Osano and Slack Webhook Creation

Overview

This document will walk through the following:

- How to create a basic app in Slack to allow incoming webhooks from Osano.
    - Note that this is not a complete guide on the creation of Slack apps and only outlines what is required for integrating with Osano webhooks. For a comprehensive guide on Slack applications, please consult Slack's documentation.
- How to create an outgoing webhook in Osano which will send a message to a specified Slack channel.

Benefits

- Improve the speed and efficiency of your Subject Rights Request process by automatically notifying necessary employees when a new request has been generated outside of the Osano application.

Requirements

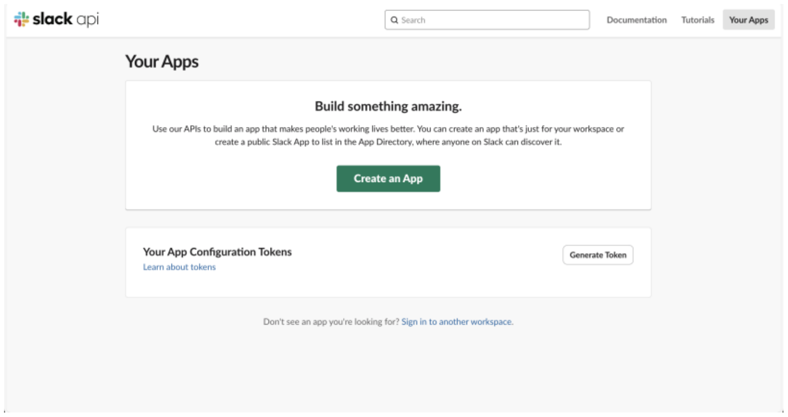
- The ability and permissions to create a new application in Slack and connect it to a channel in your workspace.
- The ability in permissions to create webhooks in Osano.
    - https://docs.osano.com/hc/en-us/articles/22469433756052-User-Roles

Create an app in Slack.

Before we can create a webhook in Osano, we first need to create an app in a Slack workspace and connect it to the channel where we want our messages to be posted when the webhook fires.

1. In your browser, navigate to api.slack.com and click on "Your Apps"

2. Click on the “Create an App” button



Your Apps

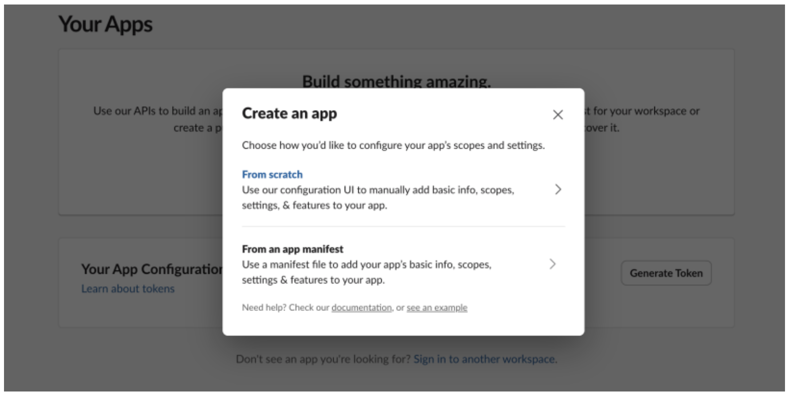
Build something amazing.

Use our APIs to build an app that makes people's working lives better. You can create an app that's just for your workspace or create a public Slack App to list in the App Directory, where anyone on Slack can discover it.

Your App Configuration Tokens

Don't see an app you're looking for? Sign in to another workspace.

3. Click on “From Scratch



Create an app

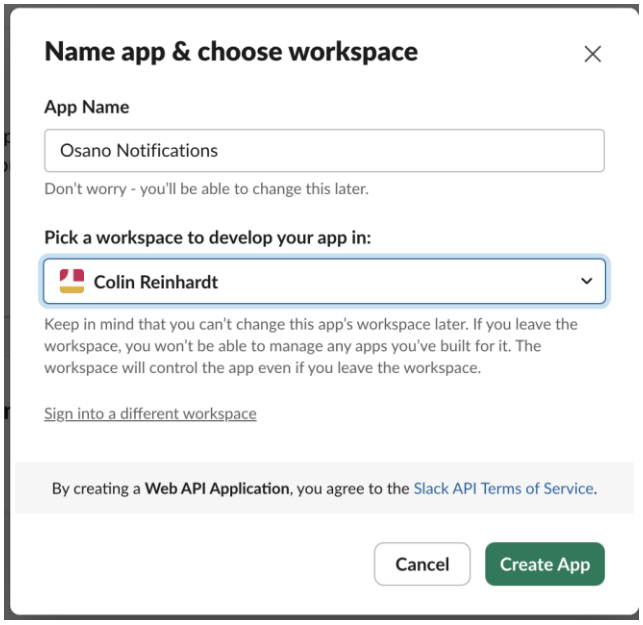
Choose how you'd like to configure your app's scopes and settings.

From scratch
Use our configuration UI to manually add basic info, scopes, settings, & features to your app.

From an app manifest
Use a manifest file to add your app's basic info, scopes, settings & features to your app.

Need help? Check our documentation, or see an example

4. Provide your app with a name, and choose the workspace in which you would like to develop in. Click the “Create App” button.



Name app & choose workspace

App Name: Osano Notifications

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

Pick a workspace to develop your app in: Colin Reinhardt

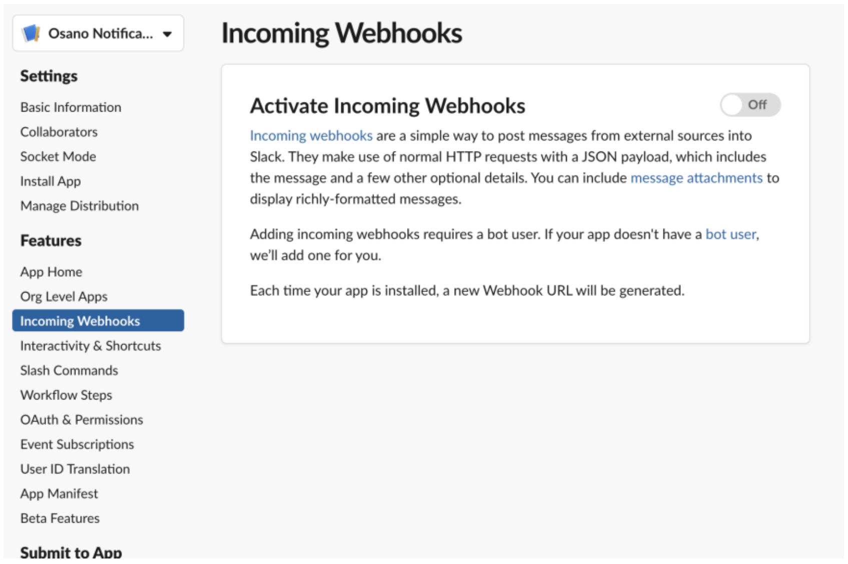
Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace.

Sign into a different workspace

By creating a **Web API Application**, you agree to the Slack API Terms of Service.

5. You will be brought to a new page where you can view the basic information, settings, features, and more for your app. Click on the “Incoming Webhooks”

section on the left-hand side.



Incoming Webhooks

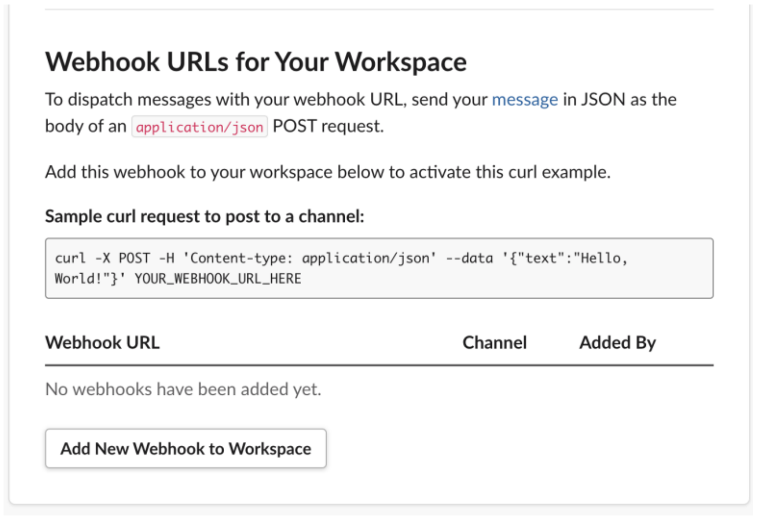
Activate Incoming Webhooks

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details. You can include message attachments to display richly-formatted messages.

Adding incoming webhooks requires a bot user. If your app doesn't have a bot user, we'll add one for you.

Each time your app is installed, a new Webhook URL will be generated.

6. Turn the Off switch to On to activate incoming webhooks for this app.
7. You will now see something like the image below. Click on the “Add New Webhook to Workspace” button.



Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your message in JSON as the body of an `application/json` POST request.

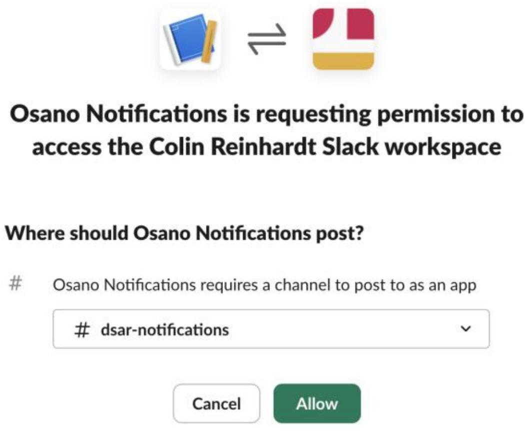
Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello,
World!"}' YOUR_WEBHOOK_URL_HERE
```

Webhook URL	Channel	Added By
No webhooks have been added yet.		

8. You will be brought to a new page where you can select the Slack channel in which this webhook will post messages. Choose your desired channel from the dropdown menu and click “Allow”.

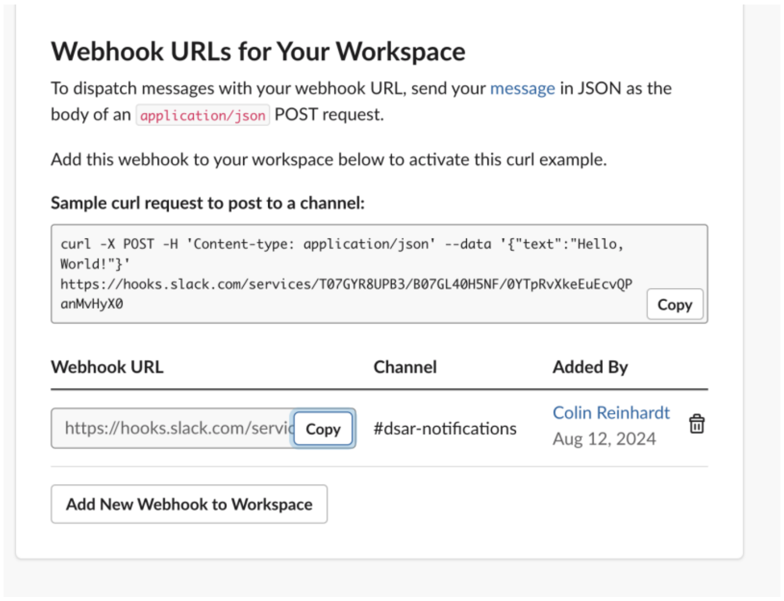


Osano Notifications is requesting permission to access the Colin Reinhardt Slack workspace

Where should Osano Notifications post?

Osano Notifications requires a channel to post to as an app

9. You will then be brought back to the previous app overview page. You will see a new “Webhook URL” field. Copy this URL.



Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your message in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

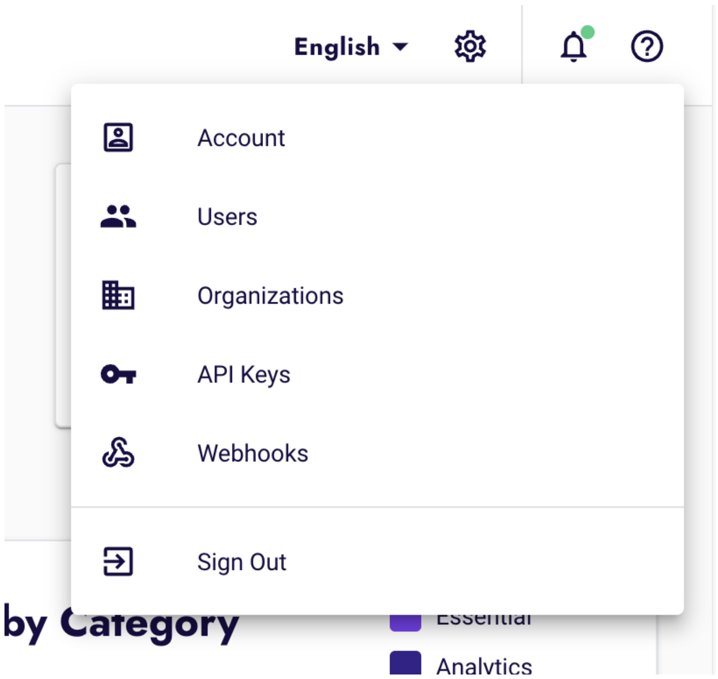
```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello,
World!"}'
https://hooks.slack.com/services/T07GYR8UPB3/B07GL40H5NF/0YTpRvXkeEuEcvQP
anMvHyX0
```

Webhook URL	Channel	Added By
https://hooks.slack.com/servic	#dsar-notifications	Colin Reinhardt Aug 12, 2024

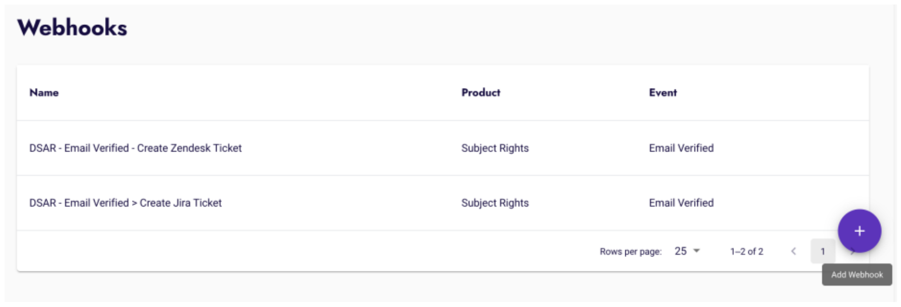
Create the webhook in Osano

Now that we have an app created in Slack and connected to a channel in the workspace, we can create a webhook to send a message to that channel with the relevant information.

1. With the URL copied from the Slack application, navigate to my.osano.com and log into your Osano account.
2. Once logged into Osano, click on the gear icon in the top right corner and choose "Webhooks".



3. Click on the purple plus button to create a new Webhook.

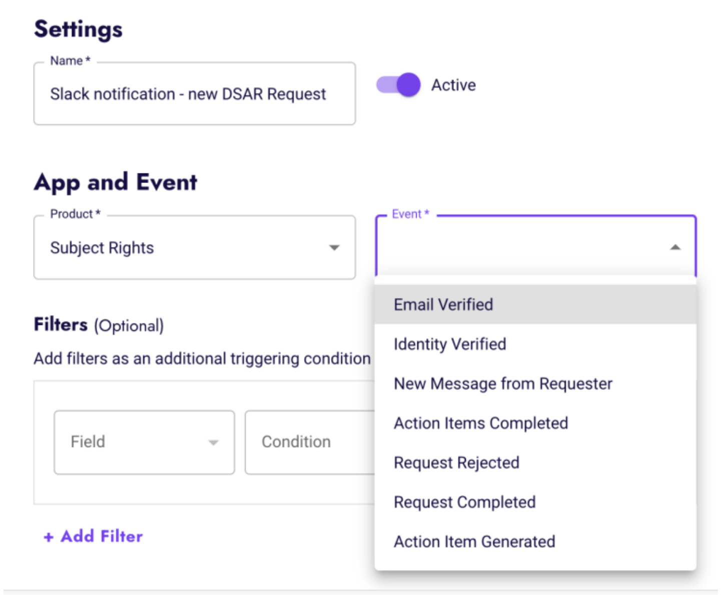


Webhooks

Name	Product	Event
DSAR - Email Verified - Create Zendesk Ticket	Subject Rights	Email Verified
DSAR - Email Verified > Create Jira Ticket	Subject Rights	Email Verified

4. On the next screen, you can specify the details of your new webhook. For this example, we will be creating a webhook that sends a new message to Slack when the Email address of a requester has been verified.

5. Provide the webhook with a name. Next, click the Product dropdown, and choose “Subject Rights. Finally, click the Event dropdown and choose “Email verified”.


Settings

Name *: Slack notification - new DSAR Request

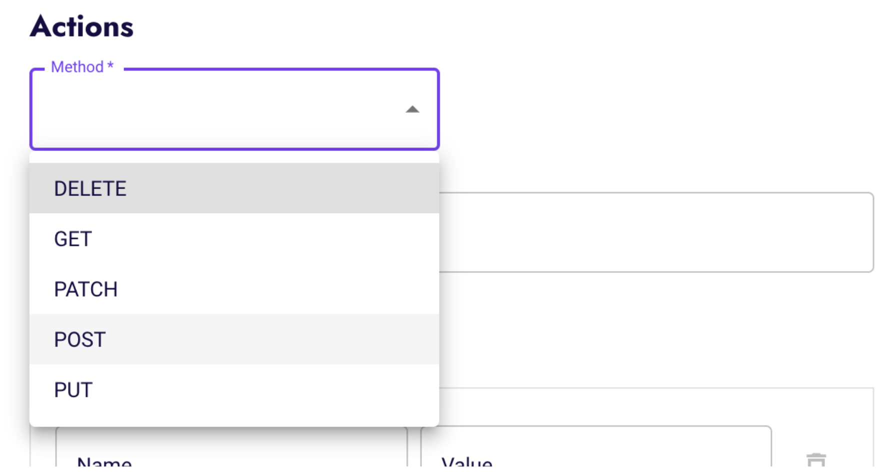
Active

App and Event

Product *: Subject Rights

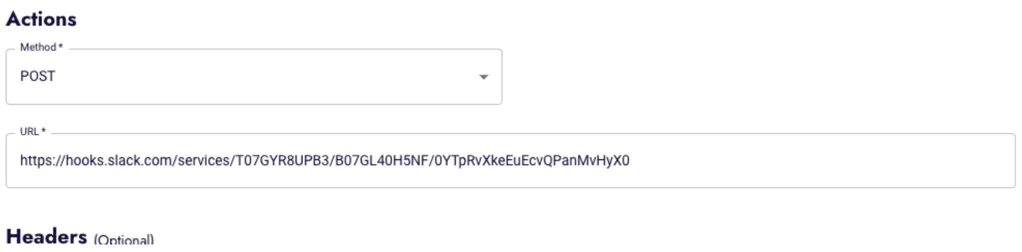
Event *

6. Scroll down and click on the “Method” drop down list and choose “POST”


Actions

Method *

7. In the URL field, paste the webhook URL you copied from Slack. If you no longer have this copied, return to api.slack.com and find your webhook URL.


Actions

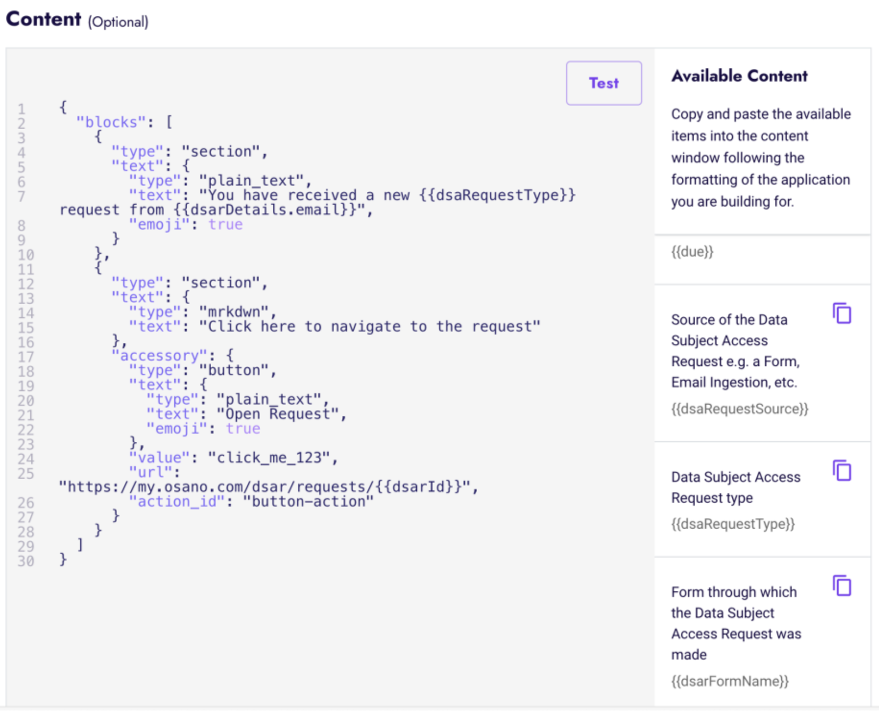
Method *: POST

URL *: https://hooks.slack.com/services/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

8. In the Content section, you will be able to enter the JSON Body of what you would like included in your Slack message. Depending on the application you are

interacting with, the format of the content body may be different from others. It is best to consult the support documentation to ensure you have the proper formatting for the outgoing webhook content. Note that to test your webhook, you do not need to include any content.

a. In the example below, we have added content in JSON format to send a message containing the requester email address, along with a button to navigate directly to the specific request.
b. Along the right-hand side, you will see a list of available content placeholders that can be used to provide details of the request in the webhook content. When the webhook fires after an actual request is sent, these placeholders will be replaced with their applicable information from the request.



```
{
  "blocks": [
    {
      "type": "section",
      "text": {
        "type": "plain_text",
        "text": "You have received a new {{dsaRequestType}}
request from {{dsarDetails.email}}",
        "emoji": true
      }
    },
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "Click here to navigate to the request"
      },
      "accessory": {
        "type": "button",
        "text": {
          "type": "plain_text",
          "text": "Open Request",
          "emoji": true
        },
        "value": "click_me_123",
        "url":
"https://my.osano.com/dsar/requests/{{dsarId}}",
        "action_id": "button-action"
      }
    }
  ]
}
```

Available Content

Copy and paste the available items into the content window following the formatting of the application you are building for.

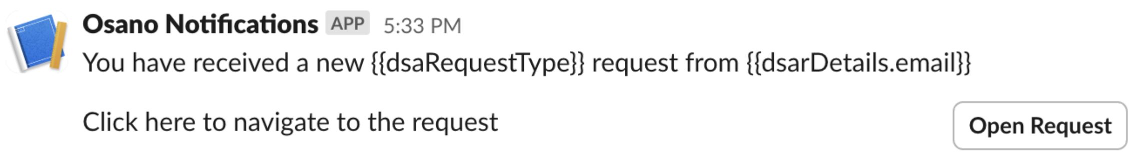
{{due}}

Source of the Data Subject Access Request e.g. a Form, Email Ingestion, etc.
{{dsaRequestSource}}

Data Subject Access Request type
{{dsaRequestType}}

Form through which the Data Subject Access Request was made
{{dsarFormName}}

9. Click on the “Test” button to send a test webhook to the provided URL. If you have properly configured your webhook, you should see a green notification in the bottom right corner. Navigate to the channel in your slack workspace to verify that the message has successfully been sent.



Osano Notifications APP 5:33 PM
You have received a new {{dsaRequestType}} request from {{dsarDetails.email}}

Click here to navigate to the request