**Universal Task Documentation**

Universal Automation Center support for Slack

UAC-SLACK Integration

Associated Activities:

Date: 22 Dec 2019

Author: Ravi Kumar Murugesan

Revision: 01

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | 02-Aug-2020 | Ravi Kumar M | Draft |
| 0.2 | 04-Aug-2020 | Ravi Kumar M | Update Overall view |

**Abstract:**

This Universal Tasks allow to send notifications to a Slack Channel. As a result, you can integrate this solution with universal controller to notify users in slack channel appropriately on job failure, Late start/run or any event notifications.

Also this task comes with a approval functional for universal controller manual tasks types, So completing the manual task can be performed from approval notification you received in your slack channel.

Contents

[1 Disclaimer 3](#_Toc47440286)

[2 Scope 3](#_Toc47440287)

[3 Introduction 3](#_Toc47440288)

[3.1 Incoming Webhooks in Slack 3](#_Toc47440289)

[3.2 Implementation Details 4](#_Toc47440290)

[4 Installation 4](#_Toc47440291)

[4.1 Software Requirements for Linux Agent 4](#_Toc47440292)

[4.2 Installation Steps 4](#_Toc47440293)

[5 Universal Task Configuration 5](#_Toc47440294)

[6 Universal Tasks for Slack Notifications Forwarding 6](#_Toc47440295)

[6.1 Send message 6](#_Toc47440296)

[6.2 Approval Notification 7](#_Toc47440297)

[7 Document References 8](#_Toc47440298)

# Disclaimer

No support and no warranty are provided by Stonebranch GmbH for this document and the related Universal Task. The use of this document and the related Universal Task is on your own risk.

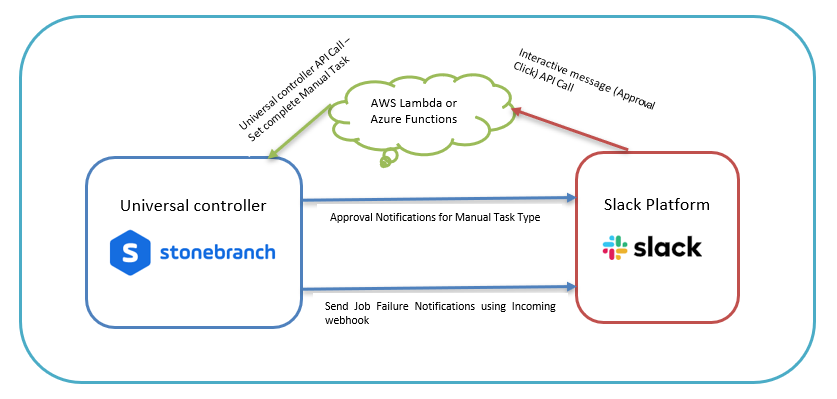
Before using this task in a production system, please perform extensive testing.

Stonebranch GmbH assumes no liability for damage caused by the performance of the Universal Tasks

# Scope

This document provides a documentation how to install and use the Universal Tasks for Slack Notifications Forwarding. If more Task will be created in the future this document will be updated accordingly.

**High Level View:**



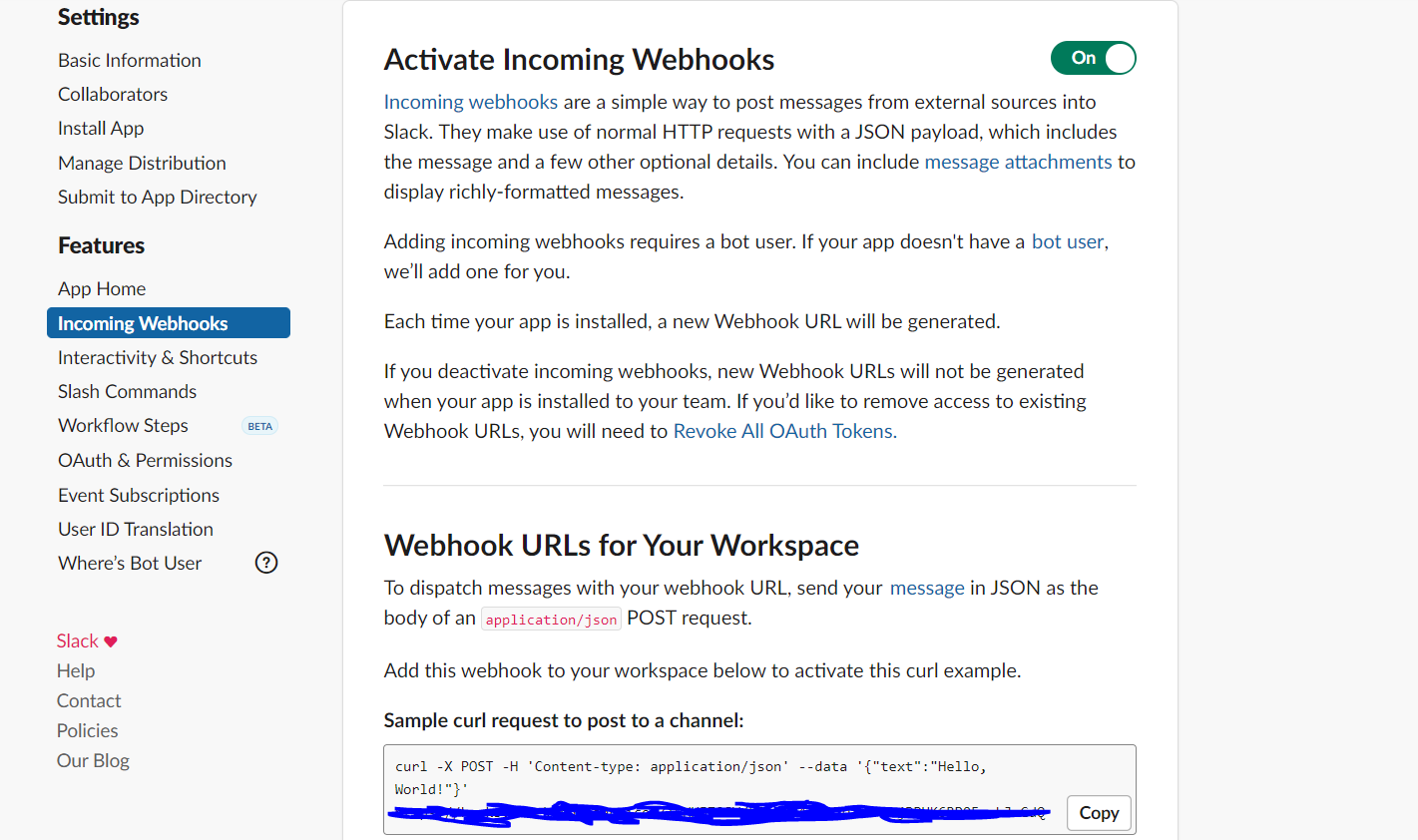
# Introduction

## Incoming Webhooks in Slack

To send notifications to slack channels from universal controller through Incoming webhooks, you may either create a new app in slack or an exist slack app incoming webhook could be used. Also to remember not all slack apps are enabled with incoming webhook, So you may activate incase if it’s not activated.

Steps to activate Incoming Webhooks as below:

* Go to Browser and provide the URL : <https://api.slack.com/apps/>
* Select or create an application that would be appropriate for sending universal controller notifications
* Click on Incoming webhooks on the left menu and activate Incoming webhooks as below



* So, the URL generated here will be used in the universal task for posting message to slack platform

## Implementation Details

Details about the universal tasks for Slack as below:

* Universal task uses requests module in python
* Accepts input parameters like slack Function, Job Name, Job Status, Execution User, Job Type, a slack incoming webhook, a message title and a text of the message
* Universal controller hosted environment should be able to post the message to Slack channel using the incoming webhooks URL generated in slack app.
* No other slack parameters would be needed apart from incoming webhook URL.
* The Universal Task supports both Universal Agent for Linux/Unix and Windows.

# Installation

## Software Requirements for Linux Agent

**Universal Task name:**  SLACK-NOTIFY

Requirements:

* Python 3.6
* For Python the following modules are required:
  + *sys, for system-specific parameters and functions*
  + *requests, to interact with a Slack channel via Api calls*

***Note: Please check requests python library is available already, if not then it needs to be added via python installer***

* *pip install requests*
* Universal Controller V6.4.7.0 or higher
* Universal Agent V6.5.0.0 or higher installed on a Linux/Windows Server

## Installation Steps

The following describes the installation steps:

1. **Check the current Python Version**

*python -V (Note: Capital “V”)*

If your Version is Python 3.6 or later all is fine. If a no python or a lower Version has been installed upgrade your python Version or install the Universal Agent with the Python binding option (--python yes). This option will install python 3.6. along with your universal agent.

e.g.

sudo sh ./unvinst --network\_provider oms --oms\_servers 7878@xxx.xxx.xx.xx --oms\_port 7878 --oms\_autostart no --ac\_netname OPSAUTOCONF --opscli yes --python yes

1. **Add the required python modules**

In a command shell run as sudo or root:

* For Python the following modules are required:
* *pip install requests*

*or in case of universal Agent with python binding:*

*/opt/universal/python3.6/bin/python3 -m pip install requests*

Only run these if not available already:

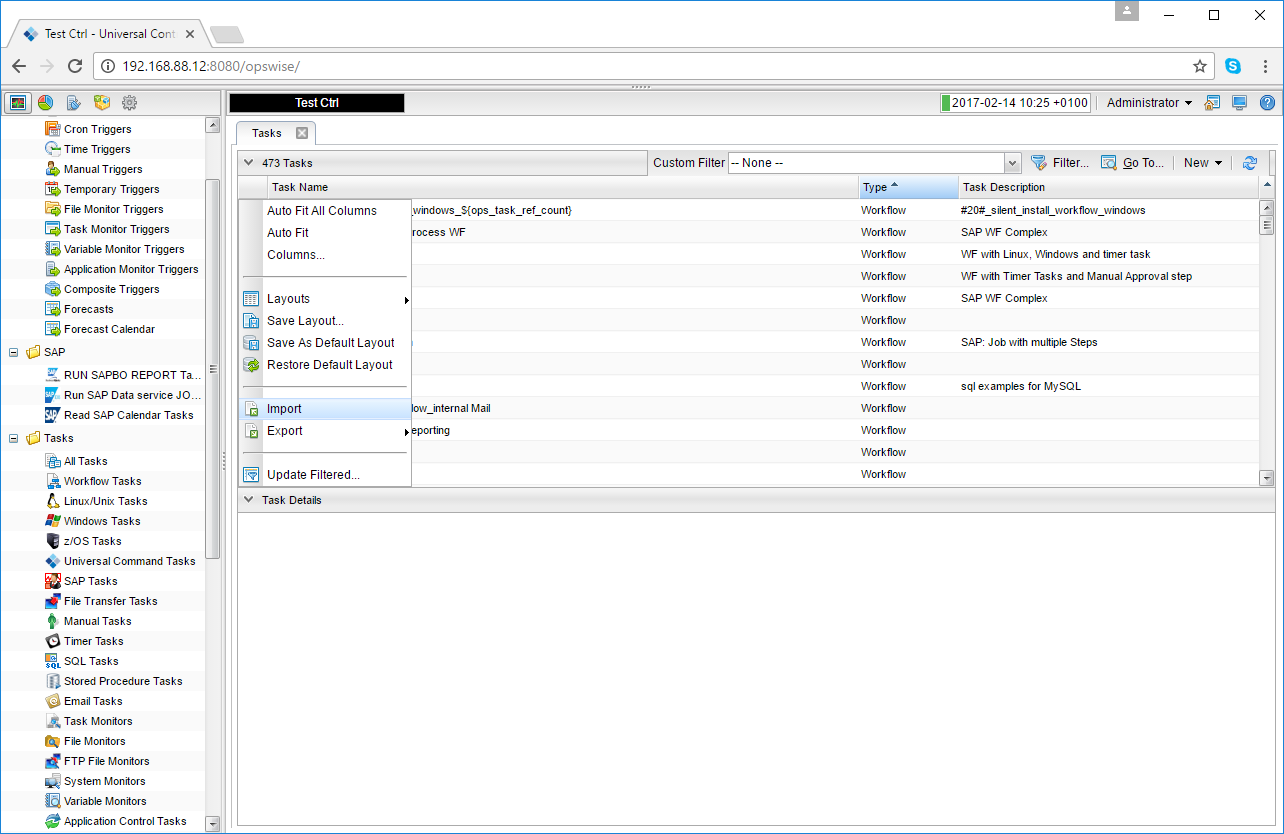
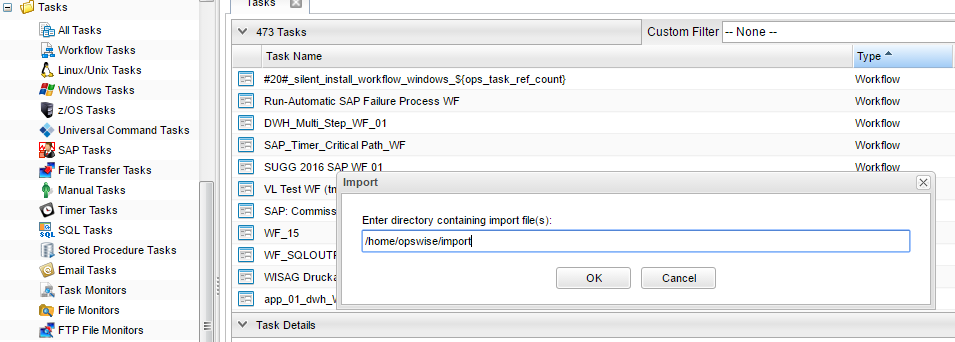
* *pip install sys*

Note:

It is assumed that the modules logging, sys, datetime, os are already available. If not install them via pip. Only the module *requests* are not part of your installation.

1. **Import SLACK-NOTIFY Universal Task including the Universal Template to your Controller**

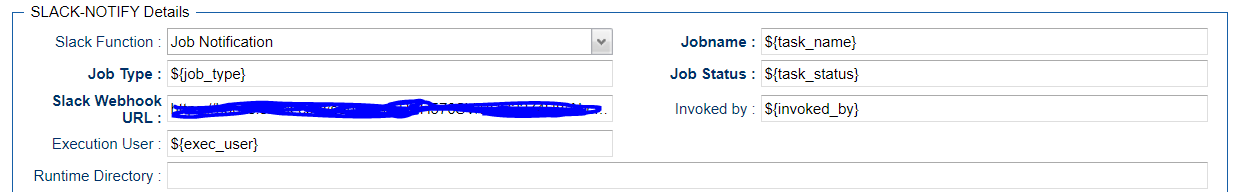
Go to “All Tasks” and load via the Import functionality the Universal Tasks configuration into the Controller.

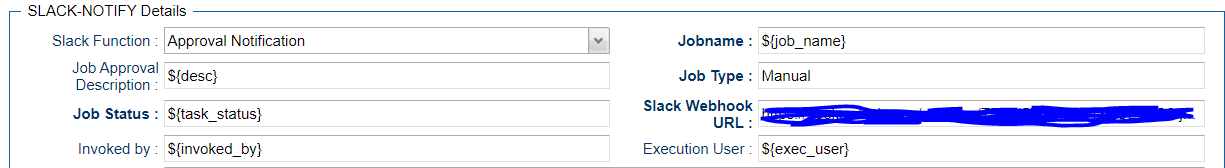
# Universal Task Configuration

1. **Fill Out SLACK-NOTIFY Universal Task:**

**1.1 Sending a notification to SLACK-NOTIFY Incoming webhook**



**1.2 Sending an Approval notification to Slack Incoming webhook**



# Universal Tasks for Slack Notifications Forwarding

The following contents describes the slack notifications through the universal task.

| UT Name | Description |
| --- | --- |
| SLACK-Notify | Send notification to Slack incoming webhook channel. |

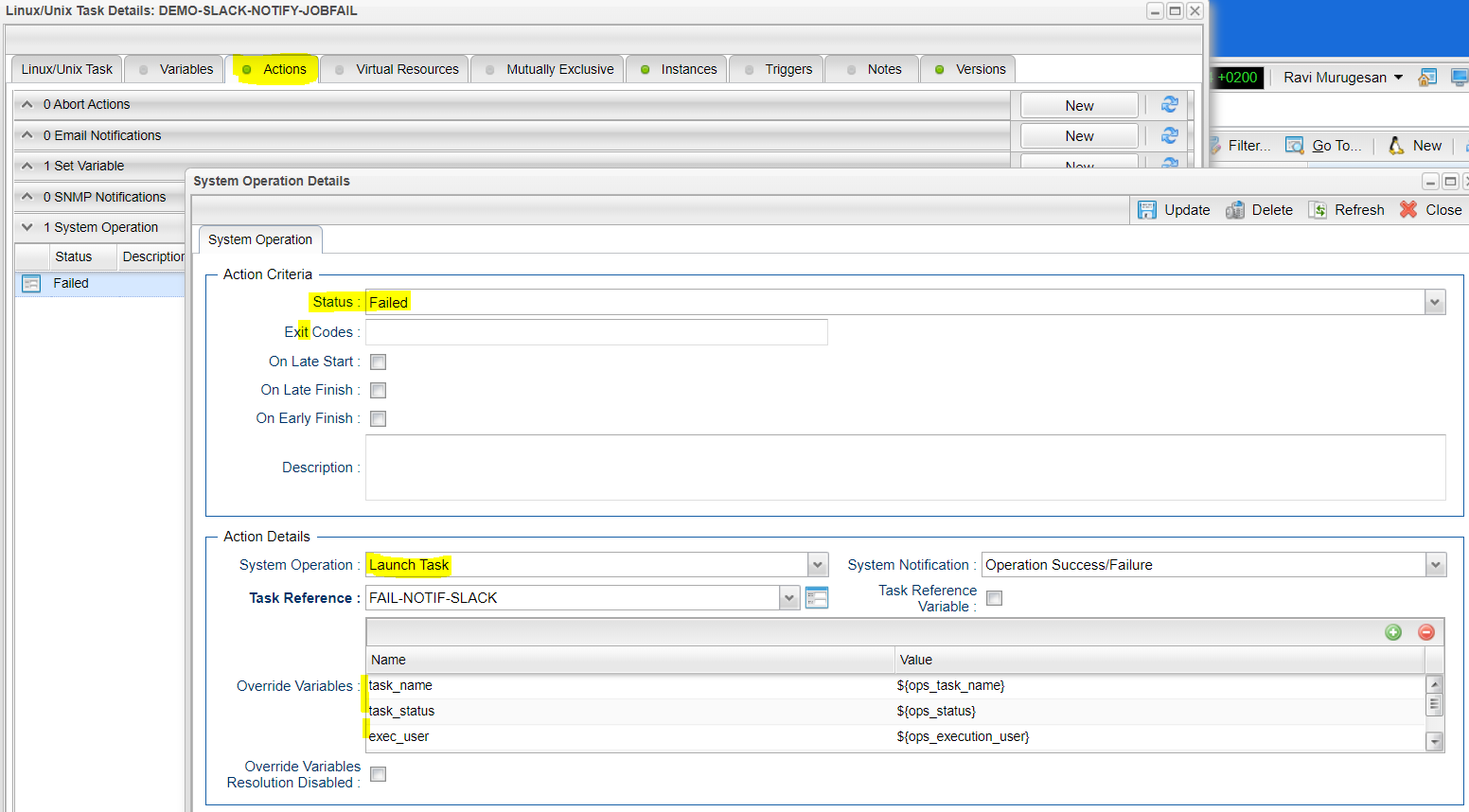
## Send message

With Send Message function we can send a notification message to the Slack channel with the current task instance details.

**Field Description:**

|  |  |
| --- | --- |
| Field | Description |
| Agent | The Agent that runs the Python script assigned to the Universal Task |
| Job Notification | send Job Notification (e.g. failure notification, late run/late finish) Approval notification (associate it with Manual Task type) |
| Job Name | Name of the job: ${ops\_task\_name} |
| Job Status | Status of the job:${ops\_status} |
| Slack Webhook URL | The incoming web hook of Slack app |
| Execution User | Details of the execution user:${ops\_execution\_user} |
| Job type | Task type of task instance: ${ops\_task\_type} |

Note: Typically these notifications can be associated to any of the task type through the Action Tab🡺System operations E.g Refer below screen shot



## Approval Notification

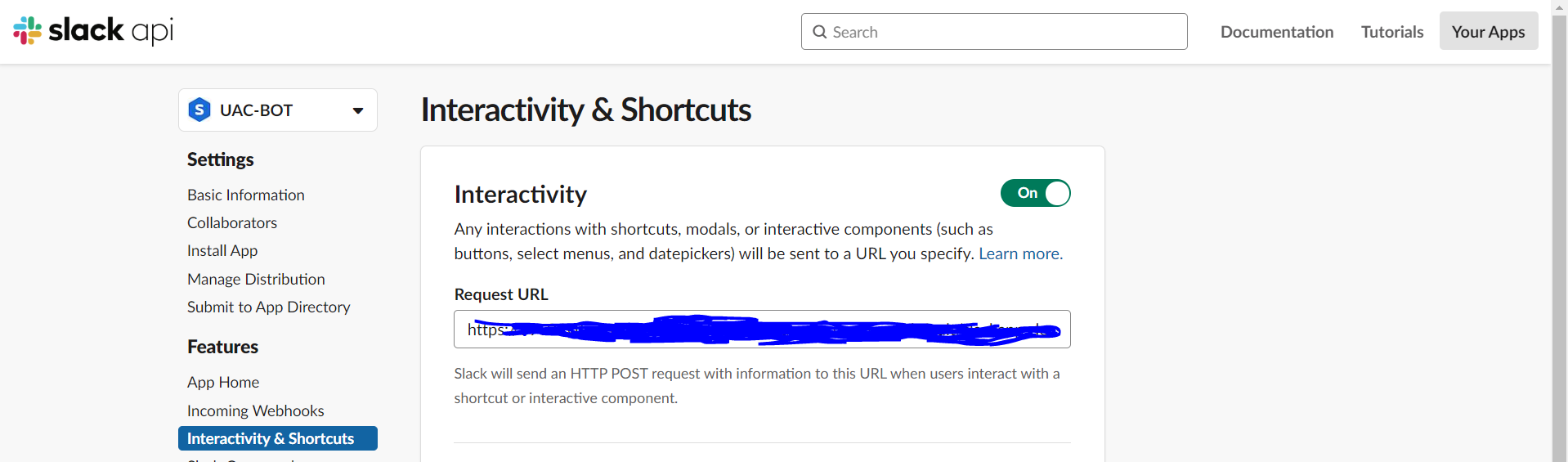
Typically Manual task types in universal controller are used, when there is a user manual intervention needed in a workflow task type and the Manual Task is completed successfully in universal controller by clicking on to the manual task command “Set Completed” traditionally .

Now there is a universal task for Microsoft teams and slack, which can provide you with a notification in the messaging channel when the workflow in controller reaches the manual task with status “Action Required”

So the Users in the slack or teams channel gets a message about the manual task that requires action and User can perform their manual action and simply click on the approve Button in the interactive message that was received in the channel from the universal controller and then the manual task in the workflow will be set to completed automatically and continue with the workflow.

Technically when the Approve or reject button is clicked in the interactive message an API call is made to a function where it can handle the event from Teams or slack. For example you may write a python function in AWS lambda + API gateway or Azure functions or any custom URL where slack or teams messaging platform can call an API POST call to handle the user action in the message as a payload and based on the posted payload data from teams or slack ,Universal controller API call would be made to set the manual task either to set complete status or No action in the function.

For slack to make the interactivity you may need to enable the interactivity as below in the slack platform: (This is required for interactive message actions in slack)



So, if you are using Lambda or Azure functions or any other Api handlers then the below parameters would need to be supplied within API function module so that it can interact with the universal controller:

* Slack or teams event payload data
* Universal controller URL
* Universal controller Username and Password (for making the API call)

**Field Description:**

|  |  |
| --- | --- |
| Field | Description |
| Agent | The Agent that runs the Python script assigned to the Universal Task |
| Slack Function | Approval Notification (ideal to associate with manual tasks) |
| Job Name | Name of the job :${ops\_task\_name} |
| Job Status | Status of the job:${ops\_status} |
| Slack Incoming Webhook | URL from the Slack platform |
| Execution User | Execution User of the Manual Task:${ops\_execution\_user} |
| Job type | Task type of task instance: ${ops\_task\_type} |

**Sample python code in AWS Lambda :**



# Document References

This document references the following documents:

|  |  |
| --- | --- |
| Ref# | Description |
| 1. Slack | <https://api.slack.com/> |
| 1. requests | <https://pypi.org/project/requests/#description> |
| 1. Universal controller API | <https://docs.stonebranch.com/confluence/display/UC67/RESTful+Web+Services+API> |