

5.3 AI and Bots

Bright Pattern Documentation

Generated: 11/28/2021 5:43 am

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
AI & Bots Tutorials Overview	4
Bot Creation	4
Integration Account Configuration	4
Putting It All Together in Bright Pattern	4
How to Create an Amazon Lex Bot	4
Procedure	5
Step 1: Create an AWS account	5
Step 2: Create an IAM user, grant administrative permissions, and save credentials	5
Step 3: Add Lex as a service and create a sample bot	6
Step 4: Review the bot workspace	6
Step 5: Edit intent properties	7
Properties	7
Step 6: Save and build	8
Step 7: Test it	8
Step 8: Publish your bot	9
Next Steps	10
How to Create a Watson Assistant	10
Procedure	11
Step 1: Create an IBM account	11
Step 2: Add Watson Assistant as a resource	11
Step 3: Add a dialog skill	11
Step 4: Create and/or add intents for your dialog skill	12
Intent Properties	13
Step 5: Create entities	13
Entity Properties	14
Step 6: Design your dialog flow	15
Dialog Properties	15
Step 7: Try the Dialog	15
Step 7: Add the Watson Assistant	16
Next Steps	17
How to Configure a Microsoft Azure Web App Bot from a Template	17
Prerequisites	18
Procedure	18
Step 1: Create a Web App Bot in Azure from a Sample Template	18
Step 2: Configure Direct Line channel	20
Next step: Edit Bright Pattern Contact Center scenario	22
How to Add a Bot/Chat Suggestions Engine Integration Account	22
Adding a New Account	22
Next Steps	24
Set up an AWS Lex Integration Account	24
Procedure	24
Step 1: Add integration account	24
Step 2: Edit properties	24
Name	25
Type	25
User ID	25
Bot name	25
Bot alias	25
Access key	25
Secret key	25
Region	25
Max API calls per day	25
Step 3: Save properties	25
Recommended Reading	26
Set up a Watson Assistant (Conversation) Integration Account	26
Procedure	26
Step 1: Add integration account	26
Step 2: Edit properties	26
Name	27
Type	27
Url	27
Workspace ID	28
Username	29
Password	30
Max API calls per day	30

Maximum suggestions	30
Test Connection	30
Step 3: Save properties	30
Recommended Reading	30
Set up a Watson Assistant Integration Account	30
A Note About the Watson Name	30
Procedure	31
Step 1: Add integration account	31
Step 2: Edit properties	31
Name	32
Type	32
Url	32
Workspace ID	34
API Key	35
How to Create a New Key	36
Max API calls per day	37
Maximum suggestions	37
Test Connection	38
Step 3: Save properties	38
Recommended Reading	38
How to Integrate Bots with Chat	38
Prerequisites	38
Procedure	38
Step 1: Create a Watson Assistant in IBM Cloud	39
Step 2: Get Your Watson Credentials	39
Step 3: Add an Integration Account	39
Properties	41
Step 4: Create or Select a Chat Scenario	41
Step 5: Edit the Scenario to Work with Your Bot	43
Step 6: Create or Select a Chat Service	50
Step 7: Create or Choose a Chat Scenario Entry	50
How to Add a New Chat Scenario Entry	51
Required Properties	51
Other Properties	52
Step 8: Edit Chat Widget Style	52
Step 9: Add a Contact Tab	52
Step 10: Get the HTML	52
Final Step: Test the Chat	52

AI & Bots Tutorials Overview

AI and Bot tutorials provide step-by-step instructions on how to create functional chatbots and integrate them into your contact center services.

Learn how to create chatbots, configure chatbot integration, and how to get your Bright Pattern scenarios to work with your integrated chatbots.

For more information about using the Contact Center Administrator application, see all [Tutorials for Admins](#).

Bot Creation

- [How to Create an Amazon Lex Bot](#)
- [How to Create a Watson Assistant](#)

Integration Account Configuration

- [How to Add a Bot or Chat Suggestions Engine](#)
- [Set up an AWS Lex Integration Account](#)
- [Set up a Watson Assistant Integration Account](#)

Putting It All Together in Bright Pattern

- [How to Integrate Bots with Chat](#)

How to Create an Amazon Lex Bot

Bright Pattern Contact Center integrates with Amazon Lex, a platform for building, testing, and deploying chatbots from the AWS Management Console. Lex provides both automatic speech recognition (ASR) and natural language understanding (NLU) technologies, enabling chatbots to recognize customers' speech and text input, understand intent, and transcribe speech input. Integration with Lex lets your contact center access Lex through chat scenarios and provide bot assistance from within chat interactions.

In this article, you will learn how to create a basic Amazon Lex bot that can be used as a conversational bot with your [configured chat service](#).



Chat showing an integrated Lex bot and suggestions for the agent

Procedure

This procedure will walk you through the process of setting up your first Amazon Lex bot. For a deeper understanding of Amazon Lex and other AWS resources, refer to [AWS's Amazon Lex Developer Guide](#).

Step 1: Create an AWS account

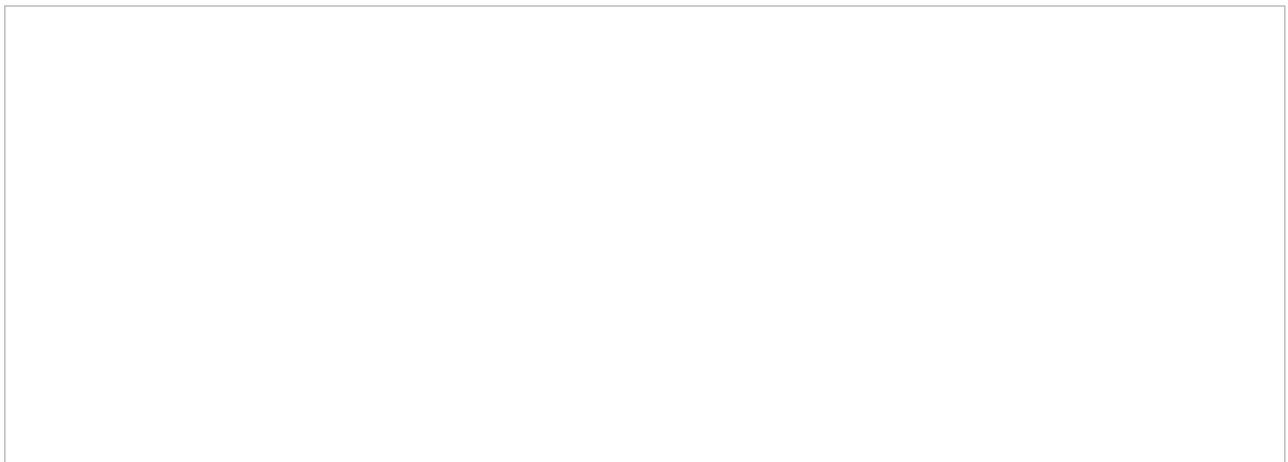
1. If you haven't already done so, [set up an AWS account](#).

Step 2: Create an IAM user, grant administrative permissions, and save credentials

Our integration accounts require access keys to connect to and use your Amazon Lex bot. Access keys are created and managed in AWS Identity and Access Management (IAM) services.

To get an access key, you need to:

1. Go to your [AWS Management Console > IAM Dashboard](#). If you don't know where it is, search AWS Services for "IAM."
2. From the IAM root menu, click **Users** and then click **Add user** to create an IAM user and grant administrative permissions. Adding a user creates credentials that are used to access AWS.



Add IAM User

3. Copy the **access key ID** (e.g., AKIAIOSFODNN7EXAMPLE) and the **secret access key** (e.g., wjaIrXUtnFEMI/K7MDENG/bPxrFicYEXAMPLEKEY). Save this for when you set up an [AWS Lex bot/chat suggestions integration account](#).

For more information, see [Managing Access Keys for IAM Users](#).

Step 3: Add Lex as a service and create a sample bot

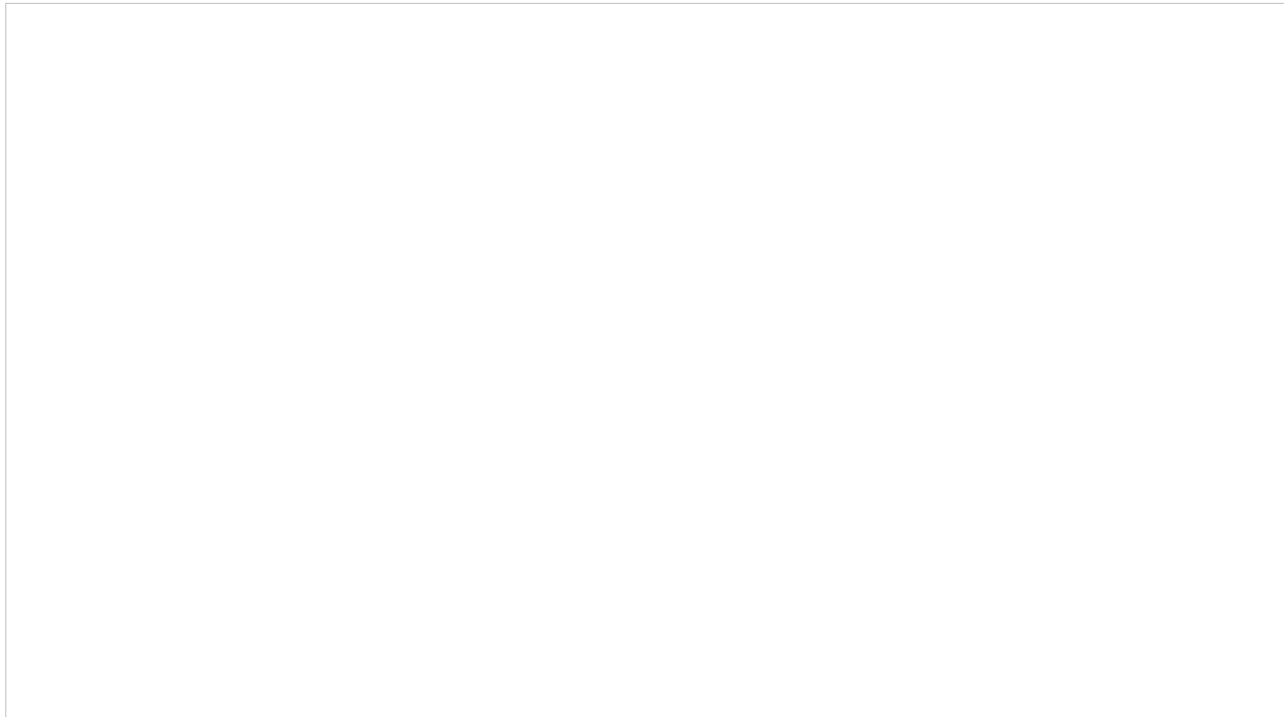
1. Sign in to the AWS Management Console and open the [Amazon Lex console](#).
2. On the *Create your bot* page, you can create a custom bot or create a ready-to-use one with a sample template. For the sake of this example procedure, select the **Book Trip** sample.
3. The Create Your Bot page will open. That is where you develop your Amazon Lex bot.



Amazon Lex Create Your Bot page

Step 4: Review the bot workspace

The bot workspace will open. Notice that there are four tabs at the top: Editor, Settings, Channels, and Monitoring. In this exercise, we will be focusing on the **Editor** tab only. You can come back to the other tabs later.



Amazon Lex Editor properties

The Editor tab includes the properties for every intent.

Step 5: Edit intent properties

Intents are actions triggered by keywords entered by your customer. You can think of intents as what customers want to do. For example, the first intent included in our sample Lex bot is "BookCar," which is what the customer wants to do (book a car) and what the bot recognizes it needs to do based on the customer's text input.

Properties

- **Sample utterances** - Phrases (i.e., keywords) that trigger the intent (e.g., "Reserve a car" or "make a car reservation" or "book a car")
- **Lambda initialization and validation** - AWS Lambda Function that validates the customer's input
- **Slots** - Data the customer must provide in order to fulfill the intent action (e.g., slot "PickUpCity" collects customer data about the city in which the car will be picked up)
- **Confirmation prompt** - Questions that confirm the intent actions and prompt the customer to input data (e.g., "Okay, I have you down for a car to be picked up in {PickUpCity}. Should I continue booking the car?")
- **Fulfillment** - AWS's business logic required to fulfill the customer's intent
 - **AWS Lambda Function** - The function used as a code hook for your Amazon Lex bot; this can perform initialization and validation, fulfillment, or both
 - **Return parameters to client** - Sends the intent information to the client application (i.e., Bright Pattern scenario) for intent fulfillment

- **Response** - Message(s) or response cards that close the intent or invoke a different one

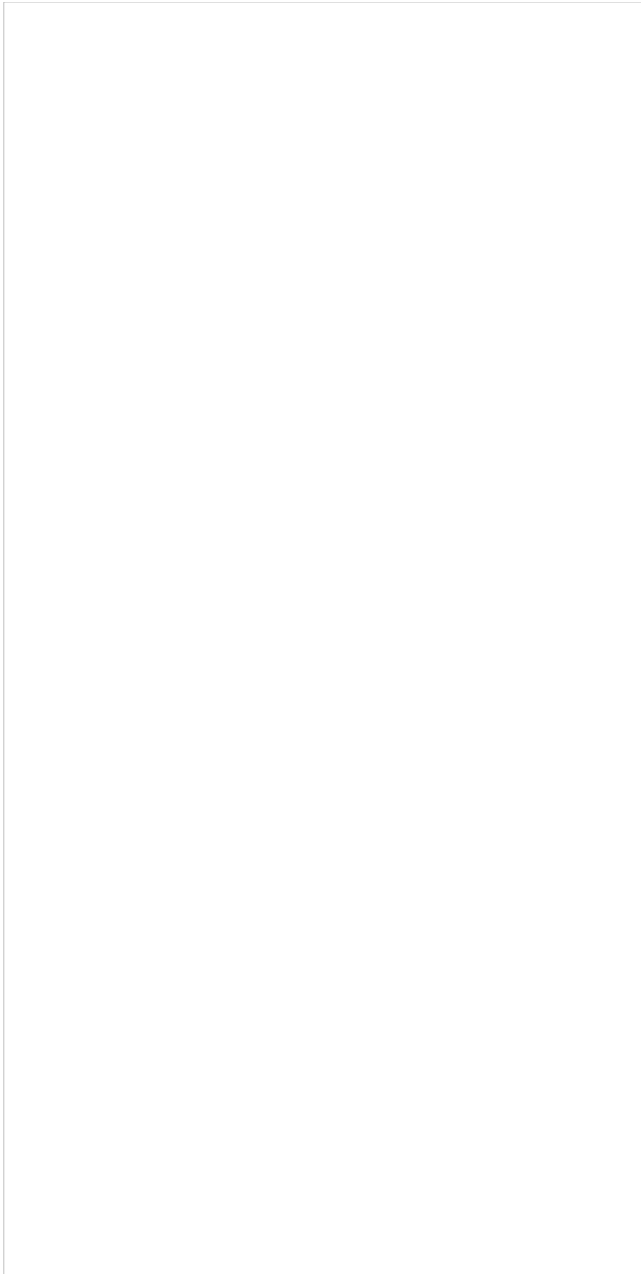
Step 6: Save and build

1. For this example procedure, leave everything on the Editor tab as-is.
2. If you did change something, be sure to click **Save** at the bottom.
3. Click **Build** at the top of the page. This builds the bot with the configured intents.

Step 7: Test it

Once the build is complete, you can test the bot in the chat window.

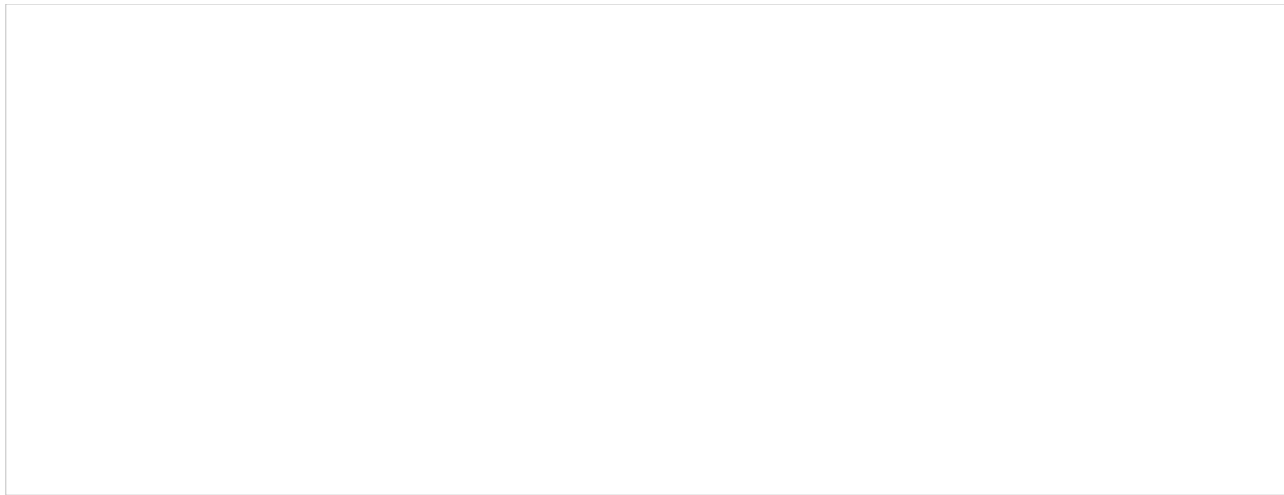
1. On the right side of the screen, click **Test chatbot** to pop out the chat window.
2. Try typing a request such as, "I want to book a trip." See what happens.



Test chat that invokes the BookCar intent

Step 8: Publish your bot

1. At the top of the page, click **Publish**.
2. In the *Publish* dialog that opens, choose or create a new alias (e.g., "TripBooker") for this bot. The *alias* is used to point to the specific version of the bot. Having multiple aliases for the same bot allows you to keep and access different intents and properties for the same bot.
3. Click **Publish** again.



Publish this version (alias) of the Amazon Lex bot

Next Steps

You have now set up a very basic Amazon Lex bot that can be integrated with Bright Pattern Contact Center. You may now:

- Read AWS documentation, explore Amazon Lex intents and slot types, and edit as desired
- Create an [AWS Lex bot/chat suggestions integration account](#)
- Edit a [chat scenario that uses your bot](#)
- [Configure web chat to work with bots](#)

How to Create a Watson Assistant

Bright Pattern Contact Center integrates with providers like IBM Watson to enable chatbots to be used in your contact center services.

In this article, you will learn how to create a basic IBM Watson Assistant that can be used as a conversational bot with your [configured chat service](#). Note that the instructions provided in this article apply to either IBM Watson Assistant or IBM Watson Assistant (Conversation).



Chat showing an integrated Watson Assistant and suggestions for the agent

Procedure

This procedure will walk you through the process of setting up your first Watson Assistant. For a deeper understanding of Watson Assistant and other IBM resources, refer to [IBM's Getting Started tutorial](#) and [IBM's API Reference](#).

Step 1: Create an IBM account

1. If you haven't already done so, [create an IBM account](#). This process creates and activates an IBMid.
2. Sign up for [IBM Cloud](#). IBM Cloud is where you will be developing and managing resources like Watson Assistant bots.

Step 2: Add Watson Assistant as a resource

1. Search IBM's catalog of resources for Watson Assistant.
2. Edit **service name**, **region**, and select a **resource group**.
3. Click **Create Assistant** to add Watson Assistant.
4. The *Assistants* page will open, showing the skills available for your account plan type.

Step 3: Add a dialog skill

Skills are the workspaces where you will be developing your bot (note that IBM skills used to be called workspaces). Skills are what provide [Natural Language Understanding](#) (i.e., sentiment analysis) for your Watson Assistant. Because you are making a conversational bot, you will be building a dialog skill for talking to customers during live chat.

1. Click **Add dialog skill**.
2. For the sake of this example procedure, select **Use sample skill**, and click **Customer Care Sample Skill**, which

is already set up for you to use and edit. Click it again and the *Assistant* page will open. That is where you develop your skill.



Use sample skill

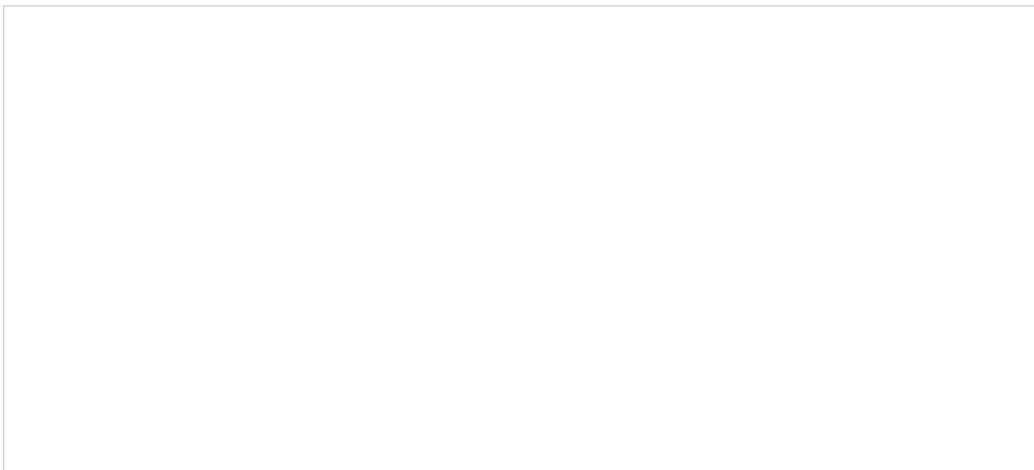
Step 4: Create and/or add intents for your dialog skill

Because you selected the sample skill, the *Assistant* page will show some preconfigured intents. *Intents* are actions triggered by keywords entered by your customer.

In Watson syntax, intents always begin with the hashtag ("#") symbol, followed by word(s) in title case (i.e., where the first letter of a word is capitalized, unless it's a preposition like "to" or an article like "the"). Multiple words are separated by underscores ("_").

Like this: **#Talk_to_Someone**

Get familiar with intents by clicking on the first one in the list. In this example, we clicked on **#Cancel** to review its properties. You can leave all the preconfigured intents as-is or add new user examples.

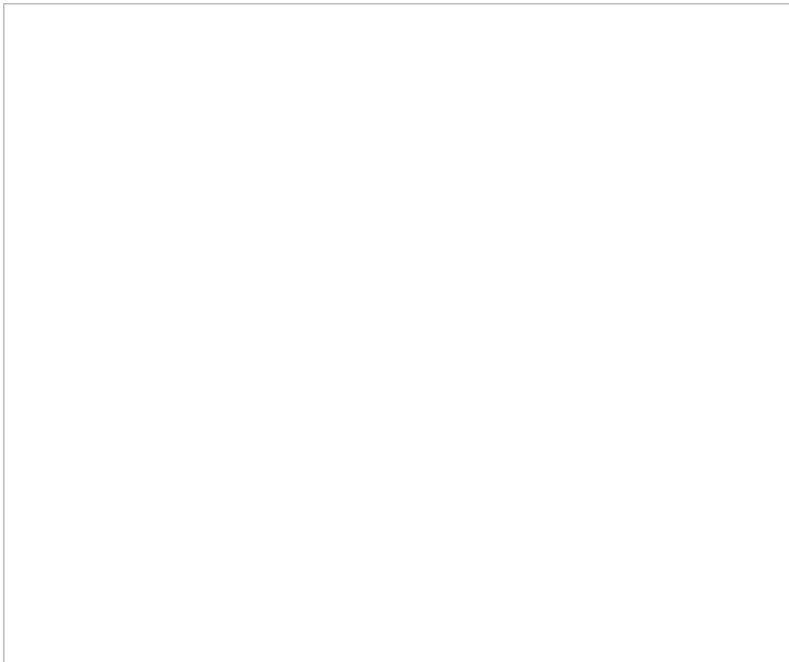


Example of Intent properties

Intent Properties

For every intent, you need to specify:

- **Intent name** - Descriptive name consisting of "#" and text (e.g., "#Cancel")
- **Description** - What this intent does (e.g., "Cancel current request")
- **User examples** - Keywords and phrases that trigger the intent (e.g., anything a customer might type during the chat); add as many examples and variations of spelling/phrases as you can, as the user examples are what make the bot compelling and helpful



User examples

Step 5: Create entities

Click on the *Entities* tab at the top of the *Assistant* page. As with intents, you will see some preconfigured entities. *Entities* are like word sets that more narrowly define the customer text that the bot recognizes.

In Watson syntax, entities always begin with the "@" symbol, followed by word(s) in lowercase. Multiple words are separated by underscores ("_").

Like this: @zip_code

Get familiar with entities by clicking on the first one in the list. In this example, we clicked on @holiday to review its properties. For this example, leave all the preconfigured entities as-is.

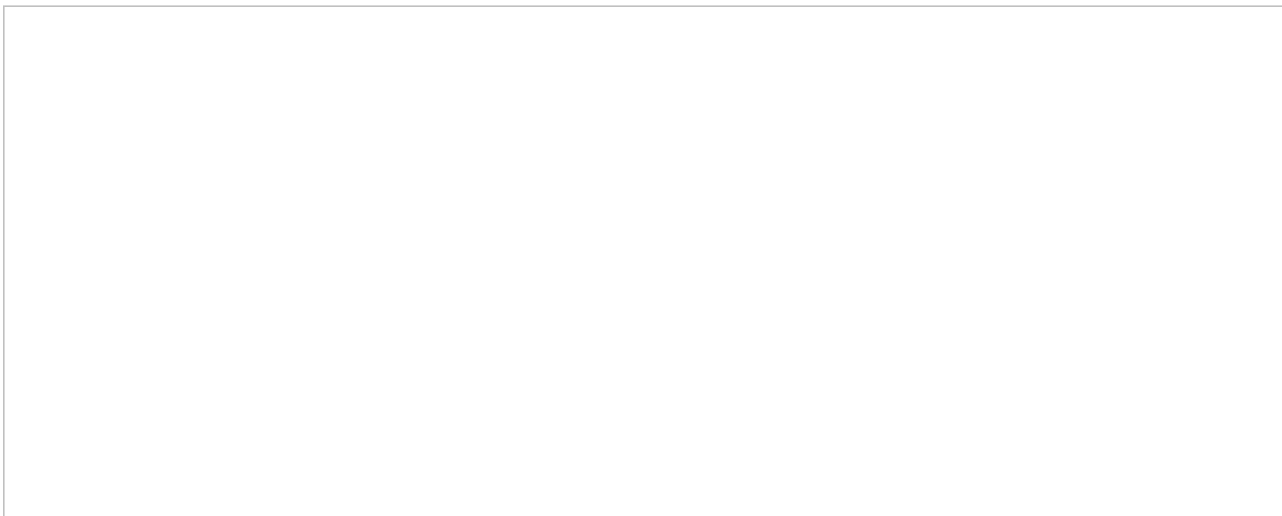


Example of entities

Entity Properties

For every entity, you need to specify:

- **Entity name** - Descriptive name consisting of the "@" symbol and text (e.g., "@holiday")
- **Value name** - A particular word that's included under the umbrella of the entity (e.g., "Hanukkah" and "Christmas" could be values for entity "@holiday")
- **Synonyms** - Related words that could be used in place of the value name (e.g., "turkey day" could be a synonym for "Thanksgiving"); if you don't know any synonyms, IBM can recommend some for you

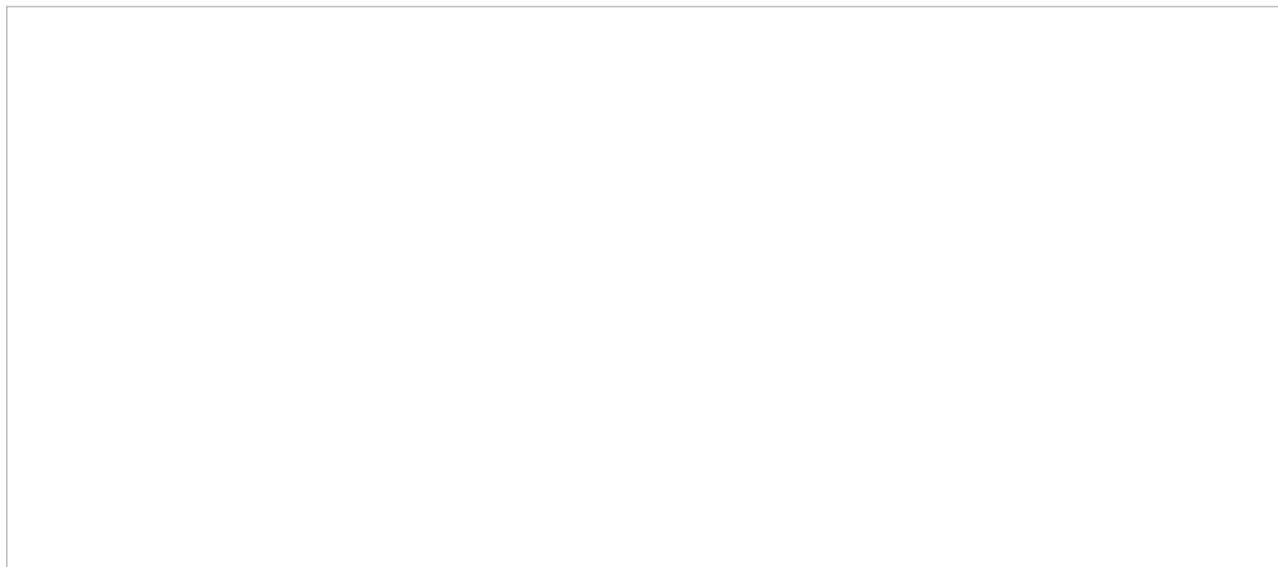


Entity values

Step 6: Design your dialog flow

Click on the *Dialog* tab at the top of the *Skills* page. The dialog for the sample skill you selected in Step 3 will be shown.

A dialog is like a [scenario](#) in that it defines what the bot does in response to a customer's text or actions. When you design your dialog flow, you are telling the Watson Assistant what to do when it recognizes defined intents and entities during an active chat. Branches of a dialog are called *nodes*, and nodes can be organized into *folders*.



Sample dialog

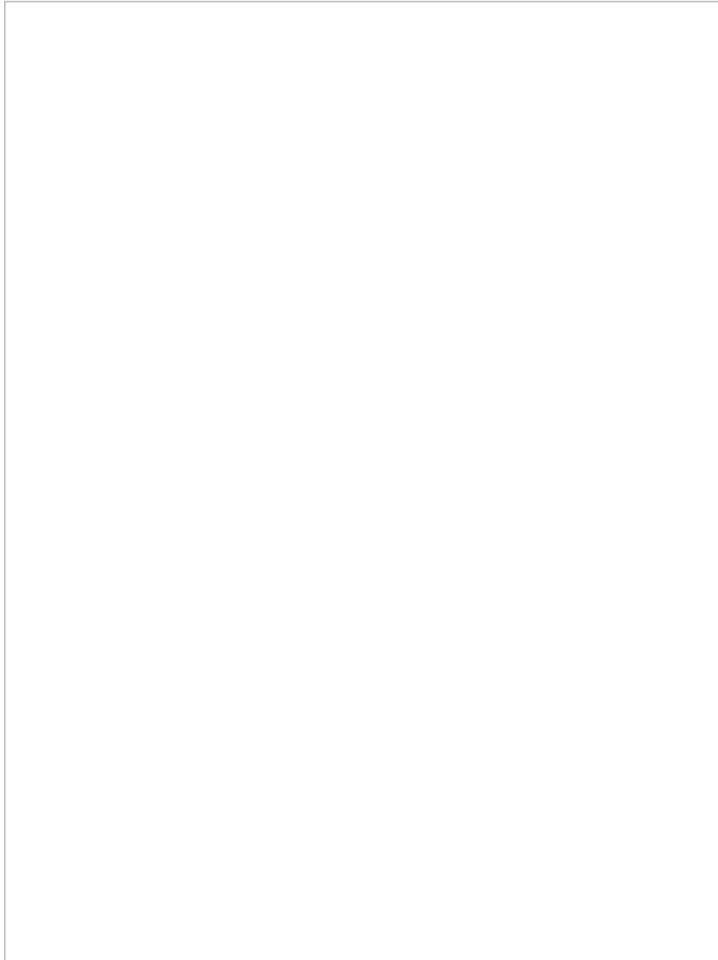
Dialog Properties

In this example, we are going to leave everything in the dialog as-is. Click on the "Hours of Operation" node to open up properties for that node:

- **Name** - The name of the node (you can edit it here)
- **If assistant recognizes** - This is where you specify intents related to the node name (e.g., for this "Hours of Operation" node, you are telling the bot what to do when it recognizes the user examples defined in the "#Customer_Care_Store_Hours" intent)
- **Then respond with** - This is where you specify entities and how the bot should respond when it recognizes them
 - **If assistant recognizes** - Enter entities here (with "@" symbol)
 - **Respond with** - This is the bot's response to the customer; in Bright Pattern's Agent Desktop, anything entered in this field becomes the bot *suggestion*
- **Wait for user input** - Select this to provide a bot response only when the customer types something

Step 7: Try the Dialog

1. Click the **Try it** button at the top of the page. This button launches the Watson Assistant you just built within a chat window.
2. Pretend to be a customer and type some text into the text input field to see how the bot responds. The Try it out tool will show the intents and entities that the bot recognizes from the text you entered. You can click on the intents shown to select other ones or mark them as irrelevant.



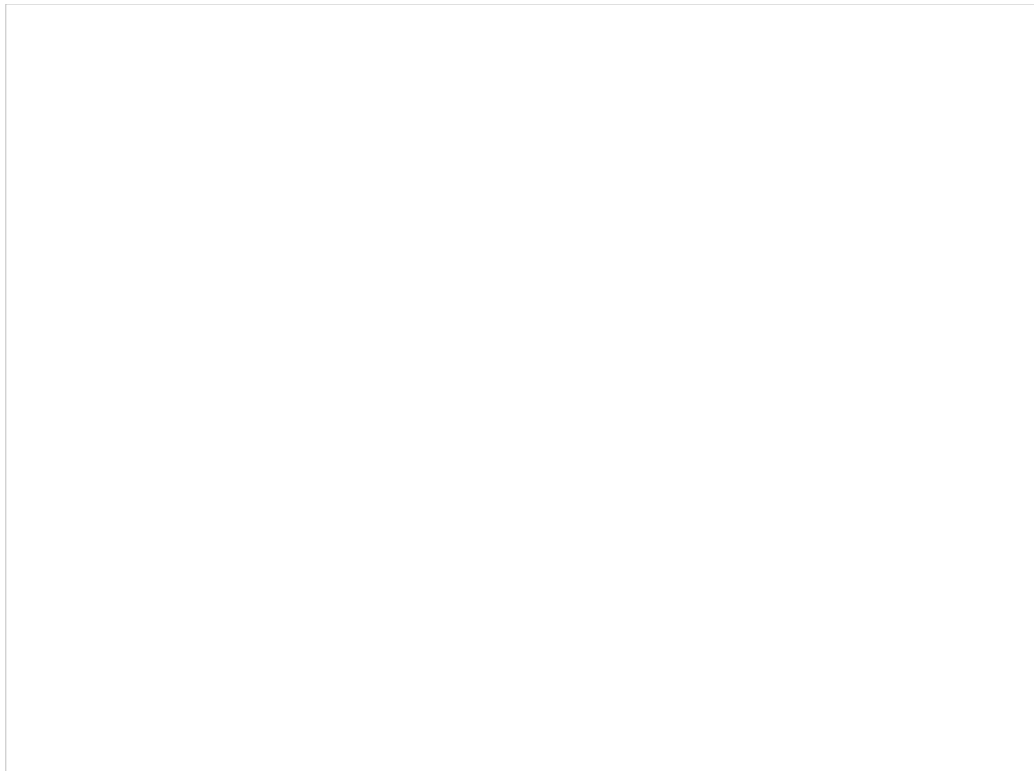
Sample dialog

Please read IBM's documentation to get a deeper understanding of Watson Assistant skills, intents, entities, and dialogs. Refer to [IBM's Getting Started tutorial](#) and [IBM's API Reference](#).

Step 7: Add the Watson Assistant

Now that you're done creating a dialog skill, it's time to assign the skill to your Watson Assistant. This last step is important because later, you may have multiple Watson Assistants and skills, and IBM needs to know which ones are assigned to each other.

1. Click **Add Assistant**.
2. Select the **Use sample skill** tab, which is the skill you just set up in this procedure. If you do not see it listed, select the **Add existing skill** tab and find it there..
3. Click on the desired skill name. That's it.



Add skill

Next Steps

You have now set up a very basic Watson Assistant that can be integrated with Bright Pattern Contact Center. You may now:

- Read IBM's documentation, explore the Watson Assistant, and edit it as desired
- Create a Watson [bot/chat suggestions integration account](#)
- Edit a [chat scenario that uses your Watson Assistant](#)
- [Configure web chat to work with bots](#)

How to Configure a Microsoft Azure Web App Bot from a Template

A Web App Bot is a web service that uses a conversational interface and communicates with the Microsoft Azure Bot Framework Service to send and receive messages and events. You can utilize a Web App Bot in Bright Pattern Contact Center to provide self-service options to customers, bot suggestions to agents, and more in chat interactions.

This application note will guide you through the process of creating a template-based Web App Bot in Azure, setting up a Bot Framework Direct Line channel, and configuring a Bright Pattern Contact Center scenario to send and receive chat messages.

Prerequisites

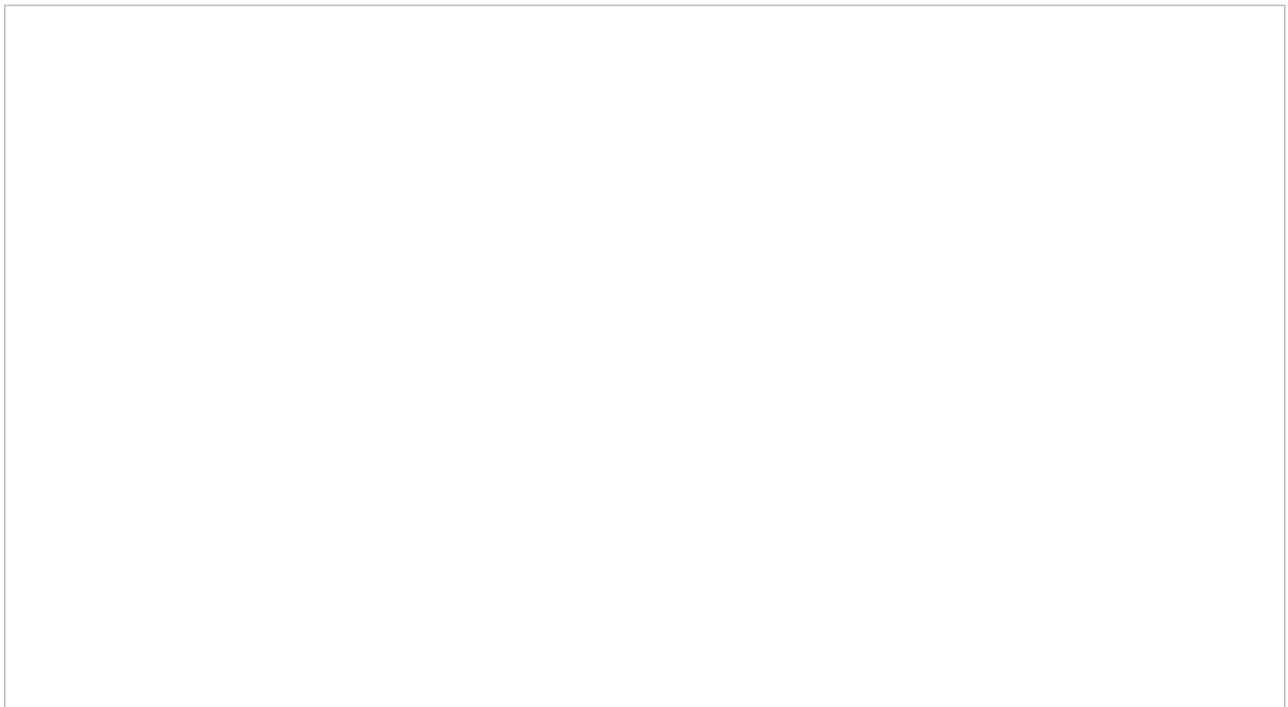
Before you begin, you must have the following:

- Microsoft account
- Active subscription to Azure (free trial OK)
- Admin access to the Bright Pattern Contact Center Administrator application

Procedure

Step 1: Create a Web App Bot in Azure from a Sample Template

1. In the Microsoft Azure Portal, in the resources and services blade menu, click+ **Create a resource**.



Click "+ Create a resource" to get started

2. In the *Azure Marketplace* > *AI + Machine Learning* section, select **Web App Bot**.



Create a "Web App Bot" bot service

3. In Web App Bot properties, fill in the required fields:
 1. **Bot handle** - The name of your bot, which can be changed later if needed. The name should be unique across Azure; you won't be able to continue if the name is already taken.
 2. **Subscription** - The subscription you want to use to pay for the service; you may also use some of your available credits.
 3. **Resource group** - The resource group for the bot to easily manage all resources created in Azure.
 4. **Location** - Choose the nearest location for your customers; this option cannot be changed later.
 5. **App name** - The name of your app, which also should be unique across Azure because a subdomain will be created to access the Web App Bot.
 6. **Bot template** - Choose **Echo Bot**.

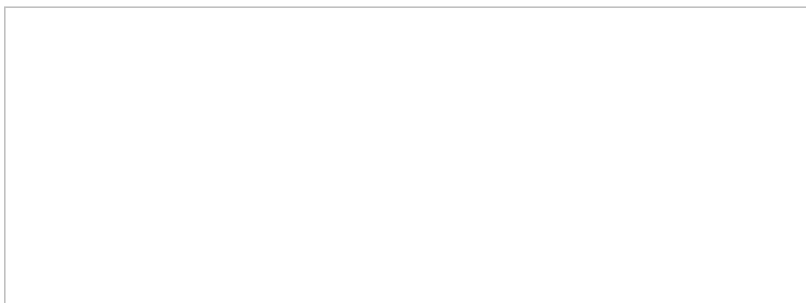


Web App Bot properties

Step 2: Configure Direct Line channel

The Bot Framework Direct Line channel lets you integrate your bot into your mobile app, webpage, or other application. In this step, we will set up a Direct Line channel and grab the secret key, which is needed for integration with Bright Pattern.

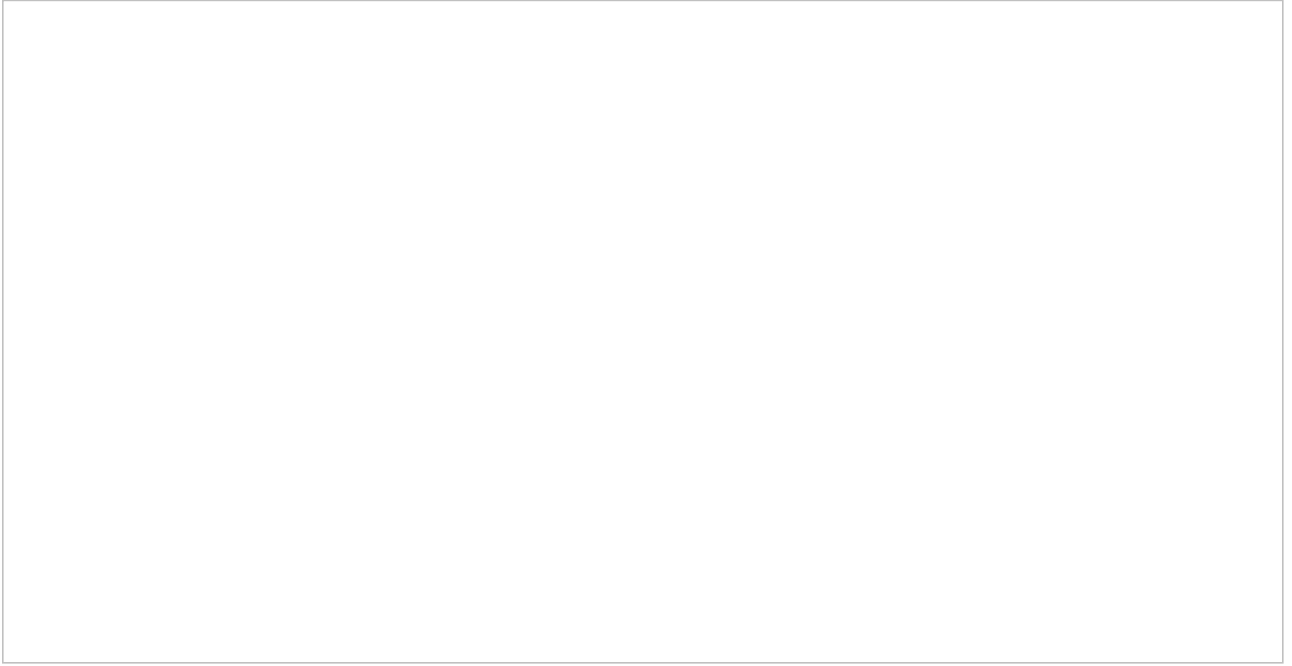
1. Go to *Home > Resource groups*, and select the group that you created or select for the bot.



Select your group

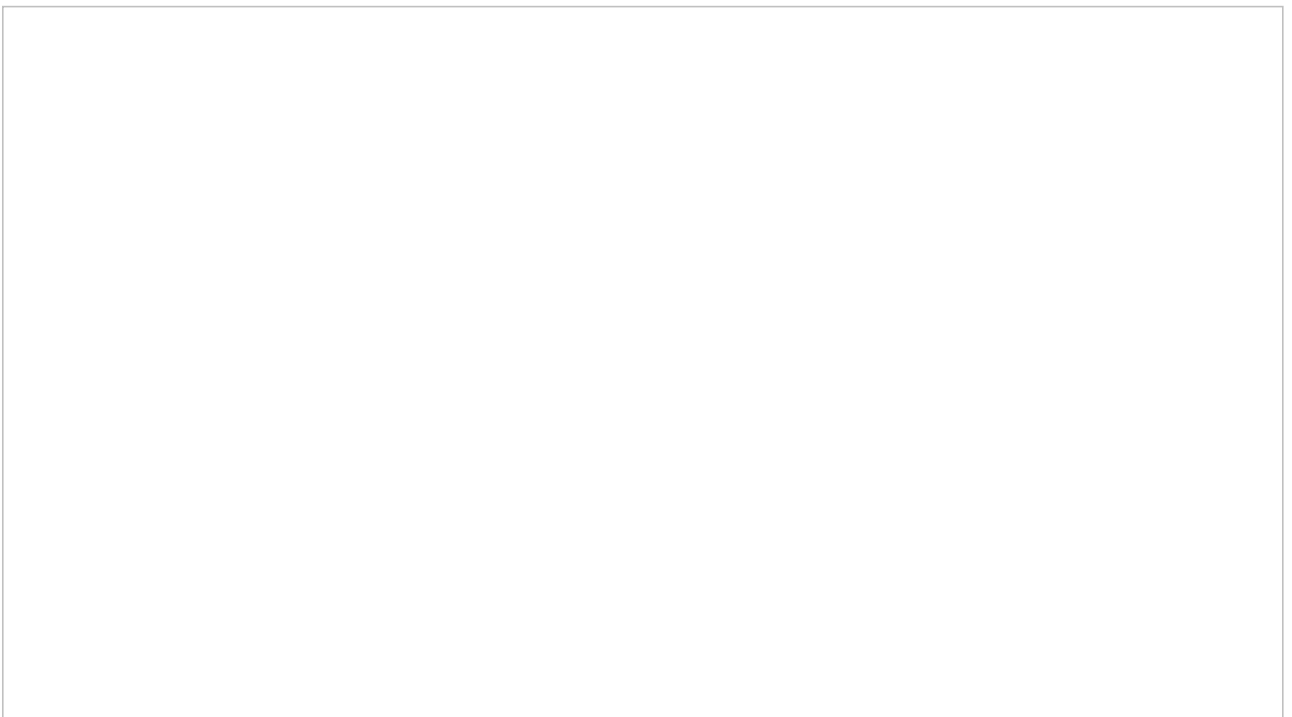
2. Now, within the bot group:

1. Select the app service you created.
2. Go to *Bot management* > *Channels*, and click **Configure Direct Line channel**.



Click "Configure Direct Line channel"

3. Copy a **Secret key**. The Bot Framework generates secret keys that your client application can use to authenticate the Direct Line API requests that it issues to communicate with your bot.



Copy a secret key

This completes Web App Bot configuration.

Next step: Edit Bright Pattern Contact Center scenario

Now that you have completed your bot configuration, you can use the bot in a Bright Pattern Contact Center chat scenario. See [How to Configure a Chat Scenario That Uses a Microsoft Azure Web App Bot](#) for detailed instructions.

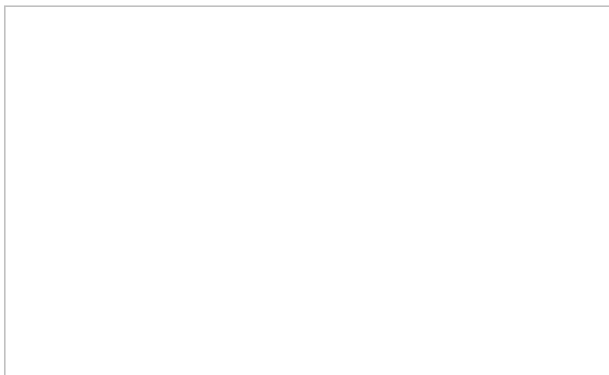
How to Add a Bot/Chat Suggestions Engine Integration Account

Bot/chat suggestions engine integration accounts allow you to use third-party bots to automate chat conversations, provide self-service options, intelligently route customers to agents, and provide meaningful suggestions to an agent during active chat sessions.

Bright Pattern Contact Center supports the following types of bot integration: AWS Lex and Watson Assistant.

Adding a New Account

1. In the Contact Center Administrator application, go to *Configuration > Call Center Configuration > Integrations Accounts*.
2. Click the **Add account (+)** button to add a new integration account.



Add new integration account

3. Select account type **Bot / Chat suggestions engine** and click **OK**.



Select "Bot / Chat suggestions engine"

4. In the *Bot / Chat suggestions engine type* dialog, select your desired type of bot instance.



5. Click **OK**. The properties for that engine will open next.

Next Steps

Now that you have a bot/chat suggestions engine integration account, you can edit its properties.

Learn more at:

- [Set up a Watson Assistant Integration Account](#)

Set up an AWS Lex Integration Account

Integrations with AWS Lex are enabled through [integration accounts](#), which store the credentials of third-party services so that Bright Pattern Contact Center can access and work with them.

In this article, you will learn how to set up your AWS Lex bot/chat suggestions engine integration account and edit its properties.

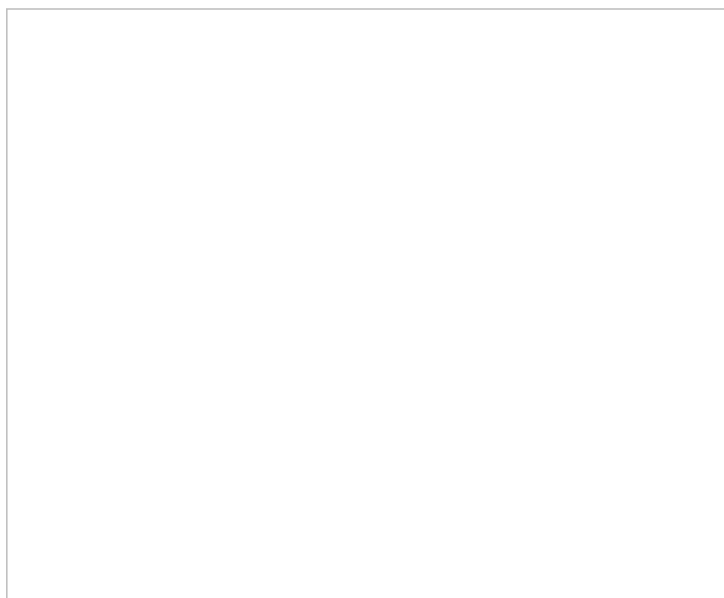
Procedure

Step 1: Add integration account

[Add a bot/chat suggestions engine integration account](#), and select type **AWS Lex**.

Step 2: Edit properties

In the Properties dialog, enter the credentials of your AWS Lex bot instance as follows. This allows Bright Pattern to access your bot and use it in chat interactions.



AWS Lex bot/chat suggestions engine integration account properties

Name

The unique name of this integration account (any). Because you can have multiple integration accounts of the same type, it is helpful to have a descriptive, memorable name.

Type

By default, the type is "AWS Lex" because you selected this type when adding the account.

User ID

Your AWS Account ID.

Find it in AWS by going to *My Account > Account Settings*.

Bot name

The bot name (e.g., "TripBooker"); note this may be different than the bot alias (see below).

Find it in AWS by going to *Amazon Lex > Bots (select the name of your bot) > Settings*.

Bot alias

The alias name (if any).

It's possible to save multiple versions of your bot, each with different intents and configurations. *Abot alias* is the name of the version of the bot.

Access key

The access key ID (e.g., AKIAIOSFODNN7EXAMPLE).

You need to set up an access key for AWS Identity and Access Management (IAM) service to get this. See [Managing Access Keys for IAM Users](#) and [AWS Management Console](#).

Secret key

The secret access key (e.g., wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY)

Region

The region for your bot instance (e.g., "us-east-1")

Max API calls per day

The maximum number of calls that can be done each day. This limit is here to keep your account from being charged for additional calls beyond what is included in a free account.

Step 3: Save properties

Click **Apply** at the bottom of the screen to save your account properties. Your AWS Lex integration configuration is now complete.

Recommended Reading

For more information on bots, see:

- [How to Create an Amazon Lex Bot](#)
- [Bot/Chat Suggestions Engine](#)
- [How to Add a Bot/Chat Suggestions Engine](#)
- [How to Integrate Bots with Chat](#)

Set up a Watson Assistant (Conversation) Integration Account

Integrations with IBM Watson Assistant (formerly Conversation) are enabled through [integration accounts](#), which store the credentials of third-party services so that Bright Pattern Contact Center can access and work with them.

Bright Pattern integrates with both IBM Watson Assistant (Conversation) (legacy version) and IBM Watson Assistant (latest version).

In this article, you will learn how to set up your Watson Assistant (Conversation) integration account and edit its properties.

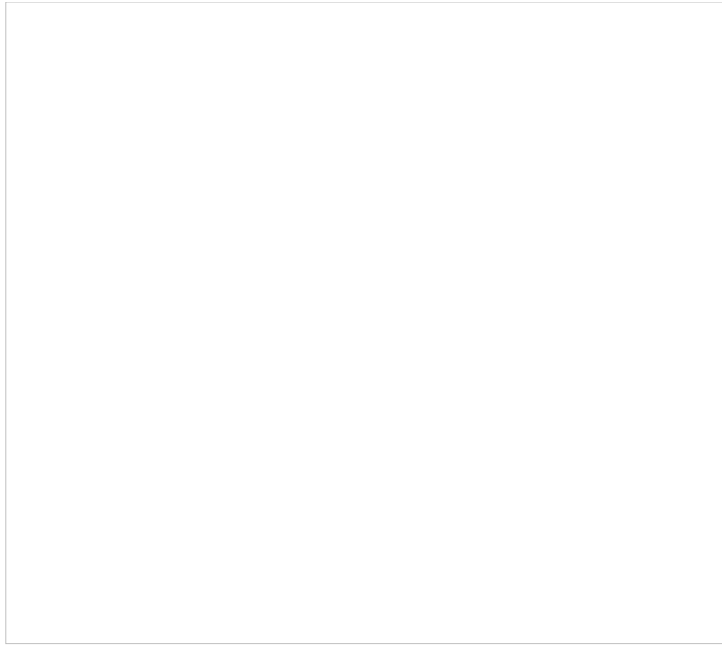
Procedure

Step 1: Add integration account

[Add a bot/chat suggestions engine integration account](#), and select type **Watson Assistant (Conversation)**.

Step 2: Edit properties

In the Properties dialog, enter the credentials of your IBM Watson Assistant (Conversation) bot instance as follows. This allows Bright Pattern to access your bot and use it in chat interactions.



Watson Assistant (Conversation) bot/chat suggestions engine integration account properties

Name

The unique name of this integration account (any).

Type

By default, the type is “Watson Assistant (Conversation)” because you selected this type when adding the account.

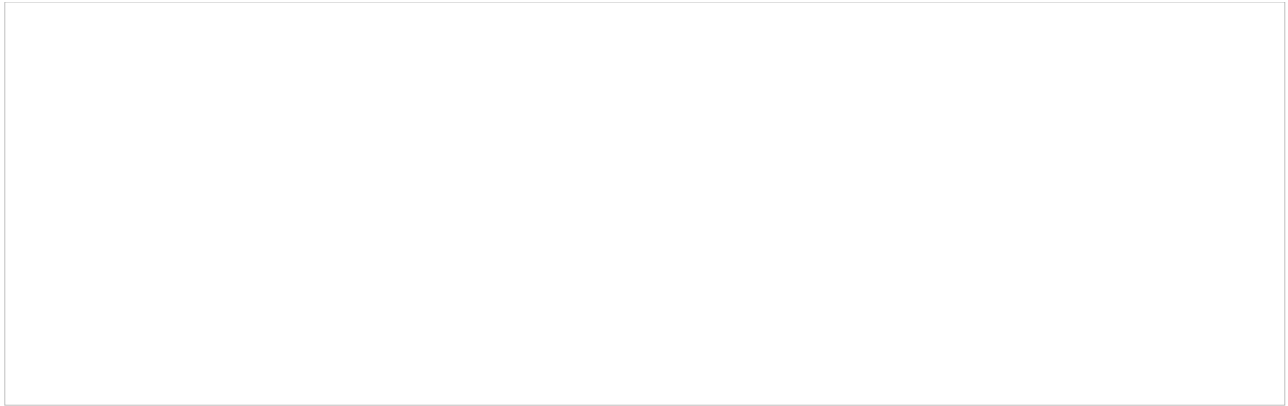
Url

The endpoint of your Watson Assistant Skill in the following format: <Legacy v1 Workspace URL>?version=2017-05-26

For example:

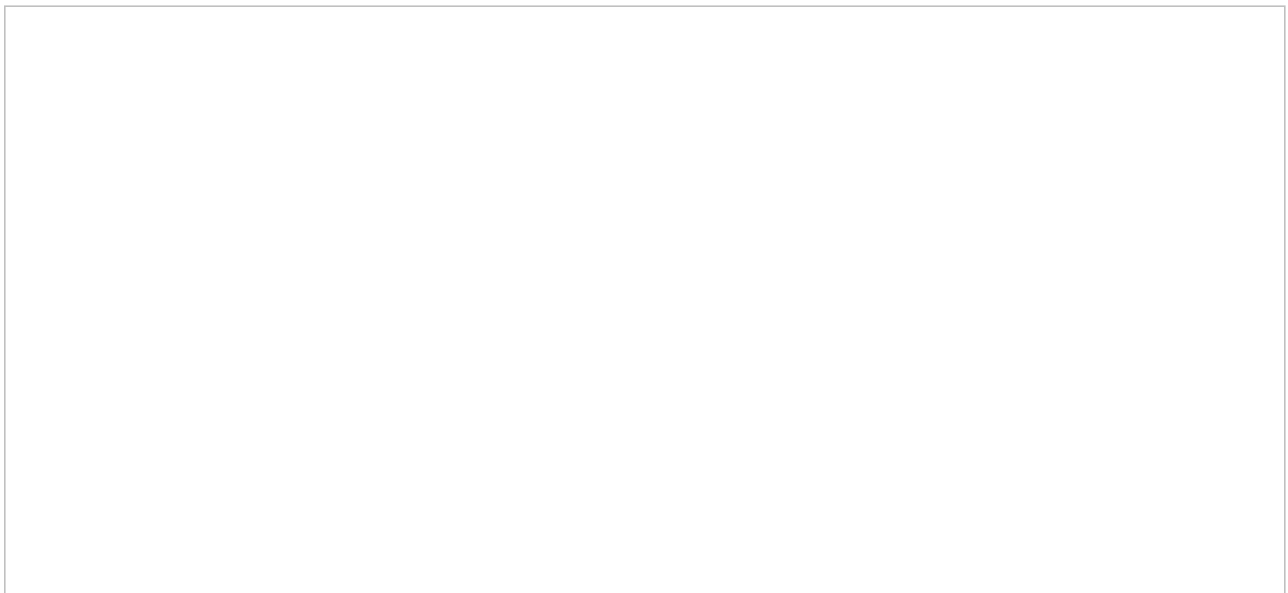
<https://gateway.watsonplatform.net/assistant/api/v1/workspaces/12a3ab45-b12b-1234-12a3-12d34aebe56c/message?version=2017-05-26>

1. To find the Legacy v1 Workspace URL, go to *IBM Watson Assistant (Conversation) > Skills*, click on the desired skill's **Actions** button, and select **View API details**.



IBM Watson Assistant > Skills

2. On the page that opens, copy your **Legacy v1 Workspace URL**.



Copy your Legacy v1 Workspace URL

Workspace ID

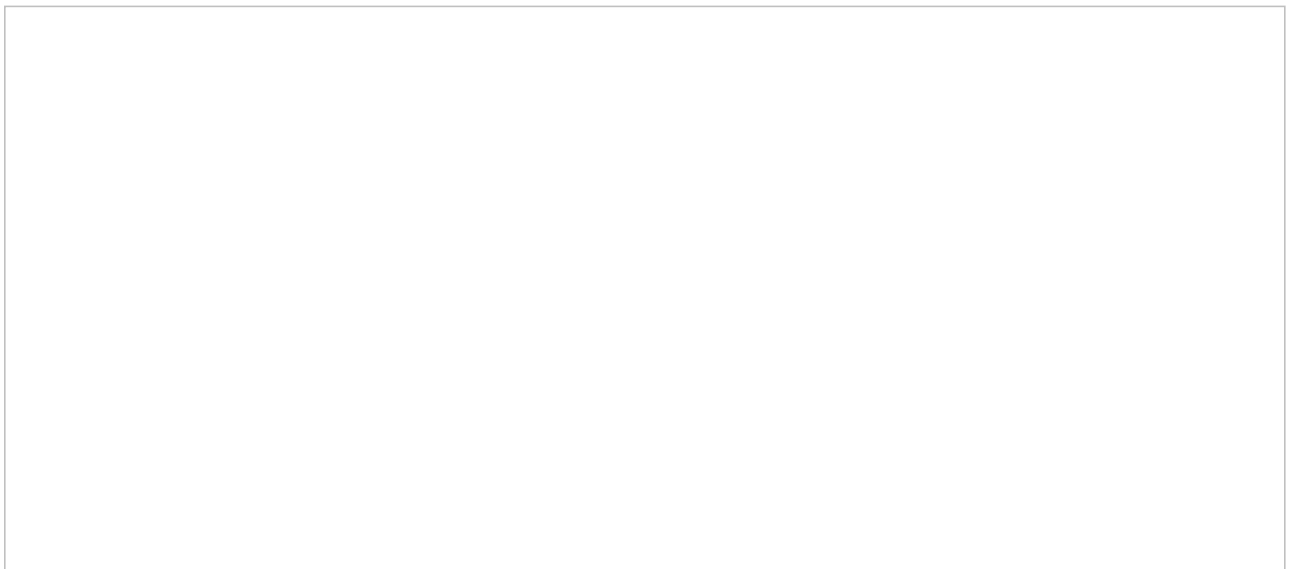
The identifying number (string) of your Watson Assistant (Conversation) Skill (i.e., instance). Note that in IBM Watson, skills used to be called workspaces.

1. To find this ID, go to *IBM Watson Assistant > Skills*, click on the desired skill's **Actions** button, and select **View API details**.



IBM Watson Assistant > Skills

2. On the page that opens, copy your **workspace ID**.



Copy your workspace ID

Username

The *username* (string) that is used to authenticate the Watson Conversation API. The username is provided in the service credentials for the service instance that you want to use.

Password

The *password* (string) used to authenticate the Watson Conversation API. The password is provided in the service credentials for the service instance that you want to use.

Max API calls per day

The maximum number of calls that can be done each day. This limit is here to keep your account from being charged for additional calls beyond the first 1,000 that are included in a free account.

Maximum suggestions

The maximum number of suggestions (e.g., 3) that can be delivered to the Agent Desktop during active chat interactions. Suggestions are the bot-generated replies that agents can select and use during chats.

Test Connection

Click to be sure Bright Pattern Contact Center can connect to your Watson Assistant (Conversation).

If the credentials are invalid, go back to the Url property and make sure you've entered it exactly as explained. Test until you see the success dialog: "Account credentials appear to be valid."

Step 3: Save properties

Click **Apply** at the bottom of the screen to save your account properties. Your Watson Assistant (Conversation) integration configuration is now complete.

Recommended Reading

For more information on Watson Assistant, see:

- [How to Create a Watson Assistant](#)
- [How to Integrate Bots with Chat](#)

Set up a Watson Assistant Integration Account

Integrations with IBM Watson Assistant are enabled through [integration accounts](#), which store the credentials of third-party services so that Bright Pattern Contact Center can access and work with them.

A Note About the Watson Name

Bright Pattern integrates with both IBM Watson Assistant--the latest version--and IBM Watson Assistant (Conversation)--the legacy version.

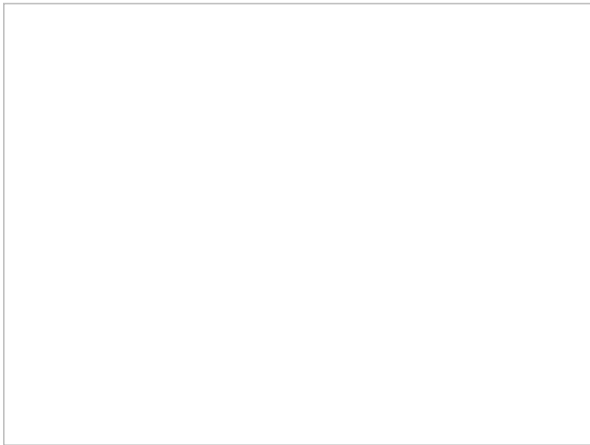
Previously, Watson Assistant was called Watson Conversation, and credentials for Watson Conversation included username and password. IBM has since renamed it Watson Assistant and has changed replaced username and password credentials with API key. For backwards compatibility, our integration accounts still work with either type of bot.

In this article, you will learn how to set up your Watson Assistant integration account and edit its properties.

Procedure

Step 1: Add integration account

[Add a bot/chat suggestions engine integration account](#), and select type **Watson Assistant**.



Bot/chat suggestions engine integration
account types

Step 2: Edit properties

In the Properties dialog, enter the credentials of your IBM Watson Assistant bot instance as follows. This allows Bright Pattern to access your bot and use it in chat interactions.



Watson Assistant bot/chat suggestions engine integration account properties

Name

The unique name of this integration account (any).

Type

By default, the type is “Watson Assistant” because you selected this type when adding the account.

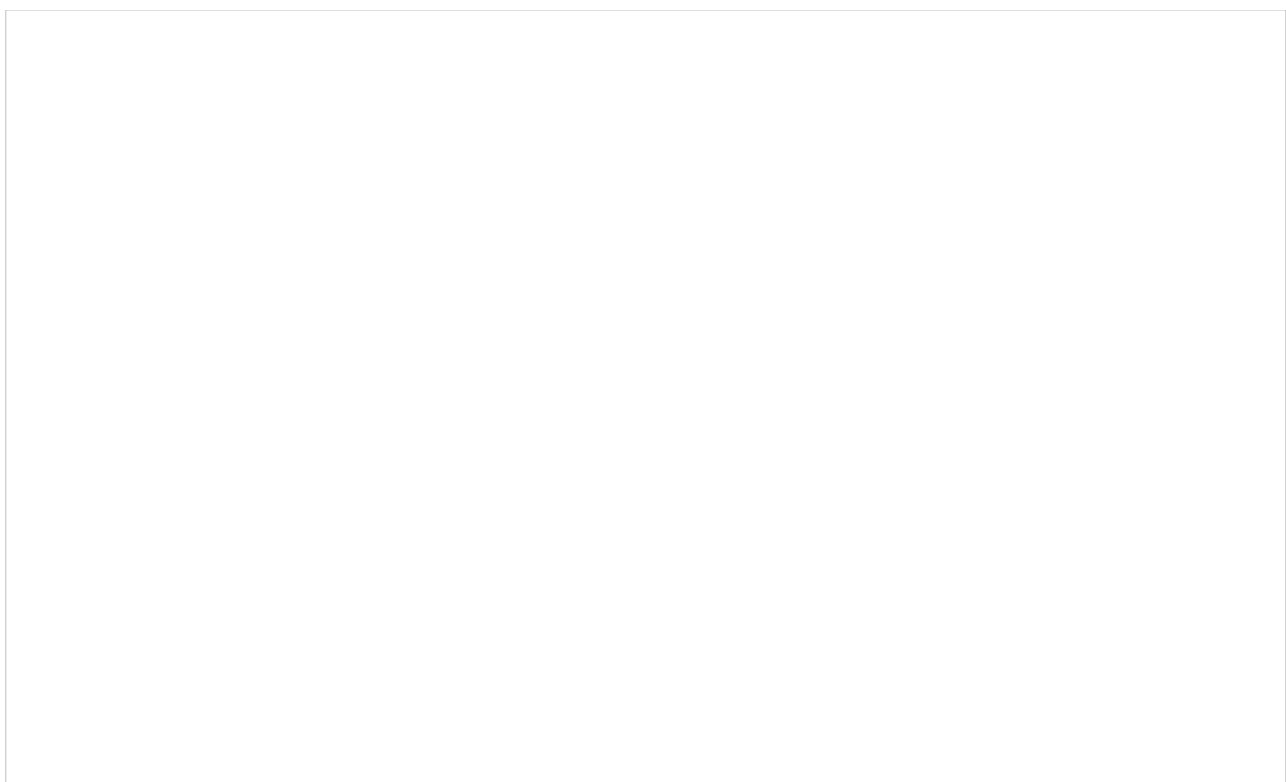
Url

The endpoint of your Watson Assistant Skill in the following format: **<Legacy v1 Workspace URL>?version=2017-05-26**

For example:

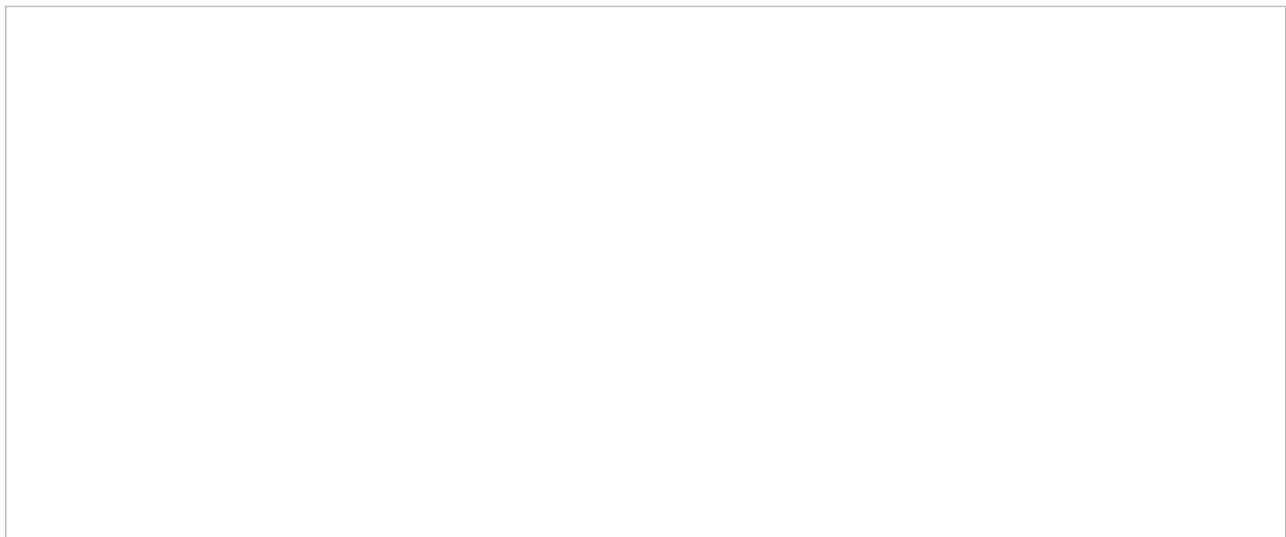
<https://gateway.watsonplatform.net/assistant/api/v1/workspaces/12a3ab45-b12b-1234-12a3-12d34aeb56c/message?version=2017-05-26>

1. You can find the URL by going to *Skills > your skill > Options > View API Details*.



IBM> Watson Skill > Options > View API Details

2. From the *Skill Details* page that opens, copy the **Legacy v1 Workspace URL**.



Copy your Legacy v1 Workspace URL

3. Then append **?version 2017-05-26** to the end of the URL. For example:

<https://gateway.watsonplatform.net/assistant/api/v1/workspaces/46a4ab18-b13b-4306-83a0-57d83aeb56c/message?version=2017-05-26>

Why do this?

Our integration accounts support requests to the V1 runtime API using the /message method, where user input is sent to Watson Assistant to get a response. We also support requests to the V2 runtime API using the /message method.

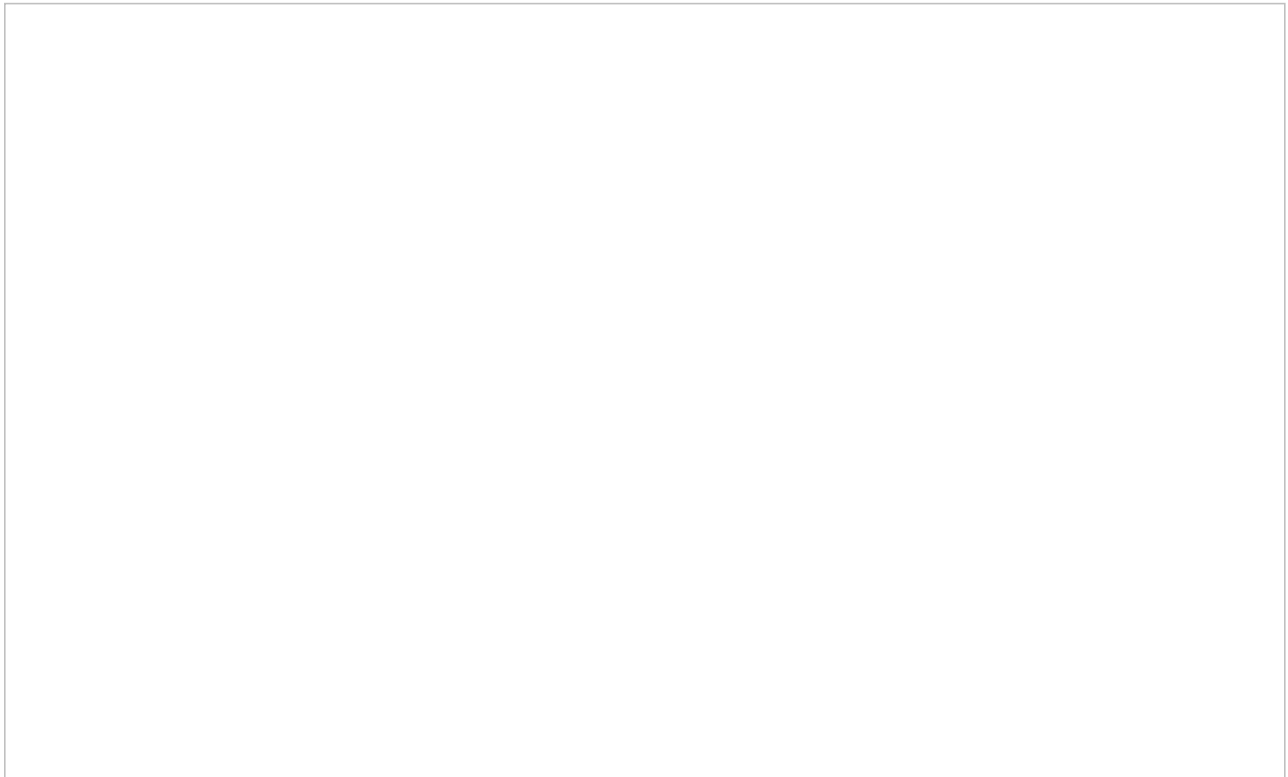
API requests require a version parameter that takes a date in the format `version=YYYY-MM-DD`. Our integration uses `/message version=2017-05-26`. Note that although the date of this version may not match the current version specified by IBM Cloud, this is the correct version to use for our integration because it is compatible with Bright Pattern applications.

When the version parameter is appended to the end of the URL property, the proper version will accompany every API request, every time the integrated Watson Assistant is used in Bright Pattern chat services.

Workspace ID

The identifying number (string) of your Watson Assistant (Conversation) Skill (i.e., instance). Note that in IBM Watson, skills used to be called workspaces.

1. To find this ID, go to *Skills > your skill > Options > View API Details*.



IBM> Watson Skill > Options > View API Details

2. From the *Skill Details* page that opens, copy the **Workspace ID**.

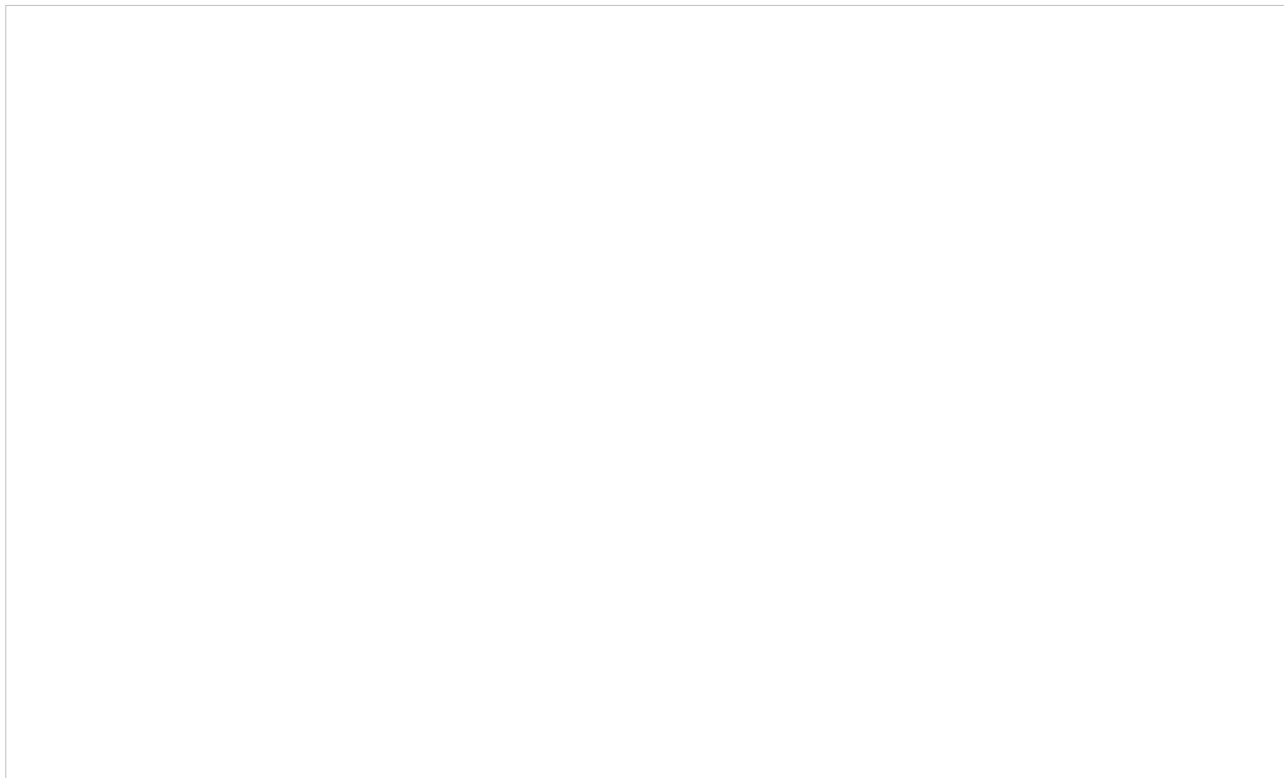


Copy your workspace ID

API Key

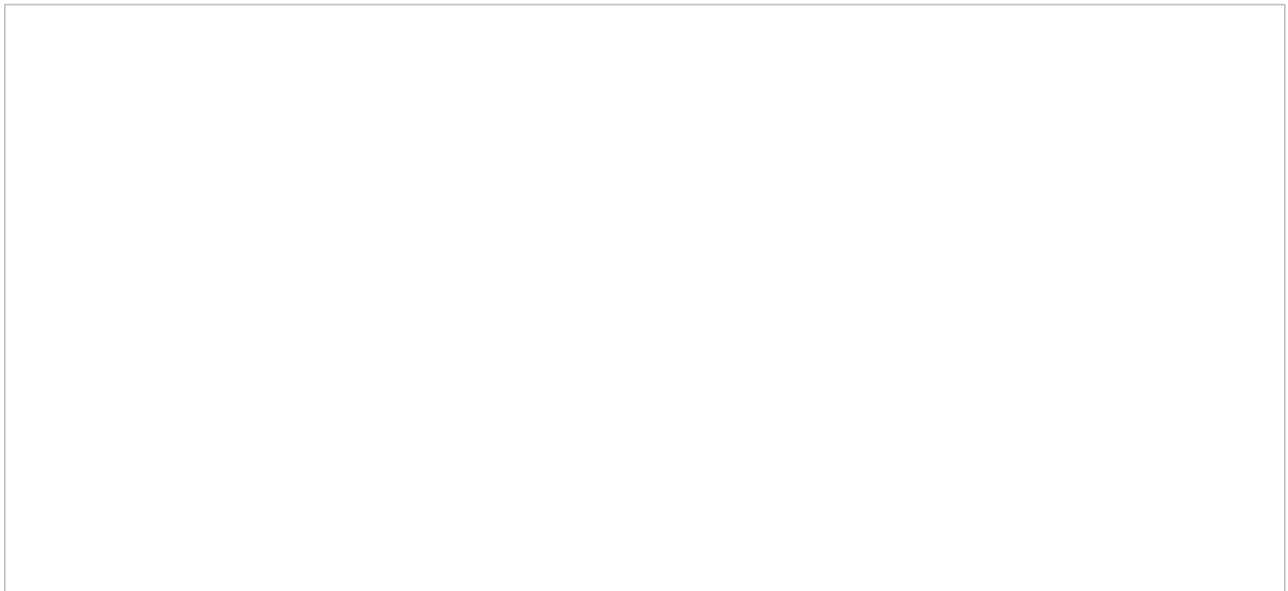
The API Key (string) used to authenticate the service (e.g., "EEE-MTX13ZS1Ta4pD8qO4rmGP_SXR7HPB8IAAnPgTwKV").

You can find the API Key by going to *Skills > your skill > Options > View API Details*.



IBM> Watson Skill > Options > View API Details

1. From the *Skill Details* page that opens, copy the **Api Key**.

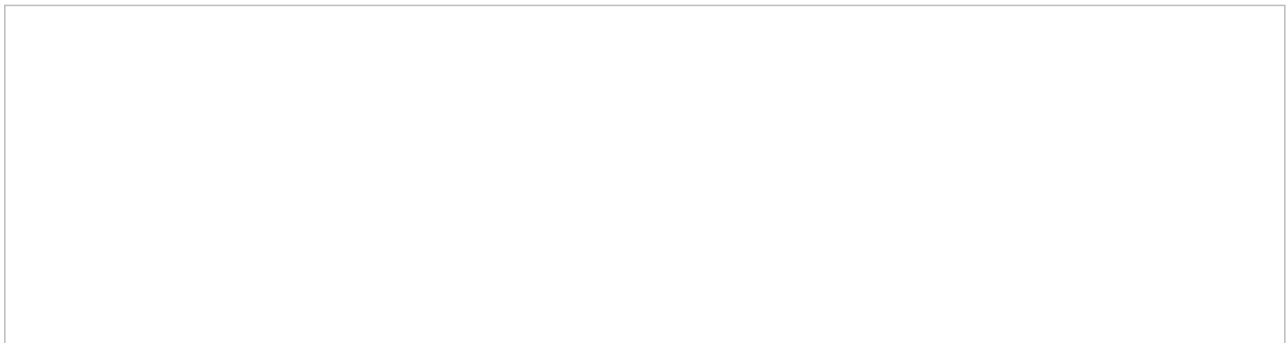


Copy your API key

The API key is either auto-generated for you or created manually in IBM Cloud.

How to Create a New Key

1. Go to IBM's website and navigate to *IBM Cloud > Access (IAM) > IBM Cloud API Keys*.
2. Click **Create an IBM Cloud API key**.



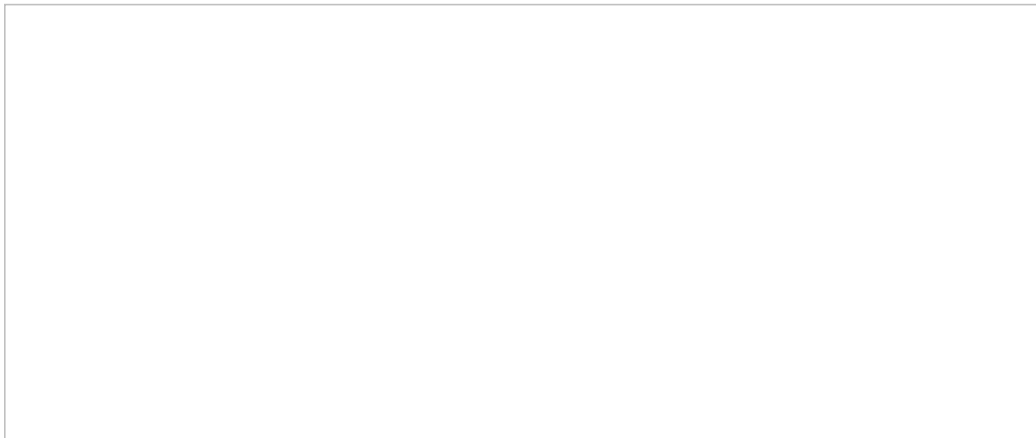
IBM Cloud API Keys

3. In **Create API Key** dialog, add name and description.



Create API Key

4. Download it and save it because the API key will be shown only once.



Create API Key

Max API calls per day

The maximum number of calls that can be done each day. This limit is here to keep your account from being charged for additional calls beyond the first 1,000 that are included in a free account.

Maximum suggestions

The maximum number of suggestions (e.g., 3) that can be delivered to the Agent Desktop during active chat interactions. Suggestions are the bot-generated replies that agents can select and use during chats.

Test Connection

Click to be sure Bright Pattern Contact Center can connect to your Watson Assistant.

If the credentials are invalid, go back to the Url property and make sure you've entered it exactly as explained. Test until you see the success dialog: "Account credentials appear to be valid."

Step 3: Save properties

Click **Apply** at the bottom of the screen to save your account properties. Your Watson Assistant integration configuration is now complete.

Recommended Reading

For more information on Watson Assistant, see:

- [How to Create a Watson Assistant](#)
- [How to Integrate Bots with Chat](#)

How to Integrate Bots with Chat

Bright Pattern Contact Center integrates with providers like IBM Watson to enable chatbots to be used in your contact center services. In this article, you will learn how to integrate an IBM Watson Assistant (i.e., a conversational bot) with your configured chat service.

In [How to Configure Web Chat](#), you learned how to set up a chat scenario, chat service, and chat scenario entry point to work together to display a chat widget on your website. A working chat widget connects a customer on a website to a live agent in your contact center.

When a bot is integrated with your chat services, customers can be connected to the bot, which can answer questions, provide assisted self-service, and connect to a live agent if needed. Even if the chat is routed to an agent, the integrated bot still runs in the background, ready to assist the agent by providing suggestions (i.e., suggested responses for the agent to select and use).

Prerequisites

This article assumes that you have:

- Read and understand [How to Configure Web Chat](#)
- Access to your company's IBM Cloud account
- Some understanding of how Watson Assistant works

Procedure

This procedure consists of the following concepts:

- Create a bot in IBM Watson

- Add an IBM Watson integration account
- Create or choose a chat scenario that incorporates bot interactions
- Create or choose a chat service for your bot
- Link the chat scenario and service to your integration account
- Test the chat

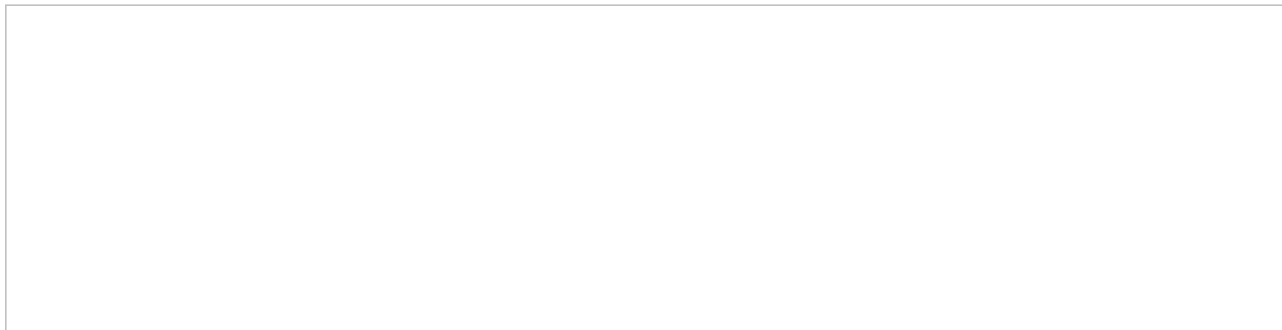
Step 1: Create a Watson Assistant in IBM Cloud

1. If you haven't already done so, sign up for an IBM account and launch IBM Cloud. IBM Cloud is the application where you will be developing and managing resources like Watson Assistant chatbots.
2. Follow all the steps in [IBM's Getting Started tutorial](#) in order to create a Watson Assistant instance. The Watson Assistant will be your bot, and you will be adding intents, entities, and a dialog to it in a workspace. For more information on Watson Assistant, see [IBM's API Reference](#).
3. You can also read our basic guidelines in [How to Create a Watson Assistant](#).

Step 2: Get Your Watson Credentials

Your credentials are necessary for integrating Bright Pattern Contact Center with your Watson Assistant. You will be using these credentials in later steps to add an integration account.

1. In your IBM Cloud console, get your Watson Assistant credentials from either of these two places:
 1. *Watson Services > Assistant > Manage* (for credentials in plain text)
 2. *Watson Services > Assistant > Service Credentials* (for JSON snippet)
2. Copy the **Url**, **Username**, and **Password** for your Watson Assistant to a separate doc, such as a text file.

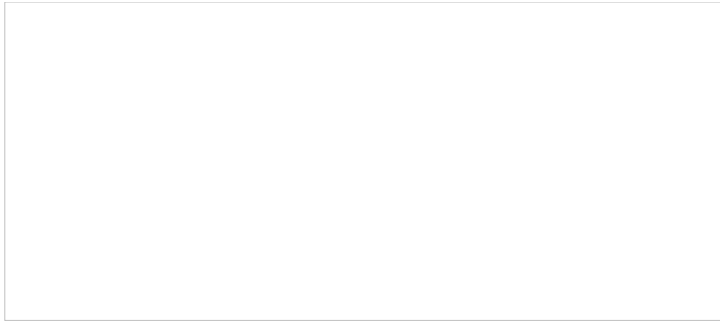


Watson Assistant credentials

Step 3: Add an Integration Account

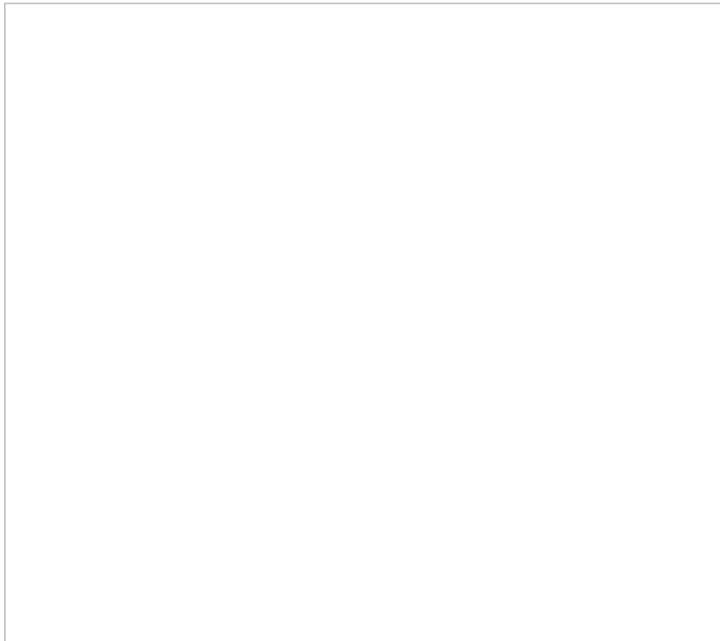
Integration accounts are what allow your contact center to operate with third-party services.

1. In the Contact Center Administrator application, go to *Call Center Configuration > Integrations Accounts*.
2. Click the **Add account (+)** button to add a new integration account.



Add new integration account

3. Select account type **Bot / Chat suggestions engine** and click **OK**.



Select "Bot / Chat suggestions engine"

4. In the *Bot / Chat suggestions engine type* dialog, select your Watson Assistant type and click **OK**.



Properties



Example of Watson Assistant integration account properties

Fill in all [Watson Assistant properties](#) or [Watson Assistant \(Legacy\) properties](#).

Click **Apply** at the bottom of the screen to save your account properties.

Step 4: Create or Select a Chat Scenario

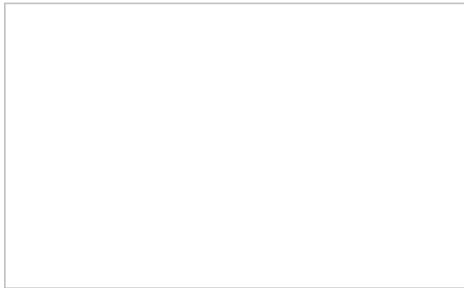
Note that this step is the same as [Web Chat Configuration Step 1](#).

1. Go to *Configuration > Scenarios > [Chat](#)*.



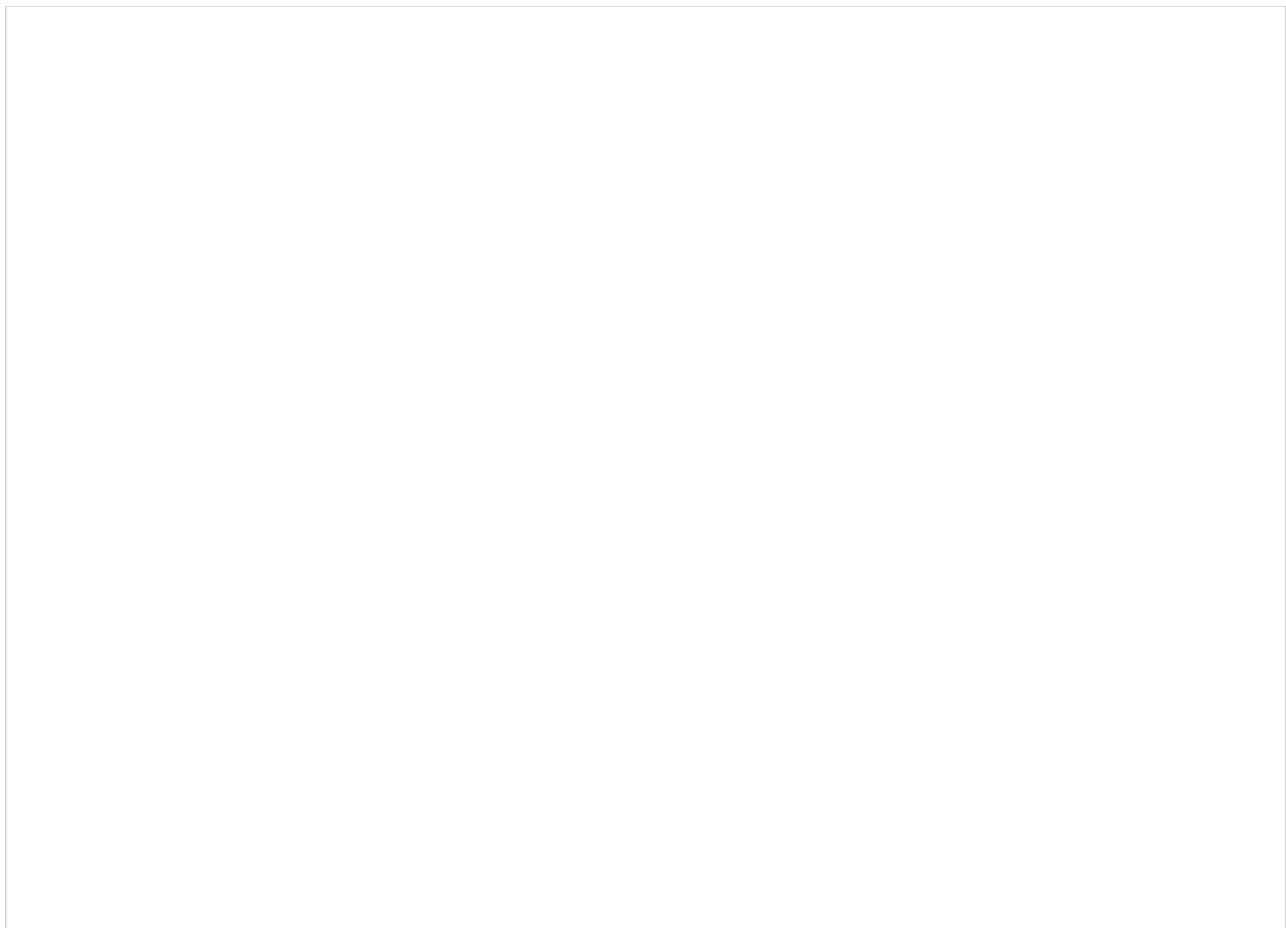
Configuration > Scenarios > Chat

2. Either select an existing Chat Scenario from the list, or click the **Add from template** button at the bottom of the screen to create a new chat scenario from the "Mobile Chat" template.



Select the "Mobile Chat" template

3. Creating a new chat scenario from a template will open the Scenario Builder application in a new browser tab or window. For the purpose of this simple setup, leave the scenario as-is and click **Save**.



Name the scenario

4. Give the scenario a unique name (e.g., "Bot Scenario") and click **Save**. Your new scenario will appear in the list of scenarios.

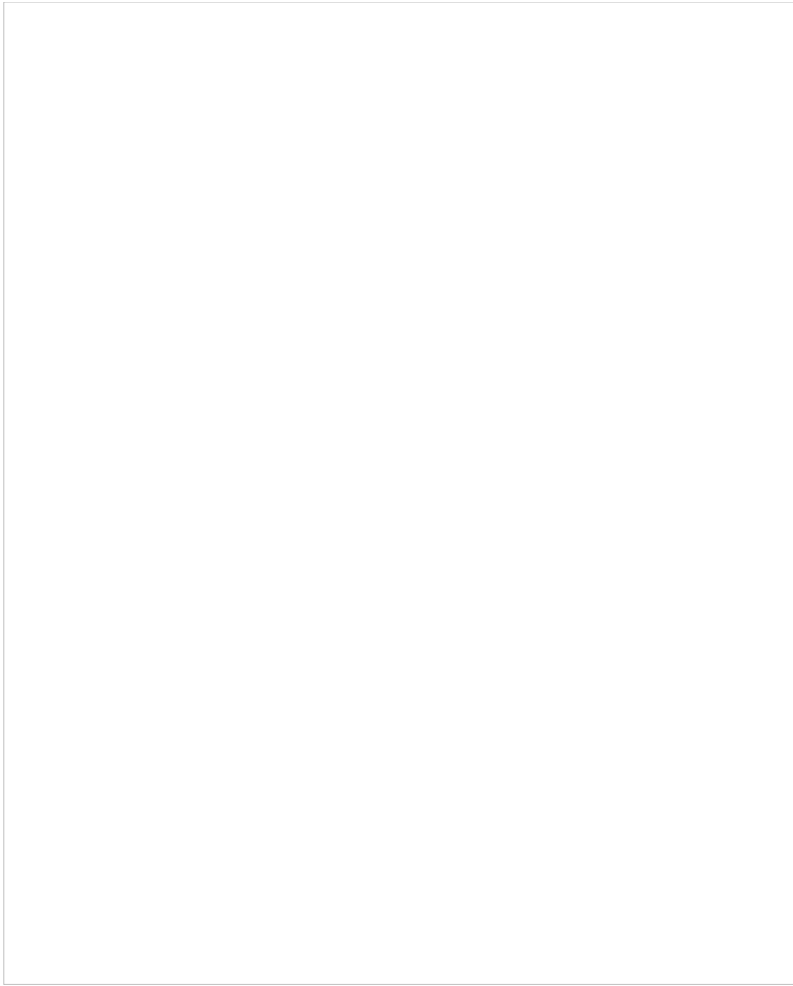
Step 5: Edit the Scenario to Work with Your Bot

To use the Watson Assistant in live web chat interactions, you'll have to edit the basic Mobile Chat scenario template to allow the bot to interact with the customer. For more information about scenarios, see the [Scenario Builder Reference Guide](#).

There are many ways to construct such a bot-enabled scenario; the instructions given in this step are provided as just one example of how to use your Watson Assistant in chat.

In this scenario-building exercise, you will be configuring the system to use the bot to:

- Automatically accept the customer's chat request
- Greet the customer
- Provide answers to the customer
- Provide suggested replies to the agent
- Connect to an agent when it's determined that the bot is not being helpful

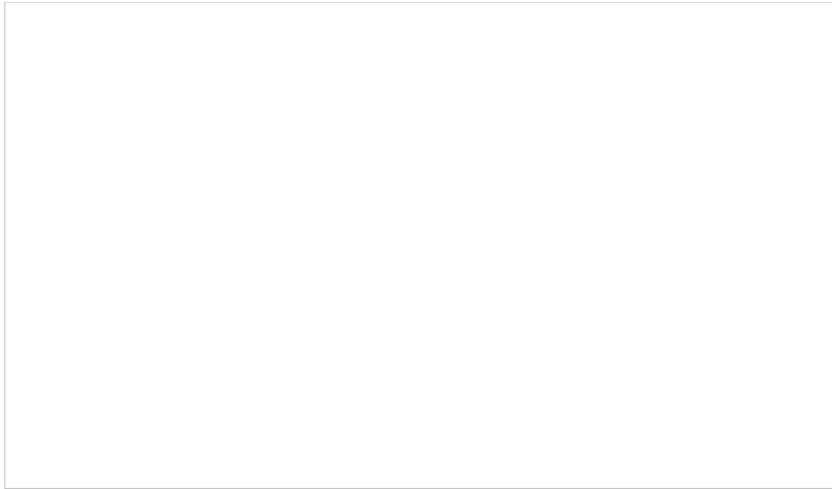


Example chat scenario in Scenario Builder

1. Download [File:App Example Bot Scenario.zip](#) and import it to your Contact Center Administrator application.
2. Select the imported scenario and click **EDIT** to open it.

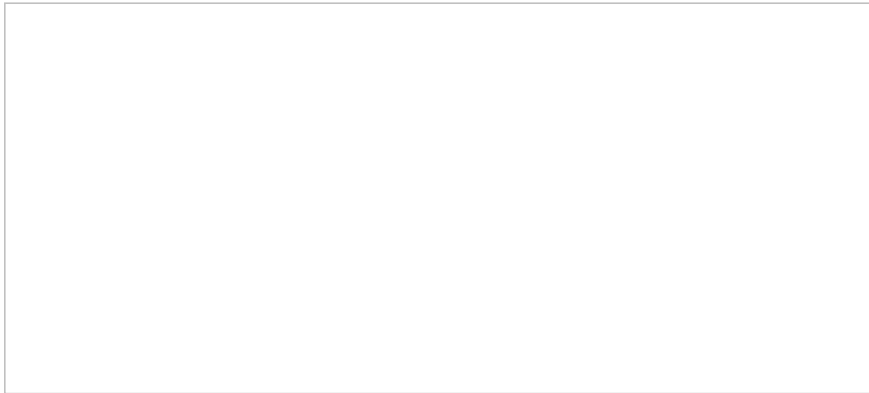
The following scenario blocks (with comments) are included in this example scenario. Be sure to read the comments that explain each scenario block's properties.

1. **Chat Bot Select Account:** This block tells the system which integrated bot will be used for this scenario.



Select an integration account

2. **Set Variable:** This establishes the customer's first name and last name. Setting this context helps to personalize the conversation.



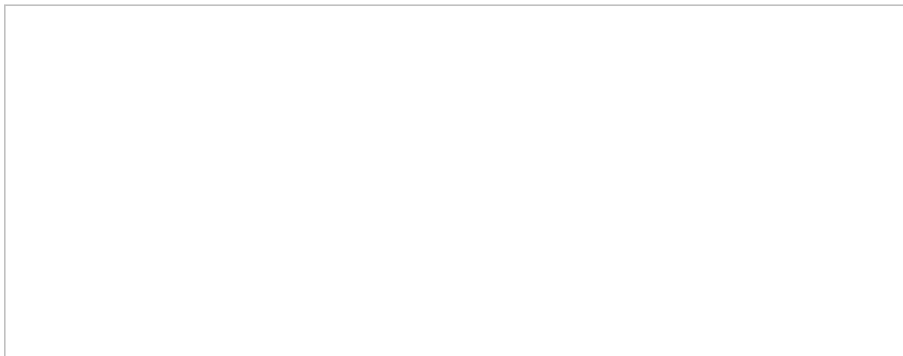
Use a variable to get the customer's name from the pre-chat form and use it in the chat interaction

3. **Send Message+:** This block delivers a chat message to the customer, incorporating the customer's name.



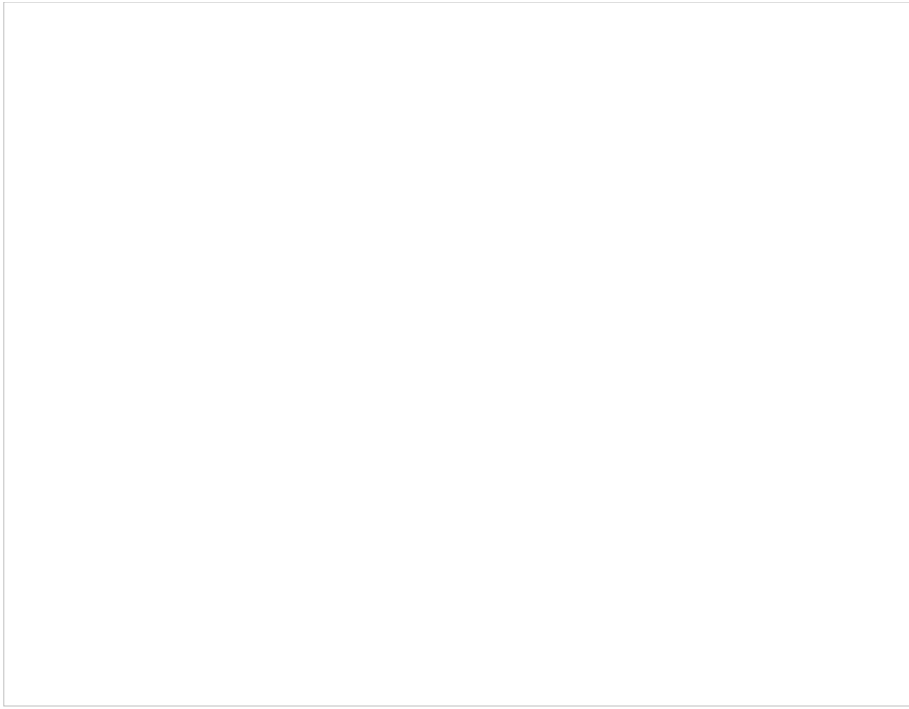
Have the bot send a message to the customer, addressing him/her by name

4. **Set Variable:** This time, we are defining a step. In this example scenario, we want the bot to respond to the customer two times before connecting to the agent. The first time is specified as "step 0" and the second time is specified as "step 1." The scenario will run through both steps. If the customer's issue is not resolved by step 1, the customer is transferred to the agent for help.



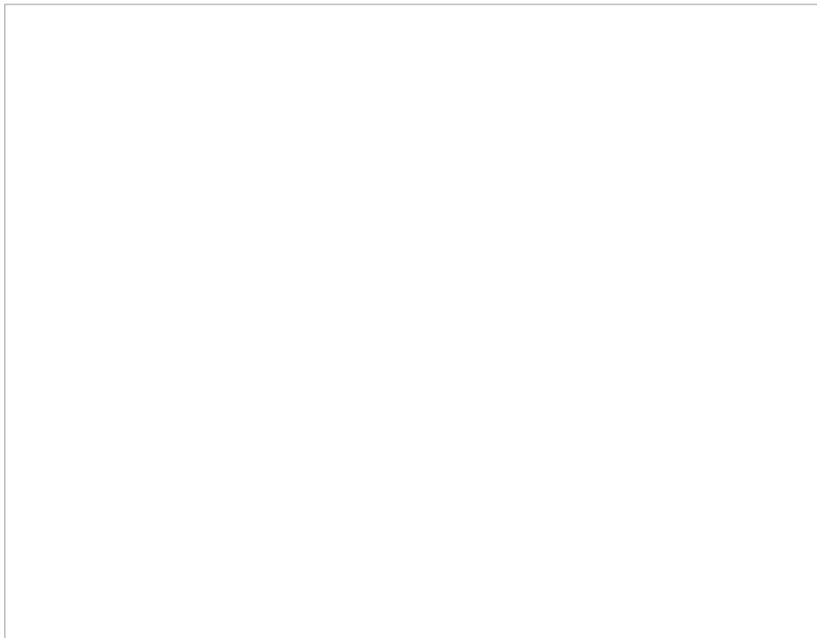
Use a variable to define this part as the first in a series of steps

5. **If:** The If block specifies what happens if the customer did not write a message in the pre-chat form's Message field. If that field is empty, the scenario tells the bot to send a message. The message aligns with what you have configured your Watson Assistant to say. We call the bot's messages suggestions.
 1. Add a branch
 2. Give it a condition of item.message is empty



If the customer's message is empty, have the bot send a message to the customer

6. **Ask a Bot:** The scenario uses this block to get suggestions (i.e., some response from the Watson Assistant) to deliver to the customer. For each of the conditional exits (Failed, Timeout, or No Data), the scenario will send a message and then connect to the agent.



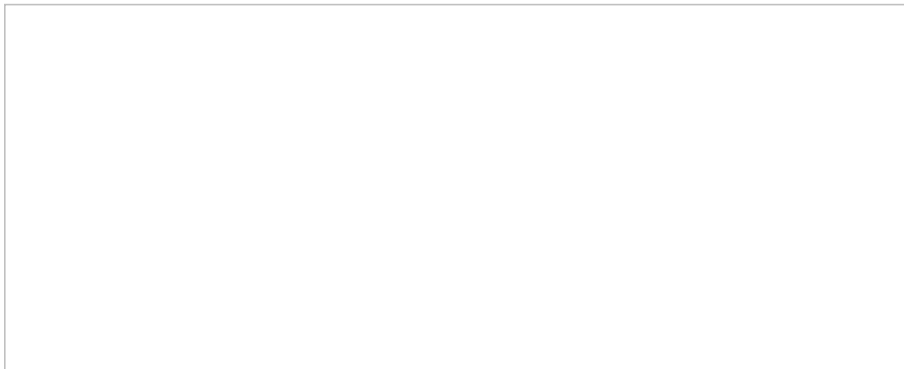
Get suggestions from the bot

7. **Send Message+:** This is used to send a message to the customer via either chat or SMS. In this example, the message is sent in the active chat interaction. If a specific message is not defined here, the standard system-wide message is sent to the customer.



Send another message if you want

8. **Set Variable:** This time, we are defining "step 1." Remember, we want the bot to respond to the customer two times before connecting to the agent. The first time is specified as "step 0" and the second time is specified as "step 1." The scenario will run through both steps. If the customer's issue is not resolved by step 1, the customer is transferred to the agent for help.



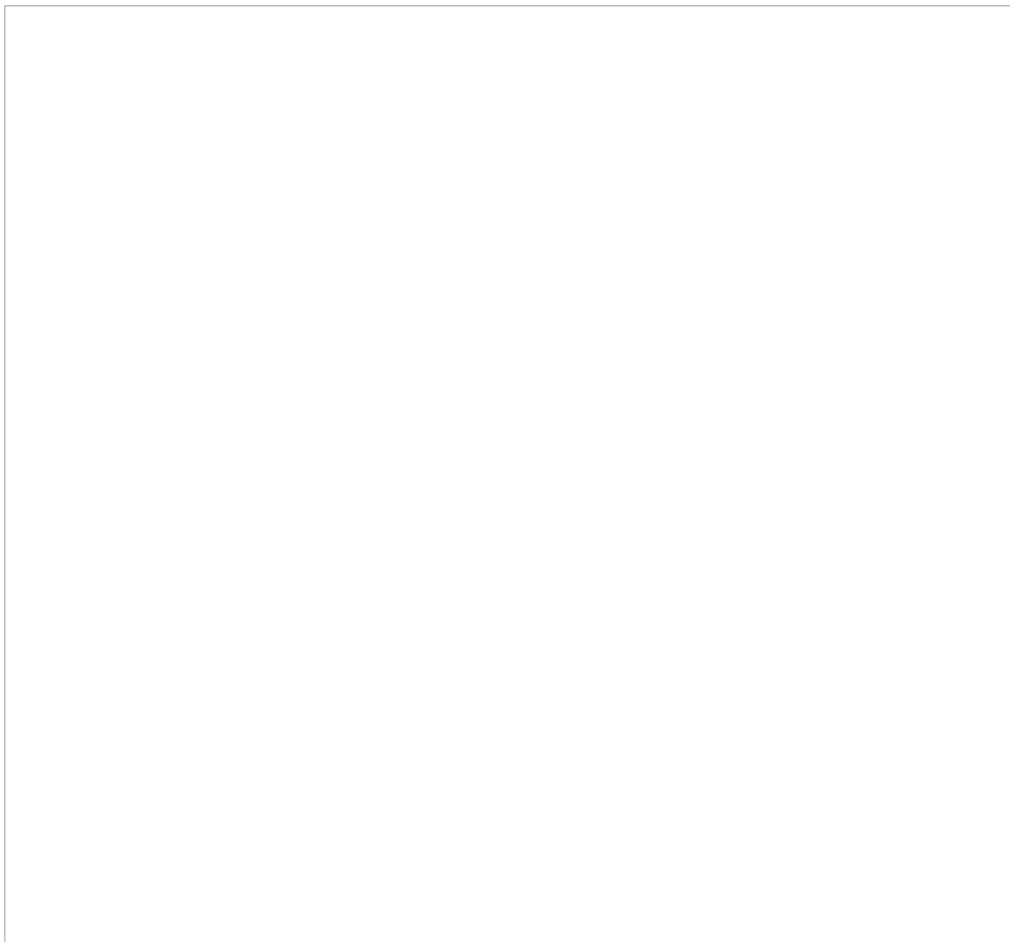
Define this part as step 1

9. **If:** In this If block, a condition is set to make sure that if the scenario takes more than two steps to resolve the customer's issue, then it will transfer the chat from the bot to the agent.
 1. Add a branch
 2. Give it a condition of "steps" > "2"



If the scenario takes more than two steps to resolve the customer's issue, then the agent steps in

10. **Find Agent:** The Find Agent block looks for the next available skilled agent to accept the chat. You can use Find Agent to set wait times and send the customer messages about estimated waiting time (EWT).



The bot couldn't resolve the customer in two steps, so the scenario looks for an agent

11. **Connect Chat:** Connect Chat is for setting the destination of the chat. If you leave the destination fields empty, the Find Agent block will find the first available agent.



You can either specify a destination for the chat or let the system find the next available agent

12. **Exit:** The Exit block completes the scenario. Without it, the scenario will loop through this configured flow until the customer ends the chat.

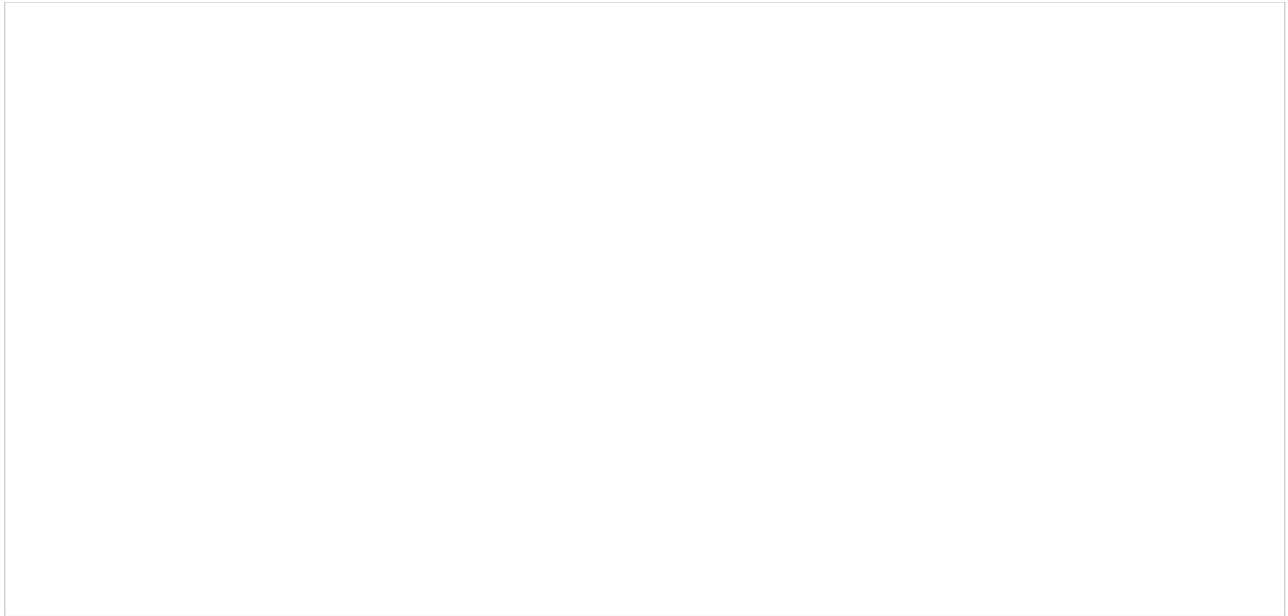
Step 6: Create or Select a Chat Service

See [Web Chat Configuration Step 2](#).

Step 7: Create or Choose a Chat Scenario Entry

Note that this step is basically the same as [Web Chat Configuration Step 3](#), with the addition of another important property.

1. Go to *Configuration > Scenario Entries > [Messaging/Chat](#)* and select a scenario entry point to use with your chat scenario. The entry point is what starts the scenario.



Messaging/Chat scenario entries list

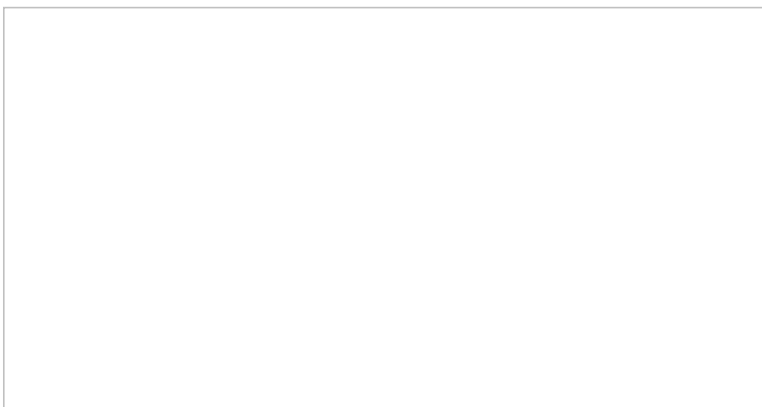
2. If there are no existing chat scenario entries shown, you have to create one.

How to Add a New Chat Scenario Entry

1. Click the **Add chat scenario entry** button.
2. In the Properties tab, fill in the properties as appropriate (see below for required and optional properties). For definitions, see the *Contact Center Administrator Guide*, section [Messaging/Chat](#).

Required Properties

In the Properties tab, make sure to address the following fields right now. These are critical to chat configuration.



Messaging/Chat scenario entry properties that you need to address for chat to work

- **Name** - The unique name (any) of the chat scenario entry (e.g., "Redwood Chat")
- **Unique identifier** - A long string of characters that identifies the chat entry. It is recommended that you change it to something short and unique (e.g., "1234").

- **Scenario** - Choose the Chat Scenario you selected or created earlier in this procedure (e.g., "Super Chat"). This is where the parts come together to make chat possible.
- **Service** - Choose the Chat Service you selected or created earlier in this procedure (e.g., "Pacific Chat"). This is where the parts come together to make chat possible.
- **Bot / Chat suggestions engine** - The engine that will provide suggested replies to the agent during chat. Note that in order to use a bot/chat suggestions engine, you must have an integration account configured for your contact center.

Other Properties

The [other remaining properties](#) are optional at this time. You can come back to them later.

Step 8: Edit Chat Widget Style

See [Web Chat Configuration Step 4](#)

Step 9: Add a Contact Tab

In order for this chat scenario entry to work, you must add a contact tab for the chat widget. See [How to Add a Contact Tab](#).

Step 10: Get the HTML

See [Web Chat Configuration Step 6](#)

Final Step: Test the Chat

You have now successfully set up your contact center service, scenario, and chat widget to work in an integrated manner with your Watson Assistant. You should test it to make sure that the:

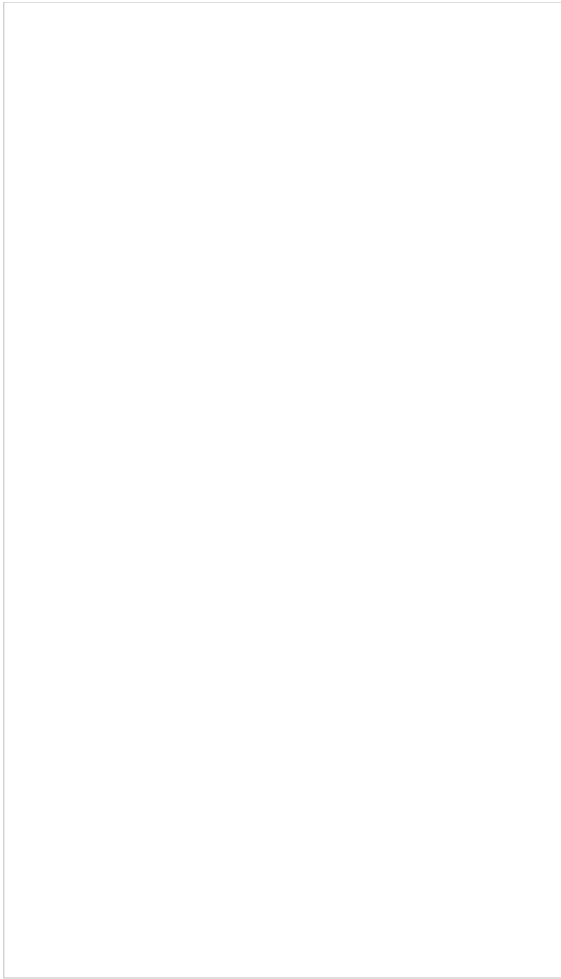
- Customer's name entered on the pre-chat form comes through during the chat
- Bot is able to automate the conversation with the customer
- Customer is connected to an agent if the bot couldn't help within two tries
- Agent is able to see the bot's suggestions during the active chat

The following screenshot shows a service chat (on the Agent Desktop side) where the bot provides two responses to the customer and then connects to an agent. You can see the bot's suggestion below the agent's text input field.



Example chat as seen on Agent Desktop

And the following is how the chat looks to the customer:



Example chat as seen by customer on
webpage