



Slack Virtual Agent

Motadata ServiceOps



11-Apr-22	Slack Virtual Agent	
	Motadata ServiceOps	

Table of Contents

1.	Overview	1
1.1.	What is Slack and Virtual Agent App?	1
1.2.	Prerequisites	1
2.	Creating and Configuring Slack App	1

11-Apr-22	Slack Virtual Agent	
	Motadata ServiceOps	

1. Overview

1.1. What is Slack and Virtual Agent App?

Slack is an application to communicate with your team members. Using Slack for incident creation and management will become easier if you add the ServiceOps Virtual Agent as an app within Slack. Once added, you can generate or review requests from Slack without signing-in to the Support Portal or accessing the virtual chat agent from the portal.

You can integrate a custom-built Slack app that connects your ServiceOps Portal, enabling the Virtual Agent to send responses to the requester over the Slack application.

This functionality is applicable from version 7.7 and above.

1.2. Prerequisites

- ServiceOps hosted on a live URL
- SSL Certificate enabled
- Requester created in the ServiceOps portal
- Slack account created with the same requester email ID as created in the above point.

2. Creating and Configuring Slack App

To create a Slack application, follow the below steps:

Step 1: Create a [Slack account](#).

Step 2: Create a **Workspace** to be associated with your account.

Once created, you will get redirected to the Slack API page.

Step 3: **Create an App**, and the following pop-up will appear.

Select the option "From an app manifest".

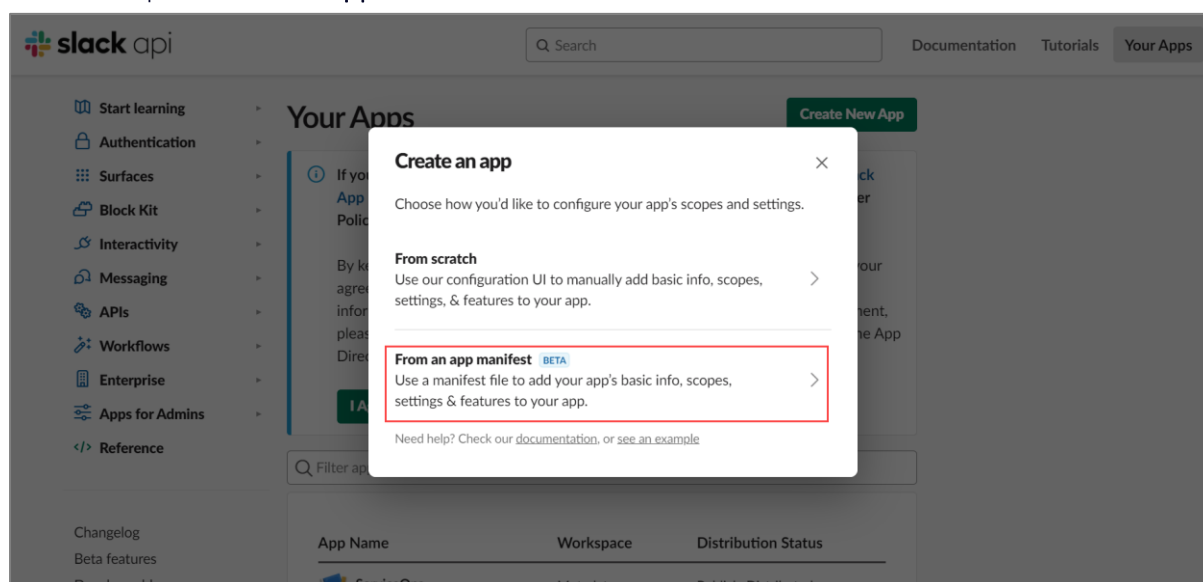


Fig 1: Select the option "From an app manifest"



Step 4: Select a workspace and click **Next**.

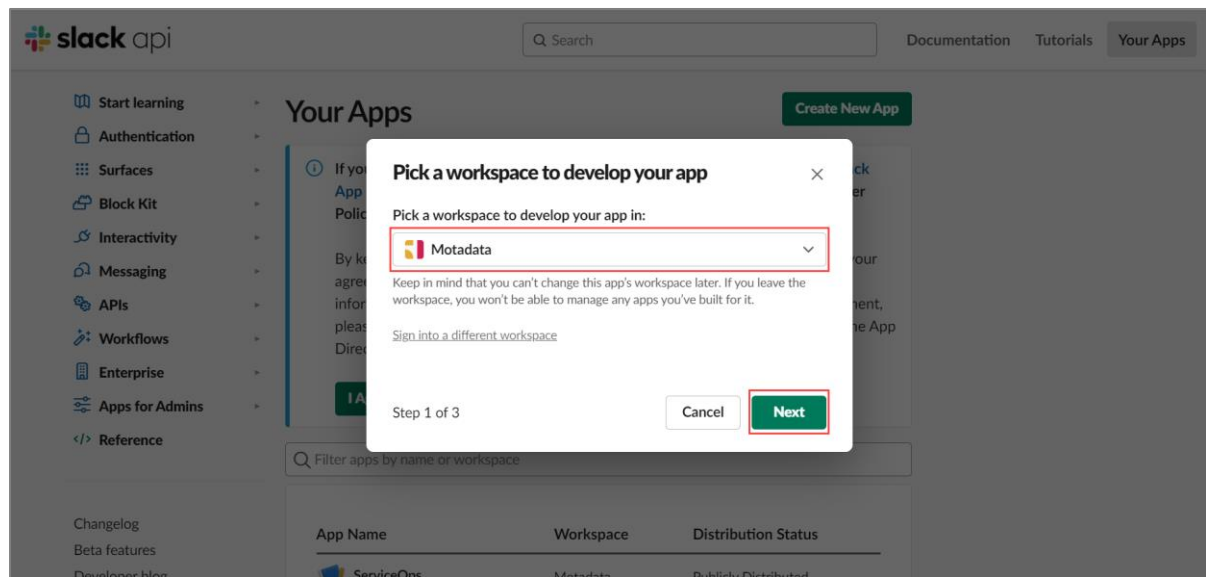


Fig 2: Select Workspace

Step 5: Use the below sample YAML code and replace all the highlighted URLs with the ServiceOps URL. Once done, copy the entire code.

```
display_information:
  name: ServiceOps
features:
  bot_user:
    display_name: ServiceOps
    always_online: true
  slash_commands:
    - command: /create_itsm_ticket_command
      url: https://sso.motadataserviceops.com/api/public/slack/events/command
      description: create tickets
      usage_hint: write subject
      should_escape: false
  oauth_config:
    redirect_urls:
      - https://sso.motadataserviceops.com/api/public/slack/install
  scopes:
    bot:
      - app_mentions:read
      - channels:history
      - chat:write
      - commands
      - groups:history
      - im:history
      - mpim:history
      - team:read
      - users.profile:read
  settings:
    event_subscriptions:
      request_url: https://sso.motadataserviceops.com/api/public/slack/readMessage
```



11-Apr-22	Slack Virtual Agent	
	Motadata ServiceOps	

bot_events:

- message.im
- message.mpim

interactivity:

is_enabled: true

request_url: <https://sso.motadataserviceops.com/api/public/slack/events>

org_deploy_enabled: false

socket_mode_enabled: false

token_rotation_enabled: false

Step 6: In the “Enter app manifest below” section, replace the default YAML code with the above modified and copied code (step 6), and click **Next**.

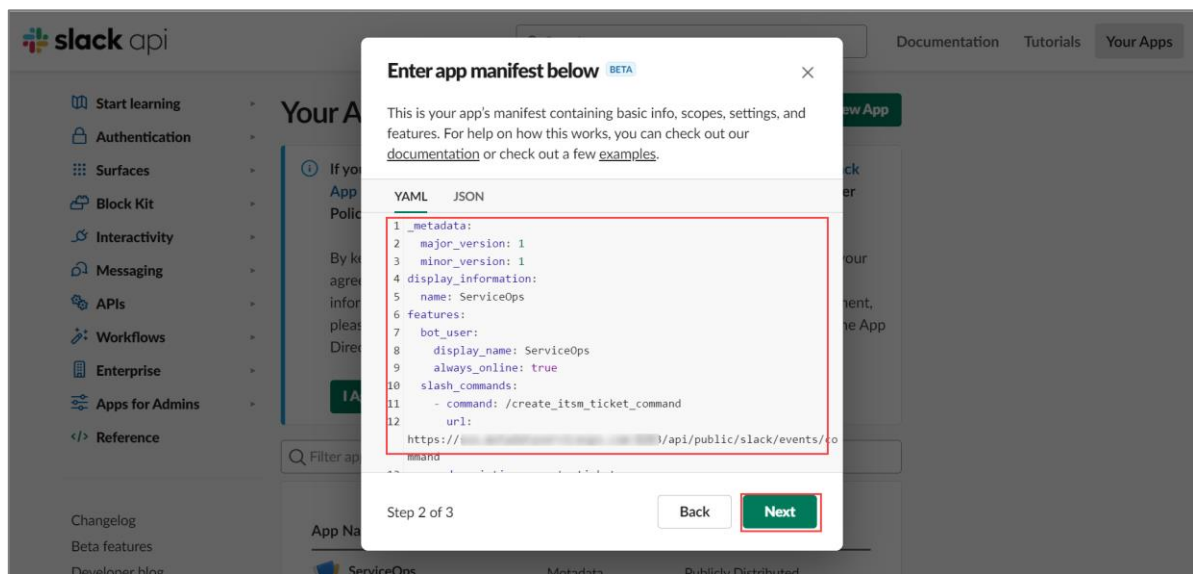


Fig 3: Copying and pasting the YAML code

Step 7: Click **Create**.

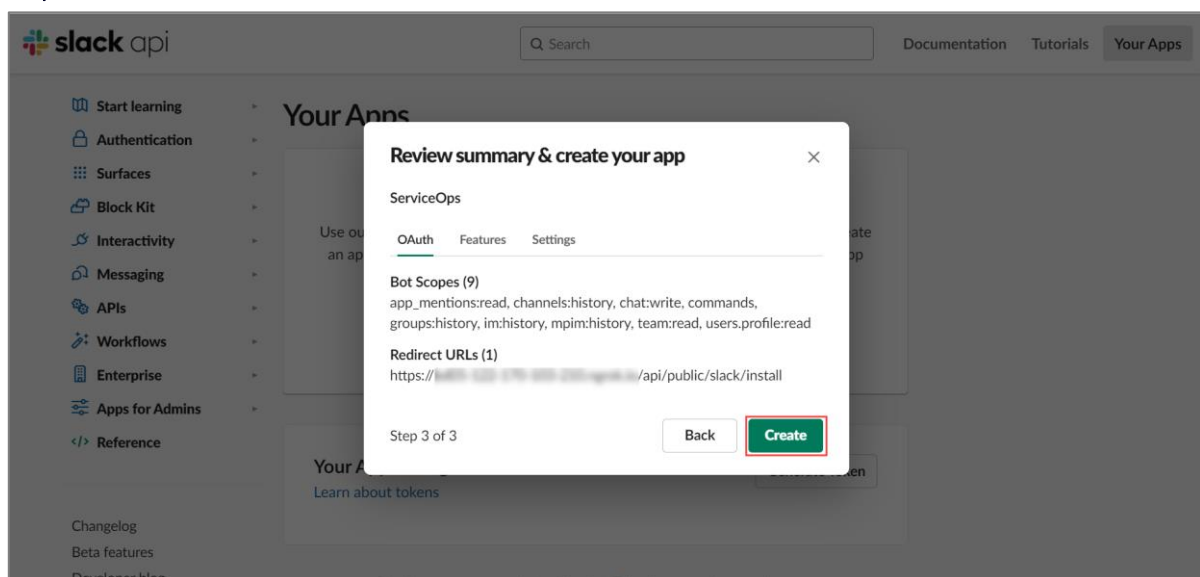
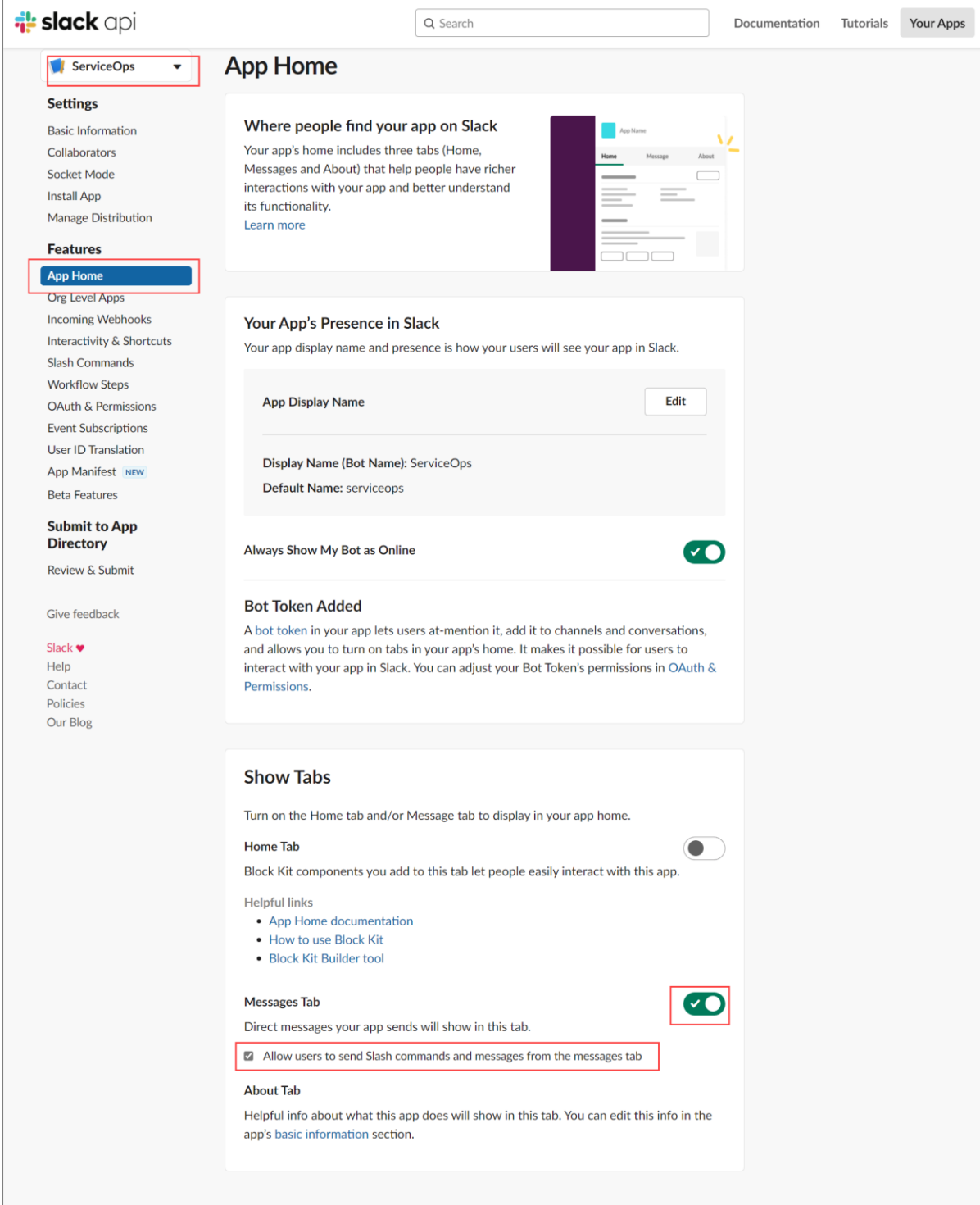


Fig 4: Create App



Step 8: Select the created app and navigate to **Features > App Home > Show Tabs > Messages Tab** section.

Enable the option “**Allow users to send Slash commands and messages from the messages tab**”, as shown below.



The screenshot shows the Slack API interface for configuring an app named 'ServiceOps'. The left sidebar has a 'ServiceOps' dropdown at the top. Under the 'Features' section, 'App Home' is selected. The main content area is titled 'App Home' and contains several sections:

- Where people find your app on Slack:** Explains that the app's home includes three tabs (Home, Messages, and About) to help users interact with the app.
- Your App's Presence in Slack:** Allows setting the 'App Display Name' (currently 'ServiceOps') and a toggle for 'Always Show My Bot as Online' (which is turned on).
- Bot Token Added:** Provides information about the bot token and its permissions.
- Show Tabs:** A section where you can turn on the 'Home Tab' and 'Messages Tab'. The 'Messages Tab' toggle is turned on. Below this, the checkbox 'Allow users to send Slash commands and messages from the messages tab' is checked and highlighted with a red box.

Fig 5: Enable the option



- Select the **"Remove Hard Coded Information"** section and enable the option **"I've reviewed and removed any hard-coded information"**.
- Activate the application using the **"Activate Public Distribution"** button.

ServiceOps

Settings

Basic Information

Collaborators

Socket Mode

Install Your App

Manage Distribution

Features

App Home

Org Level Apps

Incoming Webhooks

Interactivity & Shortcuts

Slash Commands

Workflow Steps

OAuth & Permissions

Event Subscriptions

User ID Translation

App Manifest

Beta Features

Submit to App Directory

Review & Submit

Give feedback

Slack

Help

Contact

Policies

Our Blog

Search

DocumentationTutorialsYour Apps

Manage Distribution

Share Your App with Your Workspace

You can use the URL and the Add to Slack button below to share your app with the Motadata workspace. Activate distribution (below) to share your app with any workspace.

Embeddable Slack Button

Add to Slack

 Copy

Sharable URL

https://slack.com/apps/A0HJQ8P7TCT?ref=org_apps_manage_distribution Copy

App Suggestions HTML

Add this tag to suggest your app to new users when links from your domain are mentioned in Slack. Learn more.

<script src="https://slack.com/apps/A0HJQ8P7TCT?ref=org_apps_manage_distribution"></script> Copy

Share Your App with Other Workspaces

Public Distribution

Activate this distribution option to allow all workspaces to install your app, and to view the steps to submit your app to the Slack App Directory.

Before activating distribution, please complete the steps listed below.

Enable Features & Functionality

Add OAuth Redirect URLs

Remove Hard Coded Information

Before activating public distribution, please remove any hard-coded information from your app, such as OAuth tokens and webhook URLs. Going forward, you should authorize your app for use on your workspace with OAuth, just like anybody installing your app on their own workspace would do.

☒ I've reviewed and removed any hard-coded information.

Use HTTPS For Your Features

Activate Public Distribution

Fig 6: Manage Distribution Page

Step 10: Sign-in to the ServiceOps Portal and configure the Slack App settings.

- Navigate to **Admin > Support Channel > Virtual Agent > Slack App Config** page.

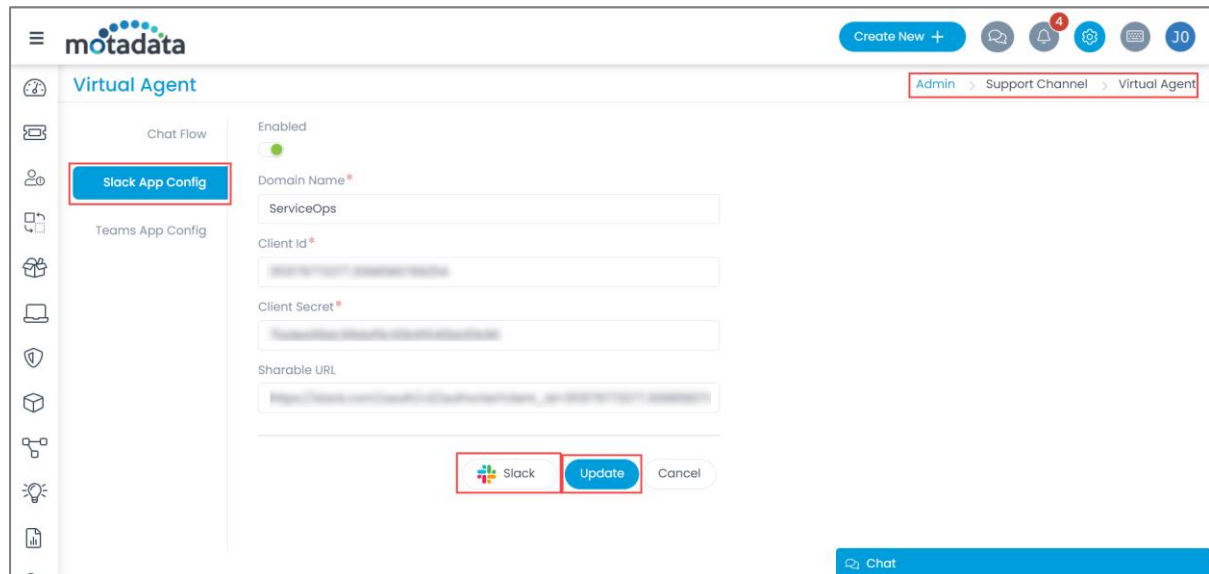


Fig 7: Slack App Config Page

- Enable** the Slack functionality to allow requesters to chat via the Slack application.
- Domain Name:** Provide the domain name on which the user will use the application. The domain name can be found under the Workspace name from the Slack account as shown below.

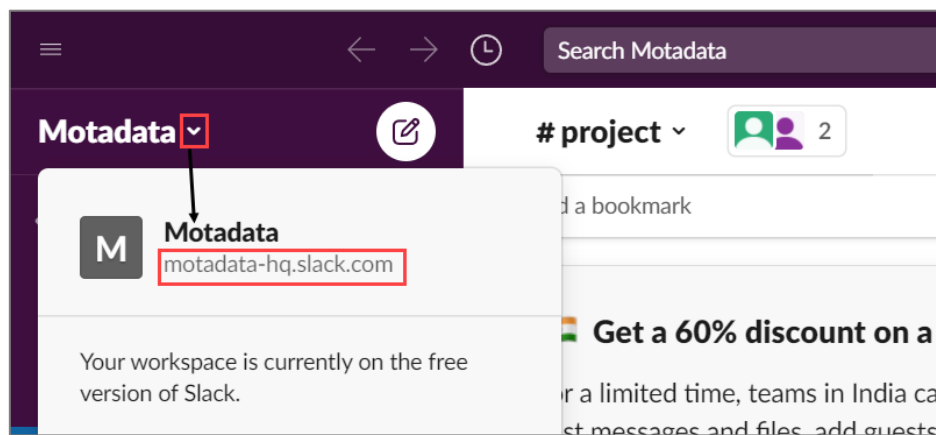
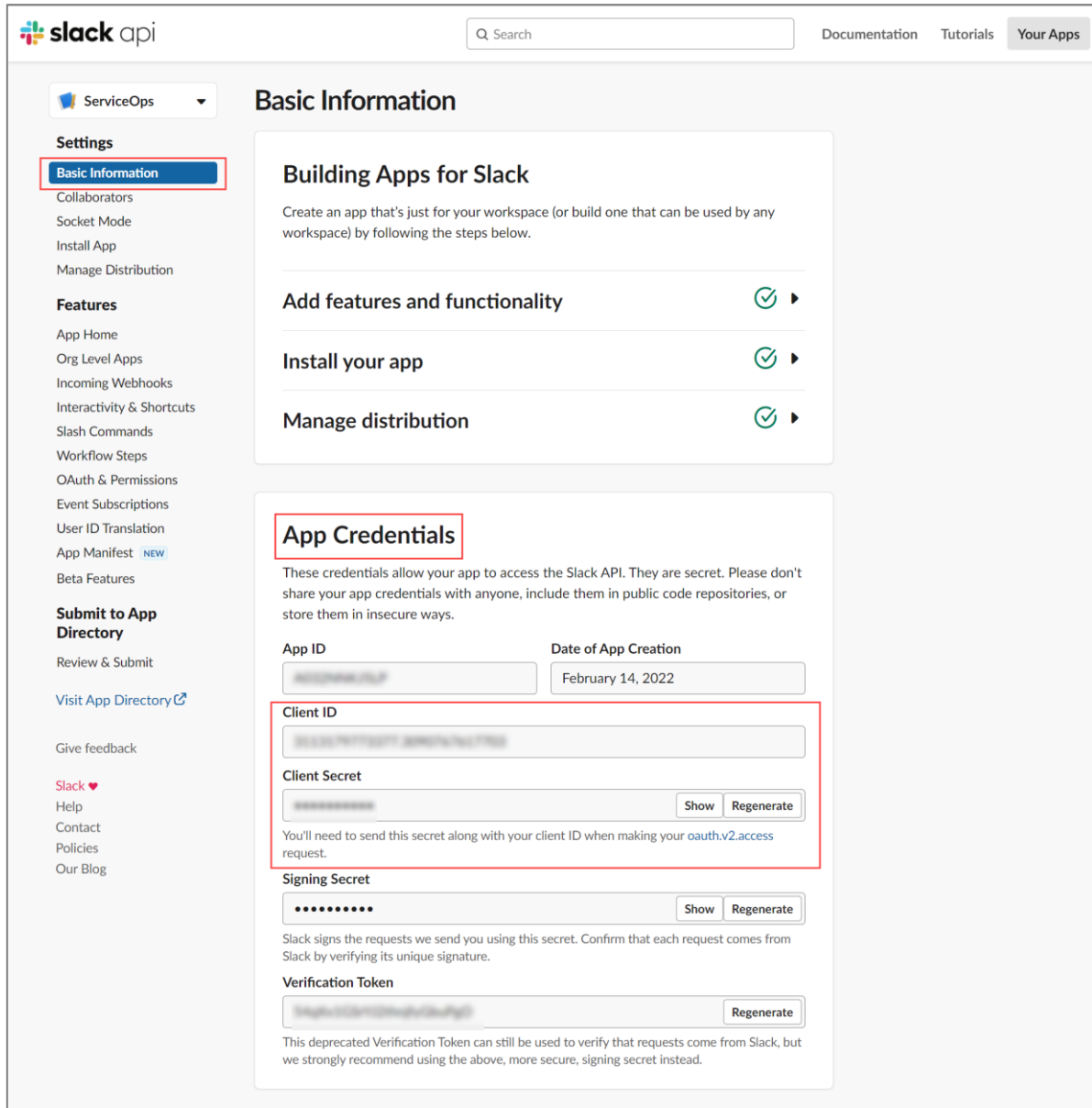


Fig 8: Domain Name

- Client ID:** Provide the Slack client ID.



- **Client Secret:** Provide the secret ID of the Slack client. You can get these details from the Slack API page, as shown below.



The screenshot shows the Slack API 'Basic Information' page. The left sidebar contains a 'Settings' menu with 'Basic Information' highlighted. The main content area is titled 'Basic Information' and 'Building Apps for Slack'. It includes a list of steps: 'Add features and functionality', 'Install your app', and 'Manage distribution', each with a green checkmark. Below this is the 'App Credentials' section, which is highlighted with a red box. It contains fields for 'App ID' and 'Date of App Creation'. The 'Client ID' field is also highlighted with a red box. Below it is the 'Client Secret' field, which is highlighted with a red box and includes 'Show' and 'Regenerate' buttons. A note states: 'You'll need to send this secret along with your client ID when making your `oauth.v2.access` request.' Below the 'Client Secret' field is the 'Signing Secret' field, which is highlighted with a red box and includes 'Show' and 'Regenerate' buttons. A note states: 'Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.' Below the 'Signing Secret' field is the 'Verification Token' field, which is highlighted with a red box and includes a 'Regenerate' button. A note states: 'This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.'

Fig 9: Client ID and Client Secret parameters

- **Sharable URL:** This is the shareable URL received from the Slack client. You can get this from the Slack API page, as shown below. Once configured, the Virtual Agent can send responses to the Slack application.

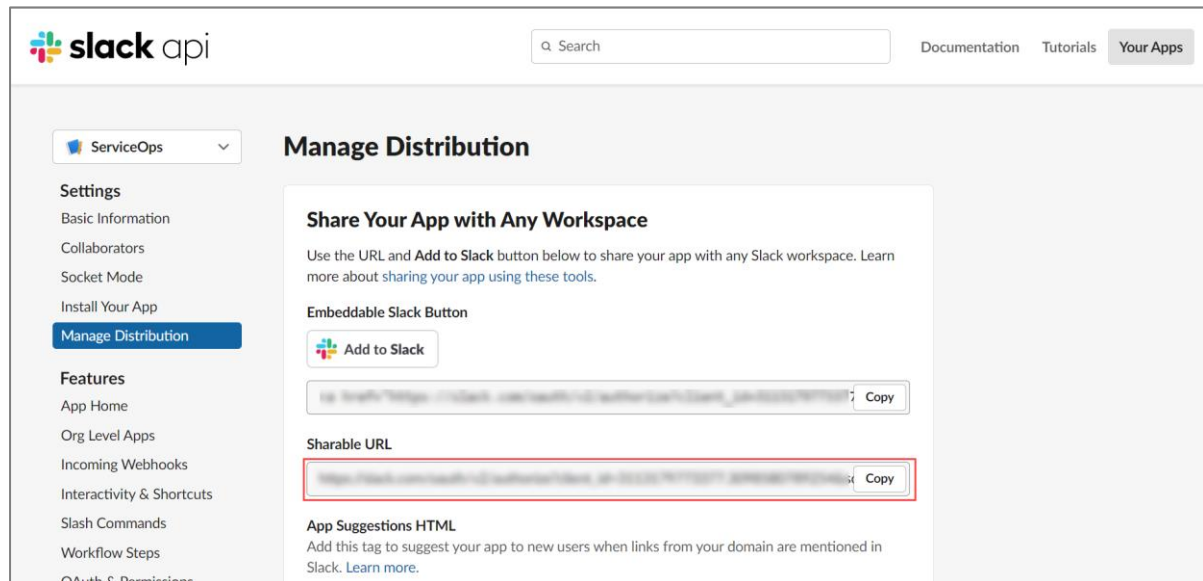


Fig 10: Sharable URL

Step 11: Click **Update**, and a confirmation message will appear.

Step 12: Click **Slack** and the below page will appear.

Click **Allow**, and a confirmation message “**Success! Successfully installed Slack App**” will appear.

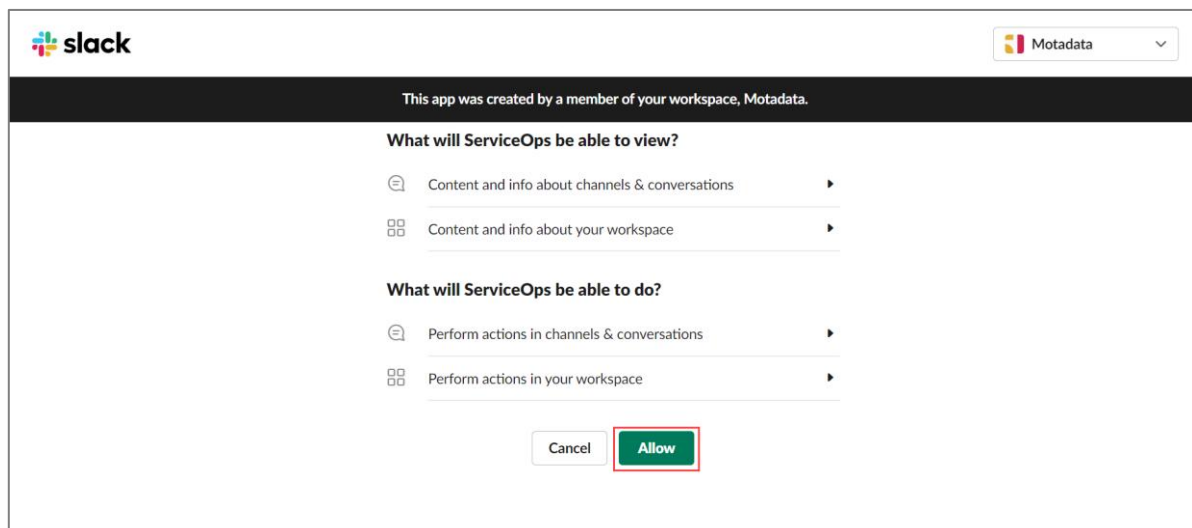


Fig 11: Allow ServiceOps to communicate with Slack



Step 13: Open the Slack application. You will find the ServiceOps App in the **Apps** section, as shown below.

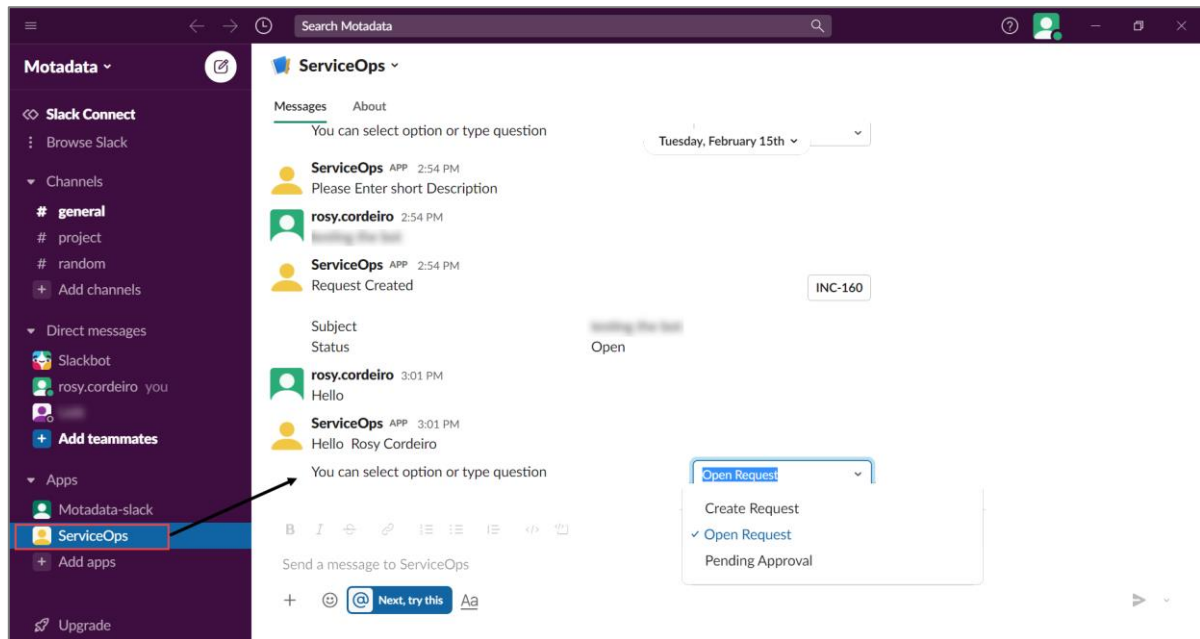


Fig 12: ServiceOps Virtual Agent in Slack

Now, you can chat with the ServiceOps Virtual Agent.



CONNECT WITH US



www.motadata.com



support@motadata.com

© 2021 Mindarray Systems Pvt. Ltd. All rights reserved.

