

Slack

Dialogflow's Slack Integration makes it easy to create your own Slack apps and bots and train them to understand natural language.

When using a third party integration, requests and end-user data are sent to both Google and the third party. Terms of service, SLA, pricing, and quotas [for your agent edition](/dialogflow/docs/editions) (/dialogflow/docs/editions) are applicable for Google Cloud of these requests. The third party's terms of service, SLA, pricing, and quotas are applicable for the third party of these requests.

Setting Up Slack

In order to set up the Slack integration for your agent, you'll need the following:

- a Slack account
- a [Slack Team](https://slack.com/create#email) (https://slack.com/create#email)

Create a Slack App

1. Navigate to the [Slack Developer Console](https://api.slack.com/apps/new) (https://api.slack.com/apps/new)
2. Enter a name for your app
3. Choose a Team you would like the app associated with
4. Click the **Create App** button

Create an App ✕


App Name


My Weather

Don't worry; you'll be able to change this later.

Development Slack Team

Development Slack Team ▼

 Garth's Group

 Sign in to another team

[Service.](#)

Cancel Create App

Add a Bot User

1. Click on **Bot Users** in the left hand menu
2. Click the **Add a Bot User** button
3. Enter a name for your Slack bot (this is what users will see they add your bot)
4. Enable **Always Show My Bot as Online**
5. Click the **Add Bot User** button

Bot User

You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about [how bot users work](#).

Default username

If this username isn't available on any team that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

Always Show My Bot as Online

When this is off, Slack automatically displays whether your bot is online based on usage of the RTM API.

On

Add Bot User

Enabling Integration in Dialogflow

In the Slack Developer Console, click on **Basic Information** in the left hand menu and scroll down to the **App Credentials** section. Make note of the **Client ID**, **Client Secret**, and **Verification Token**.

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

Client ID

Client Secret

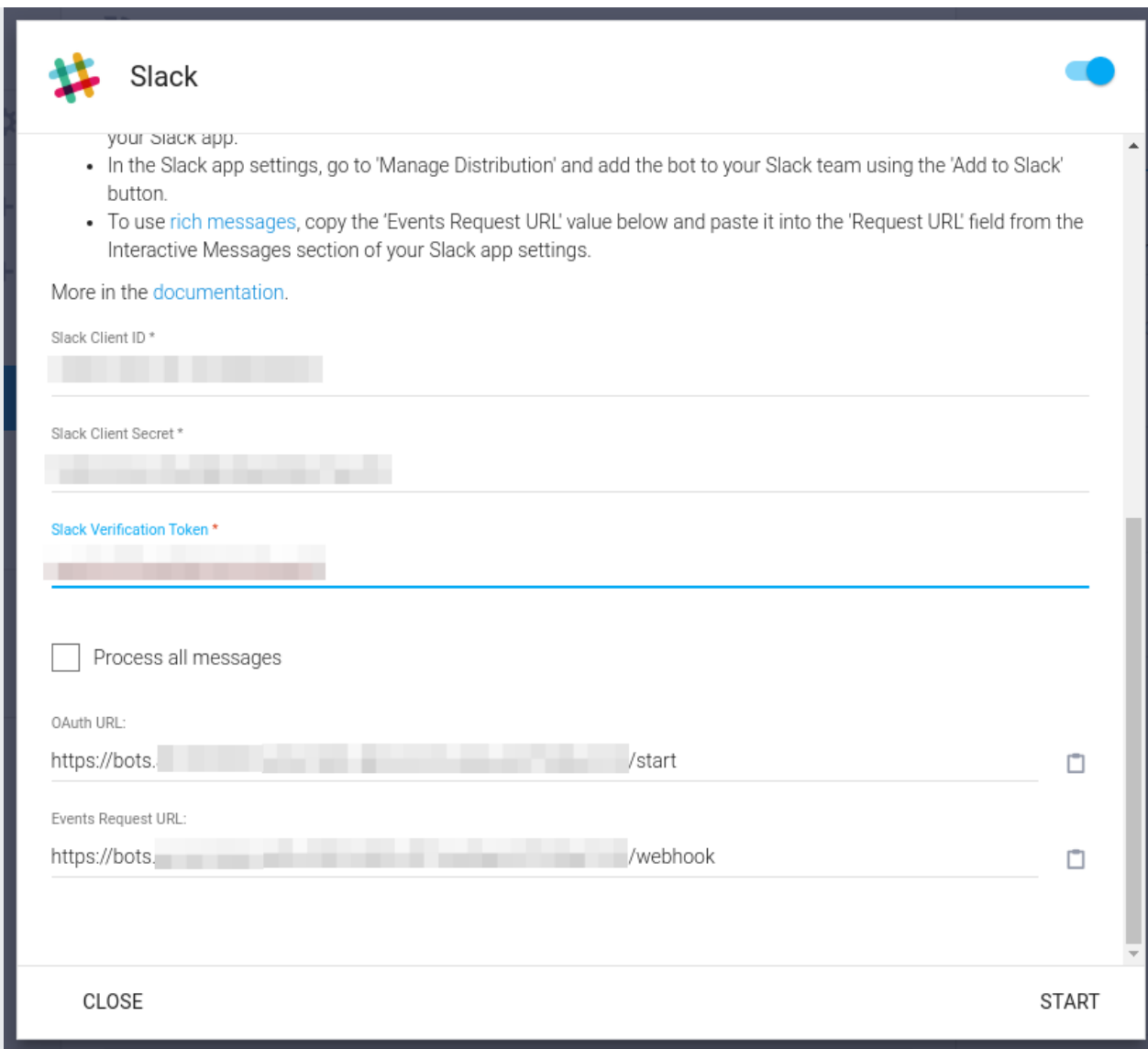
You'll need to send this secret along with your client ID when making your [oauth.access](#) request.

Verification Token

For interactive messages and events, use this token to verify that requests are actually coming from Slack. Slash commands and interactive messages will both use this verification token.

Link Slack to Dialogflow

1. In Dialogflow, go to **Integrations** in the left hand menu
2. Click on the **Slack** tile
3. Enter the related values into the following fields:
 - a. **Client ID**
 - b. **Client Secret**
 - c. **Verification Token**
4. Click **Start**



The screenshot shows a 'Slack' configuration window. At the top left is the Slack logo and the word 'Slack'. A blue toggle switch is on the top right. Below the header, there is a section for instructions: 'your Slack app.' followed by a bulleted list: '• In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.' and '• To use rich messages, copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.' Below this is a link: 'More in the [documentation](#).' The form contains three input fields: 'Slack Client ID *', 'Slack Client Secret *', and 'Slack Verification Token *'. Below these is a checkbox labeled 'Process all messages'. There are two URL input fields: 'OAuth URL:' with the value 'https://bots. /start' and 'Events Request URL:' with the value 'https://bots. /webhook'. Each URL field has a copy icon on the right. At the bottom of the form are two buttons: 'CLOSE' on the left and 'START' on the right.

Continuing Slack Setup

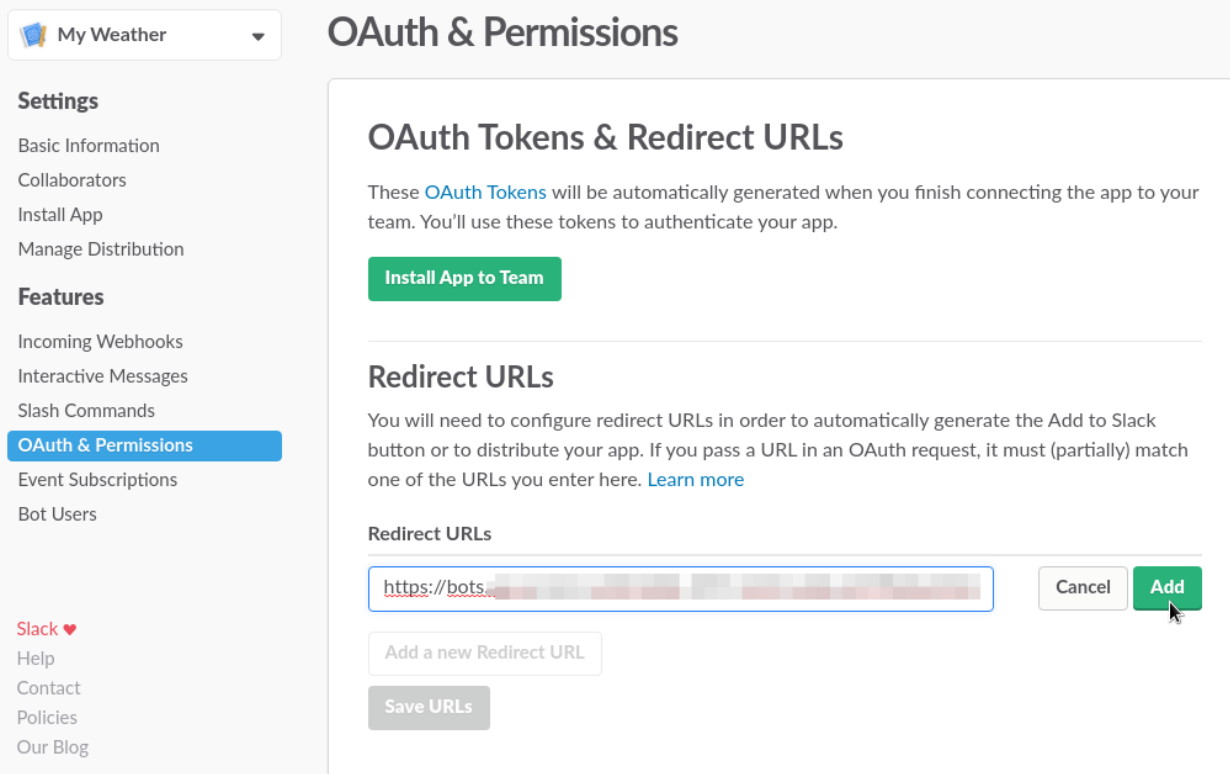
Now that you've set up Slack and Dialogflow, you'll enable OAuth, Event Requests URLs (provided by Dialogflow) and Event Subscriptions.

Add OAuth URL

1. Copy the **OAuth URL** from Dialogflow's Slack setting page
2. Go to the Slack Developer Console and click on **OAuth & Permissions** in the left hand menu

3. Click on the **Add a new Redirect URL** button and paste the OAuth URL

4. Click **Add**



My Weather OAuth & Permissions

Settings

- Basic Information
- Collaborators
- Install App
- Manage Distribution

Features

- Incoming Webhooks
- Interactive Messages
- Slash Commands
- OAuth & Permissions**
- Event Subscriptions
- Bot Users

Slack ♥
Help
Contact
Policies
Our Blog

OAuth Tokens & Redirect URLs

These [OAuth Tokens](#) will be automatically generated when you finish connecting the app to your team. You'll use these tokens to authenticate your app.

[Install App to Team](#)

Redirect URLs

You will need to configure redirect URLs in order to automatically generate the Add to Slack button or to distribute your app. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. [Learn more](#)

Redirect URLs

[Cancel](#) [Add](#)

[Add a new Redirect URL](#)

[Save URLs](#)

Add Event Request URL

1. Copy the **Event Request URL** from Dialogflow's Slack setting page
2. Go to the Slack Developer Console and click on **Event Subscriptions** in the left menu
3. Click on the switch to enable Events, and paste the Event Subscription URL in to **Request URL** field

After a few moments you should see a green "Verified" above the Request URL field.

The screenshot shows the Slack settings interface for an app named "My Weather". On the left is a navigation menu with sections for "Settings" (Basic Information, Collaborators, Install App, Manage Distribution) and "Features" (Incoming Webhooks, Interactive Messages, Slash Commands, OAuth & Permissions, **Event Subscriptions**, Bot Users). The main content area is titled "Event Subscriptions" and contains two sections: "Enable Events" and "Subscribe to Team Events".

Enable Events section:

- A toggle switch is turned "On".
- Text: "Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)"
- "Request URL Verified" status with a green checkmark.
- A text input field contains "https://bots. [redacted] /webhook" and a "Change" button.
- Text: "We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. [Learn more.](#)"

Subscribe to Team Events section:

- Once you're done updating these two settings, click the **Save Changes** button at the bottom of the page.

Enable Event Subscriptions

- Click on **Event Subscriptions** in the left menu
- Click the switch to enable the option
- Under the **Subscribe to Bot Events**, click on the **Add Bot User Event** button

This will reveal text field and list of events. You can enter terms into this field to search for events you want your bot to respond to. Here's a list of events we suggest adding:

Event Name	Event Description
message.im	A message was posted in a direct message channel
message.groups	A message was posted to a private channel
message.channels	A message was posted to a channel
im_created	A DM was created

Subscribe to Bot Events

Bot users can subscribe to events related to the channels and conversations they're part of.

Event Name

Description

No events added yet.

Add Bot User Event

messa

message.channels

A message was posted to a channel

message.im

A message was posted in a direct message channel

message.groups

A message was posted to a private channel

message.mpim

A message was posted in a multiparty direct message channel

link_shared

A message was posted containing one or more links relevant to your application

You'll receive events for any subdomain, or path of the selected domains or hosts. Ports are ignored, so requests on any port will trigger an event. Using IP addresses is not allowed. [Learn more.](#)

4. Once you've added the events, click the **Save Changes** button at the bottom of the screen

Add Your Slack Bot to a Team

1. In the Slack Developer Console, click on **Manage Distribution** in the left menu
2. Under the **Share Your App with Your Team** section, click the **Add to Slack** button

My Weather

Manage Distribution

Share Your App with Your Team


You can use the URL and the **Add to Slack** button below to share your app with the Garth's Group team. Activate distribution (below) to share your app with any team.

Embeddable Slack Button

```
<a href="https://slack.com/oauth/authorize?&client_id=[redacted]&scope=bot">
```



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

3. On the next page, click the **Authorize** button to add the agent



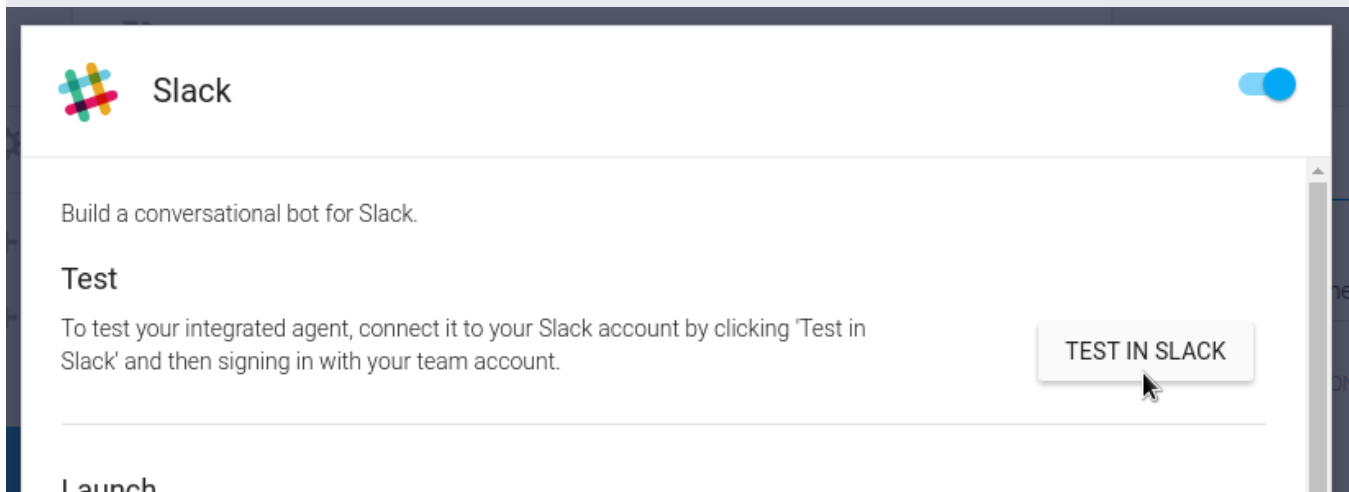
On Garth's Group, My Weather would like to:

Confirm your identity on Garth's Group

 Add a bot user with the username `@my_weather` 

Testing

You can test your agent without creating a public Slack bot. To do so, go to the **Integrations** page for your agent and click the switch on the **Slack** tile. In the pop-up window, click the **Test In Slack** button.



You'll be prompted to log in to Slack. Once logged in, navigate to the instance and locate the bot user called **@dialogflow-bot**. You can direct message the agent to test it or add it to channels in your instance.

In order to change the name of the agent or add a logo, you'll need to make the bot public.

Events

The following [events](/dialogflow/docs/events-overview) are invoked by this integration:

Event	Description
SLACK_WELCOME	Triggered when a user starts a conversation with your Slack bot.

Rich response messages

The following [rich response messages](/dialogflow/docs/intents-rich-messages) are supported.

Image responses

Supported formats and sizes:

- gif
- jpeg
- png
- bmp

Card responses

There are no mandatory fields for cards in Slack – any combination of elements can be used.

Buttons with a URL and with a text postback are supported. Buttons with a URL are displayed as hyperlinks.

Quick reply responses

Quick replies are supported in one-click Slack integrations and correspond to a text version of [Slack interactive buttons](https://api.slack.com/docs/message-buttons) (<https://api.slack.com/docs/message-buttons>).

Custom payload responses

Formatted text in one-click Slack integration

Your Slack bots can send [formatted text](#)

(https://api.slack.com/docs/message-formatting#message_formatting), as the following example shows:

```
ack": {  
text": "This is an example of *bold*, _italic_, and `code`."
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-08-20 UTC.