[PRD] Update User Status API

Target release	Q2 2025
Priority	P1
Objective	Support update user status via API
Summary	• SOP: provide an API for apps to call and update user's status • Core BE: support update status with API
Epic	⚡ **SPSTKSZ-84719** - Update User Status API `WAITING`
Document status	
Product Owner	Elaine Zhang
Developers	yingpeng.huang@shopee.com
QA	miao.zheng@shopee.com

Update History			
Date	**Version**	**By Whom**	**Description**
07 Apr 2025	1.0	Elaine Zhang	Creation

Background & Objective

Background

SeaTalk currently allows users to set their status to reflect availability. However, there are several limitations in how these statuses are updated:

1. **"In a meeting" status**: this status can currently only be updated manually, which requires extra effort from user. Users often have limited time when rushing into meetings, making it easy to forget or skip updating their status.

   - Auto-update this status from Google Calendar events is highly requested features by users.
   - With the releases of Google Calendar Bot, we've established the connection with Google Calendar. As the next step, we plan to support automatic status updates based on calendar events.

-
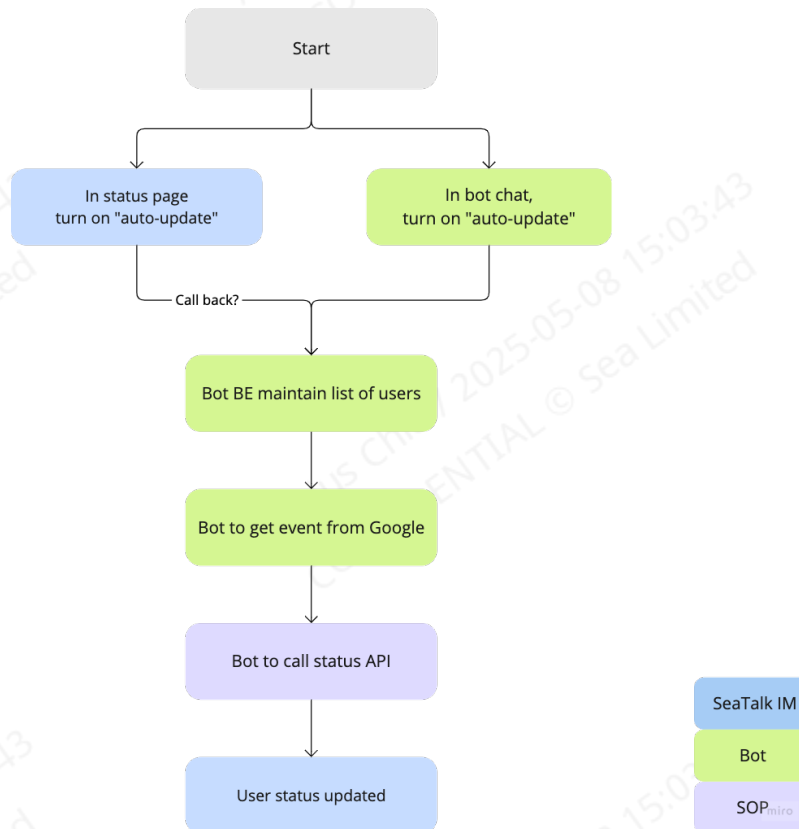
2. **Auto-update Public holiday status**: currently there is no API support for syncing public holiday data from our HRIS. As a result, any updates or changes to holiday schedules must be manually reflected in the SeaTalk database, which is a tedious and error-prone process that lacks scalability and automation.

Assumption: The auto-update status function is primarily intended for organization users (currently auto-update setting entry is hidden for non-org users)

Objective

A standardised, automated status update channel that can integrate seamlessly with 3rd party systems like HRIS and Google Calendar

Solution Overview
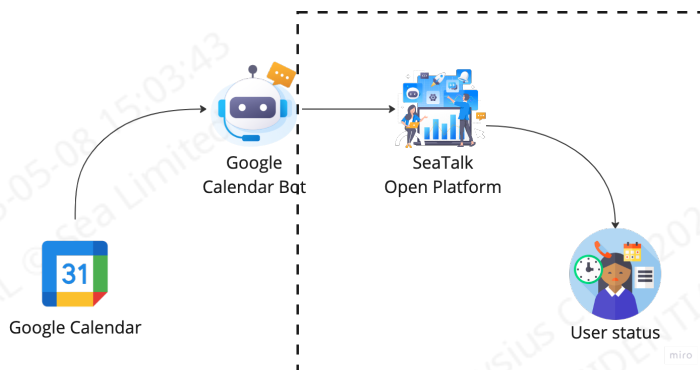
1. SOP to provide an API for status update
2. Status module to support status update by SOP API

Feature Requirement (SOP BE)

#	Title	Description
1	**New** API - "Update User Status"	A new SOP Open API, "**Update User Status**", enables a bot/an app to update users' statuses under its service scope

Checklist

Item	Decision
Admin approval	Required
API Visibility	All organisations
SOP Portal - Preset	No
SOP Portal - Display in List	Last one under "Employee" category • Description: Update a user's status • Limit by: Service scope
Bot Capability	Not Required

API Design:

Request Method: Post

End Point: https://openapi.seatalk.io/contacts/v2/update_status

----------------------- *Request* -----------------------

• Header

Parameter	Type	Mandatory	Description	Default	Sample
Authorization	string	Yes	Obtained through the Get App Access Token API	N/A	Bearer c8bda0f77ef940c5bea9f23b2d7fc0d8
Content-Type	string	Yes	Request header format	N/A	application/json

• Body

Parameter	Type	Mandatory	Length Limit	Description	Default
user_statuses	[]object	Yes	(TBD with tech)	List of user status	N/A
L employee_code	string	Yes	-	The employee code of the user	N/A
L status_text	string	Yes	Max: 15 characters Min: 1 character	The status text of the user Emoji is not allowed	N/A
L status_expiration	int	No	-	The expiry timestamp of the status, UNIX format 0 if no expiry time -1 if to withdraw the status set previously	0

- Request Body Sample

```
{
    "user_statuses": [
        {
            "employee_code": "e_12345678",
            "status_text": "In a meeting",
            "status_expiration": 1532627506
        },
        {
            "employee_code": "e_12345690",
            "status_text": "In a meeting",
            "status_expiration": 1532627500
        }
    ]
}
```

If to withdraw the status set earlier

```
{
    "user_statuses": [
        {
            "employee_code": "e_12345678",
            "status_text": "In a meeting",
            "status_expiration": -1
        },
        {
            "employee_code": "e_12345690",
            "status_text": "In a meeting",
            "status_expiration": -1
        }
    ]
}
```

------------------------ *Response* ------------------------

- Response Parameter

Parameter	Type	Description
code	int	Refer to Exception Handling section below. 0 if updating message successfully.
update_failed	[]object	List of employees whose status is failed to be updated
∟ code	int	Error code of the user whose status is failed to update
∟ employee_code	int	The employee_code of the user whose status is failed to update

- Response Sample

```
{
    "code": 0,
    "update_failed": [
        {
            "code": 3001,
            "employee_code": "9120"
        }
    ]
}
```

------------------------ *Exception Handling* ------------------------

API Rate Limit: https://confluence.shopee.io/x/owOjb

API Logic Checks:

1. The employee must under the App's service scope
2. status_text should not exceed 15 characters (otherwise treated as invalid input)
3. status_expiration timestamp has to be in the future or 0

Error Codes:

Category	Error Code	Description	Resolution
Generic error		Follows standard server API error code in [Tech] SOP Error Code Standardisation	
Update status specific	3001	User not found with the current code	Check whether the code belongs to a valid employee within the app's service/data scope
	102	Request body contains invalid input	Ensure that the request body has all the fields required and inputs them correctly
	2006	Invalid timestamp	Ensure that timestamp must be either a Unix timestamp in the future or 0/-1
	TBD by tech	Unable to update status	- Unable to override user's existing status - Unable withdraw the status with a different status_text

Feature Requirement (Core Backend)

	Title	Description
1	Status category and priority	**Category** 1. Platform auto-update status: On leave, Public holiday 2. User manual-update status 3. SOP open API-update status **Priority** Platform auto-update status **=** user manual-update status **>** API-update status • Between 2 platform statuses: On leave > Public holiday **Logics** 1. Statuses with higher priority can overwrite those with lower priority, but not the other way around 2. Status with the same priority can overwrite each other 3. ↻ indicator is only displayed for platform auto-update status

2	Support update user status via API	**Validation check**
		
		1. the user is an active user
		2. status_text should not exceed 15 characters
		3. existing validation of status_text applies
		   a. no blacklisted special character
		   b. no emoji
		4. status_expiration timestamp has to be in the future or 0/-1
		
		Update status
		
		1. Do not overwrite existing user status if any
		2. Indicate source of status in status DB table by using SOP `app_id`
		
		Match to suggested status
		
		• If `status_text` matches with any suggested status (exact match and case sensitive), mark it as a suggested status (such that status viewer can see status as per their language setting)
		
		Cancel the auto-updated status
		
		When "status_expiration" = -1
		
		• Check if `status_text` matches with existing status
		  • if yes, clear the existing status
		  • else, no change to the status
		
		Notify SOP of the status update result (success/fail) and error

Competitor References

	Update status	Get status

Slack	• Set a user's custom status by using their access token with `users.profile.set` and the `users.profile:write` scope.
	• To unset a user's custom status, set both `status_text` and `status_emoji` to empty strings: `""`.
	• <pre>"status_text": "riding a train",
	"status_emoji": ":mountain_railway:",
	"status_expiration": 0</pre>
		• Directly retrieve status for a specific user with `us‑s.profile:read` scope
		• A user's custom status is represented by the `stat‑ tatus_expiration` profile attributes.
		<pre>"status_text": "Watching cold brew stee‑
		"status_emoji": ":coffee:",
		"status_emoji_display_info": [],
		"status_expiration": 0,</pre>
		• To be notified when status and other profile fields `ge` events in the Events API or use the RTM API to

Mattermost	**Request**
	
	> Security: Bearer Auth
	
	Path Parameters
	
	`user_id` string — required
	User ID
	
	Body application/json ⌄
	
	Custom status object that is to be updated
	
	— `emoji` string — required
	Any emoji
	
	— `text` string — required
	Any custom status text
	
	— `duration` string
	Duration of custom status, can be `thirty_minutes`, `one_hour`, `four_hours`, `today`, `this_week` or `date_and_time`
	
	— `expires_at` string
	The time at which custom status should be expired. It should be in ISO format.
	
	• Permissions:
	  • Must have `edit_other_users` permission for the team.
	  • Must be logged in as the user whose custom status is being updated.
	
	``
	{
	  "emoji": "string",
	  "text": "string",
	  "duration": "string",
	  "expires_at": "string"
	}
	``
	
	https://developers.mattermost.com/api-documentation/#/operations/UpdateUserCustomStatus
		Get user status by id from the server.
		
		Permissions
		
		Must be authenticated.
		
		AUTHORIZATIONS: › *bearerAuth*
		
		PATH PARAMETERS
		
		⊣ `user_id` required — string
		User ID
		
		Responses
		
		⌄ 200
		User status retrieval successful
		
		RESPONSE SCHEMA: application/json
		
		⊣ `user_id` — string
		⊣ `status` — string
		⊣ `manual` — boolean
		⊣ `last_activity_at` — integer <int64>
		
		``
		{
		  "user_id": "string",
		  "status": "string",
		  "manual": true,
		  "last_activity_at": 0
		}
		``
		
		https://api.mattermost.com/#tag/status/operation/Ge‑