

Task 4: Deploying to Facebook Messenger

Facebook Messenger is a messaging platform that, among personal uses, can help businesses and customers communicate with each other. You can automate the communication with your customers by deploying your Watson-powered chatbot to Facebook Messenger for your business. This way, you'll promptly respond to customer queries even when you are not around.

For this lab you will integrate the basic chatbot that we built in the previous module of this course with Facebook Messenger on your Facebook page.

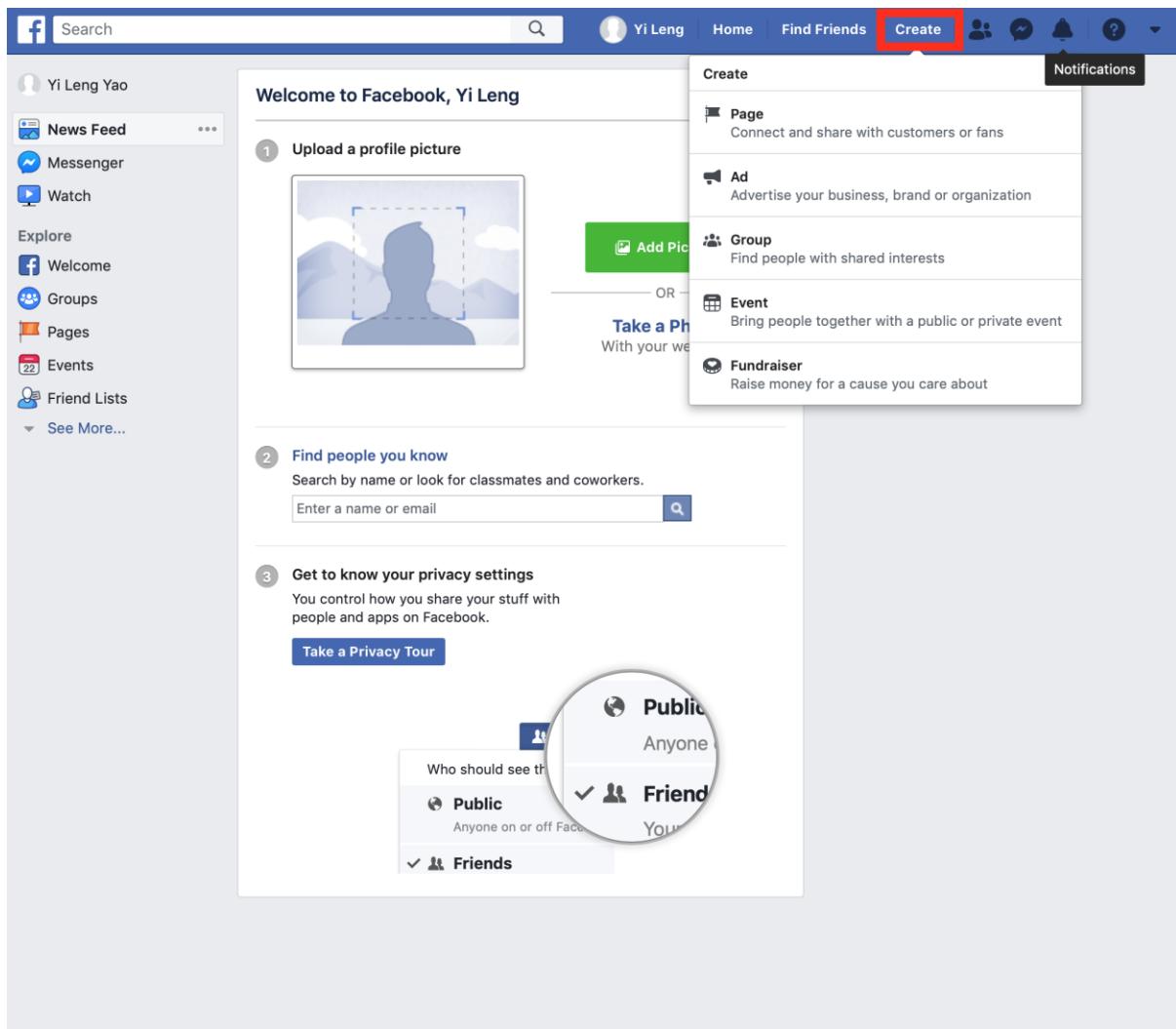
Create a Facebook Page and App

We'll need a business page in order to deploy our chatbot to Facebook. So let's get that done right away.

1. **Sign up for a Facebook account.** If you already have a Facebook account you can skip this step.

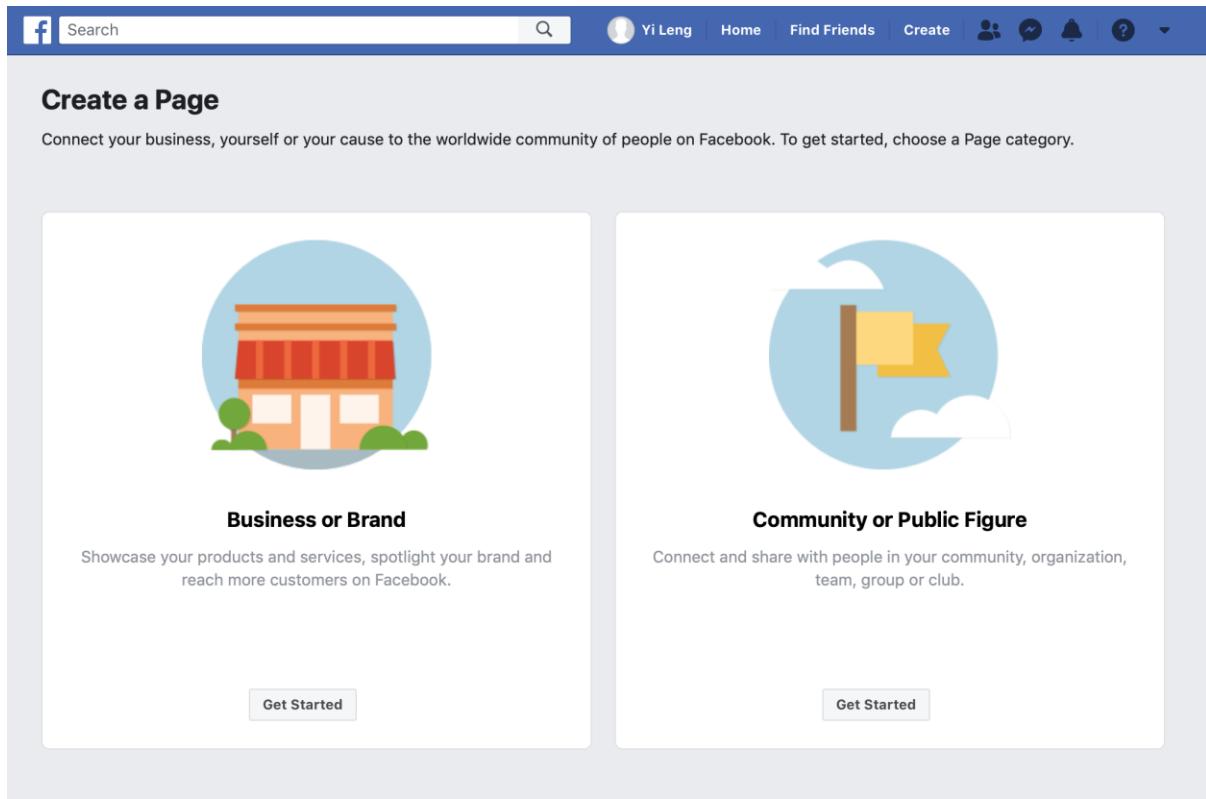
The screenshot shows the Facebook sign-up interface. At the top, there is a blue header with the word "facebook" in white. To the right of the logo are fields for "Email or Phone" and "Password", followed by a "Log In" button and a "Forgot account?" link. Below the header, the word "Sign Up" is prominently displayed in large, bold, black font. Underneath it, the text "It's free and always will be." is written in a smaller, gray font. On the left side, there is a section titled "Connect with friends and the world around you on Facebook." with three bullet points: "See photos and updates from friends in News Feed.", "Share what's new in your life on your Timeline.", and "Find more of what you're looking for with Facebook Search.". Each bullet point has a small blue icon next to it. To the right of these points, there are input fields for "First name", "Last name", "Mobile number or email", and "New password". Below these fields are dropdown menus for "Birthday" (set to May 21, 1994) and gender selection ("Female" or "Male"). A link "Why do I need to provide my birthday?" is located near the birthday field. At the bottom of the form, there is a green "Sign Up" button. Below the button, a small link "Create a Page for a celebrity, band or business." is visible.

2. **Create a Facebook Page** to host your Watson Assistant Chatbot. To do so, in the top right corner of Facebook, you'll find a **Create** button. Click on it and then select **Page** from the drop down menu that appears.



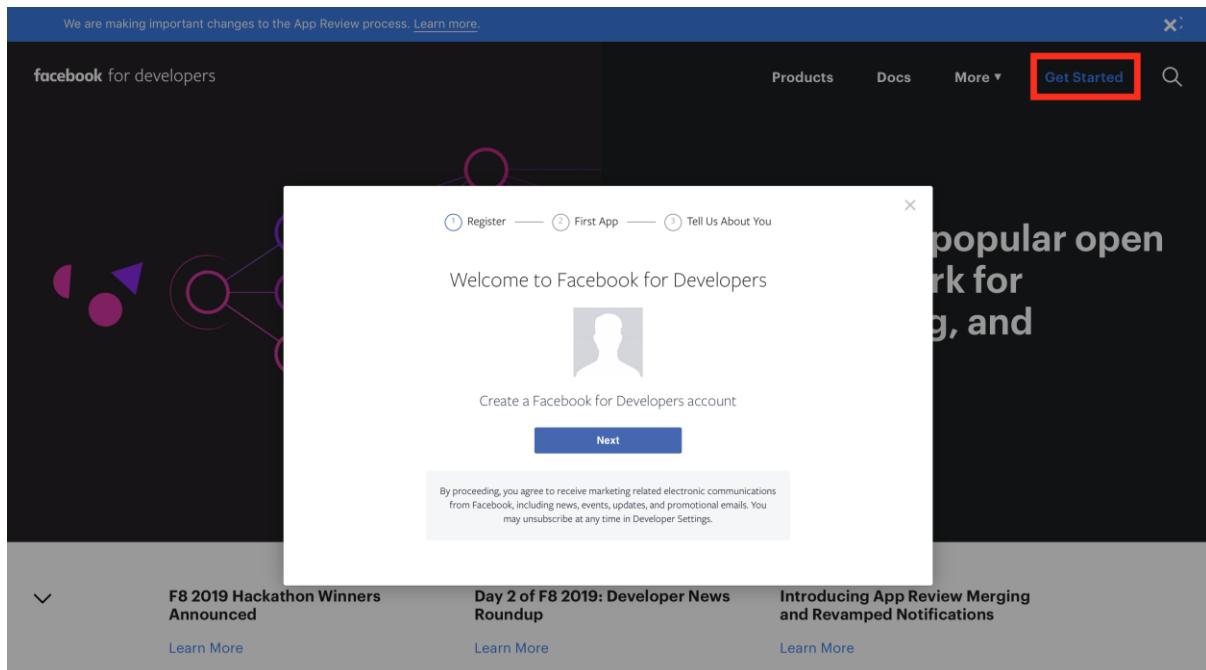
The screenshot shows the Facebook homepage for user 'Yi Leng Yao'. The top navigation bar includes links for 'Search', 'Home', 'Find Friends', 'Create' (which is highlighted with a red box), and 'Notifications'. On the left, there's a sidebar with links for 'News Feed', 'Messenger', 'Watch', 'Explore', 'Welcome', 'Groups', 'Pages', 'Events', 'Friend Lists', and a 'See More...' button. The main content area displays a 'Welcome to Facebook, Yi Leng' message, steps for 'Upload a profile picture' (with a placeholder image), 'Find people you know' (with a search bar), and 'Get to know your privacy settings' (with a link to 'Take a Privacy Tour'). A circular callout highlights the 'Public' option under privacy settings, which is set to 'Anyone'. To the right of the main content is a 'Create' menu with options: 'Page' (Connect and share with customers or fans), 'Ad' (Advertise your business, brand or organization), 'Group' (Find people with shared interests), 'Event' (Bring people together with a public or private event), and 'Fundraiser' (Raise money for a cause you care about).

You are now presented with two categories of pages that you can create, **Business or Brand** and **Community or Public Figure**. Just pick one, as either category is fine for the purpose of this lab.

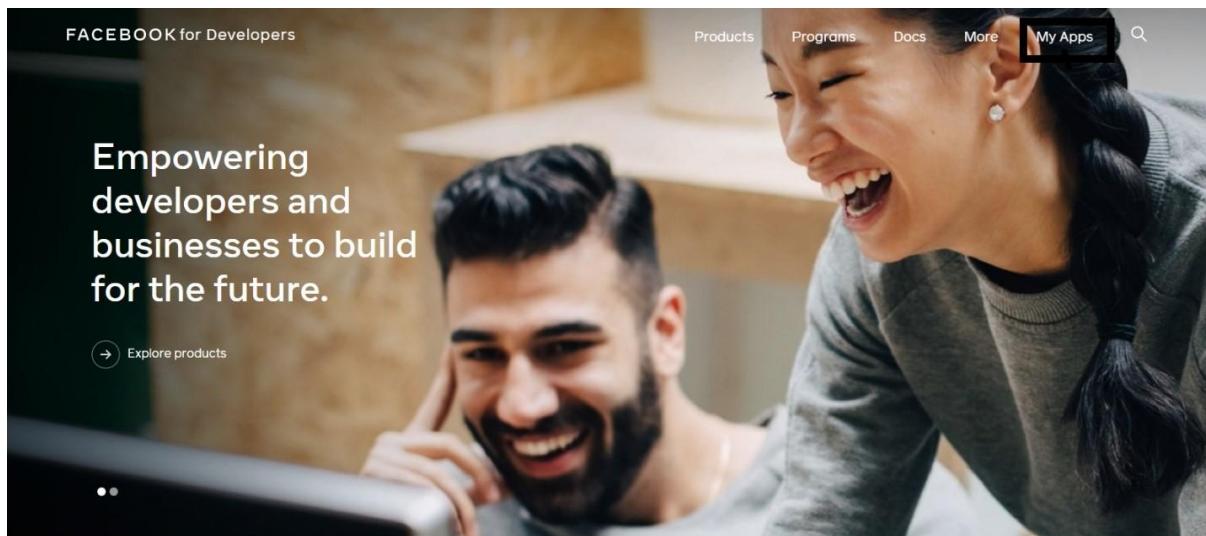


Now that you created your Facebook page, you'll need to create a Facebook application that will integrate with our chatbot.

3. **Create or login in your Facebook Developer account.** If you don't have an account you can create one by clicking Get Started in the upper right corner of the Facebook for Developers page, as shown in the figure below.



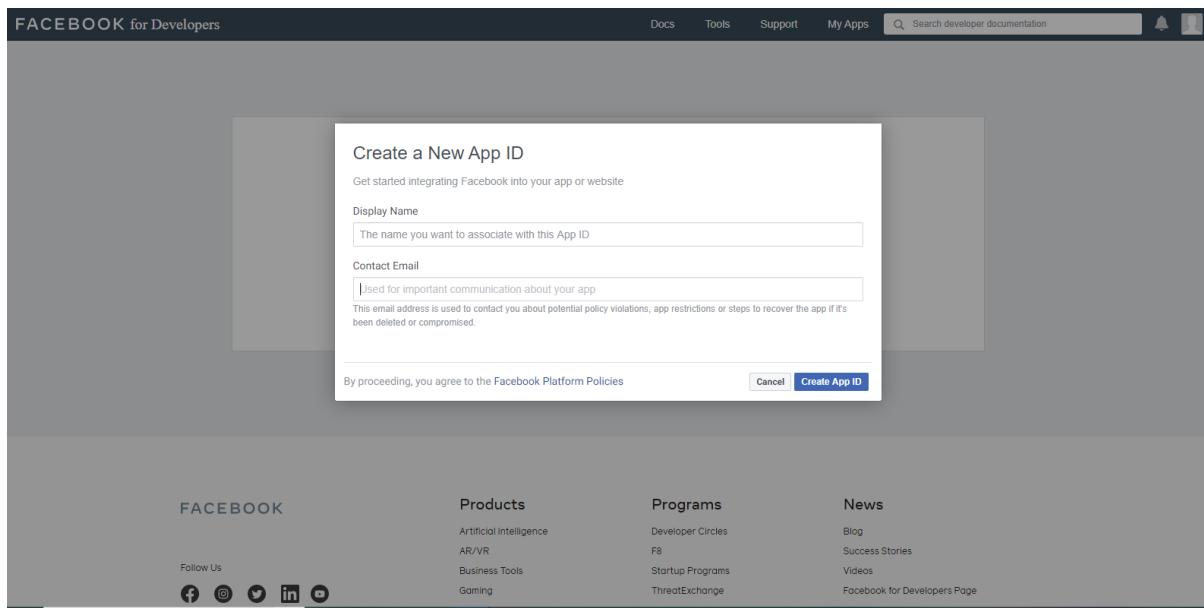
4. Once you have a Facebook Developer account you can **create your Facebook App** by clicking on **My Apps** in the top right corner of the webpage.



DEVELOPER TOOLS

This screenshot shows the "No Apps Created" page under the "My Apps" section of the developer tools. It features a central message "No Apps Created" with a small icon above it, followed by the subtext "You currently have no apps created." and a "Create App" button. The page has a light gray background with some darker gray sections on the sides.

5. You'll be asked to fill in the Display Name and the Contact Email, fill the required details and click on Create App ID.



You'll be asked to select a scenario. For our purposes, feel free to skip this step by clicking on the **Skip** button.

The screenshot shows the 'Select a Scenario' page. It lists four options: 'Implement Marketing API', 'Get Started with the Ads Insights API', 'Integrate Facebook Login', and 'Get Started with the Pages API'. Each option has a description and a list of benefits. At the bottom right, there are 'Skip' and 'Confirm' buttons, with 'Skip' being highlighted by a red box.

6. In order for your Watson chatbot to connect to your Facebook app, you'll need the App Secret for your Facebook app. Click on **Settings** in the sidebar and then select the **Basic** option in the dropdown menu.

The screenshot shows the Facebook Developers Dashboard for the app "Watson-Chatbot-Demo". The sidebar on the left has "Settings" selected, indicated by a red box. The main area shows the "Application-Level Rate Limit" and "User-Level Rate Limit" sections. The Application-Level Rate Limit shows 0% of the limit used with 100% remaining. The User-Level Rate Limit shows 0 users throttled. Below these are sections for API Stats, Calls, Errors, and Average Request Time, all showing "No data is available".

From this section, obtain your **App Secret** credentials.

The screenshot shows the Facebook Developers Dashboard for the app "Watson-Chatbot-Demo". The sidebar on the left has "Basic" selected, indicated by a red box. The main area displays basic app settings: App ID (333315910664704), App Secret (redacted and highlighted with a red box), Display Name (Watson-Chatbot-Demo), Namespace, App Domains, Contact Email (yilengyao@gmail.com), Privacy Policy URL, Terms of Service URL, App Icon (with a placeholder image and dimensions 1024 x 1024), Category (Choose a Category), and Business Use options (Support my own business, Provide services to other businesses). The "App Secret" field is specifically highlighted with a red box.

7. Now we need to connect Watson Assistant to our Facebook page. Click on **PRODUCTS** in the sidebar on the left.

Screenshot of the Facebook for Developers app settings page for "Watson-Chatbot-Demo".

The left sidebar shows navigation options: Dashboard, Settings (Basic selected), Advanced, Roles, Alerts, App Review, and PRODUCTS (highlighted with a red box).

The main content area displays the following fields:

- App ID: 333315910664704
- Status: In Development
- View Analytics
- Help

Form fields (Basic tab):

- App ID: 333315910664704
- App Secret: (redacted)
- Show
- Display Name: Watson-Chatbot-Demo
- Namespace: (empty)
- App Domains: (empty)
- Contact Email: yilengyao@gmail.com
- Privacy Policy URL: Privacy policy for Login dialog and App Details
- Terms of Service URL: Terms of Service for Login dialog and App Details
- App Icon (1024 x 1024): (image placeholder)
- Category: Choose a Category
- Find out more information about app categories here

Business Use section:

This app uses Facebook tools or data to

- Support my own business
- Provide services to other businesses

Data Protection Officer Contact Information:

The General Data Protection Regulation (GDPR) requires certain companies doing business in the European Union to designate a Data Protection Officer who people can contact for information about how their data is being processed.

This contact information will be available to people on Facebook along with other information about your app or website. [Learn More](#).

Scroll down to the **Add a Product** section. Click on **Set Up under the Messenger option** as shown below.

facebook for developers

Docs Tools Support My Apps Search developers.facebook.com

Watson-Chatbot-De... APP ID: 333315910664704 OFF Status: In Development View Analytics Help

Dashboard Settings Roles Alerts App Review

PRODUCTS + Add a Product

Account Kit

Facebook Login

Audience Network

Analytics

Messenger

Webhooks

Instant Games

Marketing API

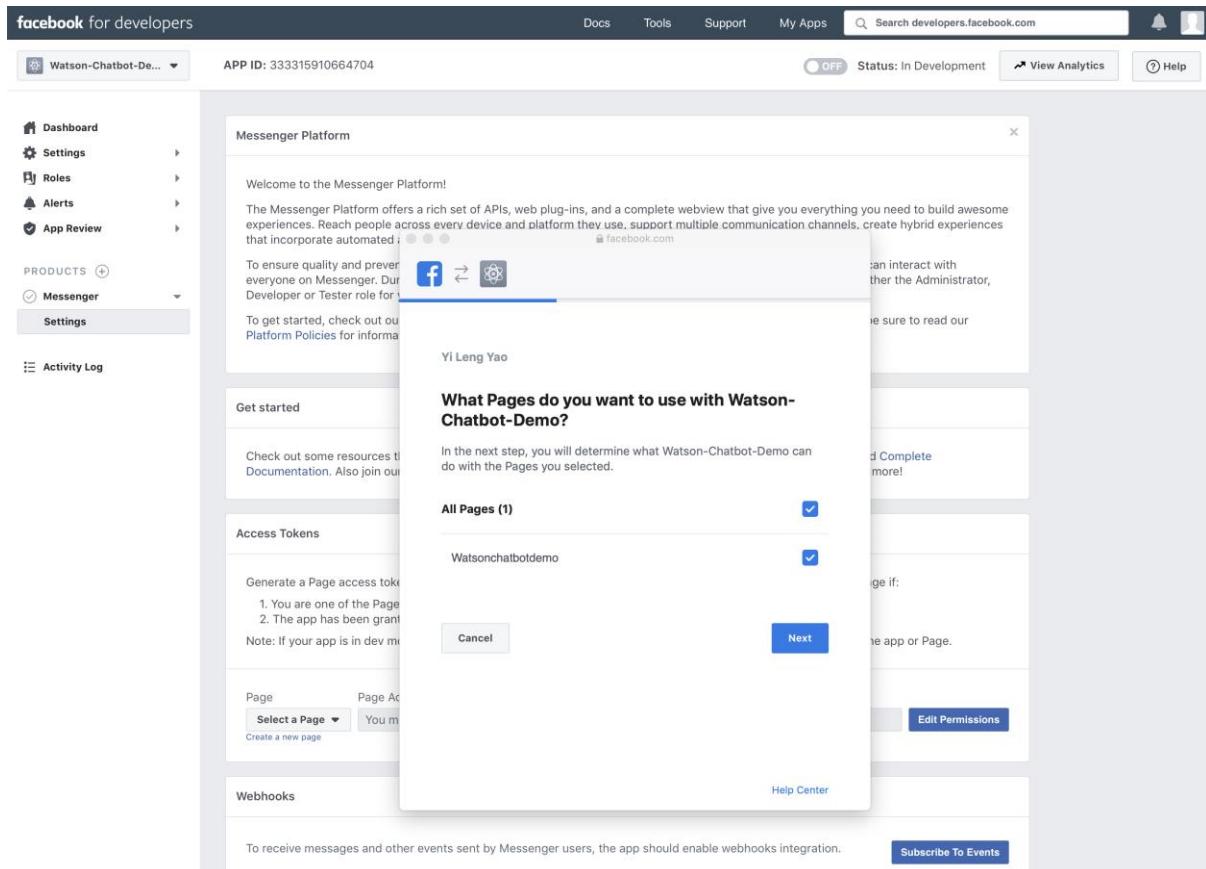
App Center

8. In the **Access Tokens** section click on **Add or Remove Pages** then click on **Continue**.

The screenshot shows the Facebook for Developers dashboard with the following details:

- Left Sidebar:** Includes links for Dashboard, Settings, Roles, Alerts, App Review, Products (Messenger selected), and Activity Log.
- Header:** Shows "Watson Assistant;Co..." as the app name, "App ID: 3226568850720418", and "In development".
- Main Content:**
 - Access Tokens:** Instructions for generating a page access token, noting that the app is in dev mode. A note states: "If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page."
 - No page permissions granted:** A message stating: "You'll need to connect pages and grant them the required permissions in order for tokens to be generated." A "Add or Remove Pages" button is present.
- Messenger Platform Modal:** A modal window titled "Messenger Platform" with the following content:
 - Welcome to the Messenger Platform!
 - The Messenger Platform offers a rich set of APIs, web plug-ins, and a complete webview that give you everything you need to build awesome experiences. Reach people across every device and platform they use, support multiple communication channels, create hybrid experiences that incorporate automated and live chat, and more, all in Messenger.
 - To ensure quality and prevent abuse of the Platform, all Messenger apps must be submitted for review before they can interact with everyone on Messenger. If your app is in dev mode, it will be reviewed by either the Administrator, Developer or Tester role.
 - To get started, check out our Platform Policies for info.
- Bottom Buttons:** "Cancel" and "Continue as Yi Leng Yao" (highlighted with a red box).

9. Select the Facebook page that you want to integrate your app with.



10. Generate the Page Access Token for your Facebook page. In the Access Tokens section click on **Generate Token**. Then save the Page Access Token.

Pages ↑	Tokens
Watsonchatbotdemo 111104543883619	—

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Now that we have the **APP SECRET** and **Page Access Token** for our Facebook app, the next step is to integrate it with Watson Assistant.

Integrate your Facebook App with Watson Assistant

1. Launch your Watson Assistant instance and click on the Assistants section. When there, **select the Student Advisor Chatbot**.

2. Click on the **Add Integrations button** on the right side of the page.

The screenshot shows the 'Student Advisor Chatbot' details on the left and the 'Integrations' section on the right. The 'Add integration' button is highlighted with a red box.

Skills
A skill is building block of your assistant. A dialog skill is created by authoring intents, entities, and nodes. A search skill helps your dialog skill by providing answers from linked documents or web pages. [Learn more](#)

Student Advisor

LANGUAGE: TRAINED DATA: VERSION: CREATED: UPDATED:
English (US) 13 Intents | 1 Entities | 15 Dialog Nodes Development May 30 2019, 11:51 -07:00 Jun 10 2019, 09:45 -07:00

LINKED ASSISTANTS (1): Student Advisor Chatbot

Integrations
Choose a channel to deploy your Assistant.
[Add integration](#)

Preview Link

Then select Facebook Messenger.

The screenshot shows the 'Add Integration' modal with the 'Facebook Messenger' option highlighted.

IBM Watson Assistant Cookie Preferences ? ?

Add Integration
Select a deployment method that is managed for you and can be configured within the tool or learn about other ways to deploy the assistant.

Managed integrations

- Slack
Make the assistant available to customers by adding it to a Slack app as a bot user.
- Preview Link
Embed the assistant in a chat widget hosted on an IBM-branded web page that your team can use for testing.
- Facebook Messenger
Make the assistant available to customers through Facebook Messenger on the web or on native mobile clients.

Other integrations

- [Custom application](#)
- [Voice Agent \(Telephony\)](#)
- [WordPress plug-in](#)

3. Paste the APP SECRET credentials of your Facebook App into the Application secret box.



Facebook Messenger Integration

Name

Facebook Messenger

Description

An integration with the Facebook Messenger platform.

Configuration for Facebook Messenger

GDPR Warning: Facebook integrations are not currently GDPR compliant. [Learn more](#)

Step 1

Create a Facebook application

Go to <http://developers.facebook.com> and log in with your Facebook credentials. Click **Add a New App** and complete the steps to create a new app ID.

*Note: If you have already created the app you want to use, select it from the **My Apps** menu.*

In the navigation pane, click **Settings -> Basic**. Under **App Secret**, click **Show**. Copy the app secret and paste it here:

Application secret

4. Proceed to Step 2 on the page, and **paste the Page Access Token from your Facebook app** in the **Page access token** box.

And then save the **Generate verify token**.



Step 2

Connect Watson Assistant to Facebook

- 1 On the Facebook app page, click the plus sign (+) next to **Products** in the left side navigation.
- 2 Under **Add a product**, find the **Messenger** tile and click **Set Up**.
- 3 In the Messenger settings, scroll down to **Token Generation**.
- 4 Click **Select a Page** and choose the Facebook page you want to use for your app.

*Note: If you do not already have a page for your app, click **Create a new page**. After you finish creating the page, return to the Facebook apps page and navigate back to the Messenger settings for your app. You can then select the page you created.*

Copy the page access token and paste it here:

Page access token

The **Generated verify token** field contains a generated verify token that Facebook can use to verify your webhook URL.

Generated verify token

Step 3

Configure Facebook webhooks

[Generate callback URL](#)

In the Facebook Messenger settings, scroll to the **Webhooks** section. Click **Setup Webhooks**.

5. Proceed to Step 3. There, **click on Generate callback URL and save your callback URL**.



Step 3

Configure Facebook webhooks

[Generate callback URL](#)

In the Facebook Messenger settings, scroll to the **Webhooks** section. Click **Setup Webhooks**.

Step 4

Connect your assistant

In the Messenger settings, go back to the **Webhooks** section and click **Select a Page**. Select the same page you selected in step 2, and then click **Subscribe**.

Note: Subscribe to only one page. Multiple-page subscriptions are not currently supported.

Congrats! You should be able to test your bot now in Messenger. You must be a Page admin to interact with it until you make it public.

Step 5

Start app review process

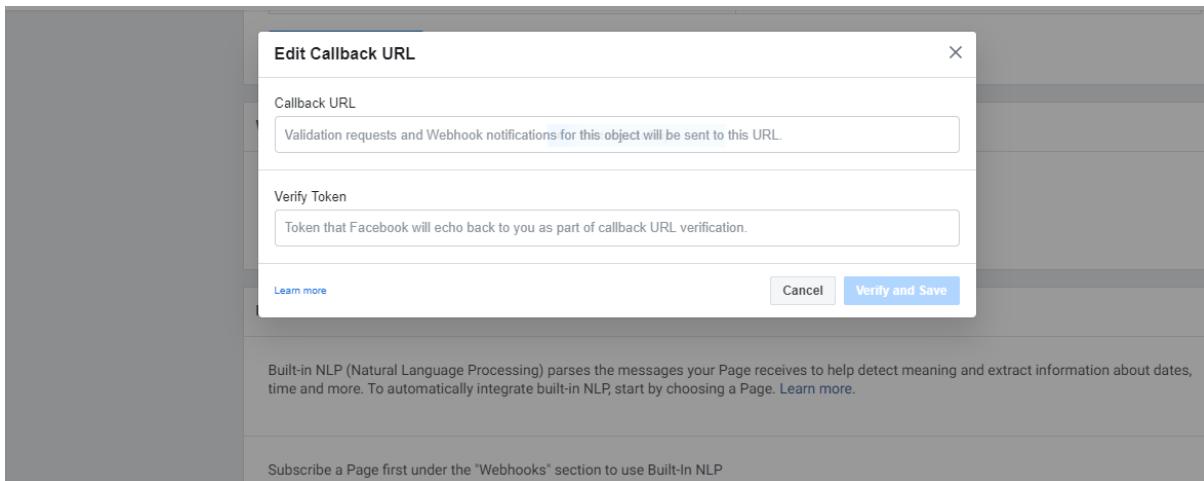
Once you have confirmed your bot is working in Messenger, and you would like to make the bot public, go to <https://developers.facebook.com/docs/messenger-platform/app-review/> and follow the instructions to start the app review process.

6. Go back to the page for your Facebook App, if you closed it you can find it again from [this link](#).

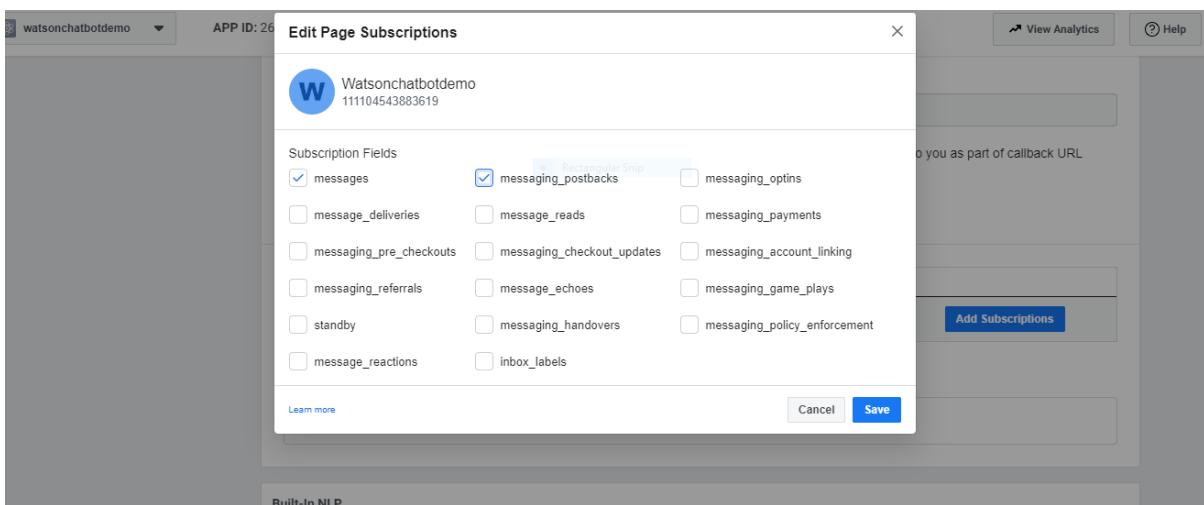
7. In the left sidebar, under the **PRODUCTS** section click on **Settings**. Then under the **Webhooks** section **click on Add Callback URL**.

The screenshot shows the Facebook for Developers Settings page. The left sidebar has sections for Dashboard, Settings, Roles, Alerts, App Review, PRODUCTS (Messenger), and Activity Log. The Messenger section is expanded. The main content area has tabs for Add or Remove Pages, Webhooks, Built-In NLP, and App Review for Messenger. The Webhooks tab is active, showing a section to receive messages and other events via webhooks integration. It includes an 'Add Callback URL' button. Below it is the Built-In NLP section, which parses messages for dates, time, and more. The App Review for Messenger section explains the need for app approval and has a 'pages_messaging' radio button selected, with a note about enabling messaging. There are also 'View Analytics' and 'Help' buttons at the top right.

8. Paste the **callback URL** and **Verify Token** from your Watson Assistant. Then click **Verify and Save**



9. Then click on **Add Subscription** and select **messages** and **messaging_postbacks**. Then click save.



10. Congratulations, your Watson Assistant chatbot is now integrated with Facebook Messenger.

In the meantime, you can **test your chatbot app** on Messenger, go to the page for your Facebook app, click on **Roles** in the sidebar on the right. Then you add can add testers that can communicate with your chatbot during this testing phase.

The screenshot shows the Facebook for Developers Roles page. The sidebar on the left has a red box around the 'Roles' section. The main content area has four sections: Administrators, Developers, Testers, and Analytics Users. Each section has a 'Add [Section]' button. The Administrators section shows one user: Yi Leng Yao. The Developers section says 'There are no developers for this app.' The Testers section shows one user: Yi Y (Pending). The Analytics Users section says 'There are no Analytics users for this app.'

Go head and test your chatbot on the Facebook Page you designated. Bear in mind that your chatbot app is not public yet. Only testers get to have access to your chatbot. Please refer to [these instructions from Facebook](#) should you want to make your app public.

Please note that Facebook Messenger **does not support embedding arbitrary HTML**. So if a chatbot deployment target includes Facebook Messenger, it might make sense to simply have a link in your responses instead of the HTML code for said link (e.g., `...`).

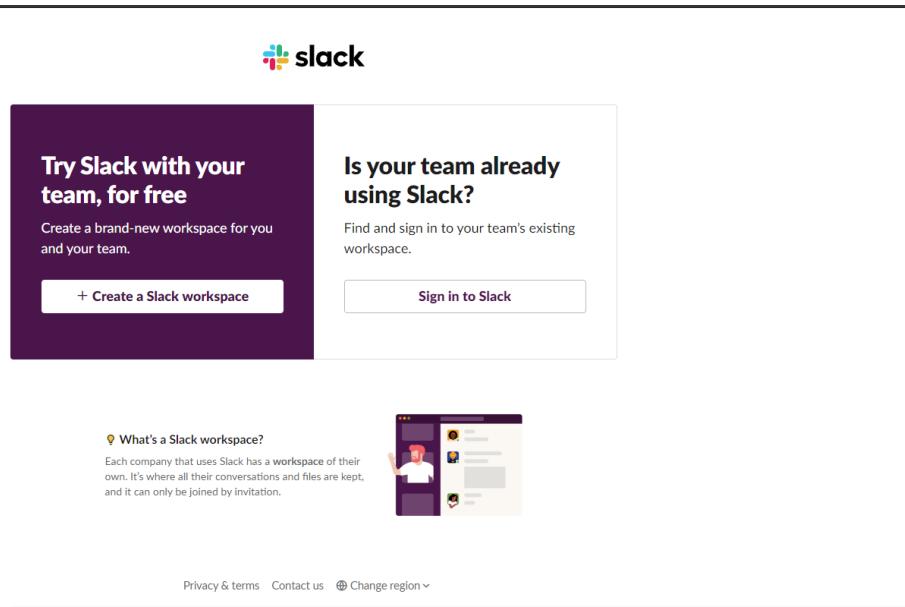
Task 5-Deploying Your Chatbot on Slack

Slack is a communication tool, typically used by people working in a team. It can also be a channel for you to connect with friends, family or anyone that you want to share information with. You can improve the workflow in your team or set reminders of important events with your friends and family by deploying chatbots on Slack.

For this lab you will integrate our chatbot with Slack.

Create a Slack App

1. Head over to [Slack.com](https://slack.com), click on the **Try Slack** button, then click on **Create a Slack Workspace**.



2. Next, enter your email address and click on the **Confirm** button. You might be asked to enter a confirmation code that will be sent to the email address you specified.
3. Specify a team and project name of choice. Feel free to skip inviting teammates.

Step 1 of 3

What's the name of your company or team?

e.g. A1 or A1 Marketing 255

From tiny teams to entire companies, everyone can use Slack
Start with the people that you work with every day – you can always expand and change this name later.

Next

Chatbot-Demo
Anamika Step 2 of 3

What would you like to use Slack for first?

Choose a few. We'll use these to help you get up and running in Slack.

Working on a project Staying connected Sharing ideas
Making announcements Replacing a meeting

Pick at least two Skip this step

Chatbot-Demo
Anamika Step 2 of 3

What would you like to use Slack for first?

Choose a few. We'll use these to help you get up and running in Slack.

Working on a project Staying connected Sharing ideas
Making announcements Replacing a meeting

In Slack, each of these can be a channel
Channels keep your conversations organised. They give each project, topic and team a dedicated space for messages and files.

Next Skip this step

The image consists of two vertically stacked screenshots of the Slack application interface.

Top Screenshot: This screenshot shows the "Add team members" step of a process. The title is "Who would you like to use Slack with?". A text input field contains "Example ellis@gmail.com". Below the input field are two buttons: "Add team members" (in a purple button) and "Skip this step" (with a black border). The background shows a dark sidebar with the workspace name "Chatbot-Demo" and a user profile for "Anamika".

Bottom Screenshot: This screenshot shows the Slack workspace interface. The left sidebar lists channels: "# ideas", "# watson-assistant-chatbot" (which is selected and highlighted in blue), and "# welcome". The main area displays the "# watson-assistant-chatbot" channel. The channel header is "# watson-assistant-chatbot" with a description: "This channel is for working on a project. Hold meetings, share docs and make decisions together with your team.". Below the header, there are two buttons: "Share a project file" and "Share daily updates". A message from "Anamika" at 15:21 is shown: "joined #watson-assistant-chatbot". At the bottom of the channel view, there is a message input field with placeholder text "Message #watson-assistant-chatbot" and a set of rich text editing icons.

Now that we have a Slack workspace and channel, we need to proceed to create a Slack App to use to communicate with our existing chatbot.

4. Check your email to find the URL of your slack as shown in figure.



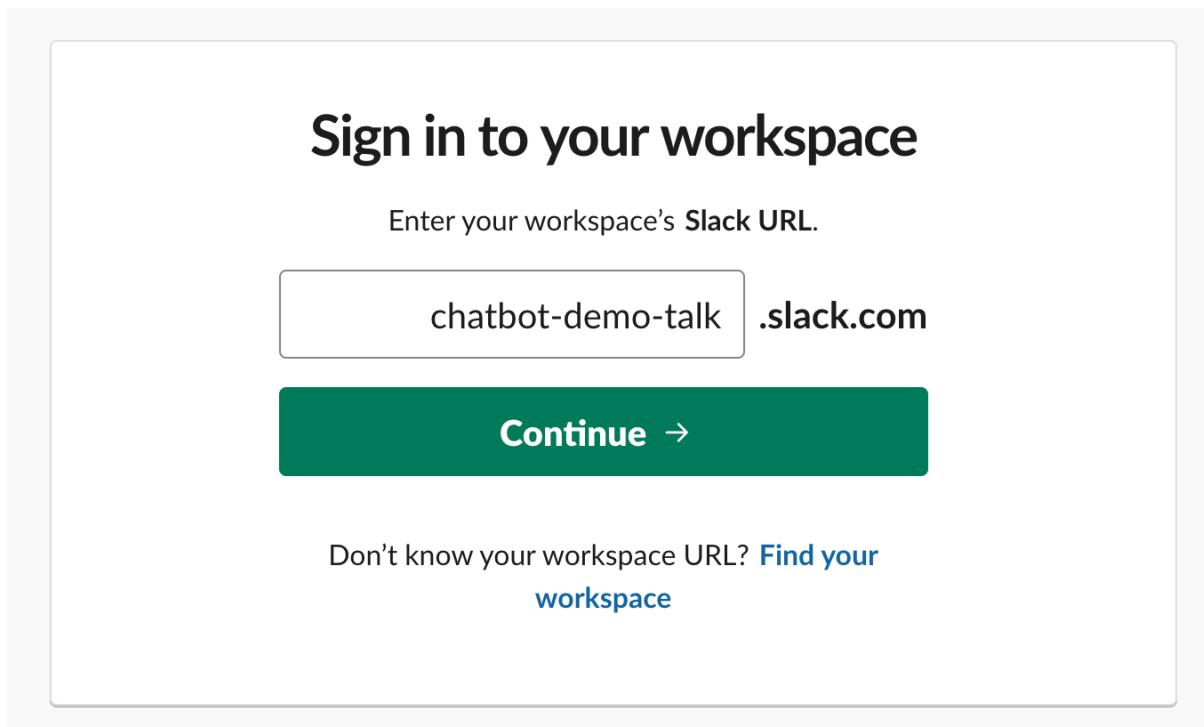
Welcome to Slack!

You've created the new Slack workspace **Chatbot Demo**. Here are your account details:

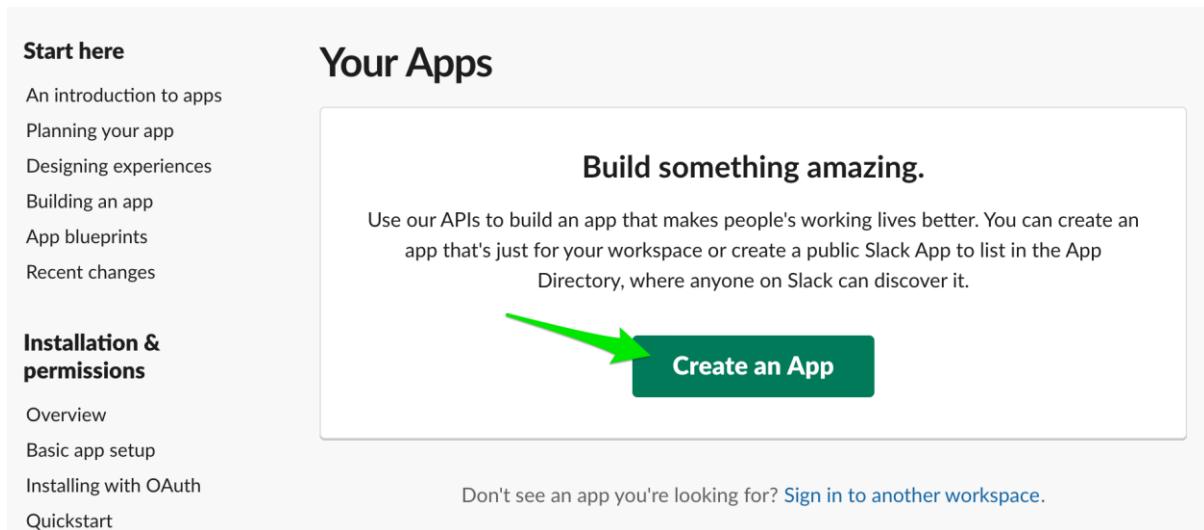

Chatbot Demo
URL: chatbot-demo-talk.slack.com
Email: acangiano@gmail.com

Your workspace is on the **free plan**, with unlimited messaging and the ability to search the 10,000 most recent messages.

-
5. Sign into your slack workspace with the following link: <https://slack.com/signin?redir=/api/apps>. **Enter your URL** (minus the .slack.com part) and **click Continue**.



6. Once you are signed in, **click on *Create an App***.



The screenshot shows the 'Your Apps' section of the Slack developer portal. On the left, there's a sidebar with 'Start here' and 'Installation & permissions' sections. The main area has a heading 'Build something amazing.' with a descriptive paragraph about using APIs to build apps. A prominent green 'Create an App' button is at the bottom right of this section. A green arrow points to this button. At the bottom of the main area, there's a note: 'Don't see an app you're looking for? [Sign in to another workspace](#)'.

7. Give your app a name and specify the workspace that you want to deploy this app on. Then **click on *Create App***.

Create a Slack App

X

App Name

Slack Chatbot Demo

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

Development Slack Workspace



Chatbot Demo



Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel

Create App

Add Permissions

Now that we have a Slack App (e.g., *Slack Chatbot Demo*), we need to ensure that we have the chatbot will have the right permission.

1. Click on **App Home** in the sidebar and then scroll to the Scopes section. Click on **Review Scopes to Add**.

Slack Chatbot ... ▾

Settings

- Basic Information
- Collaborators
- Install App
- Manage Distribution
- Submit to App Directory

Features

App Home

- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- Where's Bot User

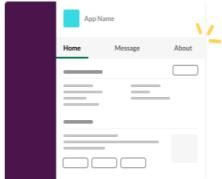
Slack ❤️
Help
Contact
Policies
Our Blog

App Home

Where people find your app on Slack

Your app's home includes three tabs (Home, Messages and About) that help people have richer interactions with your app and better understand its functionality.

[Learn more](#)



First, assign a scope to your bot token

A [bot token](#) makes it possible for users to interact with your app. A bot token lets users @mention it, and add it to channels and conversations. It also allows you to turn on tabs in your app's home.

Scopes govern an app's capabilities and permissions. You'll need to add at least one to your bot token to show the Home or Message tab in your app home. You can review and add the appropriate scopes under [Scopes](#) section in [OAuth & Permissions](#).

[Review Scopes to Add](#)

2. Scroll to scopes section. You'll want to **click on Add OAuth Scope button under Bot Token Scopes** as shown in figure.

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
You haven't added any OAuth Scopes for your Bot token.	

[Add an OAuth Scope](#)



User Token Scopes

Scopes that access user data and act on behalf of users that authorize them.

OAuth Scope	Description
You haven't added any OAuth Scopes for your User token.	

[Add an OAuth Scope](#)

Scopes define the [API methods](#) an app is allowed to call, which information and capabilities are available on the workspace it's installed on. Many scopes are restricted to specific [resources](#) like channels or files.

3. Select **app_mentions:read**. The new scope will be added as shown in the figure below. This permission enables the bot to detect mentions from Slack users.

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	
app_mentions:read	View messages that directly mention Slack Chatbot Demo in conversations that the app is in	

[Add an OAuth Scope](#)

4. Repeat the process of clicking on Add an OAuth Scope under Bot Token Scopes, to add the following permissions: **chat:write**, **im:history**, and **im:read**. Once you are done, the list will be shown as seen in the image below.

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	
app_mentions:read	View messages that directly mention Slack Chatbot Demo in conversations that the app is in	
chat:write	Send messages as Slack Chatbot Demo	
im:history	View messages and other content in direct messages that Slack Chatbot Demo has been added to	
im:read	View basic information about direct messages that Slack Chatbot Demo has been added to	

[Add an OAuth Scope](#)

5. Click on **Basic Information** in the left sidebar. Then copy and make note of your **Verification Token**. We'll need this information when integrating Slack within Watson Assistant.

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.

App ID

Date of App Creation

March 2, 2020

Client ID

Client Secret

••••••••••

[Show](#) [Regenerate](#)

You'll need to send this secret along with your client ID when making your [oauth.v2.access](#) request.

Signing Secret

••••••••••

[Show](#) [Regenerate](#)

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

[Regenerate](#)

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.

6. Now we need to install our Slack app within our workspace. **Click on *OAuth & Permissions*** in the left sidebar, then **click on *Install App to Workspace***.

The screenshot shows the 'OAuth & Permissions' section of the Slack developer console. On the left, there's a sidebar with 'Settings' and 'Features' sections. The 'OAuth & Permissions' option is highlighted with a blue background. The main area is titled 'OAuth Tokens & Redirect URLs'. It contains a paragraph about OAuth tokens and a large green button labeled 'Install App to Workspace'. A green arrow points to this button. Below it is a 'Redirect URLs' section with a note about configuring URLs for automatic addition to Slack.

OAuth & Permissions

OAuth Tokens & Redirect URLs

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

Install App to Workspace

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](#).

7. Click on **Allow** when asked to confirm your authorization.

This app was created by a member of your workspace, Chatbot Demo.

Slack Chatbot Demo is requesting permission to access the Chatbot Demo Slack workspace



What will Slack Chatbot Demo be able to view?

- Content and info about channels & conversations

What will Slack Chatbot Demo be able to do?

- Perform actions in channels & conversations

Cancel

Allow

8. Copy and make note of the **Bot User OAuth Access Token** that will be provided to you.

OAuth & Permissions

OAuth Tokens & Redirect URLs

Tokens for Your Workspace

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

Bot User OAuth Access Token

Copy

[Reinstall App](#)

Integrate Slack within Watson Assistant

Now that we have the *Verification Token*, and *Bot User OAuth Access Token* for our Slack app, we can proceed to add a Watson Assistant integration for Slack.

1. From the Student Advisor Chatbot assistant within Watson Assistant, **click on Add integration** - much like we did in the previous lab - but this time **select Slack**.
2. Scroll to Step 2 on the page, and **paste your Verification Token in the Verification token input field**.
3. **Paste your Bot User OAuth Access Token** (that you made note of, earlier) **in both the OAuth access token and Bot user OAuth access token input fields**.

Step 2

Connect Watson Assistant to Slack

On the Slack app settings page, go to the **Basic Information** tab and find the **App Cred**

Verification token

Enter Verification Token

Go to the **OAuth & Permissions** tab. Click **Install App to Workspace**, and then click **Authz**. Enter the OAuth access token and bot user OAuth access token to the fields below.

OAuth access token

Bot user OAuth access token

Enter Bot User OAuth Access Token for both fields

4. Scroll to Step 3 on the page, and **click on the *Generate request URL* button**.

Step 3

Configure your Slack bot

On the Slack app settings page, go to the **Event Subscriptions** tab.

 **Generate request URL**

5. This will generate a URL that we'll need to provide to Slack. (You may need to scroll back to Step 3 if the page refreshed.) **Copy and make note of the *Generated request URL*.**

Step 3

Configure your Slack bot

On the Slack app settings page, go to the **Event Subscriptions** tab. Switch the **Enable Events** toggle to the **On** position. In the Generated request URL

`https://integrations.us-south.assistant.watson.cloud.ibm.com/public/slack/messages/incoming`

6. Go back to the page for your Slack App. If you closed it, you can find it at the following URL <https://api.slack.com/apps>. In the left sidebar, under the **Features** section **click on Incoming Webhooks**. Turn **Activate Incoming Webhooks** on. In the warning message that appears, **click on reinstall your app**.

The screenshot shows the Slack App Settings page for a bot named "Slack Chatbot ...". The left sidebar has sections for "Settings" (Basic Information, Collaborators, Install App, Manage Distribution) and "Features" (App Home, **Incoming Webhooks**, Interactive Components). A green arrow labeled "2" points from the "Incoming Webhooks" link in the sidebar to the "Incoming Webhooks" section in the main content area. The main content area has a yellow header bar with a warning message: "⚠ You've changed the permission scopes your app uses. Please [reinstall your app](#) for these changes to take effect (and if your app is listed in the Slack App Directory, you'll need to resubmit it as well)." Below this is the "Activate Incoming Webhooks" section, which contains a toggle switch labeled "On". A green arrow labeled "1" points to this "On" button. The content of the "Activate Incoming Webhooks" section explains what incoming webhooks are and how they work.

⚠ You've changed the permission scopes your app uses. Please [reinstall your app](#) for these changes to take effect (and if your app is listed in the Slack App Directory, you'll need to resubmit it as well).

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.

7. You'll be asked to specify a channel (e.g., `#watson-assistant-chatbot-demo`). **Select the one that was generated for you earlier on when you specified the project name.**

Slack Chatbot Demo is requesting permission to access the Chatbot Demo Slack workspace



What will Slack Chatbot Demo be able to view?

Content and info about channels & conversations ▶

What will Slack Chatbot Demo be able to do?

Perform actions in channels & conversations ▶

Where should Slack Chatbot Demo post?

Slack Chatbot Demo requires a channel to post to as an app

watson-assistant-chatbot-demo ▾

Cancel

Allow



8. Now click on **Event Subscriptions** in the sidebar and switch on **Enable Events**. Then paste the request URL you saved from Watson Assistant.

9. Next, click on **Subscribe to bot events**, as shown by the arrow in the figure below.

Event Subscriptions

Enable Events

On

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

https://integrations.us-south.assistant.watson.cloud.ibm.com/public/slack/message/2

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 bot events

Subscribe to events on behalf of users

App unfurl domains

This will open the section that will allow you to subscribe to specific events.

10. Click on **Add Bot User Event** and select the event type **message.im**. Then repeat the process to add **app_mention**. This will enable both direct messages and mentions to trigger a response from the chatbot.

Subscribe to bot events

Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary [OAuth scope](#) for you.

Event Name	Description	Required Scope	
message.im	A message was posted in a direct message channel	im:history	
app_mention	Subscribe to only the message events that mention your app or bot	app_mentions:read	

[Add Bot User Event](#)

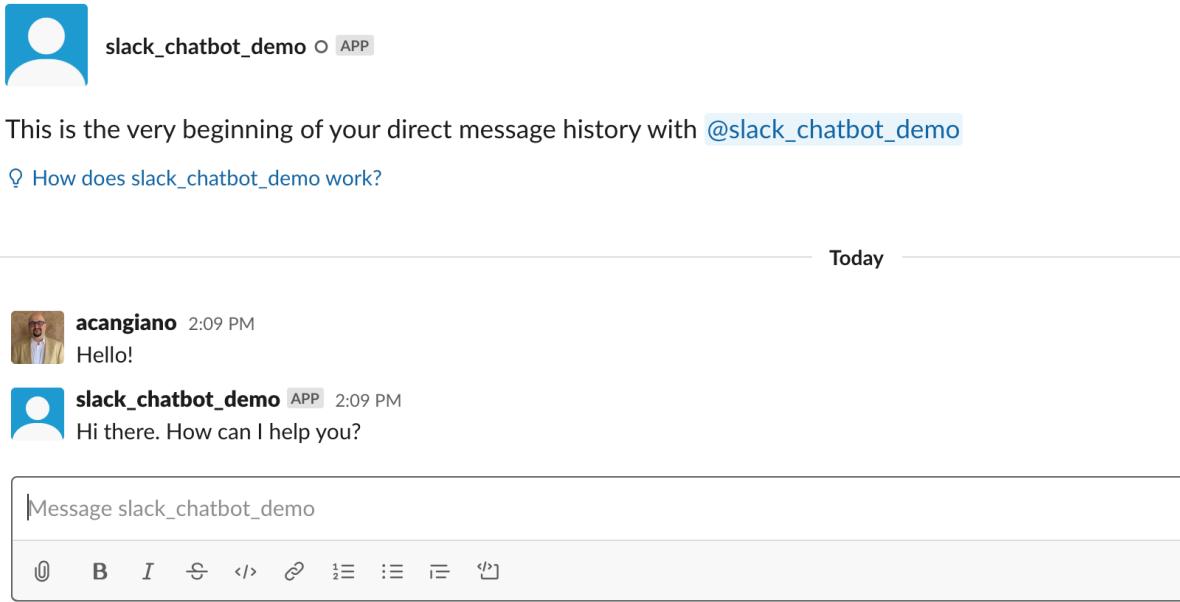
Then click on **Save Changes**.

(If the Save Changes is disabled/grey, please paste the Request URL again so that Slack can validate the URL and turn the Save Changes button to enabled/green.)

You're done! Your bot will now be available within your Slack workspace. Any direct message or mention will be sent in input to your Watson Assistant chatbot, and its response provided back to the user in Slack.

(Optional) Test your Slack App

If you are curious to see it in action, simply login into your Slack workspace, click on *Direct Messages*, and select your Slack app (e.g., *slack_chatbot_demo*). Try to say hi and you should see a response from the chatbot.



The chatbot will also respond to mentions in channels, provided you invite it to the given channel, after mentioning it as shown in the image below.