

API for Tickets

Overview

An API to list, create, and update [Tickets](#) in Administration.

Calls to the API must be authenticated with either HTTP basic authentication or [OAuth 2.0 Bearer Authentication](#).

In addition to general ticket management, we provide additional specific APIs to work with data within tickets, for example [API for Product Modeling](#).

API General Info

Response Content Type

The API only supports the `xml` response type.

Successful Responses

Response Codes

- **200** - The request was executed successfully and the server returned some data. The return content type is specified inside a response header called `content-type`. Example content types are `application/xml`, `application/json`, `application/zip` and `text/plain`.
- **204** - The request was executed successfully and the server returned no data. If the `Location` header was present then the server responded with a redirect.

Error Responses

If the API call resulted in an error then the error code will be returned along with the error object in the response body. The error object contains the attributes listed below:

- **type** - type of the error
- **cause** - the root cause of the error
- **message** - message containing details of the related error
- **uuid** - a unique identifier of the error which also may appear in the exception logs
- **timestamp** - the time of the error

Note: For chunked responses, the socket will be closed prematurely if an error occurs after the initial chunk containing the status code has been sent.

Response Codes

- **400** - The request contains invalid parameters or there are invalid settings. Consult the error message and/or check Administration for invalid settings.
- **401** - The user is not authenticated.
- **403** - The user is not authorized access to the specified endpoint.
- **404** - The specified endpoint does not exist or a request parameter is invalid. Consult the error message.
- **405** - The specified endpoint does not support the requested operation. Consult the error message.
- **409** - The specified endpoint is in a state incompatible with the requested operation. Consult the error message.
- **500** - Unknown error occurred while processing the request. Consult the CPQ logs for more information.
- **504** - The specified concurrent calls limit was exceeded and the server was not able to serve the response in a specified timeout limit. That error is always served with the xml body containing the description.

Ticket Management

List Tickets

Get a list of tickets in the system.

```
GET /api/ticket/list
```

Optional Parameters

Name	Usage	Default
where	Filter the tickets you want to list. For example <code>?where=status=in%20progress&where=summary~eto%20ticket</code> would include only tickets with the attribute 'status' set to 'in progress' and 'summary' set to some case insensitive variant of 'ETO ticket'. The comparison is case insensitive if the character ~ is used instead of =. The allowed filter keys are <code>summary</code> , <code>priority</code> , <code>description</code> , <code>owner</code> , and <code>status</code> .	

Example Response

```
<tickets>
<ticket id="T-00000002" summary="Add a new line of products"/>
<ticket id="T-00000001" summary="Update product pricing model"/>
</tickets>
```

Get Ticket

Get the details of a single ticket from the system.

```
GET /api/ticket/[ticket_id]
```

Attribute	Type	Required	Description
ticket_id	string	yes	ID of a ticket

Example Response

```
<ticket id="T-00000042" modified-by="5d033f233e924f6f81de69655b38ab1a"
  <attributes>
    <summary>Update product pricing model</summary>
    <description>This ticket adds another column to the product pricing m
    <status>in progress</status>
    <base-version>4</base-version>
    <priority>Major</priority>
    <owner>5d033f233e924f6f81de69655b38ab1a</owner>
  </attributes>
  <collaborators>
    <collaborator>5d033f233e924f6f81de69655b38ab1a</collaborator>
  </collaborators>
  <comments/>
  <attachments/>
  <transitions>
    <transition id="1">
      <target>done</target>
      <title>Mark as done</title>
    </transition>
    <transition id="6">
      <target>discarded</target>
      <title>Discard</title>
    </transition>
  </transitions>
</ticket>
```

Create Ticket

Create a ticket in the system by posting details in XML format.

```
POST /api/ticket
```

XML Format

```
<ticket>
  <attributes>
    <summary>This is a cute little test</summary>
    <description></description>
    <priority>Minor</priority>
    <base-version>1</base-version>
  </attributes>
</ticket>
```

Attribute	Type	Required	Description
summary	string	yes	The summary text of the ticket.
description	string	yes	The ticket description.
priority	string	yes	The ticket priority, is one of Blocker , Critical , Major , Minor , and Trivial .
base-version	integer/string	no	Optional base version for the ticket. The latest available version will be used if no base version is specified.

Example Response

```
HTTP/1.1 204 No Content
Location: /ticket/T-00000003
```

Update Ticket

Updates a ticket by posting an XML structure describing the attributes of the ticket.

```
PUT /api/ticket/[ticket_id]
```

XML Format

```
<ticket>
<attributes>
<summary>This is a cute little test</summary>
<description></description>
<priority>Minor</priority>
</attributes>
</ticket>
```

Attribute	Type	Required	Description
ticket_id	string	yes	ID of a ticket
summary	string	yes	The summary text of the ticket.
description	string	yes	The ticket description.
priority	string	yes	The ticket priority, is one of Blocker , Critical , Major , Minor , and Trivial .

Example Response

```
HTTP/1.1 204 No Content
Location: /ticket/T-00000003
```

Transition Ticket

Changes the state of a ticket by taking a transition. The transition has to be one of the transitions currently available for the ticket.

```
POST /api/ticket/[ticket_id]/transition/[transition-id]
```

Attribute	Type	Required	Description
ticket_id	string	yes	ID of a ticket
transition_id	string	yes	ID of a transition

Example Response

```
HTTP/1.1 204 No Content
Location: /ticket/T-00000003
```

Integration Attributes

List Integration Attributes

Lists all integration attributes in a ticket.

```
GET /api/ticket/[ticket_id]/integration-attributes
```

Attribute	Type	Required	Description
ticket_id	string	yes	ID of a ticket

Example Response

```
<integration-attributes ticket-id="T-00000028">
  <integration-attribute name="A1">
    <source part="root.part11" attribute="attr" value="name"/>
    <source part="root.sub.part12" property="prop"/>
  </integration-attribute>
  <integration-attribute name="A2">
    <source part="root.part21" attribute="attr" value="description"/>
    <source part="root.sub.part22" property="prop"/>
  </integration-attribute>
</integration-attributes>
```

Add or remove Integration Attributes in Ticket

Adds or removes integration attributes in the ticket. Duplicates are not permitted. Note that posting an empty list with the parameter 'removeNotReplaced=true' effectively removes all integration attributes in the ticket.

```
POST /api/ticket/[ticket_id]/integration-attributes[?removeNotReplaced=true]
```

XML Format

```

<integration-attributes>
<integration-attribute name="A1">
<source part="root.part11" value="name" attribute="attr"/>
<source part="root.sub.part12" property="prop"/>
</integration-attribute>
<integration-attribute name="A2">
<source part="root.part21" value="description" attribute="attr"/>
<source part="root.sub.part22" property="prop"/>
</integration-attribute>
</integration-attributes>

```

Attribute	Type	Required	Description
ticket_id	string	yes	ID of a ticket
name	string	yes	The name of the integration attribute
part	string	yes	The path to the part holding the value of the integration attribute
attribute	string	if 'property' is not specified	The attribute on the part, can be 'type' or the name of an attribute
value	string	if 'attribute' is specified, otherwise ignored	Can be '', 'name' or 'description'
property	string	if 'attribute' is not specified, otherwise ignored	The auxiliary property on the realized part or component

Parameter	Type	Required	Description
removeNotReplaced	boolean	no	Removes all not replaced records when value is set to 'true', applies to both integration attributes and their sources

Example Response



```
HTTP/1.1 204 No Content
Location: /ticket/T-00000028
```

HTTP Client Helper Features

Some HTTP Clients have quirks that make this API difficult to use. In an attempt to make it easier to use this API together with such clients, a few extensions are available:

Override HTTP Method

If your HTTP Client cannot use the HTTP Method required by some endpoint, you can just use any other method supported by your client and specify the **X-HTTP-Method-Override** header. The API will use the method value from the header if it is specified.

Example request (interpret as PATCH method):

```
GET /api/example HTTP/1.1
X-HTTP-Method-Override: PATCH
```

XML Response Instead of Header

If your HTTP Client doesn't support HTTP Status Code 204 (No Content) and doesn't give you access to the HTTP response headers, you can specify the **X-Response-As-Body** header (any value) and the API will respond with HTTP Status 200 instead. The API will also include the value of the Location header in an XML format instead.

Example Request

```
POST /api/create-something HTTP/1.1
X-Response-As-Body: 1
```

Example Response

```
<success location="/api/new_thing" />
```

Troubleshooting

List and Add Changes to the Ticket

This part of the API for Tickets has been deprecated and is no longer available to use.