extensions/**${extension-name}** where: • ${extension-name} is the name of the created extension.
Body	{ "name": "**${extension-name}**"} where:

HTTP code	201
	• ${extension-name} is the name of the created extension.

Example

Operation

```
POST /metadata/services/extensions
{
 "attributes":
   [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
   ],
 "name":"Feedback",
 "type":"single-valued"
}
```

Result

```
201 Created
{ "name": "Feedback"}
```

Change Metadata Cache

	PUT /metadata/cache/
	Available since: 8.1.300.00

Description

Enables the reload of metadata cache. Metadata cache holds information for the definition of Profile, Service, State, Task extensions, and identification keys definition.

Operation

ID	CV.WS.META.CACHE.PUT		
URI	/metadata/cache		
Method	PUT		
Parameter	**Type**	**Mandatory**	**Description**
reload	token	yes	Any value triggers a reload.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	No Content

Example

The following operation reloads the metadata cache from the database.

Operation

```
PUT /cv/metadata/cache/
{reload:true}
```

Result

HTTP/1.1 200 OK with an empty body.

<references />

Delete Metadata Identification Keys

	/metadata/identification-keys/${id_key-name}
	Available since: 8.1.300.00

Description

Allows the deletion of Identification keys. The records in metadata table are removed, and the related table is dropped (if any). An identification key can be created based on the core profile attribute, but in this case a table is not created. Therefore, no table is dropped when it is deleted.

Operation

ID	CV.WS.PROF.META.8		
URI	/metadata/identification-keys/${id_key-name}		
Method	DELETE		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameter			
${id_key-name}	string	yes	The metadata identification key name to delete. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP message	Success No Content

Example

The following operation deletes an identification key.

Operation

```
DELETE /metadata/identification-keys/idNickName
```

Result

```
HTTP/1.1 204 OK
Success No Content
```

Note: If nothing was deleted, HTTP status code 404 ("Not Found") is sent.

Delete Metadata Profile Extensions

	/metadata/profiles/extensions/${extension-name}
	Available since: 8.1.300.00

Description

Allows the deletion of a profile extension. The records in the profile extension table are deleted, the table is deleted, and the definition of the profile extension is erased. If identification keys exist based on the specified extension, it will not be deleted, and an error will be returned.

Operation

ID	CV.WS.PROF.META.5		
URI	/metadata/profiles/extensions/${extension-name}		
Method	DELETE		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	yes	The metadata extension name to delete. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP message	Success No Content

Example

The following operation deletes a profile extension.

Operation

```
DELETE /metadata/profiles/extensions/CustomName
```

Result

```
HTTP/1.1 204 OK
Success No Content
```

Note: If nothing was deleted, HTTP status code 404 ("Not Found") is sent.

Delete Metadata Services Extensions

	/metadata/services/extensions/${extension-name}
	Available since: 8.1.300.00

Description

Allows the deletion of a service extension. The records in the service extension table are deleted, the table is deleted, and the definition of the service extension is erased.

Operation

ID	CV.WS.SRV.META.3		
URI	/metadata/services/extensions/${extension-name}		
Method	DELETE		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	yes	The metadata extension name to delete. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP message	Success No Content

Example

The following operation deletes a service extension.

Operation

```
DELETE /metadata/services/extensions/relatedOffers
```

Result

```
HTTP/1.1 204 OK
Success No Content
```

Note: If nothing was deleted, HTTP status code 404 ("Not Found") is sent.

Delete Metadata States Extensions

	/metadata/states/extensions/${extension-name}
	Available since: 8.1.300.00

Description

Allows the deletion of a state extension. The records in the state extension table are deleted, the table is deleted, and the definition of state extension is erased.

Operation

ID	CV.WS.SRV.META.4		
URI	/metadata/states/extensions/${extension-name}		
Method	DELETE		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	yes	The metadata extension name to delete. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP message	Success No Content

Example

The following operation deletes a state extension.

Operation

```
DELETE /metadata/states/extensions/Feedback
```

Result

```
HTTP/1.1 204 OK
Success No Content
```

Note: If nothing was deleted, HTTP status code 404 ("Not Found") is sent.

Delete Metadata Tasks Extensions

	/metadata/tasks/extensions/${extension-name}
	Available since: 8.1.300.00

Description

Allows the deletion of a task extension. The records in the task extension table are deleted, the table is deleted, and the definition of the task extension is erased.

Operation

ID	CV.WS.SRV.META.5		
URI	/metadata/tasks/extensions/${extension-name}		
Method	DELETE		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	yes	The metadata extension name to delete. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response	
HTTP code	204
HTTP message	Success No Content

Example

The following operation deletes a task extension.

Operation

```
DELETE /metadata/tasks/extensions/Proposal
```

Result

```
HTTP/1.1 204 OK
Success No Content
```

Note: If nothing was deleted, HTTP status code 404 ("Not Found") is sent.

Get Identification Keys

	GET /metadata/identification-keys
	Available since: 8.0.100.00

Description

Returns the identification keys. This operation is available in both "production" and "maintenance" modes.
Read more about UCS Modes for Context Services.

Operation

Get Identification Keys

ID	CV.WS.PROF.META.6		
Method	GET		
URL	• All: /metadata/identification-keys • Key: /metadata/identification-keys/**${key-name}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameter			
${key-name}	string	no	The unique, case-insensitive name of the key. The name must start with a letter, and can be followed with letters, numbers, or underscores. The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on creation of index.) If you do not specify this parameter, all keys are returned.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	• /metadata/identification-keys: Identification Key[] <ref>Length can be 0.</ref> • /metadata/identification-keys/${key-name}: Identification Key

Example

The following request retrieves all the identification keys: **Operation**

```
GET /metadata/identification-keys
```

Result

Let's consider a customer profile which consists of the customer's name, the date of birth, and the external ID (the customer's ID in an external system such as a CRM). Further assume that a single extension, Contacts, include various attributes such as the phone_number. If you set up the Context Management Server to allow the customer identification via name+birthdate, external ID, or phone number, then the returned result is the following:

```
[{"name": "idNameBD", "attributes": ["name", " birthdate"]},
{"name": "idExt", "attributes": ["external_id"], "unique": true},
{"name": "idPhone", "attributes": ["phone_number"]}
{"name": "idExtension", "attributes": ["ext.code","ext.date"]}]
```

If the identification key was created off an extension, attributes that are part of the extension are prefixed with the extension name. In the previous example, "idExtension" was built on the extension "ext".

Get Metadata

	GET ${content-type} /metadata
	Available since: 8.1.000.10

Description

Returns the location for the list of available metadata resources.

You do not need to switch to a specific server mode to attempt this operation.

Operation

Get Metadata

ID	CV.WS.META.1		
Method	GET		
URL	/metadata		
HEADER	${content-type}		
Parameter	**Type**	**Mandatory**	**Description**
${content-type}	• "Content-type:text/html" • "Content-type:application/json"	yes	The formatting for the returned result.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	GET "Content-type:text/html" /metadata Returns an HTML page, containing a list of resource links as follows: `<html><body><h1>Available Resources</h1>  <a href="http://localhost:8080/metadata/${resource}">${resource-name}  </body></html>` where: • ${resource} is the location of the resource. • ${resource-name}, the name of the resource.
	GET "Content-type:application/json" /metadata Returns a JSON table, containing a list of resource links as follows: `{${resource-name-1}:${resource-1}, ${resource-name-2}:${resource-2}, ..., ${resource-name-n}:${resource-n} }` where: • ${resource_i} is the location of the resource-i. • ${resource-name_i} is the name of the resource-i.

Example

Operation

```
GET "Content-type:text/html" /metadata
```

Result

```
200 OK {
<html><body><h1>Available Resources</h1><br/>
<a href="http://localhost:8080/metadata/identification-keys">
identification-keys schemas<br/>
<a href="http://localhost:8080/metadata/tasks/extensions">Tasks Extensions<br/>
<a href="http://localhost:8080/metadata/states/extensions">States Extensions<br/>
<a href="http://localhost:8080/metadata/services/extensions">Services Extensions<br/>
<a href="http://localhost:8080/metadata/business-attributes">business-attributes mapping<br/>
</body></html>
}
```

Operation

```
GET "Content-type:application/json" /metadata
```

Result

```
200 OK
{"identification-keys schemas":"http://localhost:8080/metadata/identification-keys",
     "Tasks Extensions":"http://localhost:8080/metadata/tasks/extensions",
     "Services Extensions":"http://localhost:8080/metadata/services/extensions",
     "States Extensions":"http://localhost:8080/metadata/states/extensions",
     "business-attributes mapping":"http://localhost:8080/metadata/business-attributes"}
```

Query Business Attribute Schema

	GET /metadata/business-attributes
	Available since: 8.0.200.04

Description

Returns the schema for the list of available business attributes.

You do not need to switch to a specific server mode to attempt this operation.

Operation

ID	CV.WS.CONFIG.BA		
Method	GET		
URL	1. All: /metadata/business-attributes 2. /metadata/business-attributes/${business-attribute-name} 3. /metadata/business-attributes?cv-attribute=${attribute-full-name}		
Parameter	**Type**	**Mandatory**	**Description**
${business-attribute-name}	string	no	The name of the business attribute, in order to retrieve its schema only.
${attribute-full-name}	string	yes if ?cv-attribute= is part of the URL	The full name of a Context Services' business attribute in order to retrieve its schema. Examples: • Service.type, • Task.disposition, • State.media_type

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

HTTP code		200	
HTTP message		OK	
Parameter	**Type**	**Mandatory**	**Description**
Body Business Attribute or BusinessAttribute[]			
name	string	yes	The business attribute's name.
dbid	string	yes	The business attribute's database ID.
display_name	string	yes	The business attribute's display name.
description	string	yes	The description.
map_name Available since: 8.1.000.xx	boolean	no	• `true` means that the `Business Attribute Value Names` is returned instead of DB IDs in the responses for GET operations. • `false` means that DB IDs are returned in the responses for GET operations. See also [business-attributes] Section.
cv_attributes	string[]	no	The associated Context Services attributes.
values	BusinessAttribute[]	no	Nested possible attributes values.

Example

Operation

```
GET /metadata/business-attributes/MediaType
```

Result

```
200 OK {
{
```

```
 "display_name" : "Media Type",
 "values" : [ {
 "display_name" : "email",
 "description" : "Media EMail",
 "name" : "email",
 "dbid" : 1003
 }, {
 "display_name" : "chat",
 "description" : "Media Chat",
 "name" : "chat",
 "dbid" : 1006
 }, {
 "display_name" : "voice",
 "description" : "Media Voice",
 "name" : "voice",
 "dbid" : 1001
 }, {
 "display_name" : "fax",
 "description" : "Media Fax",
 "name" : "fax",
 "dbid" : 1014
 }],
 "description" : "Media type identifier",
 "name" : "MediaType",
 "dbid" : 1001,
 "map_name" : false,
 "cv_attributes" : [ "Service.media_type", "State.media_type", "Task.media_type" ]
}
}
```

Query Metadata Roles

	GET /metadata/roles/
	Available since: 8.1.000.10

Description

This method provides convenient access to roles as configured in Configuration Server.

Operation

-

ID	CV.WS.SRV.META.x
URI	/metadata/roles
Method	GET

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200	
HTTP message	OK	
Name	**Type**	**Description**
role_name	String	The name of the given role.
role_dbid	String	The unique ID of the given role.
role_enabled	Boolean	• enabled • disabled
task	JSON Array	A task is made up of the following:

HTTP code	200	
		• Name--The name of the task. • Capability--Indicates whether the role can perform the task.

Example

The following operation returns the configured roles with the state and capability.

Operation

```
GET /metadata/roles
```

Result

On success, the operation returns an array of zero or more JSON objects.

```
[ {
  "role_DBID" : 108,
  "role_enabled" : true,
  "tasks" : [ {
    "capableOf" : true,
    "name" : "UCS.Customer.createProfile"
  } ],
  "role_name" : "Ucs_BigBrother"
}, {
  "role_DBID" : 110,
  "role_enabled" : true,
  "tasks" : [ {
    "capableOf" : true,
    "name" : "UCS.SchemaMgt.readServiceExtensionSchema"
  }, {
    "capableOf" : true,
    "name" : "UCS.Service.readServiceExtension"
  }, {
    "capableOf" : true,
    "name" : "UCS.Service.createServiceExtension"
  }, {
    "capableOf" : true,
    "name" : "UCS.Service.deleteServiceExtension"
  }, {
    "capableOf" : true,
    "name" : "UCS.SchemaMgt.readTaskExtensionSchema"
  }, {
    "capableOf" : true,
    "name" : "UCS.Service.startService"
  }, {
    "capableOf" : true,
    "name" : "UCS.SystemMgt.readServerInfo"
  }, {
    "capableOf" : true,
   "name" : "UCS.Service.readService"
```

```
 }, {
   "capableOf" : true,
    "name" : "UCS.SystemMgt.changeServerMode"
  } ],
  "role_name" : "ucs_roleForAgents"
}, {
  "role_DBID" : 111,
  "role_enabled" : true,
  "tasks" : [ {
    "capableOf" : true,
    "name" : "UCS.SchemaMgt.handleMetadata"
  }, {
    "capableOf" : true,
    "name" : "UCS.SystemMgt.readServerInfo"
  }, {
    "capableOf" : true,
    "name" : "UCS.SchemaMgt.readGenAdminRole"
  } ],
  "role_name" : "uselessRole"
}, {
  "role_DBID" : 112,
  "role_enabled" : true,
  "tasks" : [ {
    "capableOf" : true,
    "name" : "UCS.SystemMgt.readServerInfo"
  } ],
  "role_name" : "anotherUselessRole"
} ]
```

Query Profile Extension Schema

	GET /metadata/profiles/extensions
	Available since: 8.0.100.00

Description

Queries the schemas of the given profile extension. This operation is available in both `Production` or `Maintenance` modes (read the Documentation:CS:Developer:Server Mode page for further information about server modes.)

Operation

Prerequisites: The Profile Extension Schemas resource exist. To create the profile extension schema, use the Create Profile Extension Schema operation.

Query Profile Extensions Schema

ID	CV.WS.PROF.META.3		
Method	GET		
URL	• All: /metadata/profiles/extensions • /metadata/profiles/extensions/${extension-name}		
Name	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	no	The extension name, if a specific extension should be returned. This name is **not** case-sensitive.

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	/metadata/profiles/extensions Returns an array of the following fields: • name: the unique extension name. • type: "single-valued" or "multi-valued". In v8.0.1 **only single-valued extensions** are supported. • required: true or false • attributes: Attribute Schema[]. /metadata/profiles/extensions/${extension-name} Extension Schema or Extension Schema[]

Example

Get all extension schemas

Operation

```
GET /metadata/profiles/extensions
```

Result

```
200 OK
[
{
   "name":"Phone",
   "type":"multi-valued",
   "attributes": [
       {"name":"PhoneType","type":"integer","default":0,"mandatory":"true"},
       {"name":"prefix","type":"string","length":"3","default":"555",},
       {"name":"PhoneNumber","type":"integer","length":15,"mandatory":"true"},
       {"name":"description","type":"string","length":32,"mandatory":"true"},
       {"name":"start_availabilty","type":"datetime"},
       {"name":"end_availabilty","type":"datetime", "mandatory":"false"}
       ]
},
{
   "name":"Address",
   "type":"single-valued",
   "attributes": [
       {"name":"AddressType","type":"integer","default":0},
       {"name":"Address","type":"string","length":256},
       {"name":"City","type":"string","length":32},
       {"name":"County","type":"string","length":32},
       {"name":"PostCode","type":"string", "length":10},
       {"name":"Country","type":"string","length":32}
```

```
      ]
} ]
```

Get a single extension schema

Operation

```
 GET /metadata/profiles/extensions/Phone
```

Result

```
200 OK
{
   "name":"Phone",
   "type":"multi-valued",
   "attributes": [
       {"name":"PhoneType","type":"integer","default":0,"mandatory":"true"},
       {"name":"prefix","type":"string","length":"3","default":"555",},
       {"name":"PhoneNumber","type":"integer","length":15,"mandatory":"true"},
       {"name":"description","type":"string","length":32,"mandatory":"true"},
       {"name":"start_availabilty","type":"datetime"},
       {"name":"end_availabilty","type":"datetime", "mandatory":"false"}
   ]
}
```

Query Profile Schema

	GET /metadata/profiles/
	Available since: 8.0.100.00

Description

Queries the schema which describes the customer profiles. If the profile schema is not defined, this operation returns the HTTP Status `Code 404 (Not Found)`. In that case, your application should switch to maintenance mode, see Set Server Mode, and create the profile Schema by defining business contact attributes in the Configuration Manager. See Configuration Options for further details.

Operation

Prerequisites:

- The UCS<ref name="ucs"></ref> is in production mode.
- The profile schema is already defined.

Query Profile Schema

ID	CV.WS.PROF.META.1
Method	GET
URL	/metadata/profiles/

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	Attribute Schema[]

Example

Operation

```
GET /metadata/profiles/
```

Result

```
200 OK
[{"encrypt":false,"name":"PIN","length":256,"type":"string"},
{"encrypt":false,"name":"Title","length":256,"type":"string"},
{"encrypt":false,"name":"CustomerSegment","length":256,"type":"string"},
{"encrypt":false,"name":"LastName","length":256,"type":"string"},
{"encrypt":false,"name":"AccountNumber","length":256,"type":"string"},
{"encrypt":false,"name":"FirstName","length":256,"type":"string"},
{"encrypt":false,"name":"PhoneNumber","length":256,"type":"string"},
{"encrypt":false,"name":"ContactId","length":256,"type":"string"},
{"encrypt":false,"name":"EmailAddress","length":256,"type":"string"},
{"encrypt":false,"name":"Title","length":256,"type":"string"},
{"encrypt":false,"name":"PIN","length":256,"type":"string"},
{"encrypt":false,"name":"ContactId","length":256,"type":"string"},
{"encrypt":false,"name":"homeaddress","length":256,"type":"string"},
{"encrypt":false,"name":"EmailAddress","length":256,"type":"string"},
{"encrypt":false,"name":"PhoneNumber","length":256,"type":"string"},
{"encrypt":false,"name":"AccountNumber","length":256,"type":"string"},
{"encrypt":false,"name":"LastName","length":256,"type":"string"},
{"encrypt":false,"name":"FirstName","length":256,"type":"string"},
{"encrypt":false,"name":"CustomerSegment","length":256,"type":"string"},
{"encrypt":false,"name":"LastCalledAgent_EmployeeID","length":256,"type":"string"},
{"encrypt":false,"name":"LastCalledAgent_TimeStamp","length":0,"type":"datetime"},
{"encrypt":false,"name":"PreferredAgent_EmployeeID","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_workitem","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_workitem","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_workitem","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_auxwork","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_auxwork","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_auxwork","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_imchat","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_imchat","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_imchat","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_outboundpreview","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_outboundpreview","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_outboundpreview","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_vmail","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_vmail","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_vmail","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_smail","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_smail","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_smail","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_cobrowsing","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_cobrowsing","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_cobrowsing","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_webform","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_webform","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_webform","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_busevent","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_busevent","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_busevent","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_voice","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_voice","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_voice","length":256,"type":"string"},
```

{"encrypt":false,"name":"LCA_EmplID_appsharing","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_appsharing","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_appsharing","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_email","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_email","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_email","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_chat","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_chat","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_chat","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_any","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_any","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_any","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_fax","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_fax","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_fax","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_video","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_video","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_video","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_whiteboard","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_whiteboard","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_whiteboard","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_alert","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_alert","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_alert","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_voip","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_voip","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_voip","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_sms","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_sms","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_sms","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_EmplID_callback","length":256,"type":"string"},
{"encrypt":false,"name":"LCA_TimeStamp_callback","length":0,"type":"datetime"},
{"encrypt":false,"name":"Pref_EmplID_callback","length":256,"type":"string"}]

Query Service Extension Schema

	GET /metadata/services/extensions
	Available since: 8.0.200.00

Description

Queries an extension schema used for service resources. Your application must create a schema which defines the state extension before this extension can be used. For instance, if your application needs to fill in feedback data in some state, the first step is to create the corresponding schema which details the lists of attributes that compose this extension. See Extension for further details.

Operation

ID	CV.WS.SRV.META.2		
Method	GET		
URL	• all: /metadata/services/extensions • /metadata/services/extensions/**${extension-name}**		
Name	**Type**	**Mandatory**	**Description**
URI Parameter			
${extension-name}	string	no	The unique name of the extension. If you do not specify the name, all the service schema are returned. The name is **not** case-sensitive.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further

details on the possible codes that this operation can return.

Response	
HTTP code	200
HTTP message	OK
Body	Extension Schema or Extension Schema[]

Example

Retrieve all the service extension schemas

Operation

```
GET /metadata/services/
```

Result

```
  200 OK
 [
{
  "attributes": [
    {
      "default": "no discount",
      "encrypt": false,
      "length": 100,
      "mandatory": false,
      "name": "coupon",
      "type": "string",
      "unique": false
    },
    {
      "default": 5,
      "encrypt": false,
      "length": 0,
      "mandatory": true,
      "name": "amount",
      "type": "integer",
      "unique": false
    }
  ],
  "name": "couponExtension",
  "type": "single-valued"
 },
{
"name" : "relatedOffers",
"attributes" : [ {
  "unique" : false,
  "encrypt" : false,
  "mandatory" : true,
  "name" : "offer_name",
  "length" : 50,
  "type" : "string"
}, {
  "unique" : false,
  "encrypt" : false,
```

```
  "mandatory" : false,
  "name" : "type",
  "length" : 50,
  "type" : "string"
}, {
  "unique" : false,
  "encrypt" : false,
  "mandatory" : false,
  "name" : "comments",
  "length" : 1024,
  "type" : "string"
} ],
"type" : "multi-valued"
}
]
```

Retrieve a single schema

Operation
Retrieve the *couponExtension* schema.

```
 GET /metadata/services/couponExtension
```

Result

```
 200 OK
{
  "attributes": [
    {
      "default": "no discount",
      "encrypt": false,
      "length": 100,
      "mandatory": false,
      "name": "coupon",
      "type": "string",
      "unique": false
    },
    {
      "default": 5,
      "encrypt": false,
      "length": 0,
      "mandatory": true,
      "name": "amount",
      "type": "integer",
      "unique": false
    }
  ],
  "name": "couponExtension",
  "type": "single-valued"
 }
```

Query State Extension Schema

	GET /metadata/states/extensions
	Available since: 8.0.200.00

Description

Retrieves one or more schema for state extensions. This operation is possible in either maintenance and production mode.

Operation

Prerequisites: Universal Contact Server is in maintenance mode.

Query State Extension Schema

ID	CV.WS.SRV.META.6		
Method	GET		
URL	1. All: /metadata/states/extensions 2. /metadata/states/extensions/${extension-name}		
Name	**Type**	**Mandatory**	**Description**
URL Parameters			
${extension-name}	string	no	The unique name of the extension, to retrieve a single extension schema. If you do not specify this name, all the state extension schemas are returned. Note: The name is **not** case-sensitive.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	Extension Schema or Extension Schema[]

Examples

Several results

Operation

```
GET /metadata/states/extensions
```

Result

```
200 OK
[{
 "attributes": [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"Feedback",
 "type":"single-valued",
},
{
 "attributes": [
     {"name":"priority","type":"integer","mandatory":"true"},
     {"name":"customerNotes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"PriorityFlag",
 "type":"single-valued",
}
]
```

Single result

Operation

```
GET /metadata/states/extensions/feedback
```

Result

```
 200 OK
{
 "attributes": [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"Feedback",
 "type":"single-valued",
}
```

Query Task Extension Schema

	GET /metadata/tasks/extensions
	Available since: 8.0.200.00

Description

Retrieves one or more schema of task extensions. This operation is possible in either maintenance and production mode.

Operation

ID	CV.WS.SRV.META.8		
Method	GET		
URL	1. All: /metadata/tasks/extensions 2. /metadata/tasks/extensions/**${extension-name}**		
Name	**Type**	**Mandatory**	**Description**
URL Parameters			
${extension-name}	string	no	The unique name of the extension, to retrieve a single extension schema. If you do not specify this name, all the task extension schemas are returned. Note: The name is **not** case-sensitive.

<references />

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	200
HTTP message	OK
Body	Extension Schema or Extension Schema[]

Examples

Several results

Operation

```
GET /metadata/tasks/extensions
```

Result

```
200 OK
[{
 "attributes": [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"Feedback",
 "type":"single-valued",
},
{
 "attributes": [
     {"name":"priority","type":"integer","mandatory":"true"},
     {"name":"customerNotes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"PriorityFlag",
 "type":"single-valued",
}
]
```

Single result

Operation

```
GET /metadata/tasks/extensions/feedback
```

Result

```
 200 OK
{
 "attributes": [
     {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
     {"name":"rating","type":"integer","mandatory":"true"},
     {"name":"notes","type":"string","length":256,"mandatory":"false"}
     ],
 "name":"Feedback",
```

```
  "type":"single-valued",
}
```

Schema Resources

Purpose

Groups the Schema resources.

Attribute Schema

	Purpose: Describes an attribute schema, which can be part of an Extension Schema or of a Profile Schema.
	Available since: 8.0.100.00

Description

Your application can retrieve or manage the definitions of the attributes that compose some resources of the Context Services. Each attribute schema defines an attribute, its name, its type, and associated properties such as the default value of the attribute (if it exists). For instance, the following attribute schema defines that the Address Type attribute accepts integer values and is equal to 0 by default:

```
{"name":"AddressType","type":"integer","default":0}
```

Profile Attributes

Profile attributes correspond to a recorded schema which details the attribute name, type, and content. In the 8.0.1 release, the configuration of a new attribute schema is performed in the Configuration Manager. Select in "Tenant" the item *specific configuration*, and under "Business Attributes" the item 'Contact Attributes'. Then, create a new Business Attribute Value. By default, the new Attribute is of type string. To assign another type to the schema, in Annex, create a section "settings", and a key "type" with the value "string" or "date". The default predefined attributes visible in the Configuration Server are the following:

- First Name (string)
- Last Name (string)
- Phone Number (string)
- E-mail Address (string)
- Title (string)

Default 'internal' Attributes (not visible in the Configuration Server) are the following:

- LastCalledAgent_EmployeeID (string)
- LastCalledAgent_TimeStamp (date)
- PreferredAgent_EmployeeID (string)

Additional attributes are available for each active media defined in the Business Attribute "Media Type" of the Configuration Manager:

- LCA_EmplID_<media> (string)

- LCA_TimeStamp_<media> (date)
- Pref_EmplID_<media> (string)

Attributes of type date/time are formatted in ISO 8601 UTC <ref>Wikipedia ISO 8601</ref>: [YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z
Example: "1981-03-17T02:00:00.000Z"

Service, State, and Task extension attributes

In 8.0.2, your application can define service, states, and tasks extension resources, based on extension schema which use an array of Attribute Schemas to define the content of the extension. See Extension Schema for further details.

Resource

Attribute Schema

Field	Type	Mandatory	Description
name	string	yes	The name is unique for each attribute within the given extension. • Starts with a letter, and can be followed with letters, numbers, or underscores. • Is restricted to a maximum of 30 characters (Maximum RDBMS<ref name= "RDBMS">http://en.wikipedia.org/wiki/Relational_database_management_s shared limit.) In profile attribute schemas, the name is **not** case-sensitive.
type	token • boolean • string • integer • long	yes	1. The constraints on these types (the minimum and maximum values supported for the integer type) vary with the RDBMS<ref name="RDBMS" />.

Field	Type	Mandatory	Description
	• double • date • dateTime • currency		2. Customer profiles are built on top of UCS' Contact Attributes, and support only String and DateTime attribute types. However, all attribute types are supported for profile extensions.
default	numeric or string	no	Provides a default value for the attribute. For numeric attributes, you do not need quotation marks. Quotation marks are for string, date, and datetime attributes. By default, the date/time format is ISO 8601 UTC <ref>Wikipedia ISO 8601</ref>:`[YYYY]-[MM]-[DD]T[HH]:[mm]:[ss].[SSS]Z`.
length	integer	no	Reserved to attributes of type `String`. Defines the maximum number of characters for the string. • Default length is 256 • Maximum length is 4000.
mandatory	boolean	no	• *false* by default; • *true* if the attribute is mandatory when inserting a new Profile Extension record.

<references />

Extension Schema

	Purpose: Describes the schema for any service, state, or task extension.

Description

Describes one of the Extension resources.

- Profile extensions are supported. Before your application can start adding extensions to a given profile, your application must define this extension's schema, which includes its nested attributes.
- Service extensions are supported and multi-valued extensions are allowed.

Resource

Profile Extension Schema

Field	Type	Mandatory	Description
name	string	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters (Maximum RDBMS shared limit on creation of index).
type	token • "single-valued" • "multi-valued" • "classifier"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer. • Multi-valued: multiple extension records can be associated with a

Field	Type	Mandatory	Description
			given customer. • Classifier: the single-valued extension has no attribute.
attributes	Attribute Schema[]	no	The array of *zero or more* attributes. Empty when type="classifier".
unique	string[]	no	Lists attributes which are unique in the scope of a given customer. The customer profile cannot include several extension records with identical values for these attributes. For example, let's consider a contact extension with the attributes "phone_num", "ext", "time_of_day". To ensure that a given customer does not have two contact extensions with the same phone number, set the following in the extension schema: unique = ["phone_num", "ext"] Each item of this array is either a single attribute, or an array of two or more attributes.

Service/State/Task Extension Schema

Schema for Service/State/Task Extensions

Name	Type	Mandatory	Description
name	string	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed with letters, numbers, or underscores (_). The name is restricted to a maximum of 26 characters.
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated

Name	Type	Mandatory	Description
			with a given customer. • Multi-valued: multiple extension records can be associated with any given customer.
attributes	Attribute Schema[]	yes	The attributes that define the extension.
unique	string[]	no	Lists attributes (separated with commas) which are unique in the scope of a given service, state, or task. The resource cannot include several extension records with identical values for these attributes. For example, let's consider a bill extension with the attribute "bill_id". To ensure that a given service does not have two bills with the same "bill_id", set the following in the extension schema: unique = ["bill_id"] Each item of this array is either a single attribute, or an array of two or more attributes.

Examples

Schema of a Single-Valued Extension

```
{
"attributes": [
    {"name":"FeedbackType","type":"string","length":"10","mandatory":"true"},
    {"name":"rating","type":"integer","mandatory":"true"},
    {"name":"notes","type":"string","length":256,"mandatory":"false"}
    ],
"name":"Feedback",
"type":"single-valued"
}
```

Schema of a Multi-valued Extension

```
  {
```

```
    "attributes": [
        {"name":"car type","type":"string","length":"50","mandatory":"true"},
        {"name":"price","type":"integer","mandatory":"true"},
        {"name":"seats","type":"integer"},
        {"name":"comments","type":"string","length":"1024"}
        ],
    "name":"Proposal",
    "type":"multi-valued"
  }
```

See Query State Extension Schema for further examples.

Schema of a Profile Extension

The following schema describes the Address extension.

```
"name":"Address",
"type":"single-valued",
"attributes": [
    {"name":"AddressType","type":"integer","default":0},
    {"name":"Address","type":"string","length":256},
    {"name":"City","type":"string","length":32},
    {"name":"County","type":"string","length":32},
    {"name":"PostCode","type":"string", "length":10},
    {"name":"Country","type":"string","length":32}
    ]
```

Identification Key

Purpose

Describes the Identification Key Resource.

Description

The Identification Key is a combination of attributes used to identify a customer. These attributes (one or more) belong to the Customer Profile or to its extensions. Your application is responsible for creating the identification keys that are alter use to identify customers. See Create Identification Key for further details.

Identification keys cannot be issued from both the profile and its extensions.

Some identification key samples:

- An identification key consisting of attributes "name" and "birthdate" in the Profile.
- A key based on attribute "pin" of a single-valued extension used to hold customer pass codes.
- A key based on attribute "number" of a multi-valued extension used to record the phone numbers we have seen a customer call from.

When the UCS receives a new customer profile and the associated extension data, it builds the indexing structures based on the specified identification keys, in order to ensure efficient customer identification.

Resource

Identification Key

Field	Type	Mandatory	Description
name	string	yes	The unique, case-insensitive name of the key. The name must start with a letter, and can be followed with letters, numbers, or underscores. The name is restricted to a maximum of 26 characters. (Maximum RDBMS shared limit on

Field	Type	Mandatory	Description
			creation of index.)
source	string	no	Can take on one of the following values: • *profile* (default value): indicates that the attributes specified in parameter "attribute" belong to the core customer profile. • The unique name of the given Extension, used to create the Identification Key. • If none, it means that the attributes specified in the parameter *attributes* are not sourced from either the profile or an extension. In this case, you must manage manually the identification table in UCS.
attributes	string[]	yes	The array names of one or more attributes of the extension or of the profile.
unique	bool	no	• *false* by default. • *true* to indicate that the given attributes of the profile or extension are unique across all customers.

Example

The following identification key is a phone number.

```
{ "name": "idPhone",
 "attributes": ["phone_number"]}
```

The following sample defines "FirstName" and "LastName" as a "primary key" instead of "indexed

key".

```
{"name":"idFirstNameLastName", "unique":true, "attributes":["FirstName","LastName"]}
```

The combination of both parameters is unique. If your application tries to insert twice the same values, the operation will return an error. The use of a "primary key" instead of an "indexed key" makes the identification queries faster. It also prevents from creating two profiles with the same email address for example.

Related operations

- Get Identification Keys
- Create Identification Key