

Staff API Fetching Information


Created by Unknown User (drew.lederman), last modified by Unknown User (akashdeep.sethi) on 28/05/15

- Client request
 - POST arguments
- Server response
 - Root node
 - Product info
 - Ticket data
 - Ticket counts
 - Ticket filters
 - Department listing
 - Status listing
 - Priority listing
 - Type listing
 - Email queues
 - Tickets macros
 - Custom fields
 - Chat data
 - Departments
 - Chat skills
 - Canned responses
 - Staff groupsStaff
 - Staff user listing
 - Avatars
 - Closing root node

Fetches information about the desk, such as departments, statuses, email queues, staff info, etc.

API supported in version > v4.60

Client request

-  To retrieve desk information, perform an **HTTP POST** request to the following URL:
- staffapi/index.php?/Core/Default/GetInfo

POST arguments

Argument name	Value
sessionid	The unique session identifier sent in the log in response from the server
wantticketdata	Specifies whether the response should include ticket related data such as departments, statuses, priorities, etc. (0 or 1; default = 1)
wantmacros	Specifies whether the response should include macro data (0 or 1; default = 1) Note: if <i>wantticketdata</i> =0, this argument has no effect.
wantchatdata	Specified whether the response should include live chat related data such as departments, canned responses, etc. (0 or 1; default =1)
wantstaff	Specifies whether the response should include staff data (0 or 1; default = 1)

Argument name	Value
<code>wantavatars</code>	Specifies whether the response should include staff avatar data (0 or 1, default 0)

Server response

A standard [server response](#) containing a UTF-8 encoded XML payload in the following format:

Root node

The parent node for the entire XML payload. If you do not locate this node at the top level, it is likely that the server returned an error response. See [error codes](#) for further information.

```
<kayako_staffapi>
```

Product info

```
<uniqueid>...</uniqueid>
<version>...</version>
<product>...</product>
<companyname>...</companyname>
```

Node name	Value
<code>uniqueid</code>	The unique license identifier
<code>version</code>	The human-readable form of the server's version (e.g. '4.0.0'). You may use this to test for feature compatibility.
<code>product</code>	The product name, e.g. 'Fusion'
<code>companyname</code>	The company name specified in settings, e.g. 'Kayako Infotech Ltd.'

Ticket data

Ticket counts

A list of all relevant ticket counts based on the assigned departments of the current user.

```
<ticketsummary inbox="..." mytickets="..." unassigned="..." trash="..." />
<ticketcount departmentid="..." statusid="..." new="..." count="..." />
```

Node name	Purpose	Attributes
<code>ticketsummary</code>	A summary of ticket counts that match certain criteria	inbox, mytickets, unassigned, trash: A numerical value specifying the number of tickets in each group

Node name	Purpose	Attributes
	Only sent for departments or statuses with a count of 1 or more tickets; departments or statuses without any tickets won't be included.	department identifier. statusid : The unique status identifier. If 0, this count is for an entire department (see departmentid), regardless of status. new : 1 if there are new tickets since the last update; 0 otherwise count : The total number of tickets

Ticket filters


A list of available ticket filters. Ticket filters are used by the client to request a set of tickets from the server that meet certain criteria.

```
<ticketfilter id="..." title="..." />
```

Node	Value	Attributes
ticketfilter	None; attributes contain all data	id : The unique numeric identifier of this ticket filter title : The human-readable title of this ticket filter (e.g. 'My Open Tickets')

Department listing

A list of departments and their sub-departments. Your application should store this data in persistent storage upon each newly performed log in operation.

 Departments can be nested, but only up to one level deep. (i.e. a top-level department can have children, but the child departments cannot)

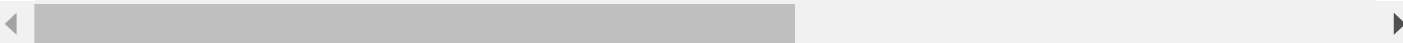
```
<department id="..." title="..." assigned="..." parentdepartmentid="0" >  
  <department ... />  
</department>
```

Node name	Value	Attributes
department	None; contains information about a department and its sub-departments	id : The unique numeric identifier of this department. title : The title of this department (e.g. 'Sales') assigned : 1 if the authenticated staff user has membership to the department, 0 otherwise parentdepartmentid : This will be the ID of the parent department, although the sub department will be nested, this value can be used to do quick lookups.

Status listing

A list of all statuses that a ticket may be assigned. Each department has its own list of statuses which is reflected in the departmentid attribute.

```
<ticketstatus id="..." departmentid="..." title="..." fgcolor="..." bgcolor="..." displayon
```



Node name	Value	Attributes
status	None; attributes contain all data	departmentid : The unique numeric identifier of the department to which this status belongs, or 0 if the status is department-agnostic title : The human-readable title of this status (<i>e.g.</i> 'Pending Reply') fgcolor : Hexadecimal web format of the color to use when rendering the text (foreground) of this status in a GUI. (<i>e.g.</i> '#AABBCC') bgcolor : Hexadecimal web format of the color to use when rendering the background of this status in a GUI. displayorder : When sorting statuses, use this numeric value (ascending) markasresolved : 1 if a ticket with this status is considered 'resolved,' 0 otherwise. canchangestatus : 1 if the authenticated user has permission to set tickets to this status, 0 otherwise. iconurl : The HTTP URL of the icon that can optionally be displayed to represent this status in a GUI; PNG format, 16x16 pixels.

Priority listing

A list of priorities that tickets may be assigned.

<code><ticketpriority id="..." title="..." fgcolor="..." bgcolor="..." displayorder="..." iconurl="..." /></code>

Node name	Value	Attributes
priority	None; attributes contain all data	id : The unique numeric identifier of this priority title : The human-readable title of this priority (<i>e.g.</i> 'Urgent') fgcolor : Hexadecimal web format of the color to use when rendering the text (foreground) of this priority in a GUI. (<i>e.g.</i> '#AABBCC') bgcolor : Hexadecimal web format of the color to use when rendering the background of this priority in a GUI. displayorder : When sorting priorities, use this numeric value (ascending) iconurl : The HTTP URL of the icon that can optionally be displayed to represent this priority in a GUI; PNG format, 16x16 pixels.

Type listing

A list of all ticket types that a ticket may be assigned. A ticket type can be linked to a department which is reflected in the departmentid attribute.

<code><tickettype id="..." title="..." departmentid="..." displayorder="..." iconurl="..." /></code>
--

Node name	Value	Attributes
type	None; attributes contain all data	title : The human-readable title of this type (<i>e.g.</i> 'Feature Request') departmentid : The unique numeric identifier of the department to which this type belongs, or 0 if the type is department-agnostic displayorder : When sorting types, use this numeric value (ascending) iconurl : The HTTP URL of the icon that can optionally be displayed to represent this type in a GUI; PNG format, 16x16 pixels.

Email queues

A list of email queues that the client may use as its point of origination for new tickets or replies.

<code><emailqueue id="..." email="..." departmentid="..." /></code>

Node	Value	Attributes
emailqueue	None; attributes contain all data	id : The unique numeric identifier of this email queue email : The email address that will be used as the origination of emails sent from this queue departmentid : The unique numeric identifier of the department that this email queue is associated with

Tickets macros

A list of predefined macros that can be executed with a single click/tap.

- Unlimited nesting depth**
Note that categories and macros can be nested without limit. Make sure to recursively parse category nodes.

```
<macrocategory id="..." title="...">
  <macro id="..." creationdate="..." totalhits="..." lastusage="..." departmentid="..." own
  <macro id="..." creationdate="..." totalhits="..." lastusage="..." departmentid="..." own
</macrocategory>
```

Node name	Value	Attributes
macrocategory	None; represents a macro category	id : The unique numeric identifier of this macro category title : The title of this macro category (e.g. 'Greetings')
macro	Textual contents of the macro	id : The unique numeric identifier of this macro staffid : Unique numeric identifier of the staff member who created the macro creationdate : Unix time stamp of macro creation totalhits : Number of times the macro has been executed lastusage : Unix time stamp of the last time the macro was executed departmentid : The unique numeric identifier of the department to which the the ticket should be moved, or 0 if it should not be moved ownerstaffid : The unique numeric identifier of the staff member that should become the new owner of the ticket when the macro is executed, or 0 if no change in ownership should occur tikettypeid : The unique numeric identifier of the type the ticket should be marked as, or 0 if the type shouldn't be modified ticketstatusid : The unique numeric identifier of the status that the ticket should be set to, or 0 if the status shouldn't be modified priorityid : The unique numeric identifier of the priority that the ticket should be assigned, or 0 if the priority shouldn't be modified tags : A space-delimited list of tags associated with this macro

Custom fields

A listing of available custom field groups and fields within those groups. Individual tickets will contain information about the values of the fields in each group.

```
<customfieldgroup id="..." title="..." type="..." displayorder="...">
  <customfield id="..." title="..." type="..." isrequired="..." usereditable="..." staffedi
  ...
</customfield>
</customfieldgroup>
```

customfieldgroup	Contains a list of field nodes that describe the fields within the group.	id : Unique numeric identifier of the group. title : The human-readable title of the custom field group, (e.g. 'Account Information') type : The type (area of the product that the group is utilized), which is described in custom field group types . displayorder : When sorting groups in a GUI, use this numeric value (ascending).
customfield	The field's default value to use when none has been entered.	id : The unique numeric identifier of this field. The uniqueness of this value is irrespective of the parent group. title : The human-readable title of the field, (e.g. 'Account Number') type : The type of the field, which are described in custom field types . isrequired : 1 if the field must contain valid input before proceeding past the entry form, or 0 otherwise. usereditable : 1 if end users can change the value of the field after they've entered it for the first time, or 0 otherwise. staffeditable : 1 if staff users can change the value of the field after it has been first entered, or 0 otherwise. regex : If a regular expression@wikipedia is required to validate the input for the field, it is set here. displayorder : When sorting fields in a GUI, use this numeric value (ascending). desc : The human-readable description of the field, (e.g. 'Your XYZ Company account number')

Chat data

Departments

This section contains a list of live chat departments. The data should be stored for the duration of the client/server session.

-  Departments can be nested, but only up to one level deep. (i.e. a top-level department can have children, but the child departments cannot)

```
<chatdepartment id="..." title="..." assigned="...">
  <chatdepartment id="..." title="..." assigned="..." parentdepartmentid="..." />
</chatdepartment>
```

Node name	Value	Attributes
chatdepartment	May contain one or more chatdepartment nodes	id : Numeric ID of this department title : String title of this department (e.g. 'Sales') assigned : Whether or not the user is assigned to this department ('1' or '0') parentdepartmentid : Parent department's numeric ID, if this is a sub-department

Chat skills

This section contains chat skill information.

```
<chatskill id="...">...</chatskill>
```

Node name	Value	Attributes
chatskill	String title of this chat skill (e.g. 'Unix expert')	id: Numeric ID of this chat skill

Canned responses

Canned responses are organized in a tree-like structure. The parent node, as well as child nodes, may contain any number of categories and responses, so the tree should be parsed recursively.

Canned data should be stored for the duration of the client/server session.

```
<canned>
  <category id="..." title="...">
    <response id="..." title="...">
      <url>...</url>
      <image>...</image>
      <code lang="...">...</code>
      <message>...</message>
    </response>
  </category>
</canned>
```

Node name	Value	Attributes
canned	May contain one or more category or response nodes	(none)
category	May contain one or more category or response nodes	id: Numeric ID for this category title: String title for this category
response	May contain one of each of any of the following nodes: url, image, code, message (<i>canned responses may contain multiple 'actions'</i>)	id: Numeric ID for this response title: String title for this response
url	A URL to be pushed when the response is executed	(none)
image	A URL of an image to be pushed when the response is executed	(none)
code	A code snippet to be sent when the response is executed	lang: Type of code. See Language codes
message	A text message to be sent when the response is executed	(none)

Staff groups


This section contains staff group information.

```
<staffgroup id="..." title="..." />
```

Node name	Attributes
	title: String title of this staff group (e.g. <i>Administrators</i>)

Staff user listing

A list of staff users present in the server's database.

<code><staff id="..." firstname="..." lastname="..." fullname="..." username="..." designation="..."</code>


Node name	Value	Attributes
staff	None; attributes contain all data	id: The unique numeric identifier of this staff user firstname: This staff user's first name (e.g. <i>'Thomas'</i>) lastname: This staff user's last name (e.g. <i>'Paine'</i>) fullname: This staff user's full name (e.g. <i>'Thomas Paine'</i>) username: This staff user's user name on the helpdesk (e.g. <i>'tpaine'</i>) designation: This staff user's title or position within your organization (e.g. <i>'Vice President'</i>) email: This staff user's e-mail address mobilenumber: This staff user's mobile telephone number, without symbols or spaces (e.g. <i>'12085551212'</i>) lastvisit: Unix time stamp of this staff user's last login to the helpdesk timezone: The PHP timezone that this staff user has set in their settings departments: A comma-delimited list of unique department identifiers that this staff user has permission to access.

Avatars

Note: Avatar data will only be returned if the post argument 'wantavatars' is set to '1'

<code><avatar staffid="..." type="..."></code> <code><!-- base64 encoded data --></code> <code></avatar></code>

Node name	Value	Attributes
avatar	Base-64 encoded image data	staffid: Numeric ID of the staff user to whom this avatar belongs type: Type of image. One of the following: 'gif' , 'png' , 'jpg'

Closing root node

The final XML node is simply the end of the original root node, completing the XML document.

<code></kayako_staffapi></code>
