

The REST API provides programmatic access to read / write Symphony data. For instances:

1. Post a wall post
2. Create a chat room
3. Add users to a chat room
4. Search for messages in a chat room
5. Follows a user

Common Parameters

Parameters that are common to all the API endpoints

Name	Required / Optional	Description
appid	Required	AppID of your application
referrer	Optional	Origin of the request

Users

Operations related to users

HTTP Request	Parameters (if any)	Description
URIs relative to https://dev6-webcontroller.symphony.com/webcontroller, unless otherwise noted		
GET /api/v3/user_info		Retrieve the authenticated user's info
GET /search/Search?type=people	q: (required): search token	Search for a user. The value of the parameter q could be user id or pretty name or screen name
POST /maestro/followees	followee: (required) [{ "type": "DEFINITION", "definitionType": "USER_FOLLOW", "id": "412316860443", "text": "nicholas", "connectorId": "lc" }]	Follow a user
DELETE /maestro/followees/lc/{numeric user ID to unfollow}		Unfollow a user

Threads

Operations related to threads (also known as conversations)

HTTP request	Parameters (if any)	Description
URIs relative to https://dev6-webcontroller.symphony.com/webcontroller, unless otherwise noted		
POST <code>/api/v3/room_create</code>	room: (required) JSON representing the room to be updated. See sample below: { "name": "2awesome", "publicRoom": false, "memberAddUserEnabled": true, "discoverable": false, "readOnly": false }	Create a chatroom. publicRoom: defines the room type, public or private memberAddUserEnabled: determines if room members can add other users discoverable: determines if a room would show up in search results readOnly: determines whether a room is read only or not. Only owner can post in a read only room
POST <code>/api/v3/room_deactivate</code>	threadid: (required) Id of the room	Deactivate a room.
GET <code>/api/v3/room_search</code>	roomname: (optional) Name of the room. Must be an exact match showmember: (required) true if the response should include members info of the room	Search for a room based on the room name. If roomname is not specified, it returns all room the authenticated user is in.
POST <code>/api/v3/room_update</code>	threadid: (required) ID of the room room: (required) JSON representing the room to be updated. See sample below: { "name": "2awesome" }	Update a room. Only room owner can perform this action. Thread ID can be obtained through the <code>/room_search</code> endpoint
POST <code>/api/v3/room_user_remove</code>	threadid: (required) Id of the room to have the user removed userid: (required) numeric ID of the user to be removed. The numeric ID can be obtained using <code>/search/Search?type=people</code> endpoint	Remove an user in a room. Only room owner or the users themselves have the rights to perform this action
POST <code>/api/v3/room_user_add</code>	threadid: (required) Id of the room to have the user added userid: (required) numeric ID of the user to be added. The numeric ID can be obtained using <code>/search/Search?type=people</code> endpoint	Remove an user in a room. Only room owner or the users themselves have the rights to perform this action
POST <code>/api/v3/post_to_room</code>	parseentities: (required) true if text should be parsed for URLs, @mentions, keywords, etc. messagepayload: (required) JSON representing	Post to a chat room. Only room members can post to a room.

HTTP request	Parameters (if any)	Description
	<p>the message. See Sample below:</p> <pre>{ "sendingApp": "Ic", "text": "I love #hackaton", "version": "SOCIALMESSAGE", "threadId": "0+fUIMWoHi/VzN4/MogBzn///r4fRnRK" }</pre>	<p>sendingApp: the value must be “Ic”</p> <p>version: the value must be “SOCIALMESSAGE”</p> <p>threadId: ID of the room.</p>
POST /api/v3/post_to_wall	<p>parseentities: (required) true if text should be parsed for URLs, @mentions, keywords, etc.</p> <p>messagepayload: (required) JSON representing the message. See Sample below:</p> <pre>{ "sendingApp": "Ic", "text": "I love #hackaton", "version": "SOCIALMESSAGE" }</pre>	<p>Post to a chat room. Only room members can post to a room.</p> <p>sendingApp: the value must be “Ic”</p> <p>version: the value must be “SOCIALMESSAGE”</p> <p>threadId: ID of the room.</p>

Message

Operations related to messages

HTTP request	Parameters (if any)	Description
URIs relative to https://dev6-webcontroller.symphony.com/webcontroller, unless otherwise noted		
GET /api/v3/message_search_post_by	<p>fromuserid: (required) the numeric user ID of the message author. The numeric ID can be obtained using /search/Search?type=people endpoint</p> <p>startingrow: (optional) start row number of the result set returned in this batch</p> <p>maxrow: (optional) max row returned in this call, capped at 2000</p> <p>from: (optional) timestamp (milliseconds since the epoch)</p> <p>to: (optional) timestamp (milliseconds since the epoch)</p>	<p>Query messages that posted by the user id or a list of user id (separated by space)</p>
GET /api/v3/message_search_by_keyword	<p>keyword: (required) search token or a list of search tokens, separated by space character. e.g. #hackathon</p> <p>startingrow: (optional) start row number of the result set returned in this batch</p> <p>maxrow: (optional) max row returned in this call, capped at 2000</p> <p>from: (optional) timestamp (milliseconds since</p>	<p>Query messages that contain the specified keyword (hashtag / cashtag). The messages returned could be post, IMs or chat rooms messages that the authenticated user is entitled to view</p>

HTTP request	Parameters (if any)	Description
	<p>the epoch)</p> <p>to: (optional) timestamp (milliseconds since the epoch)</p>	
<pre>GET /api/v3/message_search_by_thread</pre>	<p>threadid: (required) id of the conversation or list of conversation separated by space charater.</p> <p>token: (optional) token contained in the message</p> <p>hashtag: (required) search token or a list of search tokens, separated by space character. e.g. #hackathon</p> <p>startingrow: (optional) start row number of the result set returned in this batch</p> <p>maxrow: (optional) max row returned in this call, capped at 2000</p> <p>from: (optional) timestamp (milliseconds since the epoch)</p> <p>to: (optional) timestamp (milliseconds since the epoch)</p>	<p>Query messages that contain the specified keyword (hashtag / cashtag). The messages returned could be post, IMs or chat rooms messages that the authenticated user is entitled to view</p>

Keyword

Operations related to keywords

HTTP request	Parameters (if any)	Description
URIs relative to https://dev6-webcontroller.symphony.com/webcontroller, unless otherwise noted		
<pre>POST /maestro/keywords</pre>	<p>keyword: (required)</p> <pre>[{ "type": "DEFINITION", "definitionType": "KEYWORD", "id": "#pico", "text": "#pico", "connectorId": "lc" }]</pre>	<p>Follow a keyword</p>
<pre>DELETE /maestro/keywords/lc /{keyword ID}</pre>		<p>Unfollow a keyword.</p> <p>Be sure to URL encode the keyword ID in the path parameter</p>