StaffAPI - Logging In

Created by Unknown User (ryan.lederman), last modified by Unknown User (akashdeep.sethi) on 09/05/14

- Client request
  - POST arguments
- Server response
  - Root node
  - Two Factor Key Request
  - Session data
  - Closing root node

Client request

To log in (authenticate the client), perform an HTTP POST (StaffAPI - Logging In) request to the following URL:

```
staffapi/index.php?/Core/Default/Login
```

POST arguments

Argument	Value
username	The UTF-8@Wikipedia-encoded username of the user to authenticate.
password	The user's password, urlencoded.
twofactorkey	Optional: The two factor authentication key (in plaintext). This value should be dispatched only when server requests it, which is when the staff user has enabled two factor authentication.
initialstatus	The initial user status to be set after logging in (for live chat only).

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>
```

Two Factor Key Request

The server will request a two factor authentication key if the staff user logging in has enabled two factor authentication. If your application receives this request, it should prompt the user for the two factor key and send the request again with the 'twofactorkey' POST argument as shown above.

```
<status>2</status>
<error>Please send the two factor authentication key to login.</error>
```

Node name	Value
status	The server will send '2' as the status value if the two factor authentication is enabled.
error	The error message returned by SWIFT. i.e., "Please send the two factor authentication key to login."

Session data

```
<status>...</status>
<error>...</error>
<version>...</version>
<sessionid>...</sessionid>
<sessiontimeout>...</sessiontimeout>
<staffid>...</staffid>
<uniqueid>...</uniqueid>
<option name="...">...</option>
<permission name="...">...</permission>
<variable name="...">...</variable>
```

Node name	Value
status	The status code of the login request. Any value other than '1' indicates failure.
error	If status does not equal '1', the error message returned by SWIFT. i.e., "Invalid username or password"
version	The human-readable form of the server's version (e.g. '4.0.0'). You may use this to test for feature compatibility.

Node name	Value
sessionid	The unique string identifier for this session. You should store this value, as it will be required in every other request.
sessiontimeout	The amount of time (in seconds) before the session will expire due to inactivity. You should use this to determine if your application needs to re-authenticate bef in order to avoid "session expired" errors.
staffid	The unique numeric identifier of the staff member that was authenticated *(You!)*
uniqueid	The unique license identifier
option	Represents the value of the option specified by the `name` attribute: **recordphonecalls**: Whether or not the client should send recordings of phone calls to the server (**1** or **0**). **chatqueuetype**: Type of chat queue in use by the server. One of the following: **`roundrobin`**, **`queue`**
permission	Represents the value of the permission specified by the `name` attribute (**1** or **0**): **observechats**: Whether or not the currently logged in staff is allowed to enter a chat as an observer.
variable	Represents the value of the variable specified by the `name` attribute: **maxuploadsize**: The maximum size, in octets, allowed when uploading files. Equivalent to upload_max_filesize in the server's PHP configuration.

Closing root node

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

```
</kayako_staffapi>
```