



5.8 How to Use Conversational IVR in a Scenario

Bright Pattern Documentation

Generated: 8/17/2022 12:45 am

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
AI & Bots Overview	4
Bot Creation	4
Integration Account Configuration	4
Putting It All Together in Bright Pattern	4
How to Create a Watson Assistant	4
Procedure	5
Step 1: Create an IBM account	5
Step 2: Add Watson Assistant as a resource	5
Step 3: Add a dialog skill	5
Step 4: Create and/or add intents for your dialog skill	6
Intent Properties	7
Step 5: Create entities	7
Entity Properties	8
Step 6: Design your dialog flow	9
Dialog Properties	9
Step 7: Try the Dialog	9
Step 7: Add the Watson Assistant	10
Next Steps	11
How to Create an Amazon Lex Bot	11
Procedure	12
Step 1: Create an AWS account	12
Step 2: Create an IAM user, grant administrative permissions, and save credentials	12
Step 3: Add Lex as a service and create a sample bot	13
Step 4: Review the bot workspace	14
Step 5: Edit intent properties	15
Properties	15
Step 6: Save and build	16
Step 7: Test it	16
Step 8: Publish your bot	17
Next Steps	17
Inbound Voice Service Configuration	17
Prerequisites	18
Procedure	18
1. Create and configure an inbound voice service	18
2. Assign teams to the service	19
3. Create additional skills and assign them	19
4. Create the scenario for the service	20
5. Associate the scenario with a service	21
6. Specify service level thresholds	22
7. Specify dial-out information	22
8. Set up periodic call recording exports	23
9. Configure caller ID	24
10. Customize historical reports	24
Service Configuration Quickstarts	25

1. REDIRECT [5.3:Tutorials-for-admins/AI/HowtoCreateWatsonAssistant](#)
1. REDIRECT [5.3:Contact-center-administrator-guide/CallCenterConfiguration/IntegrationAccounts/BotChatSuggestionsEngine](#)
1. REDIRECT [5.3:Contact-center-administrator-guide/CallCenterConfiguration/IntegrationAccounts/SpeechToText](#)
1. REDIRECT [5.3:Contact-center-administrator-guide/CallCenterConfiguration/IntegrationAccounts/TextToSpeech](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/SetVariable](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/ChatBotSelectAccount](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/PlayListen](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/AskaBot](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/AskaBot](#)
1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/If](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/Goto](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/SetVariable](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/WebScreenPop](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/FindAgent](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/ScenarioBlocks/Exit](#)

1. REDIRECT [5.3:Scenario-builder-reference-guide/Exercises/ScenarioBuilderBasics](#)

AI & Bots 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](#)

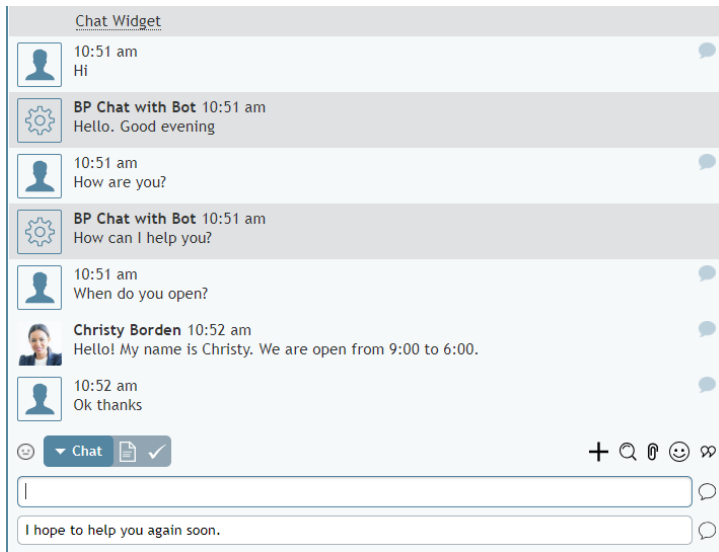
Putting It All Together in Bright Pattern

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

How to Create 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.

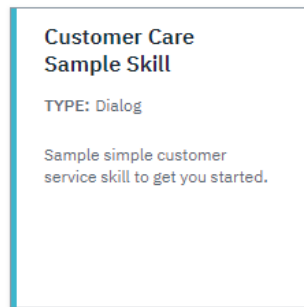
1. Click **Add dialog skill**.
2. For the sake of this example procedure, select **Use sample skill**, and click **Customer Care Sample Skill**, which

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.

Add Dialog Skill

Create a new skill, add a sample, or import one

Create skill Use sample skill Import skill



Use sample skill

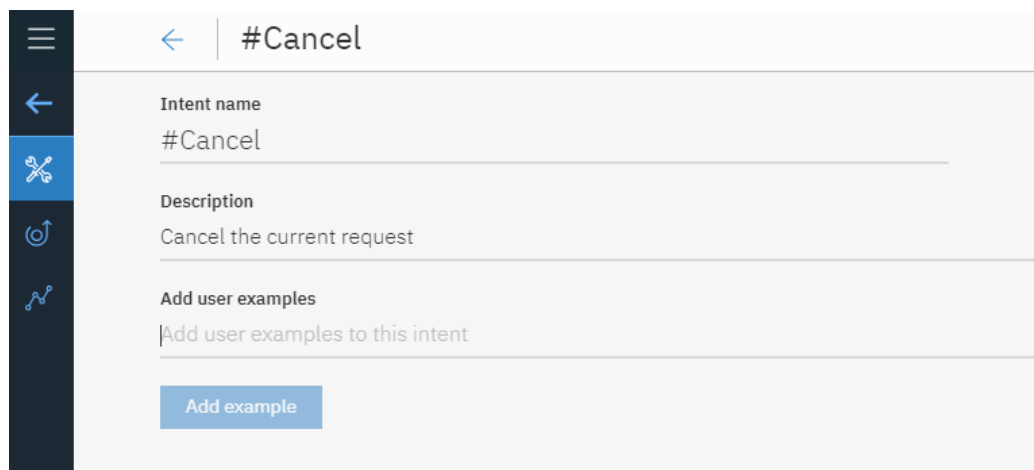
Step 4: Create and/or add intents for your dialog skill

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 (7) ▼

cancel that ✎

cancel the request ✎

forget it ✎

i changed my mind ✎

i don't want a table anymore anymore ✎

nevermind ✎

never mind ✎

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.

My entities System entities

Add entity



<input type="checkbox"/> Entity (6) ▼	Values
<input type="checkbox"/> @holiday	halloween, thanksgiving, new years eve, new years, christmas eve, labor day, independence day,
<input type="checkbox"/> @landmark	times square, empire state building, grand central
<input type="checkbox"/> @phone	US Phone pattern
<input type="checkbox"/> @reply	no, yes
<input type="checkbox"/> @specialist	Nicholas, Barbara, Brenda, Derrik, Robert, Maria
<input type="checkbox"/> @zip_code	US Zip

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

Dictionary

Annotation ^{BETA}

<input type="checkbox"/> Entity values (10) ▼	Type	
<input type="checkbox"/> christmas	Synonyms	christmas day, x man day, xmas, x mas, x-mas, x-mas day, xmas day
<input type="checkbox"/> christmas eve	Synonyms	x mas eve, x-mas eve, xmas eve
<input type="checkbox"/> halloween	Synonyms	
<input type="checkbox"/> independence day	Synonyms	7/4, fourth of july, july 4, july 4th, july fourth
<input type="checkbox"/> labor day	Synonyms	
<input type="checkbox"/> memorial day	Synonyms	
<input type="checkbox"/> new years	Synonyms	1/1, jan 1, jan 1st, jan first, january 1, january 1st, january first, new year, new year day, ne
<input type="checkbox"/> new years eve	Synonyms	12-31, 12/31, dec 31, dec 31st, new year's eve

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*.

The screenshot displays the Watson Assistant dialog editor. On the left, a tree view shows the dialog flow for 'Customer Care Sample Skill'. The 'Hours of Operation' node is selected. On the right, the configuration for this node is shown. It includes the intent '#Customer_Care_Store_Hours' and a table of response rules.

If assistant recognizes	Respond with
1 @holiday:christmas @holiday:the	We are closed on @holiday
2 @sys-date.reformatDateTime("EEE	We are open on <? @sys-date.refo
3 @holiday	We are open on @holiday regular t
4 @sys-date.reformatDateTime("EEE	Our hours on <? @sys-date.reform
5 true	Our hours are Monday to Friday 10

Sample dialog

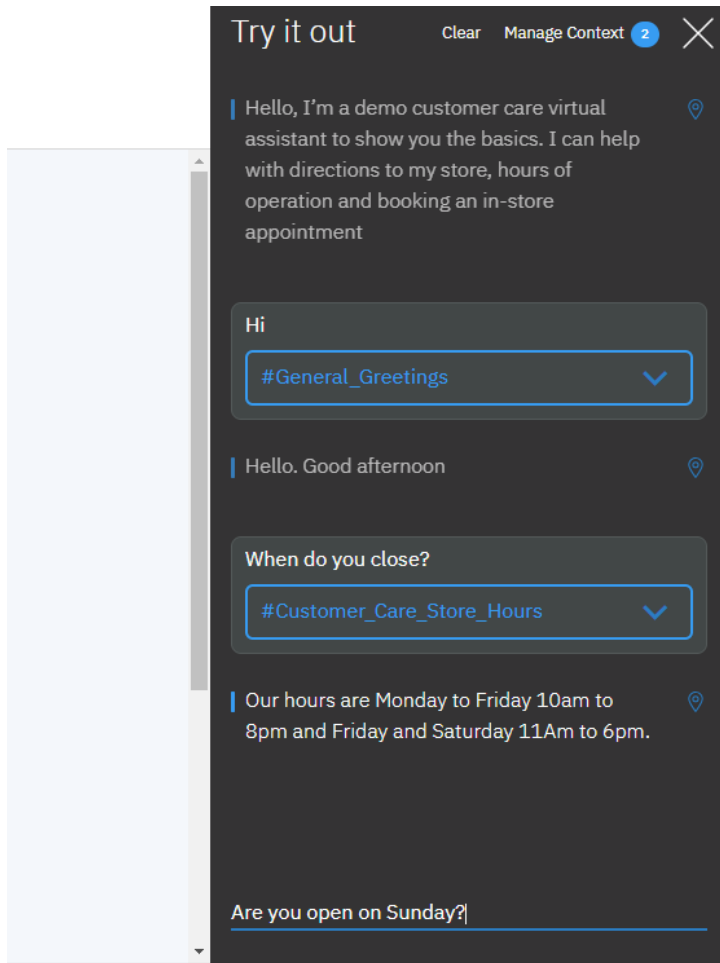
Dialog Properties

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 Dialog Skill

Create a new skill or add an existing one

[Create skill](#)

[Add existing skill](#)

[Use sample skill](#)

[Import skill](#)

Name

Product Returns

Description (optional)

Language

English (US)

Create

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:

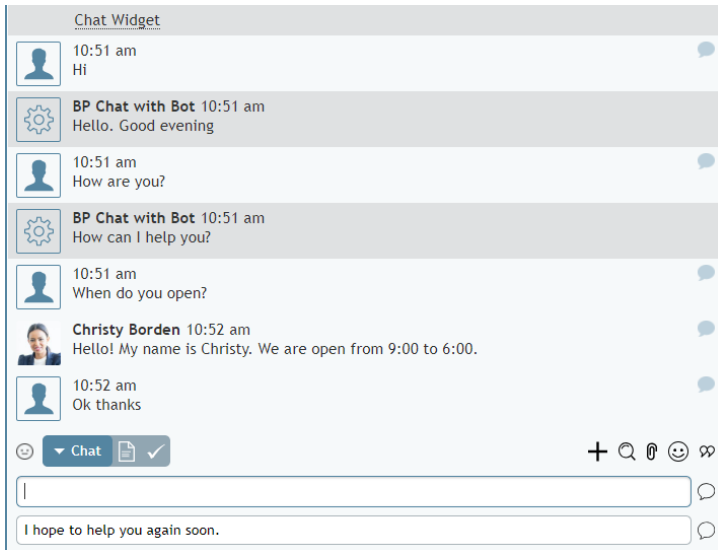
- Read IBM's documentation, explore the Watson Assistant, and edit it as desired
- Create a Watson [bot/chat suggestions integration account](#)
- Edit a [chat scenario that uses your Watson Assistant](#)
- [Configure web chat to work with bots](#)

How to 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](#).

Please note that Bright Pattern only supports version 1 Amazon Lex bots at this time.



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 user



Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [REDACTED]

Download .csv

	User	Access key ID	Secret access key	Password
▶	Tracy	[REDACTED]	***** Show	***** Show

Email login instructions

Send email

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.

Create your bot

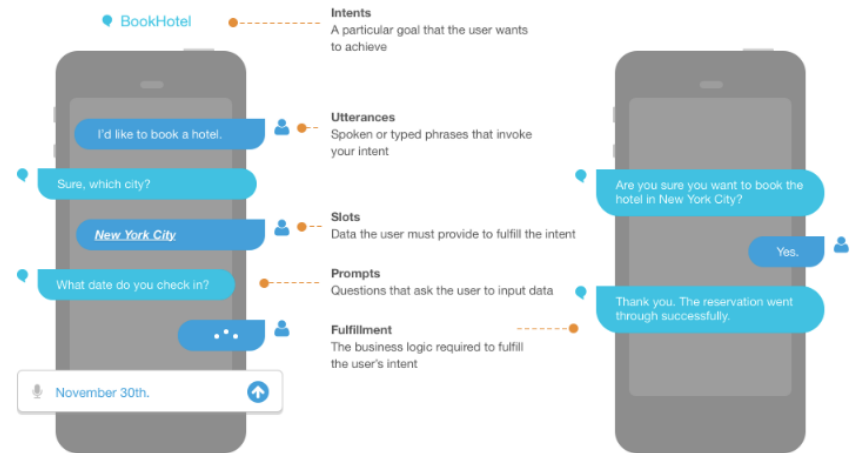
Amazon Lex enables any developer to build conversational chatbots quickly and easily. With Amazon Lex, no deep learning expertise is necessary—you just specify the basic conversational flow directly from the console, and then Amazon Lex manages the dialogue and dynamically adjusts the response. To get started, you can choose one of the sample bots provided below or build a new custom bot from scratch.

CREATE YOUR OWN TRY A SAMPLE

Custom bot BookTrip OrderFlowers ScheduleAppointment

Bot name

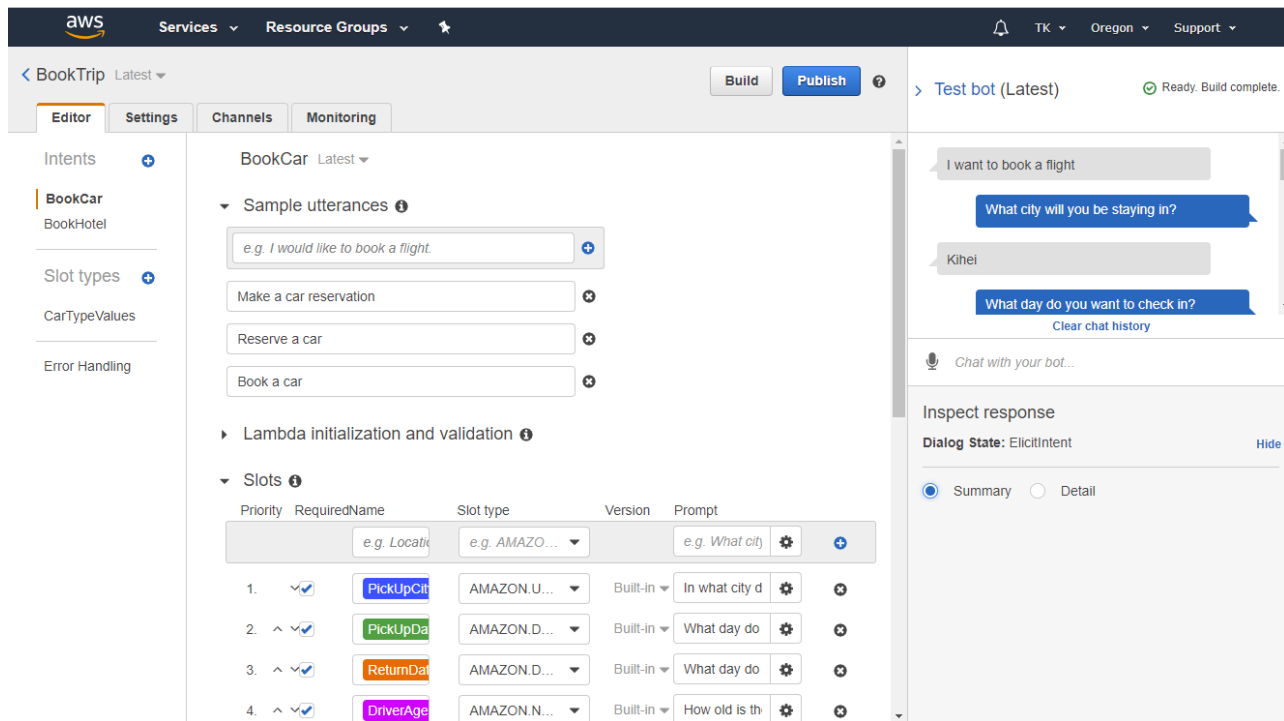
BookTrip



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

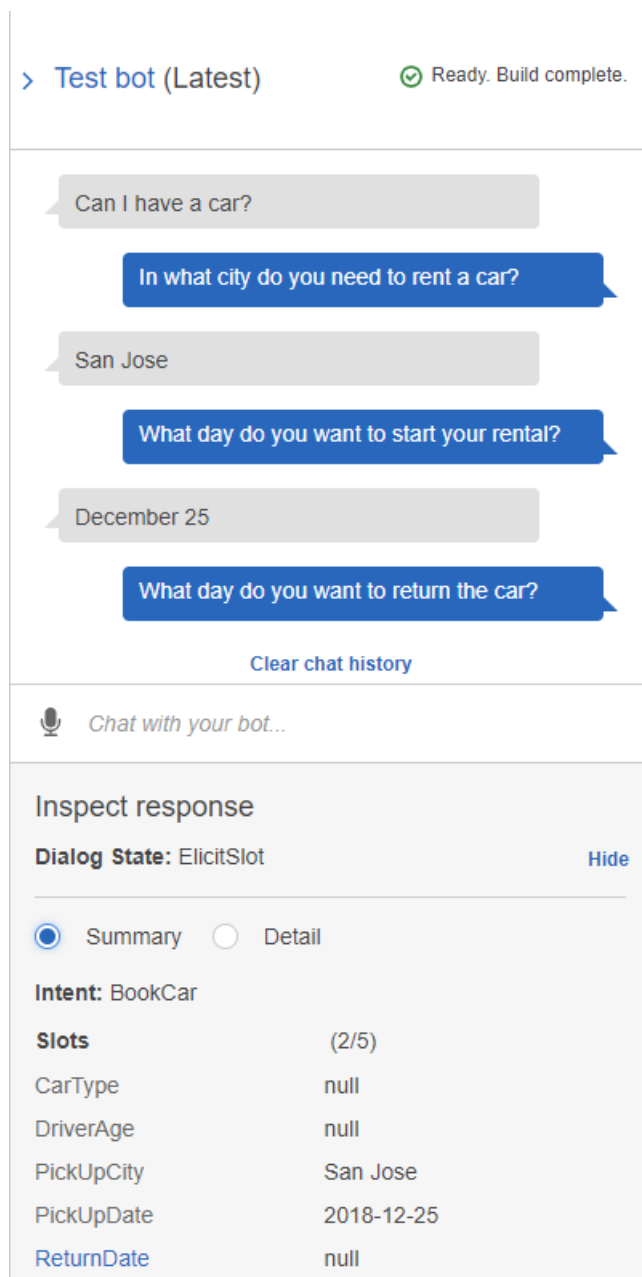
Step 6: Save and build

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 bot (Latest) 🟢 Ready. Build complete.

Can I have a car?

In what city do you need to rent a car?

San Jose

What day do you want to start your rental?

December 25

What day do you want to return the car?

Clear chat history

🎤 Chat with your bot...

Inspect response

Dialog State: ElicitSlot Hide

Summary Detail

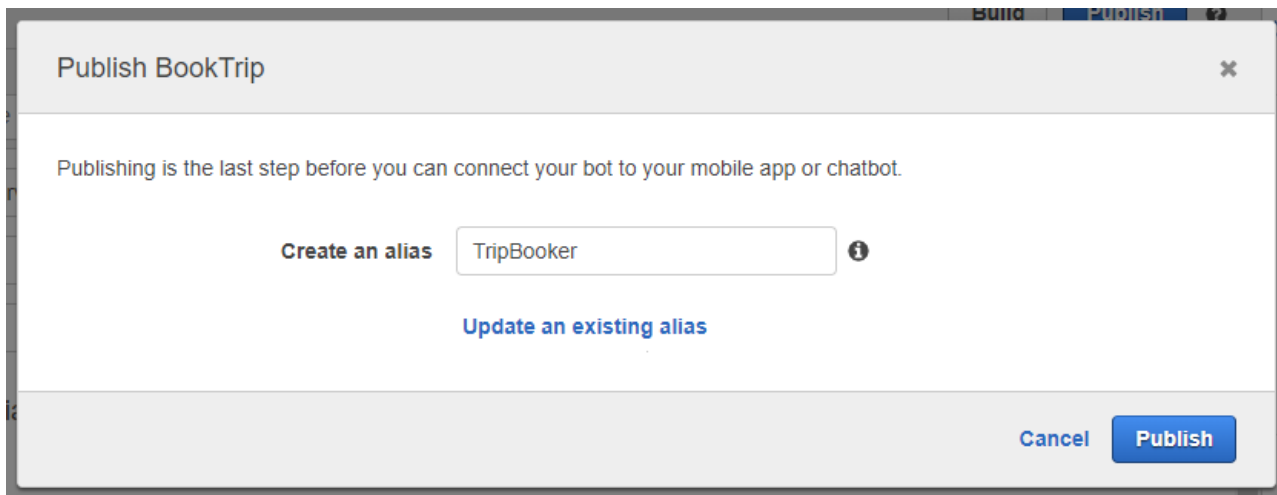
Intent: BookCar

Slots	(2/5)
CarType	null
DriverAge	null
PickUpCity	San Jose
PickUpDate	2018-12-25
ReturnDate	null

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](#)

Inbound Voice Service Configuration

This article describes how to configure inbound voice service within Bright Pattern's Contact Center Administrator application.

For inbound voice services with virtual queuing (callback option), see also Bright Pattern's [Virtual Queue Tutorial](#).

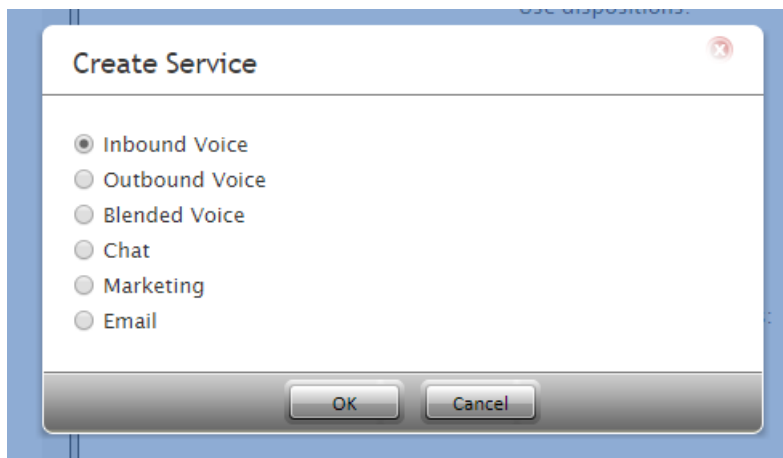
Prerequisites

This article assumes that you have already completed [initial configuration](#) for your contact center.

Procedure

1. Create and configure an inbound voice service

1. Log in to Contact Center Administrator, and go to *Services and Campaigns*.
2. Click the **Add service (+)** button to create a new inbound voice service.
3. Select **Inbound Voice** as the type of service.



Choose from the available services

4. In the [Properties tab](#) that appears, update the service properties, being sure to give the service a unique name (required). At this point, all other properties are optional. For the sake of this procedure, we are going to keep the properties simple, as shown.



Name:

Type: ▼

Hours of operation: select specify
Open 24/7
[add](#)

Override schedule on selected days:

Enable voice signature flag:

Record calls:

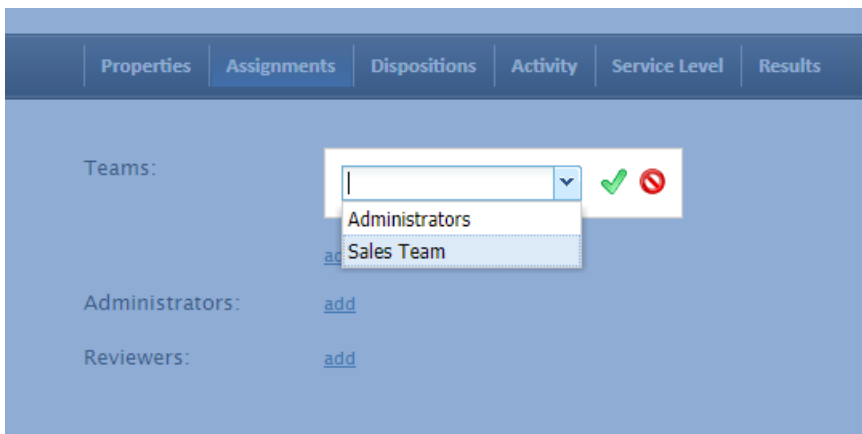
Record IVR call segment:

Inbound voice service properties

5. Click **Apply** to save your changes.

2. Assign teams to the service

1. In the [Assignments tab](#), assign teams to this service. Note that the service will be created with a default service skill that can be assigned automatically to all members of the associated teams.



Assign users to teams

2. Optionally assign administrators and reviewers to this service.

3. Click **Apply** to save your changes.

3. Create additional skills and assign them

1. If necessary, create [language skills](#) and any other [auxiliary skills](#) that may have to be taken into account when routing calls requesting this service.

Auxiliary Skills

Language

English
Japanese
Russian
[add skill](#)

[add skill group](#)

You have to add skills before you can assign them

2. In *Users & Teams > Skill Levels*, use [skill levels](#) to assign the language and/or auxiliary skills to agents who will provide this service. The higher the numeric value, the more skilled that user is.


Skill Levels

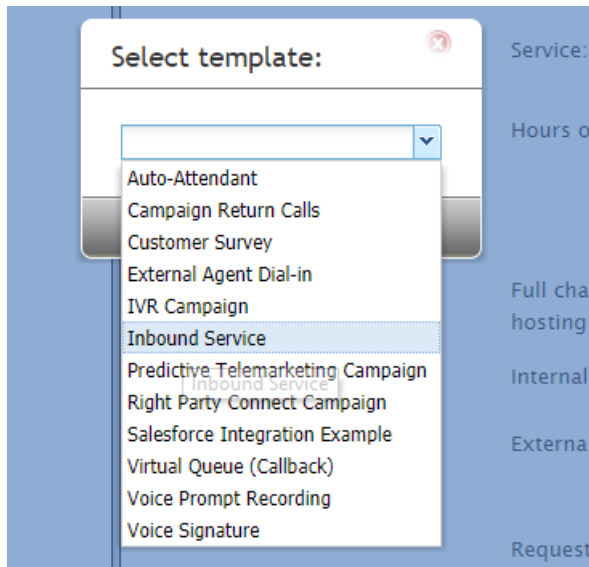
All users 6 users				Language			Services
First Name	Last Name	Team	Roles	English	Japanese	Russian	Basic Chat
Admin	User	Administrators	Agent,Campaign Administ...	100	100		
Brett	R	Sales Team	Agent	100			100
Bright	Pattern	Administrators	Agent,Campaign Administ...	100		100	
Shelby	B	Sales Team	Agent,Campaign Administ...	100			100
Ted	H	Sales Team	Agent,Campaign Administ...	100			100
Tracy	K	Administrators	Agent,Campaign Administ...	100			

Skills are assigned as percentage values (e.g., "100" or "50")

3. Click **Apply** to save your changes.

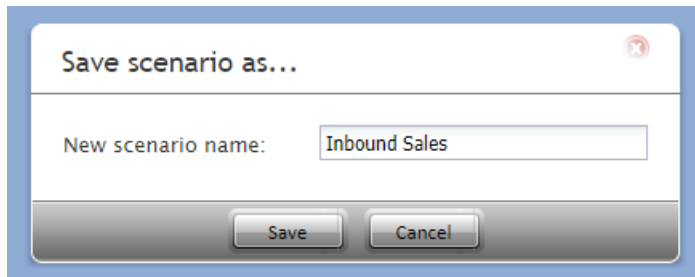
4. Create the scenario for the service

1. Go to *Configuration > Scenarios > Chat* to create the voice scenario for this service.
2. Either select an existing voice 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 "Inbound Voice" template.



Select the "Inbound Voice" template

3. Creating a new 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**. You can come back to Scenario Builder later to edit the scenario.



Name the scenario

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

5. Associate the scenario with a service

1. Go to [Scenario Entries](#) > [Dial-in](#).
2. Select the dial-in scenario entry that has been preconfigured for your contact center (typically "Voicemail").
3. In the **Service** property, select the inbound voice property that you created earlier in this procedure. This associates the scenario entry with the inbound voice service.

Name:

Scenario: [add/edit](#)

Service: [add/edit](#)

Hours of Operation: from service select specify
 Hours of operation will be taken from the service that is active at the moment.

Full chat code for local hosting:

Internal numbers: 8888
[add](#)

External numbers: [add](#)

Requested skills: [add](#)

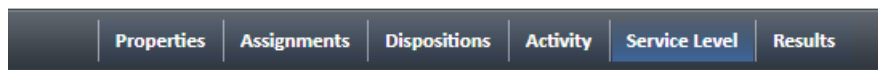
Dial-in scenario entry properties

4. Click **Apply** to save your changes.

Note: The remaining steps are optional and depend on your workflows and reporting requirements.

6. Specify service level thresholds

1. Go back to *Services & Campaigns* and select the chat service you just created.
2. In the [Service Level tab](#), set the percentage of chats associated with this service that are expected to be answered within a certain amount of time.



Service level:

Answered calls: %

Within threshold: seconds

Set service level thresholds

7. Specify dial-out information

1. Go to *Directory* > [Dial-out Entries](#) and select an entry from the list. A dial-out entry is an access number that has been assigned to your contact center by your service provider.
2. Update the entry properties, including phone number prefixes and caller ID, for outbound consultation calls.

Digits:

Is a prefix (not a complete number):

Remove prefix:

Insert prefix:

Set service (if not set by agent): [add/edit](#)

Caller ID selection (by diminishing priority)

Use customer party number on transfers and consult calls (if available):

Direct mapping of internal phones: [edit](#)

Use Caller ID from service configuration if service information is available, otherwise use this number:

Append *<extension> to caller ID:

Update dial-out entries properties

8. Set up periodic call recording exports

1. Go to *Services & Campaigns* > [Results tab](#).
2. Under **Periodic Recording Export Jobs**, click **add**.
3. Update all the properties, specifying which call recordings should be exported and where they should go.

>> Periodic Export of Recordings x

Name:

Enabled:

What to export:

Type of content: Complete call recordings
 Recordings of agent segments

Voice signature only:

Dispositions: ▼

Recording completed after: 1 / 31 / 19 03:11:20 PM (America/Los_Angeles)

Destination:

Protocol: FTP
 SFTP
 Amazon AWS

FTP/SFTP Server hostname:

Recordings: [recordings / \\$\(callDate\) / \\$\(callHour\) / \\$\(callDate\)\\$\(callTime\)_\\$\(customerPh...](#)

Upload manifest with list of files: CSV File

Authentication

Username:

Password:

Daily export time window
(set by provider) 11:00 PM – 6:00 AM (America/New_York)

Periodic Recording Export Jobs properties

9. Configure caller ID

1. Go to *Services & Campaigns* and select your inbound voice service from the list of services.
2. Click on the [Numbers tab](#) and then on the **Caller ID** button to configure a caller ID for outbound SMS communications.
3. Click the **Add caller ID (+)** button.
4. Select your phone number from the list, and enter a state/province to be displayed.
5. Click **Apply** to save your changes.

10. Customize historical reports

Review the available service reports and, if necessary, customize them to your specific reporting needs.

Your inbound voice service configuration is now complete.

Service Configuration Quickstarts

Be sure to read Bright Pattern's other service configuration quickstarts, which will help you to set up bot-assisted chat services, campaigns, and email services for your contact center:

- [Chat Service Configuration](#)
- [Outbound Campaign Service Configuration](#)
- [Email Service Configuration](#)