



Trio API

Introduction

The Trio API is an API for 3rd party developers to view, add and update data in Trio. It provides programmatic access to Contact Center configuration data, to retrieve real time Contact Center data and to create, update and delete tasks in the Contact Center queue. In addition, it provides access to read, add and update subscriber data in the Company Directory. The API can for example be used for wallboard applications, to display contact center status such as current waiting time and queue length on a website or for provisioning of users.

About this Help

This help describes the functionality, syntax and formats used in the API for handling client requests/responses. More information about requests/responses can be found by clicking the resource links. All responses have link and data collections which can be seen in detail by clicking the Type Collection link.

Authentication and Authorization

Successful authentication (login) will provide an access token which must be provided in all further requests. The API user authorization rights are controller by Trio Admin. See the Login resource for details.

Contact Center configuration changes

Changes to the Contact Center data, e.g. agents, will automatically be activated by the Queue Process which occurs one minute after the latest successful change.

Content encoding

To control character encoding for a request, specify the encoding like 'Content-Type': 'application/json; charset=iso-8859-1'.

Failover

The Data collection (ApiDataDto) in the Api resource gives information about the hostname of the primary (primaryServer) and secondary (secondaryServer) Trio servers. If the system is running in failover mode, calls to the primary server will be redirected (307 Temporary Redirect) and vice versa when running in normal mode. The client must then reauthenticate itself against the secondary server to get a new access token. Redirection can be turned off by specifying the query string '?redirect=false'

Changes to directory data (e.g. creating or updating subscribers) is disabled while the system is operating in failover mode. Reading existing directory data is however still possible when communicating with the secondary server.

Filtering

Two types of filtering exists. Either filtering on the resource itself (1) or the actual data in the resource response can be filtered (2).

1. Resources that return collections of data can be filtered by specifying the appropriate field name to the query string (see the 'Request Information' part in the resource page for field name), eg. `/api/cc/1/services?name=cust`

2. Collection fields (members) in the reply can be filtered by specifying the 'fields' query string field to a comma separated list of property names of the properties to be included in the collection reply. Eg.

`/api/cc/1/services/1/state?fields=open,servicename`. Most resources have default filtering enabled that can be changed this way.

Note that the two filtering types can be combined. Eg. `/api/cc/1/services?name=cust&fields=serviceid`

Links

All resources returns link collections that can be used to dynamically build a page tree.

Please do not hard code URIs as they may change in future releases.

Media types

Default media type is JSON (`application/json`). To tell the API that you want to work with XML, set both the HTTP 'Accept' and 'Content-Type' headers to `'application/xml'`.

To get indented (easier to read) responses when debugging/developing, specify the header `'Accept': 'text/html'`.

Paging

Resources that return collections with more than 10 elements will be using paging to reduce the amount of data sent in one reply.

The HTTP header will contain an 'X-Pagination' entry with 'currentPage', 'pageSize', 'totalCount', 'totalPages', 'previousPageLink' and 'nextPageLink' values. The default page size is 10.

Permission and Access Rights

The client will only get data that matches the permissions/access rights/groups set in Trio Administrator. Changes to the configuration of permissions/access rights/groups will be applied when the access token is refreshed, which should occur within an hour (default value), or when it is done manually.

Sorting

Resources that return collections of data can be sorted by the setting the 'sort' field to a comma separated list of the collection properties (member) names, eg. `/api/cc/1/services?sort=servicename`

Sorting can be reversed by preceeding the name with a '-' character.

URI encoding

When using non 7-bit ASCII characters in URIs, they need to be URL (percent) encoded using UTF-8 or ISO-8859-1. For example 'Smörgåsbord' -> `'Sm%C3%B6rg%C3%A5sbord'`

Resources

Api

API base URI, `http://[TESERVER]/te/api`

API	Description
GET	Information about the Api (no authentication required).

Login

The login request is used to obtain access and refresh tokens that must be used in subsequent requests. The request must have the API client key set as Basic authentication in the Authorization header. The API client key is an encrypted Base64 ISO-8859-1 encoded string that Enghouse will provide. Username and password for the API user (must be Base64 ISO-8859-1 encoded) are placed in the "form data set" as "username" and "password" with content type "application/x-www-form-urlencoded". When the client has obtained a valid access token it can do authenticated API requests with a bearer token using the 'Authorization' request header field with the 'Bearer' HTTP authorization scheme. The default access token lifetime is 3600 seconds (1 hour) and a new request should be made with the refresh token placed in the "form data set" as "refresh_token" to renew the access token before or after it has expired. The default maximum refresh token lifetime is 2592000 seconds (30 days) and the client must do a new login when the refresh token has expired.

API	Description
POST login	Get access and refresh tokens.

RefreshLogin

The refreshlogin request should be used to renew the access token before or after it has expired. The request must have the API client key set as Basic authentication in the Authorization header. The API client key is an encrypted Base64 ISO-8859-1 encoded string that Enghouse will provide. The refresh token is placed in the "form data set" as "refresh_token" with content type "application/x-www-form-urlencoded". When the client has obtained a new access token it can do authenticated API requests with a bearer token using the 'Authorization' request header field with the 'Bearer' HTTP authorization scheme. The default access token lifetime is 3600 seconds (1 hour). The default maximum refresh token lifetime is 2592000 seconds (30 days).

API	Description
POST refreshlogin	Renew access and refresh tokens.

Security

The security node is the base node for all access right related resources..

API	Description
GET security	List all accessible resource types in the security.

SecurityGroups

Security Groups Management.

API	Description
GET security/groups?subsystemId={subsystemId}&extId={extId}&groupId={groupId}&resourceTypeId={resourceTypeId}	Get list of security groups.
GET security/groups/{groupId}	Get a security group.

API	Description
POST security/groups	Create a new security group.
PATCH security/groups/{groupId}	Update a security group.
DELETE security/groups/{groupId}	Delete a security group.
POST security/groups/{groupId}/members	Create a new security group member.
DELETE security/groups/{groupId}/members/{memberId}	Delete a security group member.

SecurityAgents

Agent Rights Management.

API	Description
GET security/agents?subsystemId={subsystemId}&agents={agents}	Get list of agent rights.

SecurityServices

Services Rights Management.

API	Description
GET security/services?subsystemId={subsystemId}&services={services}	Get list of service rights.

ContactCenters

Contact Center management.

API	Description
GET cc?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}	Get all Contact Centers.
GET cc/{ccid}?fields={fields}	Get a specific Contact Center.

Agents

Contact Center agent management.

API	Description
GET cc/{ccid}/agents?page={page}&pageSize={pageSize}&sort={sort}	Get all Contact Center agents.

API	Description
<code>{sort}&id={id}&firstName={firstName}&lastName={lastName}&fields={fields}</code>	
<code>GET cc/{ccid}/agents/{agentid}?fields={fields}</code>	Get a specific Contact Center agent.
<code>PATCH cc/{ccid}/agents/{agentid}</code>	Update Contact Center agent configuration.

AgentState

Get current state for Contact Center agents.

API	Description
<code>GET cc/{ccid}/agents/state?page={page}&pageSize={pageSize}&sort={sort}&id={id}&fields={fields}</code>	Get current state for all Contact Center agents.
<code>GET cc/{ccid}/agents/{agentid}/state?fields={fields}</code>	Get current state for a specific Contact Center agent.
<code>PATCH cc/{ccid}/agents/{agentid}/state</code>	Set current state for a specific Contact Center agent.\n Note! The used agentID must be logged in by request PATCH 'cc/{ccid}/agents/{agentid}/login' before this resource can be used.
<code>PATCH cc/{ccid}/agents/{agentid}/login</code>	Set logon state for a specific Contact Center agent.

AgentGroups

Contact Center agent group management.

API	Description
<code>GET cc/{ccid}/agentgroups?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}</code>	Get all Contact Center agent groups.
<code>GET cc/{ccid}/agentgroups/{id}?fields={fields}</code>	Get a specific Contact Center agent group.

Services

Contact Center service management.

API	Description
GET cc/{ccid}/services?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&groupId={groupId}&fields={fields}	Get all Contact Center services.
GET cc/{ccid}/services/{serviceid}?fields={fields}	Get a specific Contact Center service.

ServiceState

Get current state for Contact Center services.

API	Description
GET cc/{ccid}/services/state?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}	Get current state for all Contact Center services.
GET cc/{ccid}/services/{serviceid}/state?fields={fields}	Get current state for a specific Contact Center service.

ServiceGroups

Contact Center service group management.

API	Description
GET cc/{ccid}/servicegroups?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}	Get all Contact Center service groups.
GET cc/{ccid}/servicegroups/{id}?fields={fields}	Get a specific Contact Center service group.

Cases

Contact Center case management.

API	Description
GET cc/{ccid}/services/cases?page={page}&pageSize={pageSize}&sort={sort}&vid={vid}&jobno={jobno}&cust={cust}&sid={sid}&caseClass={caseClass}&caseType={caseType}&fields={fields}	<p>Get all cases.</p> <p>Note! The response can contain up to a maximum of 20 cases per service. A specific case can however always be retrieved until the case is finished.</p>

API	Description
GET cc/{ccid}/services/cases/{jobno}?fields={fields}	Get a specific case. Note! If this request is issued directly after a POST for creation of a 'Task' an http status code 404 may be returned. In that case the request should be re-sent after a couple of seconds.
POST cc/{ccid}/services/{serviceid}/cases	Creates a case. Note! Only applies to case of type 'Task'.
PATCH cc/{ccid}/services/cases/{jobno}	Updates a case. Note! Only applies to case of type 'Task'.
DELETE cc/{ccid}/services/cases/{jobno}	Deletes a case. Note! Only applies to case of type 'Task'.

Roles

Contact Center role management.

API	Description
GET cc/{ccid}/roles?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}	Get all Contact Center roles.
GET cc/{ccid}/roles/{roleid}?fields={fields}	Get a specific Contact Center role.

Schedules

Contact Center schedule management.

API	Description
GET cc/{ccid}/schedules?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&exceptionListID={exceptionListID}&fields={fields}	Get all Contact Center schedules.
GET cc/{ccid}/schedules/{scheduleid}?fields={fields}	Get a specific Contact Center schedule.

ScheduleExceptions

Contact Center schedule exception management.

API	Description
GET cc/{ccid}/schedules/exceptionlists?page={page}&pageSize={pageSize}&sort={sort}&id={id}&name={name}&fields={fields}	Get all Contact Center schedule exception lists.

API	Description
GET cc/{ccid}/schedules/exceptionlists/{id}?fields={fields}	Get a specific Contact Center schedule exception (list).
GET cc/{ccid}/schedules/exceptionlists/{id}/exceptions?page={page}&pageSize={pageSize}&sort={sort}&fields={fields}	Get all Contact Center schedule exceptions for a list.
GET cc/{ccid}/schedules/exceptionlists/{listid}/exceptions/{id}?fields={fields}	Get a specific Contact Center schedule exception.

Skills

Contact Center skill management.

API	Description
GET cc/{ccid}/skills?page={page}&pageSize={pageSize}&sort={sort}&userId={userId}&roleId={roleId}&serviceId={serviceId}&fields={fields}	Get all Contact Center skills.
GET cc/{ccid}/skills/{uid}	Get a specific Contact Center skill.
POST cc/{ccid}/skills	Create a Contact Center skill.
PATCH cc/{ccid}/skills/{uid}	Update a Contact Center skill.
DELETE cc/{ccid}/skills/{uid}	Delete a Contact Center skill.

SkillProfiles

Get Contact Center skill profile management.

API	Description
GET cc/{ccid}/skillprofiles?page={page}&pageSize={pageSize}&sort={sort}&profileId={profileId}&name={name}&fields={fields}	Get all Contact Center skill profiles.
GET cc/{ccid}/skillprofiles/{profileId}?fields={fields}	Get a specific Contact Center skill profile.
POST cc/{ccid}/skillprofiles	Create a Contact Center skill profile.
PATCH cc/{ccid}/skillprofiles/{profileId}	Update a Contact Center skill profile.

API	Description
DELETE cc/{ccid}/skillprofiles/{profileid}	Delete a Contact Center skill profile.

SkillProfileItems

Contact Center skill profile items management.

API	Description
GET cc/{ccid}/skillprofiles/{profileid}/skillprofileitems?page={page}&pageSize={pageSize}&sort={sort}&roleId={roleId}&serviceId={serviceId}&fields={fields}	Get all Contact Center skill profile items.
GET cc/{ccid}/skillprofiles/{profileid}/skillprofileitems/{uid}	Get a specific Contact Center skill profile item.
POST cc/{ccid}/skillprofiles/{profileid}/skillprofileitems	Create a Contact Center skill profile item.
PATCH cc/{ccid}/skillprofiles/{profileid}/skillprofileitems/{uid}	Update a Contact Center skill profile item.
DELETE cc/{ccid}/skillprofiles/{profileid}/skillprofileitems/{uid}	Delete a Contact Center skill profile item.

Directory

The directory node is the base node for all company directory related resources (e.g. subscribers).

API	Description
GET directory	List all accessible resource types in the directory.

Subscribers

Subscriber Management.

API	Description
GET directory/subscribers?page={page}&pageSize={pageSize}&sort={sort}&fields={fields}&keyword={keyword}&extension={extension}&firstName={firstName}&lastName={lastName}&externalId={externalId}&systemId={systemId}&custGroup={custGroup}	List subscribers (or a sub set by applying appropriate filter conditions).

API	Description
{custGroup}&custGroupId={custGroupId}&cardKey={cardKey}&extra1={extra1}&extra2={extra2}&extra3={extra3}&extra4={extra4}&extra5={extra5}&extra6={extra6}&extra7={extra7}&extra8={extra8}&extra9={extra9}&extra10={extra10}&extra11={extra11}&extra12={extra12}&extra13={extra13}&extra14={extra14}&extra15={extra15}&extra16={extra16}&extra17={extra17}&extra18={extra18}&extra19={extra19}&extra20={extra20}&subscriberId={subscriberId}&properties={properties}	
GET directory/subscribers/{subscriberId}	Get a specific subscriber.
POST directory/subscribers	Create a new subscriber.
PATCH directory/subscribers/{subscriberId}	Update a subscriber.
DELETE directory/subscribers/{subscriberId}	Delete a subscriber.

DirectorySkills

Directory Skills Management.

API	Description
GET directory/skills	Get list of directory skills.
GET directory/skills/{skillId}	Get a directory skill.
POST directory/skills	Create a new directory skill.
DELETE directory/skills/{skillId}	Delete a directory skill.

Capabilities

Capabilities Management.

API	Description
GET directory/capabilities	Get list of Capabilities.

Departments

Departments Management.

API	Description
GET directory/departments	Get list of Departments.
GET directory/departments/{departmentId}	Get a department.
POST directory/departments	Create an new department.
DELETE directory/departments/{departmentId}?recursive={recursive}	Delete a department.

Pbxs

PBX Management.

API	Description
GET directory/pbx s	Get list of PBX.
GET directory/pbx s/{pbxId}	Get a PBX.
POST directory/pbx s	Create a new PBX.
PATCH directory/pbx s/{pbxId}	Update a PBX.
DELETE directory/pbx s/{pbxId}	Delete a PBX.

Calls

Controller that handles (voice) calls. Note! An agent must be logged in before one can use the REST resource MakeCall, POST 'cc/{ccid}/services/{serviceid}/calls'.

API	Description
GET cc/{ccid}/services/calls?page={page}&pageSize={pageSize}&id={id}&service={service}&fields={fields}	Retrieves the current state of all call cases.
GET cc/{ccid}/services/calls/{jobno}	Retrieves the current state of a call case.
POST cc/{ccid}/services/{serviceid}/calls	Creates and starts a new outgoing call. Note! The used agentID must be logged in by request PATCH 'cc/{ccid}/agents/{agentId}/login' before this resource can be used.
PATCH cc/{ccid}/services/calls/{jobno}	Updates a call to start or stop recording.
DELETE cc/{ccid}/services/calls/{jobno}	Delete a call

Calendar

Calendars Management.

API	Description
GET directory/calendarsystems	Get list of calendars.
GET directory/calendarsystems/{calendarId}	Get a calendar.
POST directory/calendarsystems	Create a new calendar.
PATCH directory/calendarsystems/{calendarId}	Update a calendar.
DELETE directory/calendarsystems/{calendarId}	Delete a calendar.

CustomerGroups

Customer groups management.

API	Description
GET directory/customergroups	Get list of customer groups.

Titles

Titles Management.

API	Description
GET directory/titles	Get list of titles.
GET directory/titles/{titleId}	Get a title.
POST directory/titles	Create a new title.
DELETE directory/titles/{titleId}	Delete a title.

Extensions

Extension Management.

API	Description
GET directory/subscribers/{subscriberId}/extensions	Get subscribers list of extensions.
GET directory/subscribers/{subscriberId}/extensions/{extensionId}	Get a subscribers specific extension.

API	Description
POST directory/subscribers/{subscriberId}/extensions	Create a new extension.
PATCH directory/subscribers/{subscriberId}/extensions/{extensionId}	Update a extension.
DELETE directory/subscribers/{subscriberId}/extensions/{extensionId}	Delete a extension.

ExtraNames

ExtraName Management.

API	Description
GET directory/extrafieldnames	Get list of extra names.
GET directory/extrafieldnames/{extraNameId}	Get an extra name.
POST directory/extrafieldnames	Create an new extra name.
PATCH directory/extrafieldnames/{extraNameId}	Update an extra name.
DELETE directory/extrafieldnames/{extraNameId}	Delete an extra name.

Referrals

Referral Management.

API	Description
GET directory/subscribers/{subscriberId}/referrals	Get subscribers list of Referrals.
GET directory/subscribers/{subscriberId}/referrals/{referralId}	Get a subscribers specific Referral.
POST directory/subscribers/{subscriberId}/Referrals	Create a new Referral.
PATCH directory/subscribers/{subscriberId}/Referrals/{ReferralId}	Update a Referral.
DELETE directory/subscribers/{subscriberId}/Referrals/{ReferralId}	Delete a Referral.

MessageChannels

Extension Management.

API	Description
GET directory/subscribers/{subscriberId}/messagechannels	Get subscribers list of message channels.
POST directory/subscribers/{subscriberId}/messagechannels	Create a new message channel.
DELETE directory/subscribers/{subscriberId}/messagechannels/{channelId}	Delete a message channel.
PUT directory/subscribers/{subscriberId}/messagechannels	Send message to message channels.
PUT directory/subscribers/{subscriberId}/messagechannels/{channelId}	Send message to a specifik message channel.

AddressSpaces

PBX Management.

API	Description
GET directory/addressspaces	Get list of adress spaces.
GET directory/addressspaces/{asId}	Get a adress space.