

Chatbot Deployment with IBM Cloud Watson Assistant-phase 5

Chatbot Deployment with IBM Cloud

Watson Assistant

OBJECTIVE:

The primary objective is to deploy an effective and efficient chatbot using IBM Cloud Watson Assistant to enhance customer support and engagement for our organization. In the era of digital transformation, businesses are increasingly turning to chatbots to enhance customer interactions and streamline operations. Deploying a successful chatbot solution begins with a well-defined problem statement and a human-centered design approach. This abstract delves into the critical aspects of problem definition and the application of design thinking principles in the context of deploying chatbots using **IBM Cloud Watson Assistant**.

1. Customer Support:

Provide efficient and accurate customer support by answering questions and resolving issues.

2. Information Retrieval:

Retrieve information from a knowledge base or database to assist users with their queries.

3. Task Automation:

Automate repetitive tasks or processes to improve efficiency.

4. Sales and Marketing:

Assist with sales and marketing efforts by engaging with potential customers and guiding them through product information or purchase processes.

5. Data Analysis:

Analyze and present data in a user-friendly manner.

6. Natural Language Understanding:

Improve user interactions by understanding and responding to natural language queries.

7. Multilingual Support:

Provide support in multiple languages to reach a diverse audience.

8. Integration with Other Systems:

Integrate with other software and systems for a seamless user experience.

9. Continuous Learning:

Continuously improve by learning from user interactions and feedback.

10. Cost Reduction:

Reduce operational costs by handling routine tasks without human intervention.

Problem Definition:

The project involves creating a chatbot using IBM Cloud Watson Assistant. The goal is to develop a virtual guide that assists users on messaging platforms like Facebook Messenger and Slack. The chatbot should provide helpful information, answer frequently asked questions (FAQs), and offer a friendly conversational experience. The project includes designing the chatbot's persona, configuring responses, integrating with messaging platforms, and ensuring a seamless user experience.

Design Thinking:

Design thinking is a user-centric approach that focuses on empathy, ideation, and prototyping to create solutions that truly meet user needs. Applying design thinking to chatbot deployment involves gaining deep insights into user behavior and preferences through research and user interviews. This understanding forms the foundation for ideating innovative chatbot functionalities.

Chatbot's Persona:

1. Define the Purpose:

Start by clarifying the primary purpose of your chatbot. What tasks will it perform, and who is the target audience?

2. Character and Name: Give your chatbot a name and define its character traits. Is it formal, friendly, or professional? Choose a persona that aligns with your brand and user expectations.

3. Voice and Tone: Decide on the chatbot's voice and tone. Will it use a casual, conversational tone or a more formal one? Ensure consistency in language throughout interactions.

4. User Interaction: Determine how the chatbot will interact with users. Will it be empathetic, humorous, or strictly informational? This should align with the user experience you want to deliver.

5. Design Visual Elements: If applicable, create a visual representation of your chatbot, such as an avatar or profile picture, to make it more relatable.

6. Script Conversations: Develop a script for various user interactions. Consider common user queries and how the chatbot should respond in a helpful and relevant manner.

7. Personalization: Incorporate personalization into the chatbot's responses by using user data and preferences when appropriate.

8. Handling Errors: Plan how the chatbot will handle errors or misunderstandings gracefully, guiding users back on track.

9. Multimodal Experience: If your chatbot operates on multiple platforms or devices, ensure consistency in its persona and behavior across all of them.

10. User Feedback: Continuously collect user feedback to improve the chatbot's persona and responses. Make adjustments based on user preferences and needs.

11. Ethical Considerations: Be aware of ethical guidelines and privacy concerns when designing the chatbot's persona, and ensure it respects user boundaries.

12. Training and Testing: Train your chatbot using real user data and conduct thorough testing to refine its persona and responses.

conversation flow

1. Welcome Message:

Start with a friendly greeting to welcome users.

Inform users about the chatbot's capabilities and how it can assist them.

2. User Intent Recognition:

Use natural language processing to understand the user's intent.

Identify keywords or phrases that indicate the user's purpose or question.

3. User Input Collection:

Prompt users to provide more details or clarify their request if necessary.

Collect any required information to fulfill the user's request.

4. Response Generation:

Generate a relevant and concise response based on the user's input and intent.

Use the chatbot's persona to craft responses that match its character and tone.

5. Follow-Up Questions:

Ask follow-up questions if more information is needed to complete the task.

Ensure that the conversation feels like a natural, back-and-forth dialogue.

6. Error Handling:

Plan for error and misunderstanding scenarios.

Provide friendly and helpful responses to guide users back on track.

7. User Confirmation and Feedback:

Confirm the action or information provided and ask if the user needs anything else.

Encourage users to provide feedback on the chatbot's performance.

8. Options and Navigation:

Offer users options for what they can do next.

Include menu items or commands for common tasks or inquiries.

9. Escalation to Human Agent:

If the chatbot can't resolve the user's issue, offer an escalation path to a human agent.

Provide a seamless transition from the chatbot to live support.

10. Closing the Conversation:

Thank the user for their interaction.

Let users know they can return if they have more questions in the future.

11. Exit Options:

Offer options for users to exit the conversation or provide additional resources.

12. Fallback Mechanism:

Implement a fallback mechanism to handle unexpected or out-of-scope queries.

Inform users when the chatbot doesn't understand the query and offer alternative actions.

13. Continuity and Context:

Maintain context between interactions, so users don't have to repeat information.

Remember prior user inputs and previous chatbot responses.

14. User Privacy and Security:

Ensure that sensitive user data is handled securely and that privacy concerns are addressed appropriately.

15. Testing and Iteration:

Thoroughly test the conversation flow with real users and refine it based on feedback.

Continuously improve the chatbot's responses and user experience.

Configuring responses

1. Response Types:

Determine the types of responses your chatbot can provide, such as text, images, links, or even actions (e.g., making reservations or purchases).

2. Response Generation:

Decide how responses are generated. This can involve using predefined answers, retrieving information from databases, or generating dynamic responses based on user input.

3. Dialog Flow:

Design a structured dialog flow that guides the chatbot's responses in various scenarios.

Use conditional logic to handle different user intents and questions effectively.

4. Natural Language Generation (NLG):

Implement NLG techniques to ensure that responses sound natural and are contextually relevant.

Adapt the chatbot's language to match the user's tone and style.

5. Personalization:

Incorporate personalization by using user data to tailor responses to individual preferences and history.

Address users by their names and remember their previous interactions.

6. Fallback Mechanism:

Set up a fallback mechanism for handling queries the chatbot doesn't understand.

Provide alternative suggestions or options to guide users back on track.

7. Error Handling:

Plan for error scenarios where the chatbot encounters issues or misunderstands the user.

Offer empathetic and helpful responses to rectify the situation.

8. Multimodal Responses:

If the chatbot supports multiple formats (text, images, etc.), configure it to provide the most appropriate response type for each scenario.

9. Integration with APIs:

If the chatbot needs to access external data sources or services, configure it to interact with APIs and fetch real-time information.

10. User Confirmation and Feedback:

Include mechanisms for users to confirm that the chatbot's response meets their needs.

Encourage users to provide feedback on the quality of the responses.

11. Rich Media and Cards:

If applicable, configure the chatbot to present information in a visually appealing way, such as using cards, carousels, or rich media.

12. Context Management:

Ensure that the chatbot maintains context within the conversation, so users don't have to repeat information.

13. Response Timing:

Set response times to be appropriate for the medium (e.g., faster for chat, slower for email).

Implement response delays to create a more natural conversational experience.

14. User Instructions:

Include instructions for users on how to interact with the chatbot effectively.

Make sure users understand what the chatbot can and cannot do.

15. A/B Testing:

Conduct A/B testing to compare different response configurations and choose the most effective ones.

16. Continuous Improvement:

Regularly analyze user interactions, gather feedback, and make iterative improvements to response generation and configuration.

Integration

For Facebook :

Facebook developers platform:

- **Go to the Facebook for Developers platform (<https://developers.facebook.com/>).**
- **Create a new Facebook App or use an existing one.**

Set Up Messenger Integration:

- **Inside your Facebook App, navigate to the Messenger tab and set up the Messenger integration.**
- **Generate a Page Access Token, which is needed to interact with Facebook Messenger.**

Configure Webhooks:

- **Set up a webhook to receive incoming messages from Facebook Messenger. This webhook should point to your server or cloud service.**

Develop a Backend Server:

- **Create a backend server that communicates with Watson Assistant. This server will receive messages from Facebook, send them to Watson Assistant, and relay the responses back to Facebook.**

Integrate with Watson Assistant:

- **Use the Watson Assistant API to interact with your chatbot. Send user messages to Watson Assistant and receive responses.**

Handle User Authentication (if necessary):

- **Depending on your use case, you might need to handle user authentication within your chatbot.**

Test and Deploy:

- **Test your integration thoroughly to ensure it works as expected.**
- **Deploy your integration to a production environment.**

Connect to Facebook App:

- **In the Facebook App settings, link your backend server (webhook) to the Messenger integration.**

Submit for Review:

- **If you plan to make your chatbot available to the public, you'll need to submit it to Facebook for review and approval.**

For Slack :

Create a Slack App:

- **Go to the Slack API website (<https://api.slack.com/>).**
- **Create a new Slack App or use an existing one.**

Configure Bot User:

- **Inside your Slack App, configure a bot user. This bot user will represent your Watson Assistant chatbot.**

Install the App to Workspaces:

- **Enable your Slack App to be installed by Slack users in their workspaces.**

Implement OAuth 2.0:

- **Set up OAuth 2.0 authentication to allow your app to access Slack workspaces.**

Integrate with Watson Assistant:

- **Use the Watson Assistant API to send and receive messages from Slack.**

Develop a Backend Server:

- **Create a backend server that listens for events and messages from Slack and communicates with Watson Assistant.**

Test and Deploy:

- **Test your integration and ensure it handles various user interactions.**
- **Deploy your integration to a production environment.**

Connect to Slack App:

- **In the Slack App settings, configure event subscriptions and interactive components to handle incoming messages.**

Submit for Distribution:

- **If you want to make your chatbot available to other Slack workspaces, submit your app for distribution through the Slack App Directory.**

DEVELOPMENT

IBM Cloud Account:

Sign up for an IBM Cloud account if you haven't already.

Set Up Watson assistant:

Create an instance of the IBM Cloud Watson assistant service in your IBM Cloud account.

1. The Chatbot Persona:

Determine the chatbot's personality and tone (e.g., formal, casual, friendly). Understand the target audience and their expectations. Start by defining your chatbot's persona. Is it a helpful assistant, a virtual customer support agent, or something else? Choose a name and character for your chatbot to make it more relatable.

2. The Purpose:

Determine the primary goals and objectives of the chatbot.

Decide what kind of queries or tasks the chatbot should handle.

3. Conversation Flow:

Identify the primary use cases or scenarios your chatbot will handle. For instance, if it's a customer support bot, consider scenarios like FAQs, troubleshooting, or product information.

Design a conversation flow for each scenario. Create a flowchart or diagram to outline how the conversation will progress, including user inputs and bot responses.

4. Configure Intents:

Create a list of user intents (queries or requests) that the chatbot should recognize.

For each intent, provide examples of user input to train the chatbot.

Assign descriptive names to the intents (e.g., "OrderPizza," "CheckWeather").

5. Define Entities:

Identify specific entities (variables or data) that the chatbot needs to extract from user input.

Define synonyms for entities to capture variations in user input.

Assign types to entities (e.g., "Location," "Product").

6. Design Conversation Flow:

Create a flowchart or diagram outlining the expected conversation path.

Define the welcome message and initial user prompt.

Determine how the chatbot responds to each intent and entity.

Plan for error handling and fallback responses.

7. Configure Dialog Nodes in Watson Assistant:

Map intents to dialog nodes. Each node should handle a specific intent.

Set conditions for when a node should be triggered based on user input.

Define responses for each node, including dynamic responses that use extracted entities.

Implement branching and context management for multi-turn conversations.

8. Test and Refine:

Test the chatbot with a variety of sample user inputs.

Make adjustments to intents, entities, and dialog nodes based on test results.

Continuously refine the chatbot's responses and performance.

9. Integration:

Integrate your chatbot with the platform or channels where it will be used, like a website, mobile app,facebook,slack or messaging service.

The screenshot shows the IBM Watson Assistant Lite interface. At the top, there are navigation links for 'IBM Watsonx Assistant Lite' (with 'Upgrade' and 'Jarvis' dropdowns), 'Learning resources', and a user icon. On the left, a sidebar titled 'Welcome to watsonx Assistant' contains sections for personalizing the assistant, choosing a channel (Facebook), telling about oneself, industry, role, and needs. On the right, a simulated conversation is displayed between a user ('Running World') and a bot ('Jarvis'). The conversation starts with a question from the user about Speed Demons availability. The bot responds with information about stock locations and asks for size and color preferences. The user then asks for a size nine in white, and the bot confirms availability and offers shipping options. The interface includes standard messaging controls at the bottom.

The screenshot shows the IBM Watson Assistant Lite interface. At the top, there are links for 'IBM Watson Assistant Lite' and 'Upgrade'. On the right, there are 'Learning resources' and a user icon. Below the header, the text 'Welcome to watsonx Assistant' is displayed, along with a 'Next' button. A horizontal navigation bar at the bottom includes 'Create' (selected), 'Personalize', 'Customize', and 'Preview'. The main content area is titled 'Create your first assistant' with a sub-instruction: 'Let's get your assistant up and running. Name your assistant, add a description, and choose a language. In following steps we'll gather more information, show you basic customizations, and give you a preview of what your assistant will look like.' There are three input fields: 'Assistant name' (Example: Banking Bot), 'Description (optional)' (Add a description for this assistant), and 'Assistant language' (English (US)).

Log in to IBM Cloud:

Go to the IBM Cloud website and log in to your account.

Navigate to Resource Groups:

Once logged in, click on the "Resources" section in the IBM Cloud Console.

Create a New Resource Group:

In the Resource Groups section, you should find an option to create a new resource group. Click on it.

Actions:

Actions are used to define custom logic or to trigger specific behaviors in response to user input. They can involve setting context variables, generating responses, or making external API calls.

Actions are custom code or logic that the chatbot can execute in response to user input. This might involve making API calls, modifying context variables, or other custom behavior.

Preview:

1. Testing Your Chatbot Logic:

When you're building or updating a chatbot in Watson Assistant, you can enter preview mode to test how the bot responds to different user inputs. This is especially useful for checking the dialog flow, intent recognition, and entity extraction.

2. Simulated Conversations:

In preview mode, you can simulate conversations by typing messages as if you were a user interacting with the chatbot. The chatbot responds based on its dialog nodes, intents, entities, and actions that you've defined.

3. Debugging and Validation:

You can use preview mode to identify and fix issues in your chatbot's responses, such as incorrect answers or unexpected behaviors. It's a valuable tool for debugging and fine-tuning your chatbot.

4. No Impact on Live Environment:

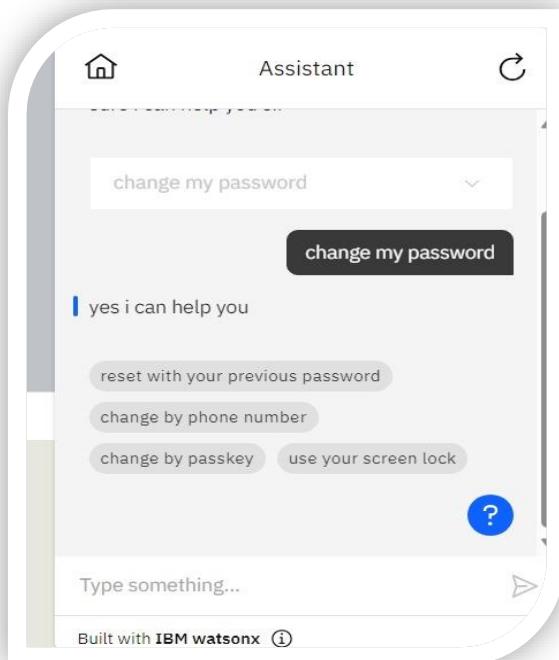
The important thing to note is that when you're in preview mode, the changes you make or the conversations you simulate do not affect the live deployment of your chatbot. It's a safe sandbox for testing.

5. Context Awareness:

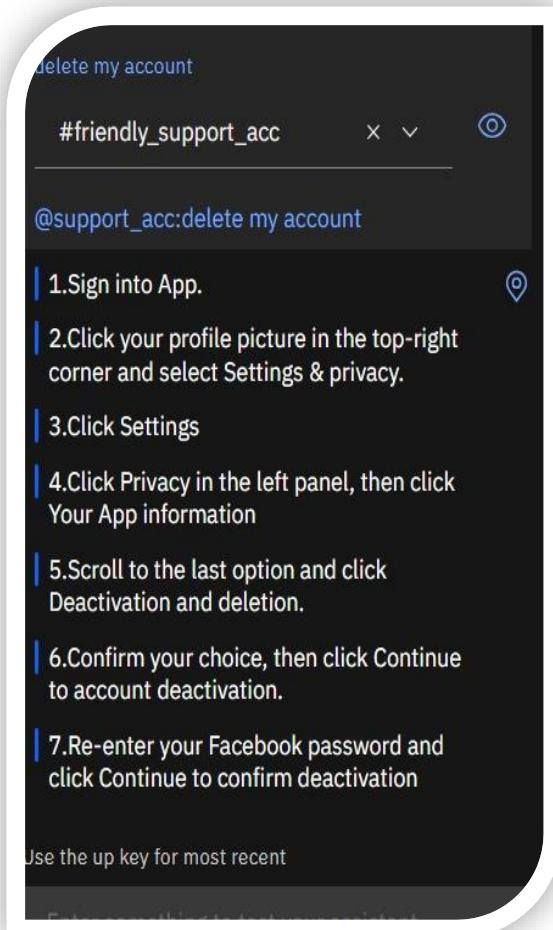
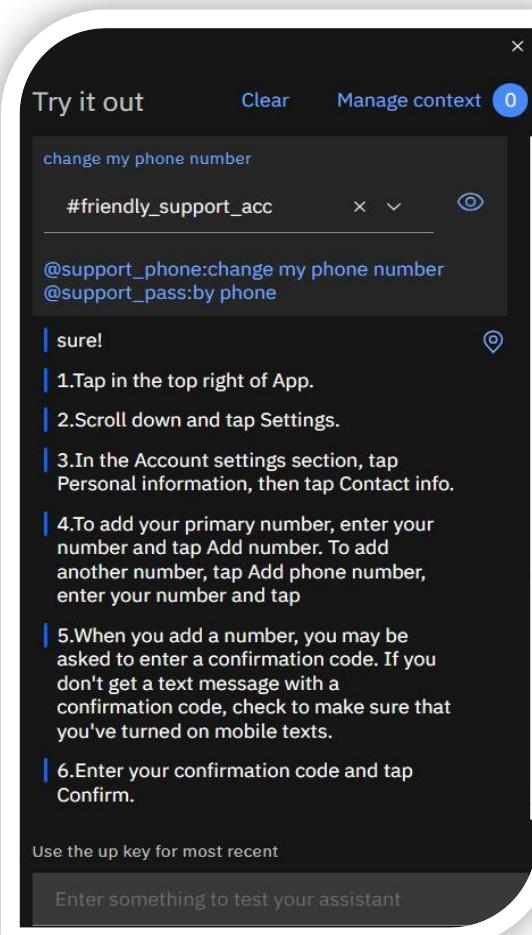
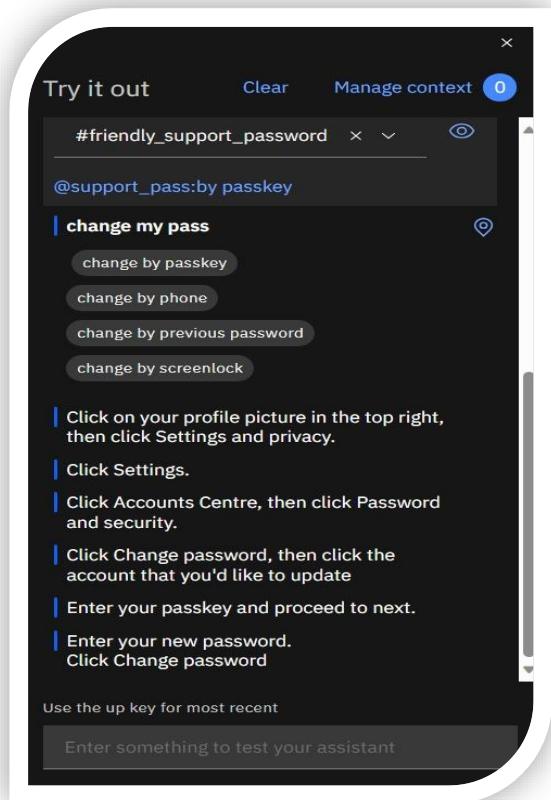
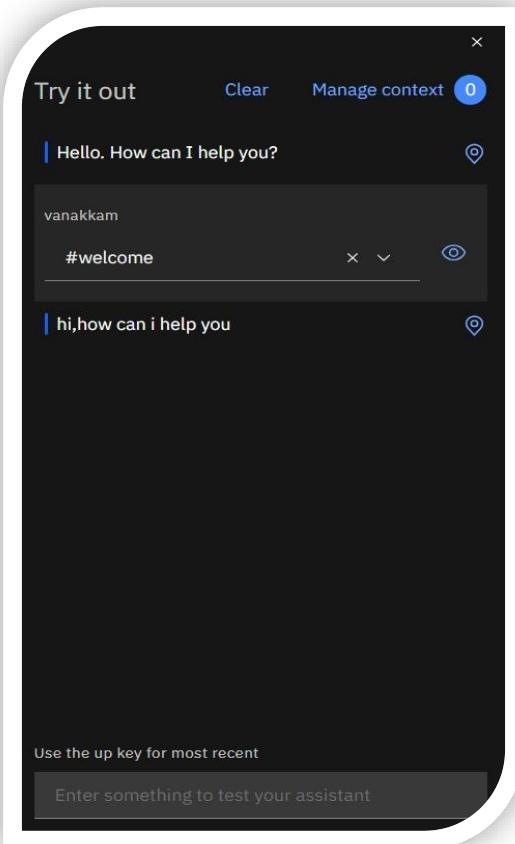
Preview mode retains context from one user message to the next, allowing you to test multi-turn conversations and ensure that context variables and actions behave as intended.

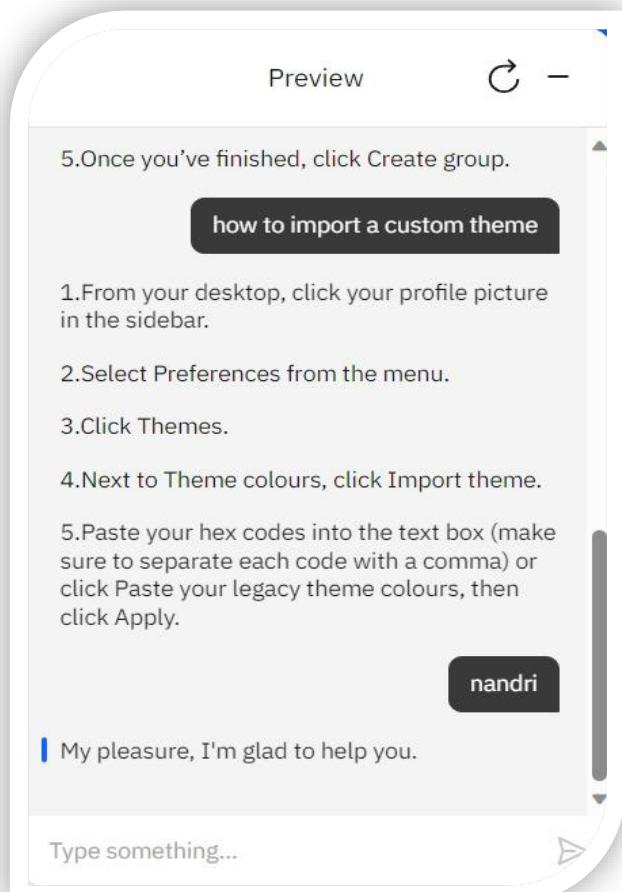
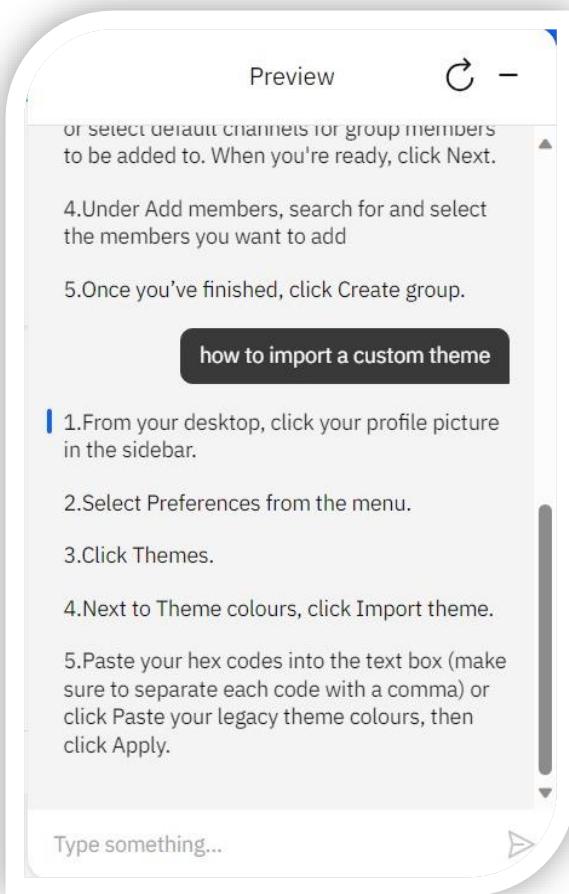
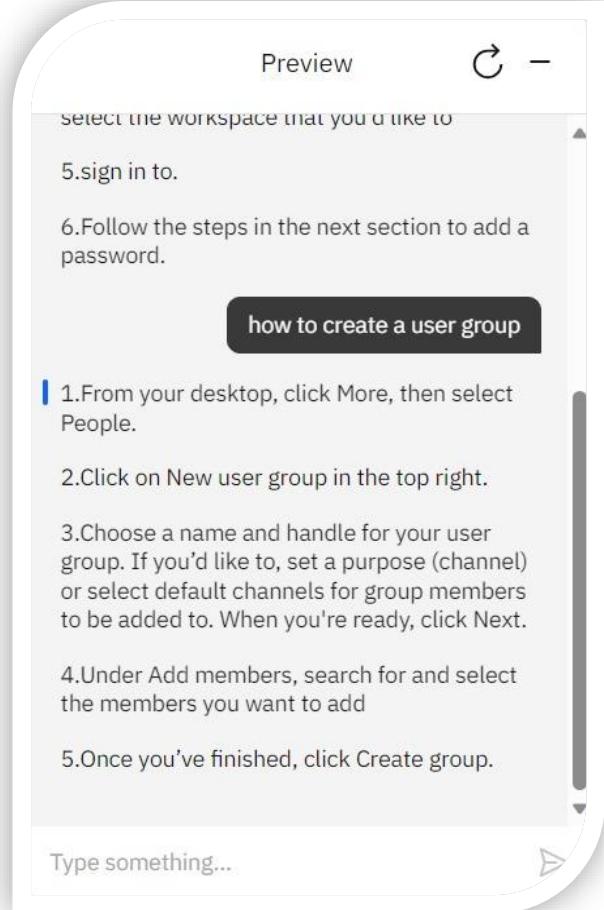
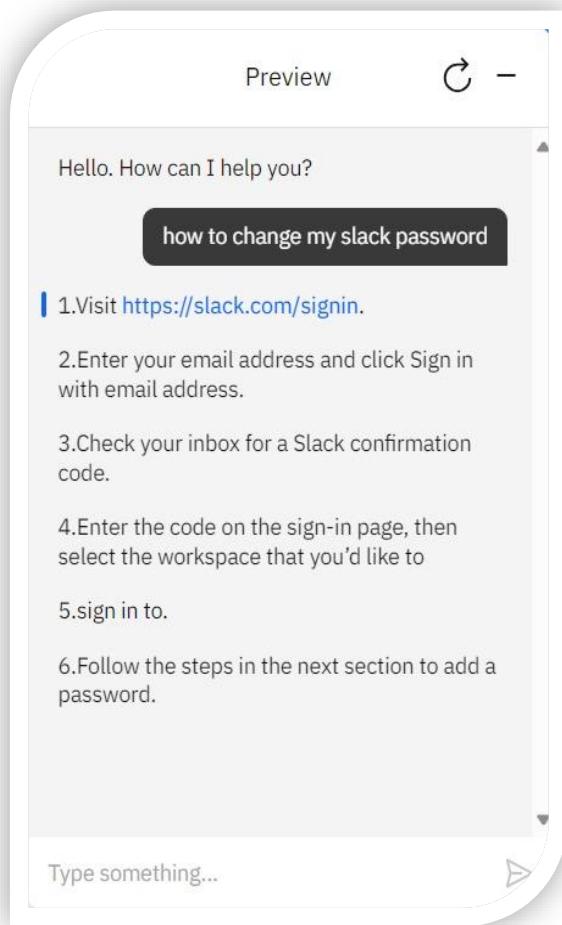
6. Preview Mode Limitations:

While preview mode is helpful for initial testing and validation, it may not fully replicate the exact user experience, especially if your chatbot relies on external integrations or real-time data.

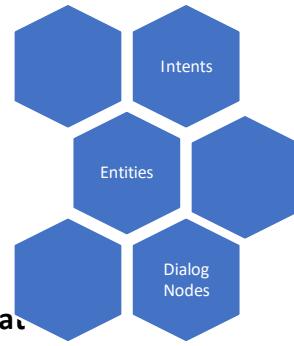


Note: Please zoom to get a clear view of images





INTENTS



1. Recognition of User Goals:

Intents are used to recognize the primary objective or goal that a user is trying to achieve through their message.

For example, if a user types, "How to change my password," the intent might be recognized as "Change password."

2. Intent Training:

To teach the chatbot to recognize intents, you provide training data that includes examples of user messages mapped to specific intents. These examples help the chatbot learn the different ways users might express the same intent.

3. Intent Matching:

When a user sends a message to the chatbot, the platform uses natural language processing to analyze the message and determine which intent it best matches. This involves considering the context of the conversation and the words and phrases used.

4. Intent Confidence:

Watson Assistant provides a confidence score associated with each recognized intent. This score indicates the platform's level of confidence in the accuracy of the intent recognition. You can use this confidence score to determine how the chatbot should respond.

5. Response Routing:

Once the intent is identified, the chatbot can route the user to the appropriate dialog node or set of responses that are designed to handle that specific intent. This is how the chatbot provides contextually relevant answers.

6. Multi-Intent Recognition:

In some conversations, users may express multiple intents within a single message. Watson Assistant can recognize multiple intents within a single message and route the conversation accordingly.

7. Customization:

You can create custom intents based on the specific goals or tasks your chatbot is designed to handle. For instance, if you're building a customer support bot, you might define intents like "login issues," "Account settings," etc...

IBM Watson Assistant Lite Upgrade Jarvis Learning resources

Dialog

Intents

Entities My Entities System Entities Dialog Options Webhooks Disambiguation Autocorrection Algorithm Version Upload / Download Content Catalog

Create intent +

Intents (6) ↑	Description	Modified ↑↓	Examples ↑↓
#friendly_support_acc	support_acc	a day ago	22
#friendly_support_password	support_password	14 days ago	8
#profile_support	profile	a day ago	8
#slack	slack_support	a day ago	12
#thankyou	end of conversation	2 days ago	11
#welcome	start of conversation	5 days ago	10

Showing 1–6 of 6 intents

1 1 of 1 pages

IBM Watson Assistant Lite Upgrade Jarvis Learning resources

#friendly_support_acc

Last updated: a day ago

friendly_support_acc

Name your intent to match a customer's question or goal

Description (optional)

support_acc

User example

Type a user example here

Add unique examples of what the user might say. (Pro tip: Add at least 5 unique examples to help Watson understand)

Add example

Annotations

Annotate entities What's this?

User examples (22) ↑ Added ↑↓

change my emailaddress	14 days ago
change my phone number	14 days ago
deactivate my account	14 days ago
delete my account	14 days ago

ENTITIES

1. Recognition of Structured Information:

Entities are used to identify and extract structured data from user messages. This structured data can be specific details such as dates, times, numbers, locations, product names, or any other relevant information.

2. Entity Types:

Watson Assistant supports various entity types, including system entities and user-defined entities. System entities are predefined for common concepts like numbers, dates, and times, while user-defined entities are created to match domain-specific data.

3. Training Data:

To teach the chatbot to recognize entities, you provide training data that includes examples of user messages containing the entities you want to extract. For instance, if you're building a chatbot for ordering pizza, you might create an entity called "PizzaTopping" and provide training examples like "I want a pizza with pepperoni."

4. Entity Values:

Within an entity type, you define entity values. For example, if you have an entity type "PizzaTopping," the entity values could be "Pepperoni," "Mushroom," and "Onion." These values represent the specific options that the chatbot should recognize within the entity.

5. Entity Matching:

When a user sends a message to the chatbot, the platform uses natural language processing to analyze the message and identify which entity values are present. For instance, if a user says, "I'd like a pizza with pepperoni and mushrooms," the chatbot might recognize the "PizzaTopping" entity with the values "Pepperoni" and "Mushroom."

6. Contextual Data:

Extracted entities are used to provide context and assist the chatbot in generating relevant responses. In the pizza example, the chatbot can use the identified pizza toppings to accurately fulfill the user's order.

7. Multiple Entities:

In a single message, users may mention multiple entities. Watson Assistant is designed to identify and extract multiple entities to enhance its understanding of user input.

IBM Watson Assistant Lite Upgrade Jarvis Learning resources

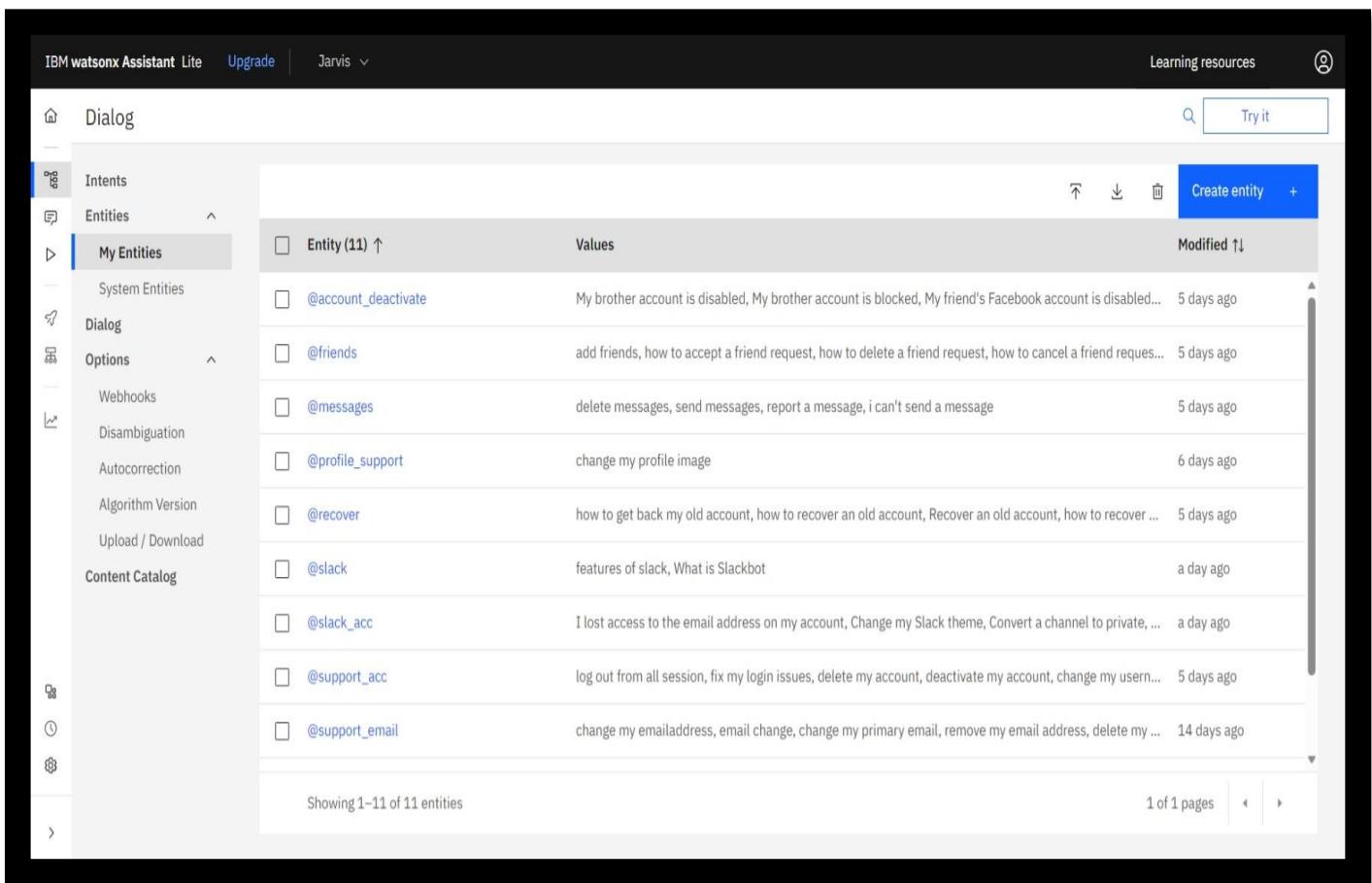
Dialog

Intents Entities My Entities System Entities Dialog Options Webhooks Disambiguation Autocorrection Algorithm Version Upload / Download Content Catalog

Entity (11) ↑ Values Modified ↑

Entity	Values	Last modified
@account_deactivate	My brother account is disabled, My brother account is blocked, My friend's Facebook account is disabled...	5 days ago
@friends	add friends, how to accept a friend request, how to delete a friend request, how to cancel a friend request...	5 days ago
@messages	delete messages, send messages, report a message, i can't send a message	5 days ago
@profile_support	change my profile image	6 days ago
@recover	how to get back my old account, how to recover an old account, Recover an old account, how to recover ...	5 days ago
@slack	features of slack, What is Slackbot	a day ago
@slack_acc	I lost access to the email address on my account, Change my Slack theme, Convert a channel to private, ...	a day ago
@support_acc	log out from all session, fix my login issues, delete my account, deactivate my account, change my user...	5 days ago
@support_email	change my emailaddress, email change, change my primary email, remove my email address, delete my ...	14 days ago

Showing 1-11 of 11 entities 1 of 1 pages



IBM Watson Assistant Lite Upgrade Jarvis Learning resources

Entity name @slack_acc Last updated: a day ago Try it

Name your entity to match the category of values that it will detect.

Value Synonyms Fuzzy matching On

Type a value Synonyms Type a synonym +

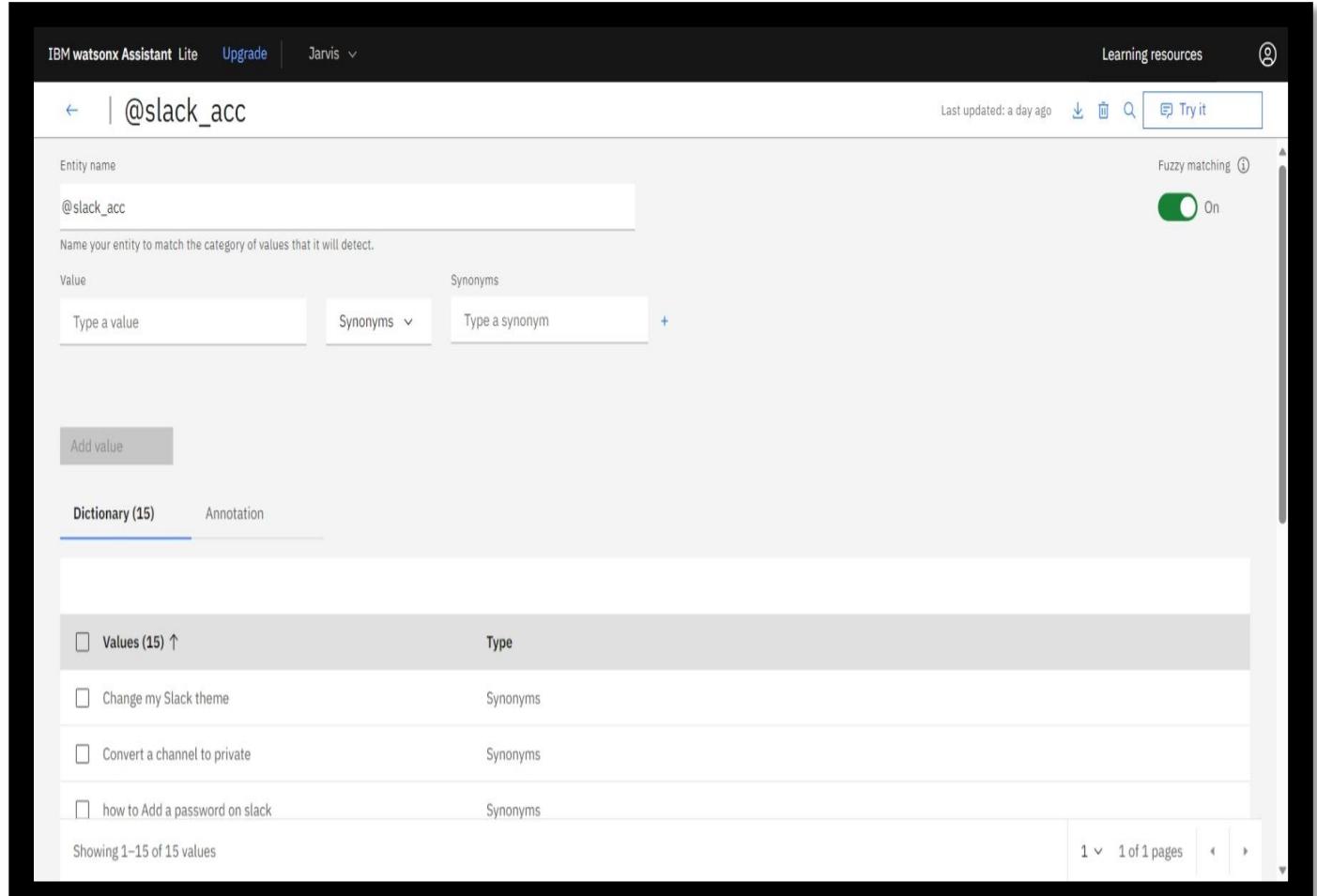
Add value

Dictionary (15) Annotation

Values (15) ↑ Type

Value	Type
Change my Slack theme	Synonyms
Convert a channel to private	Synonyms
how to Add a password on slack	Synonyms

Showing 1-15 of 15 values 1 of 1 pages



DIALOGUE

1. Purpose:

Dialogue nodes are used to define the logic and responses of your chatbot. Each node represents a specific point in the conversation where the chatbot is expected to perform an action, respond to user input, or make decisions.

2. Conditions:

Dialogue nodes are often associated with conditions that determine when they should be triggered. Conditions can be based on intents (the user's goal), entities (specific pieces of information), context variables, or other criteria.

3. Responses:

Within a dialogue node, you can specify one or more responses that the chatbot should provide when the node is triggered. Responses can include text messages, images, suggestions, or even actions to execute.

4. Contextual Information:

Dialogue nodes can set, modify, or reference context variables. These variables store information about the conversation, allowing the chatbot to remember user-specific details and maintain context throughout the interaction.

5. Branching Logic:

Dialogue nodes allow you to define branching logic, which determines how the conversation flows based on user input. You can direct the conversation to different nodes depending on the user's intent, entity, or context.

6. Conditions and Slots:

You can set up conditions and slots in dialogue nodes to gather specific information from the user in a structured way. For instance, if the user wants to order a pizza, you can use slots to collect information like pizza size, toppings, and delivery address.

7. Custom Actions:

Dialogue nodes can contain custom actions, which allow you to execute code, make API calls, and perform custom logic in response to user input. This is useful for complex tasks and integrations.

8. Error Handling:

You can use dialogue nodes to handle unexpected user input or errors gracefully. You can create nodes to provide informative error messages and guide the conversation back on track.

9. Visual Interface: Many chatbot development platforms, including IBM Watson Assistant, offer a visual interface for creating and managing dialogue nodes. This makes it easy to design and structure the conversation flow.

Add node

Add child node

Add folder

Welcome

welcome

1 Responses / 0 Context Set / Does not return

Greetings

#welcome

1 Responses / 0 Context Set / Does not return

support

#friendly_support_password

1 Responses / 0 Context Set / Jump to / Does not return

support email

@support_email

0 Responses / 0 Context Set / Jump to / Does not return

support phone

@support_phone

1 Responses / 0 Context Set / Jump to / Does not return

supportprofile

@profile_support

0 Responses / 0 Context Set / Jump to / Does not return

support account

@support_acc

6 Responses / 0 Context Set / Does not return

Add node

Add child node

Add folder

reactivate

@account_deactivate

1 Responses / 0 Context Set / Does not return

recover

@recover

1 Responses / 0 Context Set / Does not return

friends

@friends

0 Responses / 0 Context Set / Does not return

messages

#friendly_support_acc

0 Responses / 0 Context Set / Jump to / Does not return

slack account

@slack_acc

0 Responses / 0 Context Set / Jump to / Does not return

Thankyou

#thankyou

1 Responses / 0 Context Set / Does not return

Anything else

anything_else

1 Responses / 0 Context Set / Does not return

IBM Watson Assistant Lite Upgrade Jarvis Learning resources Try it

Dialog

Add node Add child node Add folder

- Intents
- Entities My Entities System Entities
- Dialog**
- Options Webhooks Disambiguation Autocorrection Algorithm Version Upload / Download Content Catalog

messages #friendly_support_acc
0 Responses / 0 Context Set / Jump to / Does not return

[Jump to messages \(Evaluate condition\)](#)

messages @messages
4 Responses / 0 Context Set / Return allowed

slack account @slack_acc
0 Responses / 0 Context Set / Jump to / Does not return

[Jump to slack_acc \(Evaluate condition\)](#)

slack_acc @slack_acc
15 Responses / 0 Context Set / Return allowed

IBM Watson Assistant Lite Upgrade Jarvis Learning resources Try it

Dialog

Add node Add child node Add folder

- Intents
- Entities My Entities System Entities
- Dialog**
- Options Webhooks Disambiguation Autocorrection Algorithm Version Upload / Download Content Catalog

slack_acc

Customize

Text	Text	Text	Text	Text	Text
1	@slack_acc:(how to reset my sl	Text, Text, Text, Text, Text, Text			
2	@slack_acc:(how to sign in to sl	Text, Text, Text			
3	@slack_acc:(how to Add a pass	Text, Text, Text, Text, Text, Text			
4	@slack_acc:(I didn't receive a p	Check your spam or junk folder. If			
5	@slack_acc:(I lost access to the	For security purposes, you must b			

4 Responses / 0 Context Set / Return allowed

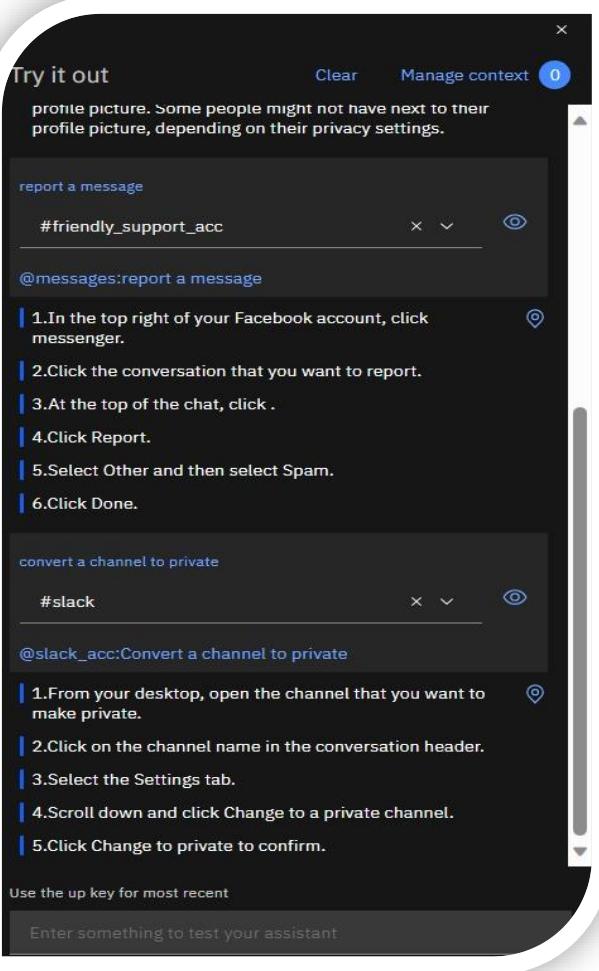
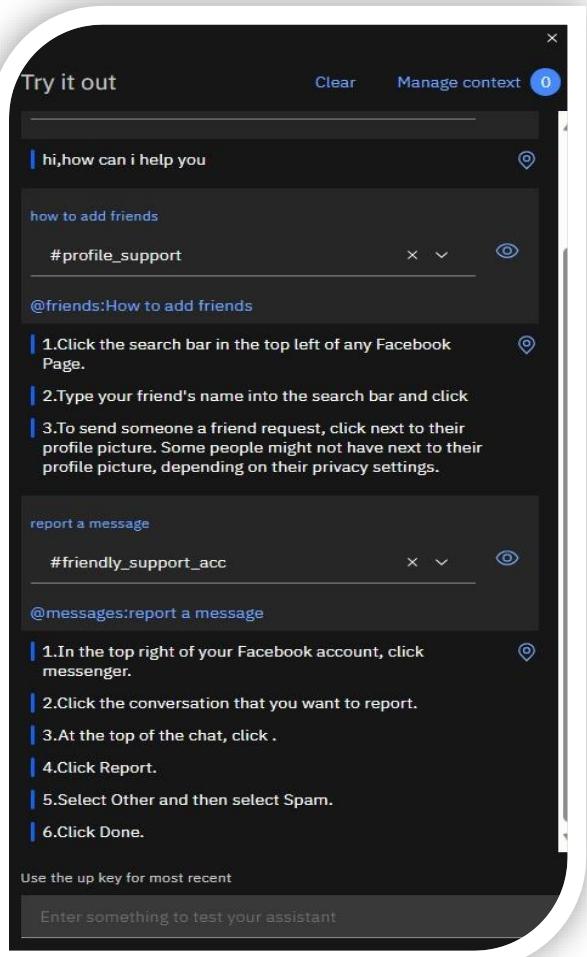
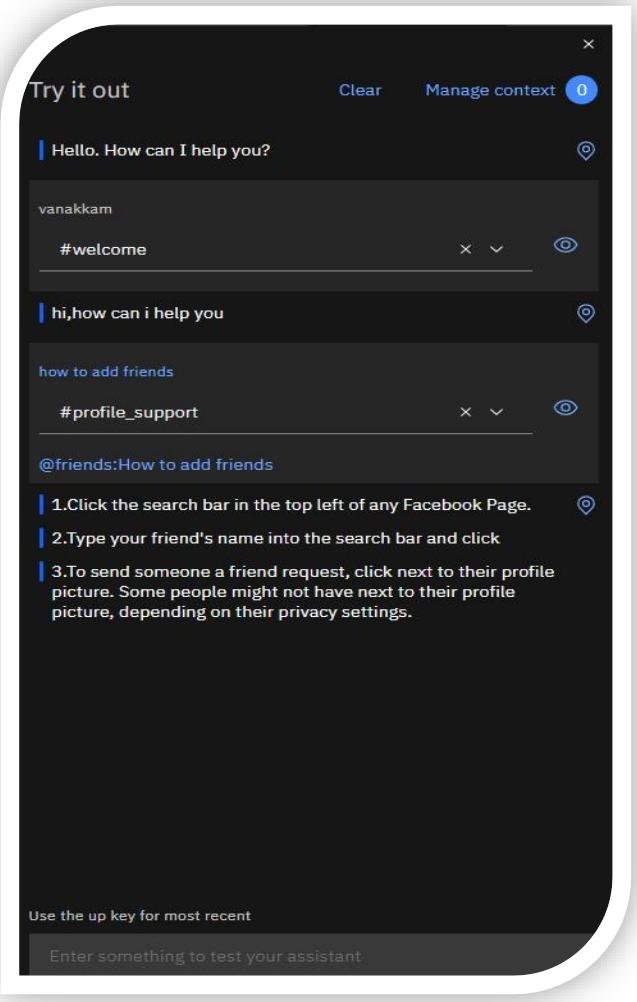
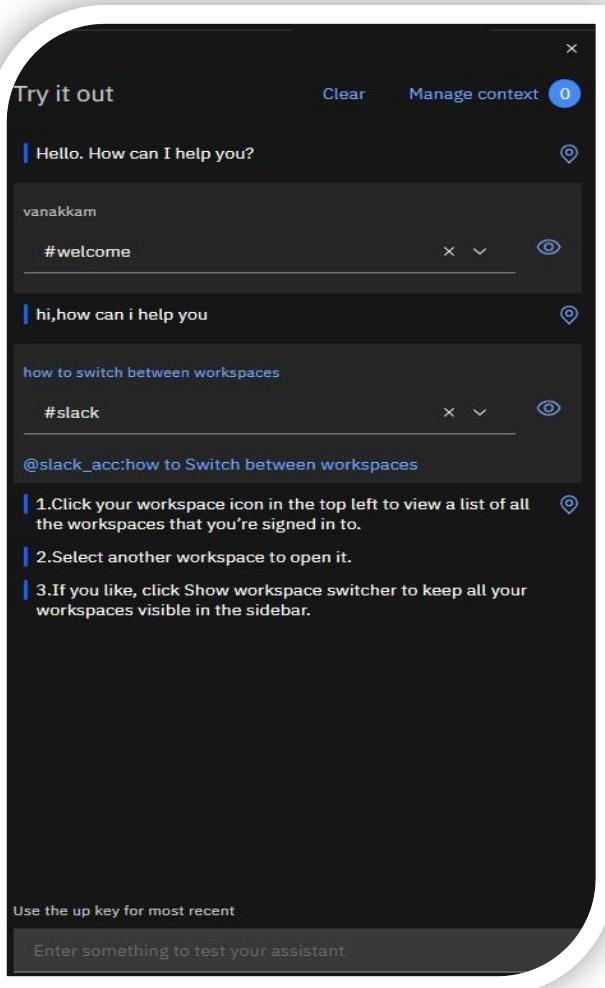
slack account @slack_acc
0 Responses / 0 Context Set / Jump to / Does not return

[Jump to slack_acc \(Evaluate condition\)](#)

slack_acc @slack_acc
15 Responses / 0 Context Set / Return allowed

Thankyou #thankyou
1 Responses / 0 Context Set / Does not return

Anything else anything_else
1 Responses / 0 Context Set / Does not return



META FOR DEVELOPERS

Meta for Developers, also known as the Meta Platforms Developer Platform, encompasses the set of tools and resources provided by Meta Platforms, Inc. (formerly Facebook) to enable developers to create applications and experiences that integrate with Meta's various products and platforms. These platforms include Facebook, Instagram, WhatsApp, Oculus, and other Meta-owned services.

1. APIs:

Meta provides application programming interfaces (APIs) that allow developers to access and interact with data and features on Meta's platforms. For example, the Facebook Graph API allows developers to work with Facebook data.

2. SDKs:

Software Development Kits (SDKs) are sets of tools and libraries that make it easier for developers to build applications for specific platforms. Meta offers SDKs for various platforms like Oculus, allowing developers to create virtual reality experiences.

3. Documentation:

Detailed documentation and resources are available to help developers understand and use Meta's tools effectively. This documentation provides guidelines, best practices, and code samples.

4. Developer Tools:

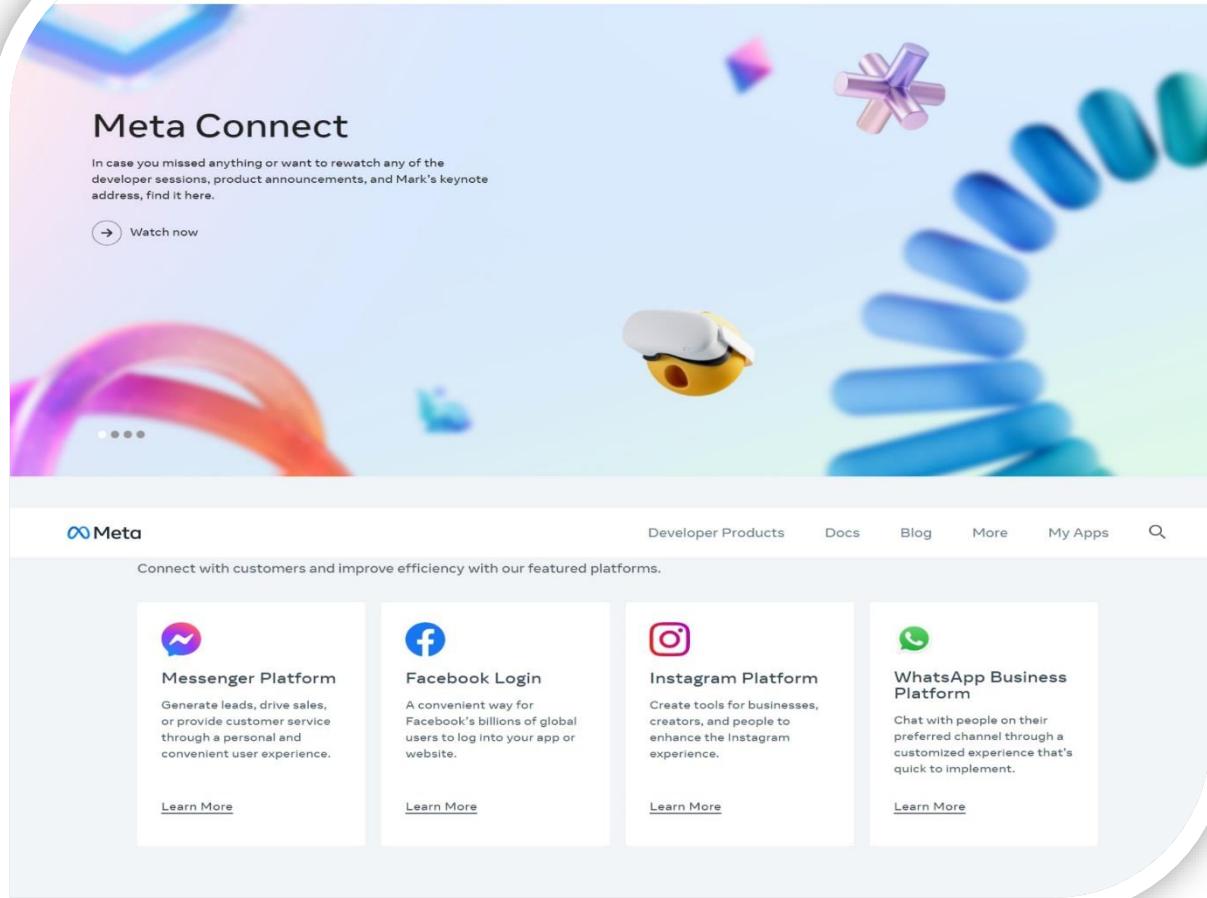
Meta may offer additional developer tools and resources, such as analytics, debugging tools, and testing environments to aid in the development and testing of applications.

5. Community Support:

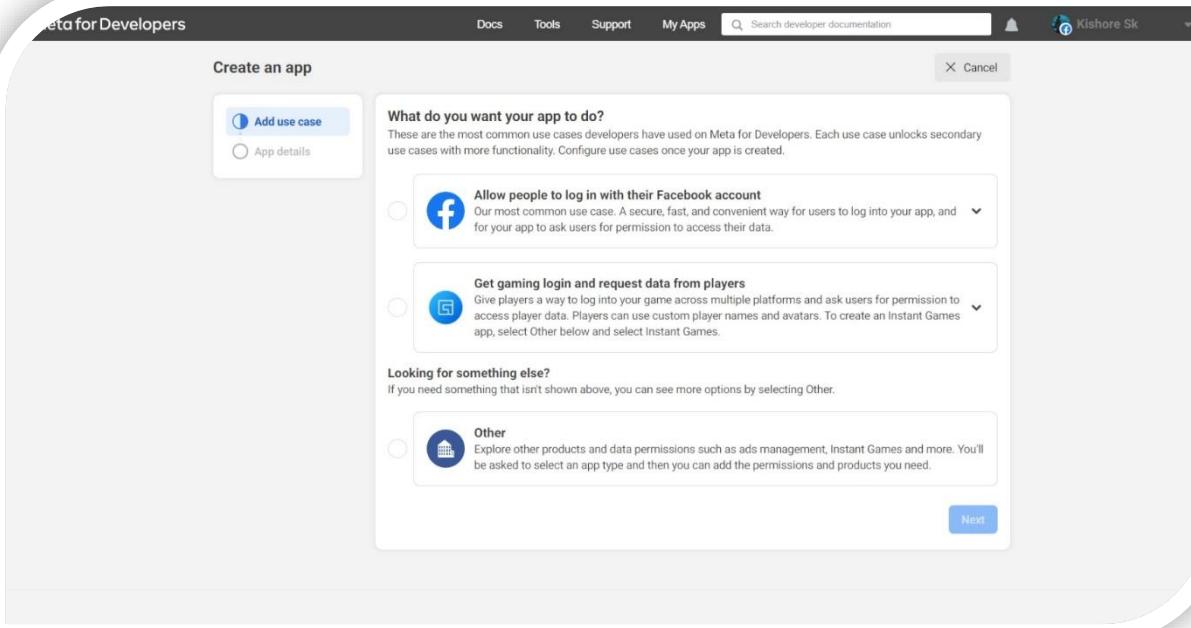
Developers can often access support through developer forums, communities, and help centers to get assistance with their projects.

6. App Review Process:

When developers create applications that integrate with Meta's platforms, they may need to go through an app review process to ensure compliance with platform policies and security standards.



The image shows the Meta Connect landing page. At the top, there's a large, colorful background graphic featuring abstract shapes like a spiral and a small yellow emoji-like character. Below the graphic, the title "Meta Connect" is displayed in a bold, black font. A subtext below the title reads: "In case you missed anything or want to rewatch any of the developer sessions, product announcements, and Mark's keynote address, find it here." A "Watch now" button with a play icon is located to the left of the text. The main navigation bar at the bottom includes links for "Developer Products", "Docs", "Blog", "More", and "My Apps", along with a search icon.



The image shows the "Create an app" interface on the Meta for Developers website. On the left, there are two buttons: "Add use case" (which is selected) and "App details". The main area is titled "What do you want your app to do?". It lists three common use cases: "Allow people to log in with their Facebook account", "Get gaming login and request data from players", and "Other". Each use case has a description and a dropdown menu. Below this, there's a section titled "Looking for something else?" with a note about selecting "Other" for more options. A "Next" button is located at the bottom right of the form.

The screenshot shows the 'App settings' section of the Facebook developer portal. The left sidebar includes links for Dashboard, Required actions, App roles, Alerts, App Review, Products, Facebook Login, Webhooks, Messenger, and Activity log. The main area displays basic app information: App ID (2036177513393580), App secret (redacted), Display name (chatbot), Namespace (empty), Contact email (empty), Privacy Policy URL (Privacy policy for Login dialog and app details), Terms of Service URL (Terms of Service for Login dialog and App Details), and App icon (1024 x 1024). A 'Verifications' section contains a 'Business verification' note and a 'Data Protection Officer contact information' form. Buttons for 'Discard' and 'Save changes' are at the bottom.

Access Tokens

[Create new Page](#)

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: 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.

Pages ↑	Tokens
 Friendly support 160296827161517	Token generated Generate token

[Add or remove Pages](#) [i](#)

Webhooks

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

Callback URL https://integrations.au-syd.assistant.watson.appdomain.cloud/public/fac...	Verify token
---	-----------------------

Validation requests and Webhook notifications for this object will be sent to this URL. Token that Meta will echo back to you as part of callback URL verification.

[Edit callback URL](#) [Show recent errors](#)

Pages ↑	Webhooks
 Friendly support 160296827161517	2 fields messages, messaging_postbacks Edit

[Add or remove Pages](#) [i](#)

INTEGRATING CHATBOT WITH MESSAGING PLATFORM

1. Create a Watson Assistant Chatbot:

Start by creating a chatbot using IBM Watson Assistant. Define the intents, entities, and dialog flows that your chatbot will handle.

2. Choose a Messaging Platform:

Select the messaging platform where you want to integrate the chatbot. Common options include Facebook Messenger, Slack, or a custom web chat.

3. Integration Method:

Depending on the platform, you may need to use different integration methods. Common approaches include using APIs, webhooks, or SDKs provided by the messaging platform. Watson Assistant also provides a webhook integration option.

4. Develop the Integration:

Write code or scripts to connect the messaging platform to Watson Assistant. This often involves setting up a middleware that handles user messages, sends them to the chatbot, and relays the responses back to the user.

5. Authentication and Authorization:

Ensure you have the necessary credentials and permissions to access both the messaging platform and Watson Assistant.

6. Message Handling:

Implement logic to parse incoming messages, understand user intent, and send the appropriate queries to Watson Assistant.

7. Response Processing:

Handle responses from the chatbot and format them for display in the messaging platform.

8. Testing:

Thoroughly test the integration to make sure the chatbot functions correctly within the messaging platform. Test various user inputs and dialog flows.

9. Deployment:

Deploy the integrated chatbot on the messaging platform for users to interact with.

10. Monitoring and Maintenance:

Continuously monitor the chatbot's performance, gather user feedback, and make improvements to the chatbot's responses and capabilities.

The screenshot shows the IBM Watson Assistant Lite dashboard. On the left, there's a sidebar with icons for Home, Channels, and Extensions. Under 'Channels', it says 'Deploy your assistant to third-party channels to expand your reach.' Below this are five cards: 'SMS' (with 'Add +'), 'Facebook Messenger' (with 'Add +'), 'Microsoft Teams' (with 'Add +'), 'Slack' (with 'Add +'), and 'WhatsApp with Twilio' (with 'Add +'). At the bottom of the sidebar, under 'Extensions', it says 'Add features or content to your assistant using custom or pre-built extensions. Learn more'. At the top right, there are 'Contact center integrations' sections for 'NICE CXone' and 'Zscaler', both with 'Open' and 'Edit' buttons. A 'Try Plus plan +' button is also visible. The top navigation bar includes 'IBM Watson Assistant Lite', 'Upgrade', 'Jarvis', 'Learning resources', and a user icon.

This screenshot shows the 'Facebook Messenger' setup wizard. At the top, it says 'Facebook Messenger' and 'Draft'. To the right are 'Close' and 'Next' buttons. Below this, there's a navigation bar with radio buttons for 'Get started', 'Application secret', 'Page access token', 'Webhook', 'Connect assistant', and 'Application review'. The main content area starts with a 'Get started' section: 'Through Facebook Messenger, your assistant is ready to join one of the best places to communicate with all the people and businesses in the world. Learn more'. Below this is a 'Steps to setting up Facebook Messenger' list: 1. Create a Facebook application, 2. Connect to Facebook page, 3. Configure Facebook webhooks, 4. Connect your assistant, 5. Start app review process.

Meta for Developers

Docs Tools Support My Apps Search developer documentation Kishore Sk

Apps

Filter by All Apps Archived Required actions

No apps yet To get started, create your first app.

Create App



Meta

Follow Us [Facebook](#) [Instagram](#) [Twitter](#) [LinkedIn](#) [GitHub](#)

Products Artificial Intelligence AR/VR Business Tools Gaming Open Source Publishing Social Integrations Social Presence

Programs ThreatExchange

Support Developer Support Bugs Platform Status Report a Platform Data Incident Facebook for Developers Community Group Sitemap

News Blog Success Stories Videos Meta for Developers Page

Terms and Policies Platform Initiatives Hub Platform Terms Developer Policies European Commission Commitments

© 2023 Meta About Create Ad Careers Privacy Policy Cookies Terms English (US)

Meta for Developers

Docs Tools Support My Apps Search developer documentation Kishore Sk Help

chatbot App ID: 3602245196730166

Dashboard

Required actions

Build your app

Use cases

App settings

App roles

Testing

Submit for review

Verification

Data handling questions

App Review

Publish

Dashboard

Build your app

Complete all the steps below to make sure your app works the way you need it to.

Use cases Add and customize use cases on your app. Pick permissions and other features to make your app work.

App settings Add platforms, domains, privacy policy, contact info, app category and more.

App roles Assign developer, analytics, tester and admin roles to people working on your app.

Testing Test your app with Graph API Explorer to make sure everything is working.

Submit for review

Meta requires developers to verify and certify their data usage before your app can go live. Developers are also required to tell us about data handling practices. We may request more information during our reviews.

Make sure your app contact email is up-to-date.

Verification This is required for apps to go live and must be completed by a person with full control of the added business account.

Data handling questions To get access and prepare for App Review, you must answer some questions about your data handling practices.

App Review App Review allows us to verify that your app is using data and use cases in an approved manner.

Publish

Publish your app so users without a role on your app can use it.

Go live Once use cases and applicable access requests have been approved, publish your app and go live.

Required actions

You don't have any required action items to display. If any of your apps need immediate attention in the future, an item will show here.

Application Rate Limit	User Rate Limit
 chatbot App ID: 2036177513393580 0% of limit used  100% Remaining	0 Users throttled

Add a product Docs Tools Support My Apps  Kishore Sk 

chatbot App ID: 2036177513393580 App Mode: Development Live App type: None

- Dashboard**
- Required actions
-  App settings
-  App roles
-  Alerts
-  App Review

Products [Add Product](#)
Activity log [Activity log](#)

App Events Understand how people engage with your business across apps, devices, platforms and websites. Read Docs Set up	Facebook Login The world's number one social login product. Read Docs Set up	Audience Network Monetize your app and grow revenue with ads from Meta advertisers. Read Docs Set up
Instagram Graph API Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts. Read Docs Set up	Instagram Basic Display The Instagram Basic Display API allows users of your app to get basic profile information, photos, and videos in their Instagram accounts. Read Docs Set up	Messenger Customize the way you interact with people on Messenger. Read Docs Set up
Marketing API Integrate Facebook Marketing API with your app. Read Docs Set up	Webhooks Subscribe to changes and receive updates in real time without calling the API. Read Docs Set up	Instant Games Create a cross-platform HTML5 game hosted on Facebook. Read Docs Set up
App Center Get your game discovered by being listed as a featured game on Facebook. Read Docs Set up	Web Payments Accept in-app payments through Facebook's secure payment system. Read Docs Set up	ThreatExchange Share and learn about potential threats to help everyone stay more secure. Read Docs Set up
Jobs Post jobs to the Facebook platform and receive applications from users. Read Docs Set up	Fundraisers Create and manage fundraisers for charities. Read Docs Set up	

Looking for something else?
Read more in our developer documentation.

[Read more](#)

Looking for additional products and permissions?

We've streamlined the app creation process by surfacing only the products and permissions needed to build your selected app type. If you need to request additional products and permissions, you can view more options by creating a new app with a different app type. [Learn More](#)

[Create a New App](#)

Assistant Lite Upgrade | Jarvis

Learning resources 

Facebook Messenger Draft

[Close](#) [Next](#)

[Get started](#) [Application secret](#) [Page access token](#) [Webhook](#) [Connect assistant](#) [Application review](#)

Create a Facebook application

Go to [Facebook for Developers](#) 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

```
.....
```

chatbot 

App ID: 2036177513393580 App Mode: Development  Live App type: None

[Learn more](#)

 [Dashboard](#)

 [Required actions](#)

 [App settings](#)

 [App roles](#)

 [Alerts](#)

 [App Review](#)

Products [Add Product](#)

Messenger 

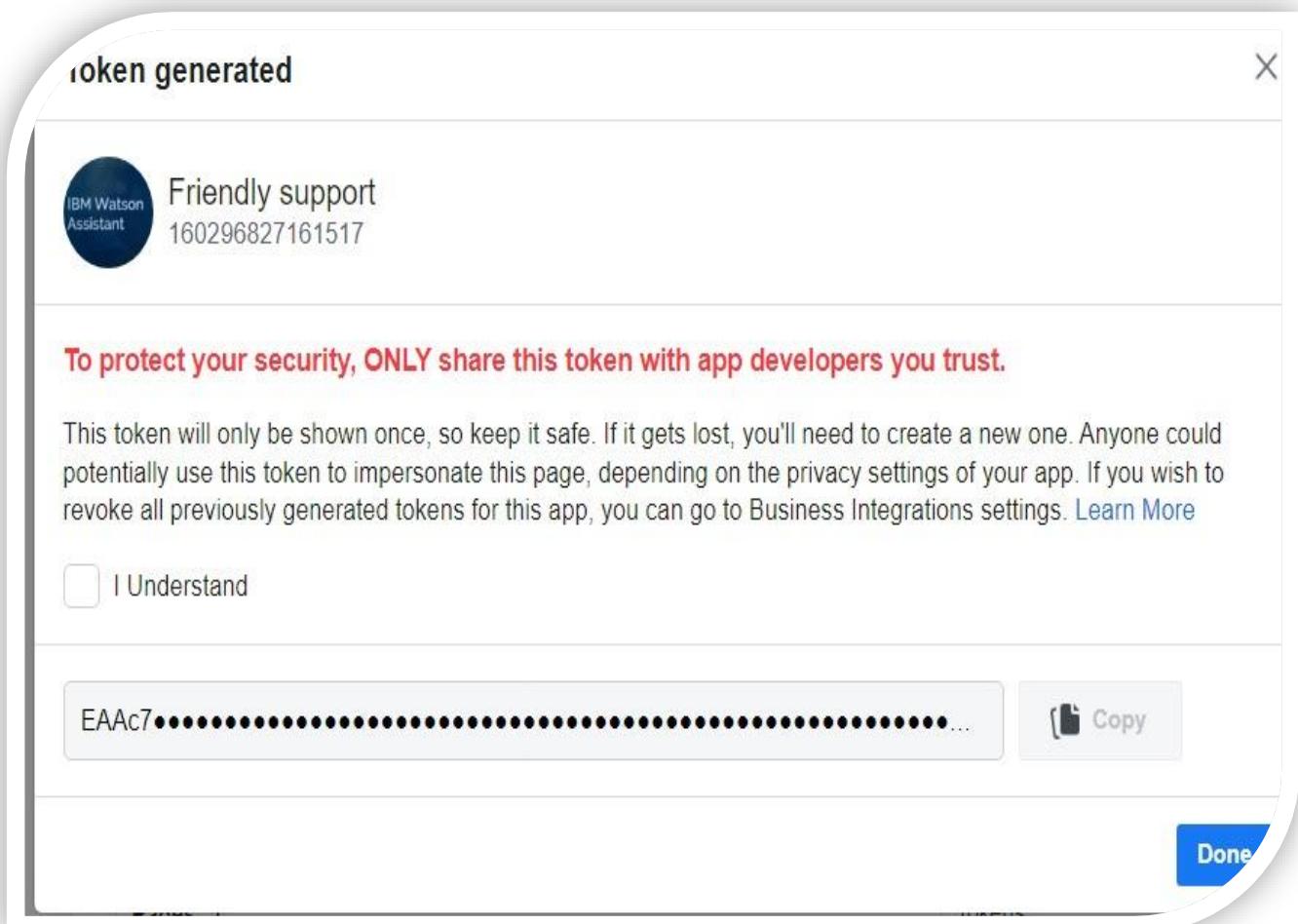
Settings

[Instagram settings](#)

[Activity log](#)

 [Add or remove Pages](#) 

Pages ↑	Tokens
 Friendly support 160296827161517	—
Generate token	



The screenshot shows a dialog for "Facebook Messenger" under the "Draft" tab. At the top, there are links for "Watsonx Assistant Lite", "Upgrade", "Jarvis", "Learning resources", and a help icon. The main content area has a title "Connect to Facebook page" and a numbered list of steps:

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 **Access Tokens**.
4. Click **Add or Remove Pages** and choose the Facebook page you want to use for your app.
5. Click **Generate Token** and check the **I Understand** checkbox to generate the page access token.

A note below the steps says: "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."

Below the note, there is a section for pasting the page access token:

Copy the page access token and paste it here:

Page access token

..... Ⓜ

Facebook Messenger Draft[Close](#)[Save and exit](#)[Facebook app](#)[Facebook page](#)[Webhook](#)[Advanced options](#)

</> Configure Facebook webhooks

The generated request URL will be using the latest service API version.

[Learn more](#)[Generated callback URL](#)

<https://integrations.au-syd.assistant.watson.appdomain.cloud>

1. In the Facebook Messenger settings, scroll to the **Webhooks** section. Click **Setup Webhooks**.
2. In the **Edit Callback URL** window, paste the generated callback URL into the **Callback URL** field.
3. In the **Verify Token** field, paste the verify token that was generated for you earlier.
4. Click **Verify and Save**.
5. Click **Add Subscriptions**. In the **Edit Page Subscriptions** window under **Subscription Fields**, select **messages** and **messaging_postbacks**. Click **Save**.

Edit page subscriptions



Friendly support
160296827161517

Subscription Fields

<input checked="" type="checkbox"/> messages	<input checked="" type="checkbox"/> messaging_postbacks	<input type="checkbox"/> messaging_optins
<input type="checkbox"/> messaging_optouts	<input type="checkbox"/> message_deliveries	<input type="checkbox"/> message_reads
<input type="checkbox"/> messaging_payments	<input type="checkbox"/> messaging_pre_checkouts	<input type="checkbox"/> messaging_checkout_updates
<input type="checkbox"/> messaging_account_linking	<input type="checkbox"/> messaging_referrals	<input type="checkbox"/> message_echoes
<input type="checkbox"/> messaging_game_plays	<input type="checkbox"/> standby	<input type="checkbox"/> messaging_handovers
<input type="checkbox"/> messaging_policy_enforcement	<input type="checkbox"/> message_reactions	<input type="checkbox"/> inbox_labels
<input type="checkbox"/> messaging_feedback	<input type="checkbox"/> messaging_customer_information	<input type="checkbox"/> group_feed

[Learn more](#)[Cancel](#)[Save](#)

CHATBOT ON FACEBOOK

1. Create a Watson Assistant Chatbot:

Start by creating a chatbot using IBM Watson Assistant. Define the intents, entities, and dialog flows that your chatbot will handle.

2. Set Up a Facebook Page:

If you don't already have one, create a Facebook Page for your business or organization. This is where your chatbot will interact with users.

3. Create a Facebook App:

You'll need to create a Facebook App to access the Messenger API. Go to the Facebook for Developers website, create a new app, and configure the app settings.

4. Get Page Access Token:

Within your Facebook App settings, obtain a Page Access Token. This token will allow your app to send and receive messages on behalf of your Facebook Page.

5. Set Up Webhooks:

Configure the webhook for your Facebook App. This is where Facebook will send incoming messages to your chatbot. You'll need to specify a callback URL for your server where you'll handle incoming messages.

6. Develop the Integration:

Write code or scripts to connect your Watson Assistant chatbot to Facebook. You'll need a server to handle the incoming messages and send them to Watson Assistant. Use the Page Access Token to send responses back to Facebook Messenger.

7. Authentication and Authorization:

Ensure you have the necessary credentials and permissions to access both Facebook and Watson Assistant.

8. Message Handling:

Implement logic to parse incoming messages from Facebook, understand user intent, and send the appropriate queries to Watson Assistant.

9. Response Processing:

Handle responses from Watson Assistant and format them for display in Facebook Messenger.

10. Deployment:

Deploy the integrated chatbot on your Facebook Page for users to interact with.

Inbox
Respond to messages, set up automations and more.

All messages Messenger Instagram Not connected Facebook comments Instagram comments

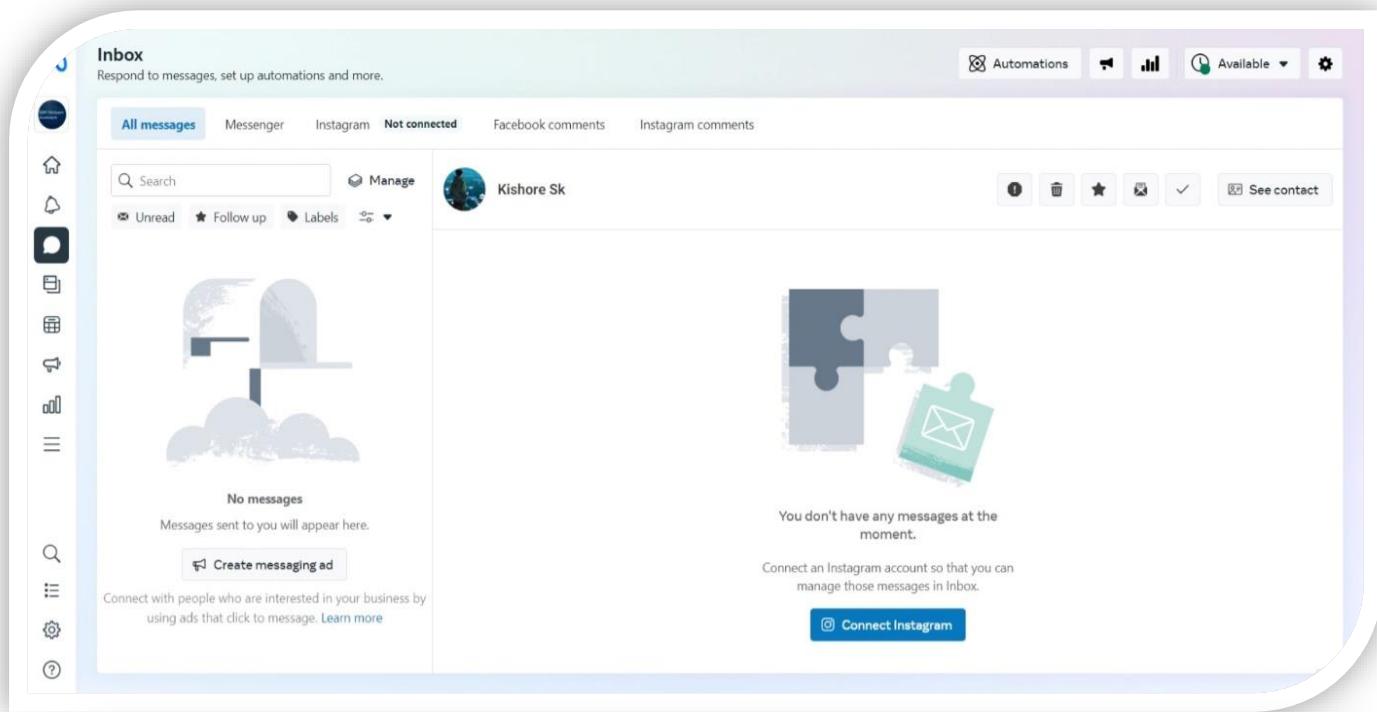
Search Manage

Unread Follow up Labels

Kishore Sk

No messages
Messages sent to you will appear here.
[Create messaging ad](#)

You don't have any messages at the moment.
Connect an Instagram account so that you can manage those messages in Inbox.
[Connect Instagram](#)



Inbox
Respond to messages, set up automations and more.

All messages Messenger Instagram Not connected Facebook comments Instagram comments

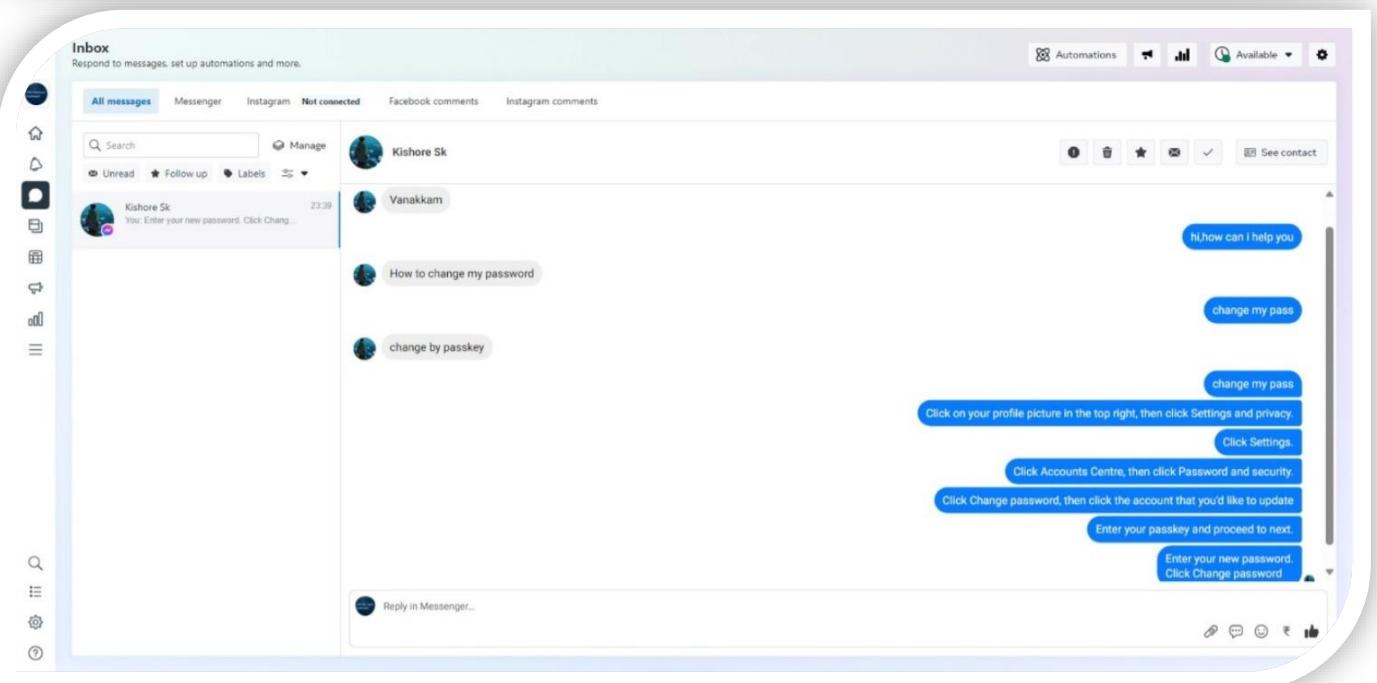
Search Manage

Unread Follow up Labels

Kishore Sk Vanakkam

hi:how can i help you
change my pass
change my pass
change my pass
Click on your profile picture in the top right, then click Settings and privacy.
Click Settings.
Click Accounts Centre, then click Password and security.
Click Change password, then click the account that you'd like to update
Enter your passkey and proceed to next.
Enter your new password.
Click Change password

Reply in Messenger...



Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

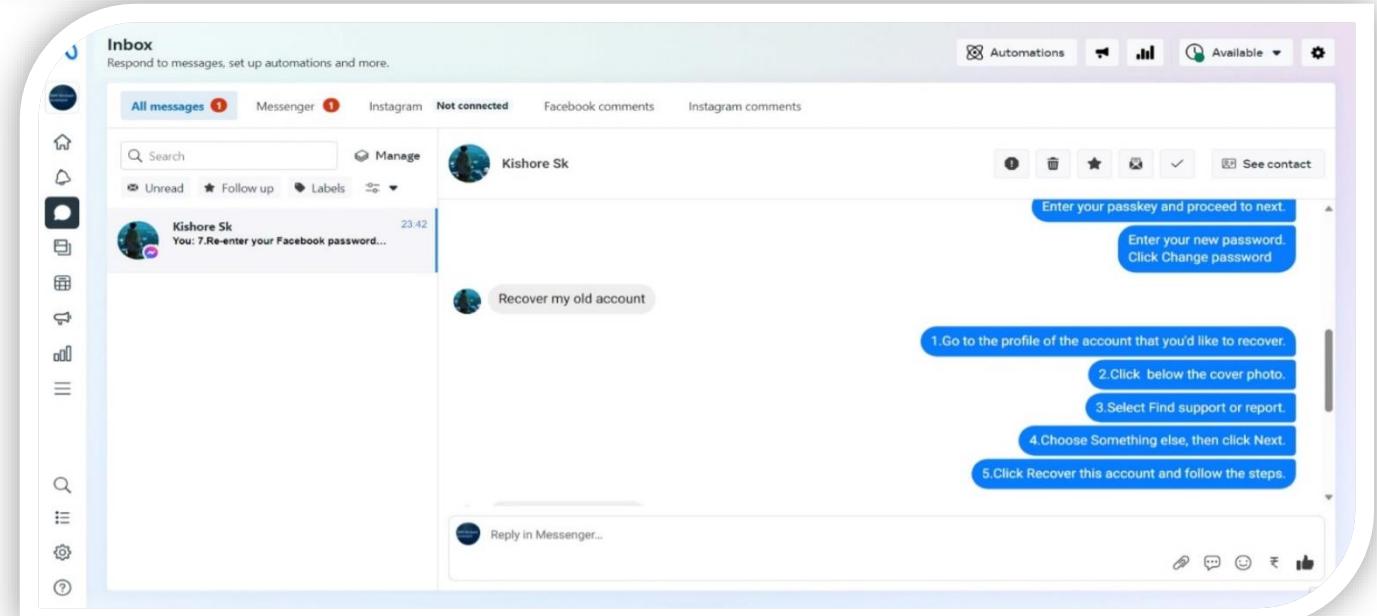
Kishore Sk

Enter your passkey and proceed to next.
Enter your new password.
Click Change password

1.Go to the profile of the account that you'd like to recover
2.Click below the cover photo.
3.Select Find support or report.
4.Choose Something else, then click Next.
5.Click Recover this account and follow the steps.

Recover my old account

Reply in Messenger...



Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: 7.Re-enter your Facebook password...

Kishore Sk How to delete my account

1.Sign into App.
2.Click your profile picture in the top-right corner and select Settings & privacy.
3.Click Settings
4.Click Privacy in the left panel, then click Your App information
5.Scroll to the last option and click Deactivation and deletion.
6.Confirm your choice, then click Continue to account deactivation.
7.Re-enter your Facebook password and click Continue to confirm deactivation

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: 6.Enter your confirmation code and...

Kishore Sk How to change my phone number

1.Tap in the top right of App.
2.Scroll down and tap Settings.
3.In the Account settings section, tap Personal information, then tap Contact info.
4.To add your primary number, enter your number and tap Add number. To add another number, tap Add phone number, enter your number and tap
5.When you add a number, you may be asked to enter a confirmation code. If you don't get a text message with a confirmation code, check to make sure that you've turned on mobile texts.
6.Enter your confirmation code and tap Confirm.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: 4.Click Delete Chat and then click...

Kishore Sk Delete message

5.When you add a number, you may be asked to enter a confirmation code. If you don't get a text message with a confirmation code, check to make sure that you've turned on mobile texts.
6.Enter your confirmation code and tap Confirm.

1.In the top right of your Facebook account, click messenger.
2.Click the conversation you want to delete.
3.At the top of the chat, click .
4.Click Delete Chat and then click Delete Chat.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100%
See how consistently your Page responds to messages and gain other valuable insights on your messaging performance.
View insights

Kishore Sk You: 3.From here, you can: Select a phot... 13:01

Kishore Sk How to change my profile image

1.Click your profile picture in the top right of Facebook, then click your profile name.
2.Click in the bottom right of your profile picture.
3.From here, you can:
Select a photo or upload a new one.
Create an avatar profile picture.
Add a frame to an existing picture.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100%
See how consistently your Page responds to messages and gain other valuable insights on your messaging performance.
View insights

Kishore Sk You: 5.Find the session that you want to... 13:01

Kishore Sk Log out from all devices

1.Tap in the top right of Facebook.
2.Scroll down and tap Settings.
3.Tap Security and login.
4.Go to the WHERE YOU'RE LOGGED IN section. You may need to tap See more to see all of the sessions where you're logged in.
5.Find the session that you want to end and tap , then tap Log Out.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100%
See how consistently your Page responds to messages and gain other valuable insights on your messaging performance.
View insights

Kishore Sk You: 8.Click Recover this account and... 13:02

Kishore Sk How to recover my old account

4.Go to the WHERE YOU'RE LOGGED IN section. You may need to tap See more to see all of the sessions where you're logged in.
5.Find the session that you want to end and tap , then tap Log Out.

1.Go to the profile of the account that you'd like to recover.
2.Click below the cover photo.
3.Select Find support or report.
4.Choose Something else, then click Next.
5.Click Recover this account and follow the steps.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: 6.Click Done. 10:02

Report a message

5.Click Recover this account and follow the steps.

1.In the top right of your Facebook account, click messenger.
2.Click the conversation that you want to report.
3.At the top of the chat, click .
4.Click Report.
5.Select Other and then select Spam.
6.Click Done.

Reply in Messenger...

About

Added details Edit
+ Phone number
+ Email address
2 November
+ Address

Facebook profile

Local time 09:48
Studied at Don bosco matriculation school karamodai
Lives in Chennai, India
From Chennai, India

Labels Manage labels
Add Label

Suggested labels
New customer Important
Today's date (10/28)

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: My pleasure, I'm glad to help you. 10:07

Nandri

1.In the top right of your Facebook account, click messenger.
2.Click the conversation that you want to report.
3.At the top of the chat, click .
4.Click Report.
5.Select Other and then select Spam.
6.Click Done.

My pleasure, I'm glad to help you.

Reply in Messenger...

Inbox
Respond to messages, set up automations and more.

All messages 1 Messenger 1 Instagram Not connected Facebook comments Instagram comments

Search Manage

Unread Follow up Labels

Your response rate is 100% See how consistently your Page responds to messages and gain other valuable insights on your messaging performance. View insights

Kishore Sk You: No worries. 13:19

Nandri

Thx

Thanks

My pleasure, I'm glad to help you. You're welcome

You're welcome

No worries

Reply in Messenger...

About

Add details about people, such as contact information. Add Details

Facebook profile

Local time 13:19
Studied at Don bosco matriculation school karamodai
Lives in Chennai, India
From Chennai, India

Labels Manage labels
Add Label

Suggested labels
New customer Important
Today's date (10/28)

Activity Add activity

Slack API

1. Set Up a Slack Workspace:

You'll need a Slack workspace to develop and test your app. If you don't have one, create a free workspace on Slack's website.

2. Create an App:

Go to the Slack API website and create a new Slack app. This will provide you with an "App ID" and other credentials.

3. Choose App Features:

Decide on the features you want to add to your app, such as chatbots, slash commands, or message buttons.

4. Use Slack APIs:

You can interact with Slack using various APIs, including the Real-Time Messaging API, Web API, Events API, and more. Use these APIs to send and receive messages, manage channels, and perform other actions.

5. Configure Permissions:

Define the permissions your app needs to access specific parts of Slack data. Request these permissions and handle the OAuth flow to authenticate your app with a user's workspace.

6. Develop Your App:

Write code to implement the functionality you've planned. You can use various programming languages and frameworks to build your app.

7. Test Your App:

Test your app in a Slack sandbox environment or a designated channel to ensure it works as expected.

8. Deploy Your App:

Once your app is ready, you can deploy it to your Slack workspace or distribute it to other workspaces if desired.

9. Submit Your App for Review:

If you want to make your app available to a broader audience, you can submit it to the Slack App Directory. Slack may review your app before it's listed.

10. Maintain and Update Your App:

Keep your app up-to-date, fix bugs, and add new features as needed. Slack may release updates or changes to its APIs, so you'll need to adapt your app accordingly.

The screenshot shows the Slack API documentation interface. On the left, there's a sidebar with various developer tools and concepts: Start learning, Automation, Authentication, Messaging, Metadata, Surfaces, Block Kit, Interactivity, APIs, Workflows, Enterprise, Apps for Admins, Gov Slack, and Reference. A 'Translated content' link is at the bottom of this sidebar.

The main area is titled 'Your Apps'. At the top right is a 'Create New App' button. Below it is a callout box containing text about the Slack App Directory Agreement and a 'I Agree' button. A search bar labeled 'Filter apps by name or workspace' is also present.

A table lists the current apps in the workspace:

App Name	Workspace	Distribution Status
Demo App	kishore	Not distributed

slack api

Our Blog

Documentation Tutorials Your Apps

Client ID
6124062283633.6113875214756

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
4ZKScgWyZqdkYTD50BTpcF6z 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.

slack api

Review & Submit Give feedback Slack ♥ Help Contact Policies Our Blog

Documentation Tutorials Your Apps

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
A063BRR6AN8	October 28, 2023

Client ID
6124062283633.6113875214756

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
4ZKScgWyZqdkYTD50BTpcF6z Regenerate

slack api

Documentation Tutorials Your Apps

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 @Demo App in conversations that the app is in
chat:write	Send messages as @Demo App
im:history	View messages and other content in direct messages that Demo App has been added to
im:read	View basic information about direct messages that Demo App has been added to
im:write	Start direct messages with people

slack api

Documentation Tutorials Your Apps

Demo App ▾

OAuth & Permissions

Settings

- Basic Information
- Collaborators
- Socket Mode
- Install 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 NEW
- Beta Features

Submit to App

Advanced token security via token rotation

Recommended for developers building on or for security-minded organizations - opting into token rotation allows app tokens to automatically expire after they're issued within your app code. [View documentation](#).

⚠️ At least one redirect URL needs to be set below before this app can be opted into token rotation

Opt In

OAuth 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 Token

xoxb-6124062283633-6108514734949-KUj3AEyKF0vNTZn8Vn77FkFU [Copy](#)

Access Level: Workspace

Demo App ▾

Settings
Basic Information
Collaborators
Socket Mode
Install 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 NEW
Beta Features**Submit to App Directory**

Review & Submit

Give feedback

Slack ❤
Help
Contact
Policies
Our Blog

OAuth & Permissions

Advanced token security via token rotation

Recommended for developers building on or for security-minded organizations – opting into token rotation allows app tokens to automatically expire after they're issued within your app code. View documentation.

⚠ At least one redirect URL needs to be set below before this app can be opted into token rotation

Opt In

OAuth 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 Tokenxoxb-6124062283633-6108514734949-KUJ3AEyKF0vNTZn8Vn77FkFU **Copy**

Access Level: Workspace

Reinstall 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.

Redirect URLs

You haven't added any Redirect URLs

Add New Redirect URL**Save URLs**

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 @Demo App in conversations that the app is in
chat:write	Send messages as @Demo App
im:history	View messages and other content in direct messages that Demo App has been added to
im:read	View basic information about direct messages that Demo App has been added to
im:write	Start direct messages with people

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.

Restrict API Token Usage

Slack can limit use of your app's OAuth tokens to a list of IP addresses and ranges you provide. Slack will then reject Web API method calls from unlisted IP addresses. Learn More.

Allowed IP Address Ranges

You haven't added any IP address ranges

Add a new IP address range**Save IP address ranges**

Revoke All OAuth Tokens

You can revoke all OAuth tokens if you want to invalidate the access any existing tokens have to Slack workspace data. Users will need to grant your app permissions again to use it.

Revoke Tokens

The screenshot shows the 'Event Subscriptions' section of the Slack API settings. On the left, a sidebar lists 'Settings' (Basic Information, Collaborators, Socket Mode, Install App, Manage Distribution), 'Features' (App Home, Org Level Apps, Incoming Webhooks, Interactivity & Shortcuts, Slash Commands, Workflow Steps, OAuth & Permissions), and 'Event Subscriptions' (selected). The main area displays the 'Enable Events' toggle switch (On) and the 'Request URL' input field (https://integrations.au-syd.assistant.watson.appdomain.cloud/public/slack). A note explains that events will be sent to this URL via HTTP POST. Below this, a callout box informs about the 'New event authorization format'.

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 [Verified](#)

https://integrations.au-syd.assistant.watson.appdomain.cloud/public/slack [Change](#)

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](#).

New event authorization format

Recent changes to Events API payloads

The Events API now sends information about authorized users and workspaces in a new, compact format. [Learn more](#).

The screenshot shows the 'Subscribe to bot events' section of the Slack API. The left sidebar includes links for Beta Features, Submit to App Directory (Review & Submit, Give feedback, Slack ❤️, Help, Contact, Policies, Our Blog), and a search bar. The main content area shows a table of bot events and a preview of event details.

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

[Add Bot User Event](#)

im_history_changed Bulk updates were made to a DM's history

message.im A message was posted in a direct message channel

groups:read

slack api

Documentation Tutorials Your Apps

Demo App ▾

App Home

Settings

- Basic Information
- Collaborators
- Socket Mode
- Install App
- Manage Distribution

Features

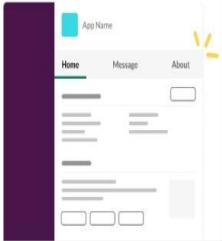
- App Home**
- Org Level Apps
- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- Workflow Steps
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- App Manifest NEW
- Beta Features

Submit to App

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](#)



Your App's Presence in Slack

Your app display name and presence is how your users will see your app in Slack.

App Display Name [Edit](#)

Display Name (Bot Name): watson chatbot
Default Name: kishore_sk

slack api

Documentation Tutorials Your Apps

Demo App ▾

Interactivity & Shortcuts

Settings

- Basic Information
- Collaborators
- Socket Mode
- Install 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 NEW

Interactivity [On](#)

Any interactions with shortcuts, modals, or interactive components (such as buttons, select menus, and datepickers) will be sent to a URL you specify. [Learn more](#).

Request URL

`https://integrations.au-syd.assistant.watson.appdomain.cloud/public/slack/66eb67`

Slack will send an HTTP POST request with information to this URL when users interact with a shortcut or interactive component.

Shortcuts

Let people take actions – like filing a bug or adding a new sales lead – while using your app in Slack. [Learn more](#).

Name	Location	Callback ID
------	----------	-------------

CHATBOT ON SLACK

1. Create an IBM Watson Assistant Service:

If you don't already have one, sign up for an IBM Cloud account and create an instance of IBM Watson Assistant, which is IBM's AI-powered chatbot development platform.

2. Design and Train Your Chatbot:

Use the Watson Assistant service to design and train your chatbot. You can define intents, entities, and dialog flows to make your bot understand and respond to user messages effectively.

3. Integrate with Slack:

To integrate your Watson chatbot with Slack, you'll typically use the Slack API. Here are the general steps:

a. Create a Slack App:

Go to the Slack API website and create a new Slack app. This app will act as the bridge between Slack and your Watson chatbot.

b. Set Up OAuth & Permissions:

Configure your Slack app to request the necessary permissions, allowing it to access Slack workspaces. You will need to define the OAuth scopes and permissions required.

c. Handle OAuth Flow:

Implement the OAuth 2.0 flow in your application to allow users to install your Slack app in their workspaces. This involves redirecting users to Slack for authorization and receiving an access token in return.

d. Receive Events from Slack:

Use the Slack Events API to listen for messages and other events in Slack channels. When a message is received, you can send it to your Watson chatbot for processing.

e. Send Responses to Slack:

Once your Watson chatbot processes a message, it can send a response back to the Slack channel through the Slack API.

4. Test Your Chatbot:

Test your chatbot within a Slack channel or a sandbox environment to ensure it interacts correctly with users.

5. Deploy Your Chatbot:

Deploy your chatbot to your desired Slack workspace(s) or make it available for others to install in their workspaces.

IBM watsonx Assistant Lite [Upgrade](#) | Jarvis ▾ Learning resources

Slack (Draft)

[Get started](#) [Slack bot](#) [Connect Slack](#) [Configure Slack](#) [Connect assistant](#)

Get started

Through Slack, your assistant is ready to join the collaboration hub that brings the right people, information, and tools together to get work done. [Learn more](#)

Steps to setting up Slack

1. Set up your Slack bot
2. Connect watsonx Assistant to Slack
3. Configure your Slack bot
4. Connect your assistant

[Close](#) [Next](#)

IBM watsonx Assistant Lite [Upgrade](#) | Jarvis ▾ Learning resources

Slack (Draft)

[Slack bot](#) [Connect Slack](#) [Configure Slack](#) [Connect assistant](#)

Set up your Slack bot

Create a [Slack app](#) if you don't have one already.

If you already have an app, go to the [Your Apps](#) page on the Slack website. Click the app you want to use.

[Close](#) [Next](#)

IBM watsonx Assistant Lite [Upgrade](#) | Jarvis ▾ Learning resources

Slack (Draft)

[Slack bot](#) [Connect Slack](#) [Configure Slack](#) [Connect assistant](#)

Connect watsonx Assistant to Slack

1. On the Slack app settings page, go to the **Basic Information** tab and find the **App Credentials** section. Copy your verification token from that section to the field below.

Verification token

2. Go to the **OAuth & Permissions** tab. In the **Bot Token Scopes** section click **Add an OAuth Scope**, and then select the following scopes:

- app_mentions:read
- chat:write
- im:history
- im:read
- im:write

[Close](#) [Next](#)

IBM Watson Assistant Lite Upgrade Jarvis Learning resources 

Slack

Draft

1. On the Slack app settings page, go to the **Event Subscriptions** tab. Switch the **Enable Events** toggle to the **On** position. In the provided text entry field, paste your request URL.

Generated request URL:

```
https://integrations.au-syd.assistant.watson.appdomain.cloud
```

1 The generated request URL will be using the latest service API version. [Learn more](#)

2. On the **Event Subscriptions** tab, find the **Subscribe to Bot Events** section. Click **Add Bot User Event**, and then select the event types you want to subscribe to. You must select at least one of the following types:

- **message.im**: Listens for message events that are posted in a direct message channel.
- **app_mention**: Listens for only message events that mention your app or bot.

3. Click **Save Changes**.

Close Next

IBM Watson Assistant Lite [Upgrade](#) | Jarvis 

Learning resources 

Slack

 [Get started](#)  [Slack bot](#)  [Connect Slack](#)  [Configure Slack](#)  [Connect assistant](#)

Connect your assistant

1. On the [App Home](#) tab, click [Edit](#) and enter a display name and default username for your virtual assistant and then click [Save](#). Enable the [Always Show My Bot as Online](#) toggle.
2. On the [App Home](#) tab, find the [Show Tabs](#) section. Enable the [Messages Tab](#) toggle and check the [Allow users to send Slash commands and messages from the messages tab](#) checkbox.
3. **Optional:** To add support for showing buttons, menus, and disambiguation options in the Slack app, go to the [Interactivity & Shortcuts](#) tab and enable the feature. Paste your request URL in the provided text entry field, and then click [Save Changes](#).

[Close](#) [Finish](#)

A screenshot of the Slack application interface. On the left, a purple sidebar shows a user profile picture 'kishore' and a navigation menu with icons for Home, Later, More, and a plus sign. The main workspace shows a channel list under 'Channels' with '# general', '# project', '# random', '+ Add channels', and a collapsed 'Direct messages' section containing 'Darkshawdowcoc In' and 'Kishore .S you'. Below these are '+ Add coworkers', 'Apps' (with 'Slackbot' listed), and 'watson chatbot' which is highlighted with a purple selection bar. The right side of the screen shows a direct message window with the title 'watson chatbot'. The window has tabs for 'Home', 'Messages' (which is selected), and 'About'. It displays a message from 'watson chatbot' (@watson chatbot) saying 'This is the very beginning of your direct message history with @watson chatbot'. Below it is a question from 'Kishore .S' at 1:08 PM asking 'How does watson chatbot work?'. A reply from 'watson chatbot' follows, saying 'hi, how can i help you'. At the bottom, there's a message input field with placeholder text 'Message watson chatbot' and a toolbar with various message formatting options like bold, italic, and code.

The screenshot shows a Slack workspace interface. On the left, the sidebar displays the workspace name "kishore" and various channels like "# general", "# project", and "# random". A "watson chatbot" app is selected. The main area shows a conversation with "watson chatbot" and "Kishore.S". The message history includes:

- Kishore.S: Vanakkam
- watson chatbot: hi,how can i help you
- Kishore.S: How to reset my slack password
- watson chatbot: 1.Visit <https://slack.com/signin>.
- 2.Enter your email address and click Sign In with email address.
- 3.Check your inbox for a Slack confirmation code.
- 4.Enter the code on the sign-in page, then select the workspace that you'd like to
- 5.sign in to.
- 6.Follow the steps in the next section to add a password.

At the bottom, there's a message input field with placeholder "Message watson chatbot" and a toolbar with various message format options.

The screenshot shows a Slack workspace interface. On the left, there's a sidebar with a purple header containing the workspace name "kishore". Below the header are sections for "Channels", "Later", "More", and "Add coworkers" (which is highlighted). Under "Apps", "Slackbot" is listed, and "watson chatbot" is selected and highlighted in purple. At the bottom of the sidebar are "Add apps" and a plus sign icon.

The main area shows a conversation with "watson chatbot". The messages are:

- Kishore .S 6:02 PM How to reset my slack password
- watson chatbot 6:02 PM 1.Visit <https://slack.com/signin>.
- 2.Enter your email address and click Sign in with email address.
- 3.Check your inbox for a Slack confirmation code.
- 4.Enter the code on the sign-in page, then select the workspace that you'd like to
- 5.sign in to.
- 6.Follow the steps in the next section to add a password.

Below this, another message from "Kishore .S" at 6:04 PM asks "How to create a channel". The "watson chatbot" responds with steps 1 through 4 to create a channel.

At the bottom of the screen, there are message input fields and various message format buttons (B, I, S, etc.) and a rich text editor toolbar.

The screenshot shows the Slack desktop application interface. On the left is the sidebar with a purple header containing the user's name 'kishore'. Below the header are sections for 'Home', 'Later', 'More', and a list of channels and direct messages. A 'Search' bar at the top right contains the query 'kishore'. The main pane displays a search result for 'watson chatbot'. The result includes a profile picture of a blue robot, the channel name 'watson chatbot', and a timestamp '6:12 PM'. Below this is a list of steps for creating a user group:

1. From your desktop, open the channel that you want to make private.
2. Click on the channel name in the conversation header.
3. Select the Settings tab.
4. Scroll down and click Change to a private channel.
5. Click Change to private to confirm.

Below this, another message from 'Kishore .S' at 6:13 PM provides instructions for creating a user group:

How to create a user group

From your desktop, click More, then select People.

Choose a name and handle for your user group. If you'd like to, set a purpose (channel) or select default channels for group members to be added to. When you're ready, click Next.

Under Add members, search for and select the members you want to add.

Once you've finished, click Create group.

At the bottom of the main pane is a message input field with placeholder text 'Message watson chatbot' and a toolbar with various icons.

This screenshot is identical to the one above it, showing the same Slack interface and search results for 'kishore'.

This screenshot is identical to the ones above it, showing the same Slack interface and search results for 'kishore'.

watson chatbot 6:17 PM

1.Click your workspace icon in the top left to view a list of all the workspaces that you're signed in to.
2.Select another workspace to open it.
3.If you like, click Show workspace switcher to keep all your workspaces visible in the sidebar.

watson chatbot 6:17 PM

1.Click your workspace icon in the top left to view a list of all the workspaces that you're signed in to.
2.Select another workspace to open it.
3.If you like, click Show workspace switcher to keep all your workspaces visible in the sidebar.

Kishore .S 6:20 PM

How to change my email address in slack

watson chatbot 6:20 PM

1. Open the app on your phone.
2. Tap on the stacked horizontal lines (hamburger menu) in the lower right hand corner of the screen to open a list of menu options.
3. Tap on the gear icon on the upper right corner of the screen to go to Settings & privacy.
4. On the Settings & privacy screen, tap Personal and account information.
5. Choose the Contact info.
6. You will then see all the phone numbers and email addresses that you've linked to your Facebook account. Next, click on Add email address.
7. On the next screen, App will require you to enter your new email address and your password. Once you're done, tap on Add Email Address, and you'll be sent a confirmation email.
8. Open the confirmation email and validate your new address. App will automatically set your newly entered email address as your primary contact

watson chatbot 6:29 PM

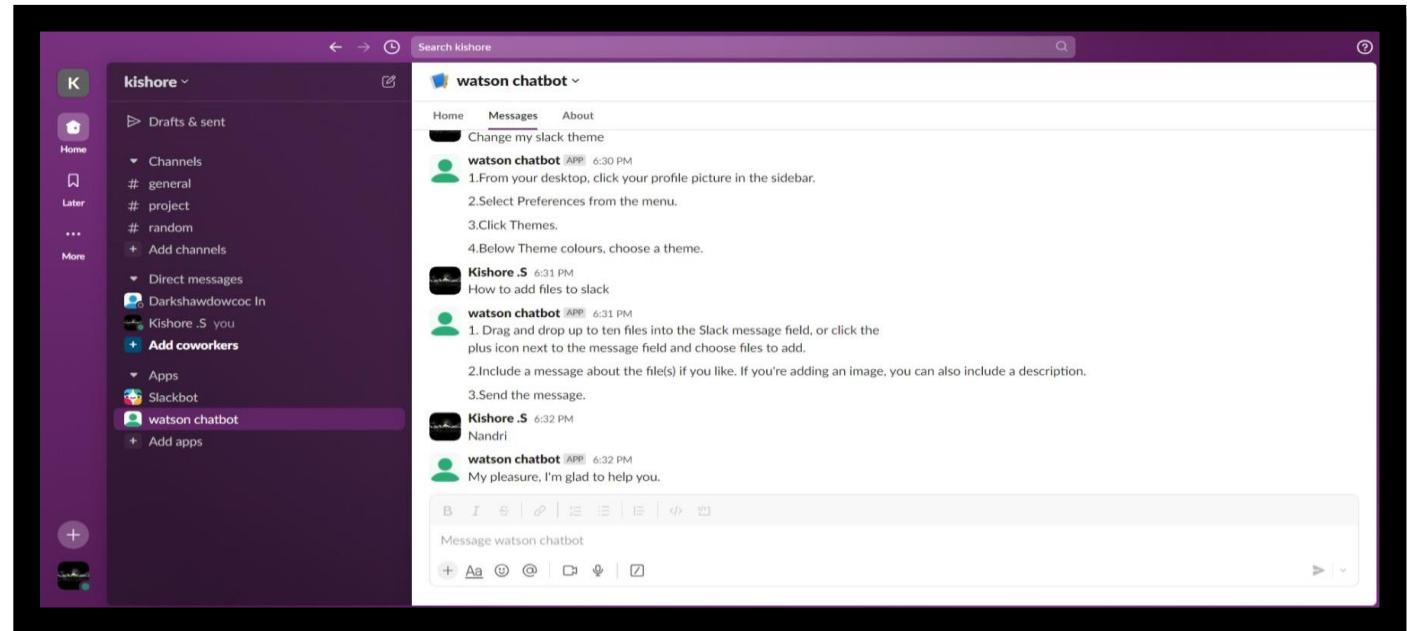
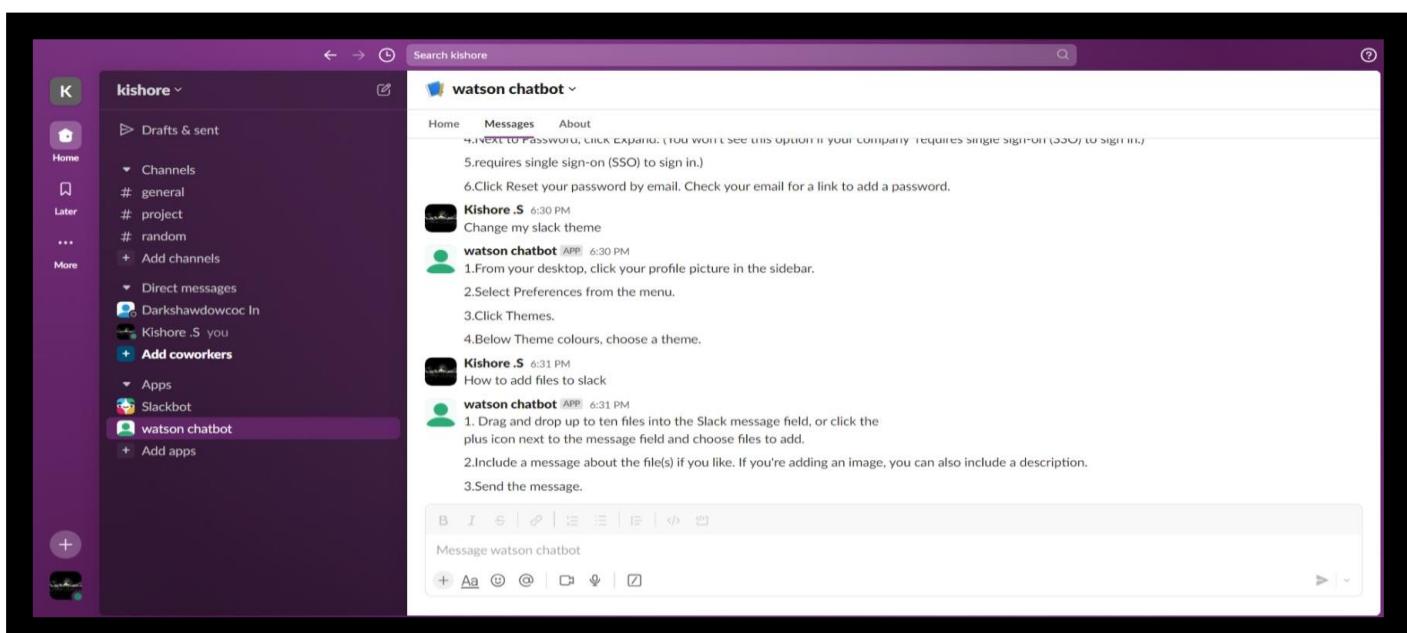
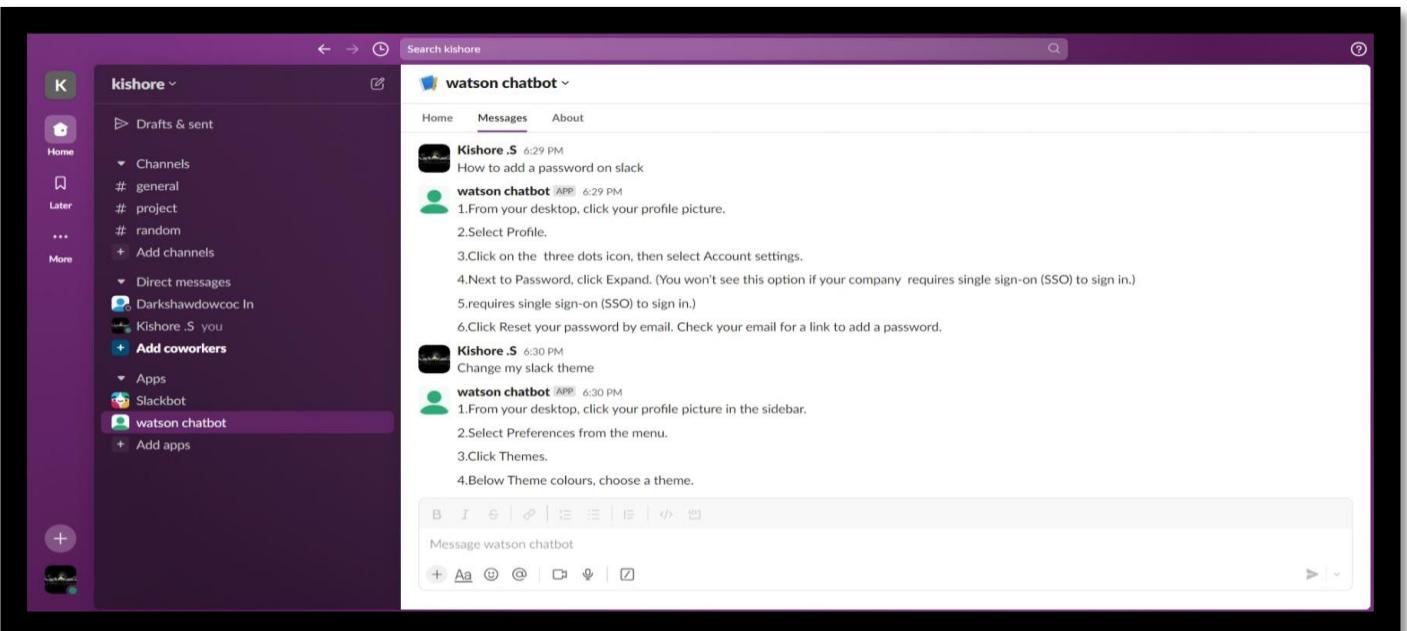
4. On the Settings & privacy screen, tap Personal and account information.
5. Choose the Contact info.
6. You will then see all the phone numbers and email addresses that you've linked to your Facebook account. Next, click on Add email address.
7. On the next screen, App will require you to enter your new email address and your password. Once you're done, tap on Add Email Address, and you'll be sent a confirmation email.
8. Open the confirmation email and validate your new address. App will automatically set your newly entered email address as your primary contact

Kishore .S 6:29 PM

How to add a password on slack

watson chatbot 6:29 PM

1.From your desktop, click your profile picture.
2.Select Profile.
3.Click on the three dots icon, then select Account settings.
4.Next to Password, click Expand. (You won't see this option if your company requires single sign-on (SSO) to sign in.)
5.requires single sign-on (SSO) to sign in.
6.Click Reset your password by email. Check your email for a link to add a password.



THANKYOU!