

Sending messages to Slack using incoming webhooks

1. Create a Slack app

- go to [Slack API: Applications | Slack](#)

- Click Create New App button

Your Apps

Create New App

- Choose "From a manifest"

Create an app ×

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

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

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


Need help? Check our [documentation](#), or [see an example](#)

- Choose a workspace

Pick a workspace to develop your app ✕

Pick a workspace to develop your app in:

Select a workspace ▾

 **AtlasX**

workspace will control the app even if you leave the workspace.

[Sign into a different workspace](#)

Step 1 of 3

Cancel **Next**

- Change App name at display_information > name

Create app from manifest ✕

This is your app's manifest containing basic info, scopes, settings, and features. For help on how this works, you can check out our [documentation](#) or check out a few [examples](#).

YAML JSON

```
1 display_information:
2   name: AtlasX Messaging
3 settings:
4   org_deploy_enabled: false
5   socket_mode_enabled: false
6   is_hosted: false
7   token_rotation_enabled: false
8
```

Step 2 of 3

Back

Next

- Click Create button

Review summary & create your app ✕

AtlasX Messaging

Features

App Home

Step 3 of 3



[Back](#) [Create](#)

2. Enable incoming webhooks

- Open App page at [Slack API: Applications | Slack](#) and click at App name

Your Apps [Create New App](#)

Filter apps by name or workspace

App Name	Workspace	App ID	App Type	Distribution Status
 AtlasX Messaging	AtlasX	A08GPQVM41J	Modern	Not distributed
 AtlasX Notification	AtlasX	A08FZ2R82R4	Modern	Not distributed

- Click Incoming Webhooks on left menu



AtlasX Messagi... ▼

Settings

Basic Information

Collaborators

Socket Mode

Install App

Manage Distribution

Features

App Home

Agents & AI Apps NEW

Workflow Steps NEW

Org Level Apps

Incoming Webhooks

Interactivity & Shortcuts

Slash Commands

Steps from Apps LEGACY

- Toggle Activate Incoming Webhooks to On

Incoming Webhooks

Activate Incoming Webhooks

On

3. Create an incoming webhook

- Click 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.

Add New Webhook to Workspace

- Choose a channel to post message to then click Allow button



AtlasX Messaging is requesting permission to access the AtlasX Slack workspace

Where should AtlasX Messaging post?

AtlasX Messaging requires a channel to post to as an app

notifications

Cancel

Allow

- Click Copy button to save Webhook URL to clipboard

Webhook URL	Channel	Added By
https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXXXX Copy	#notifications	metha.c Mar 6, 2025 

Add New Webhook to Workspace

4. Use your incoming webhook URL to post a message

Make an HTTP POST request like this:

POST <https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXXXX>

Content-type: application/json

```
{  
  "text": "Hello, world."  
}
```

C# Code Example

```
public class SlackMessagingService  
{  
    private HttpClient _httpClient;  
    private IConfiguration _config;  
  
    public SlackMessagingService(IConfiguration configuration, HttpClient  
httpClient)  
    {  
        _config = configuration;  
        _httpClient = httpClient;  
    }  
  
    public async Task<Tuple<int, string>> Send(string bodyMessage, string app =  
"Default")  
    {  
        var incomingWebhookUrl =  
_config.GetValue<string>($"WebServiceSettings:Messaging:Providers:Slack:Apps:{app  
}:IncomingWebhookUrl");  
        var slackMessageJson = $"{{\"text\": \"{bodyMessage}\"}}";  
        using var payload = new StringContent(slackMessageJson,  
System.Text.Encoding.UTF8, "application/json");  
        using var req = new HttpRequestMessage(HttpMethod.Post,  
incomingWebhookUrl) { Content = payload };  
  
        using var response = await _httpClient.SendAsync(req);  
        using var resContent = response.Content;  
        var resMessage = await resContent.ReadAsStringAsync();  
        return new Tuple<int, string>((int)response.StatusCode, resMessage);  
    }  
}
```