**LIVEPERSON CHATBOT PLATFORM**

**What is Conversational AI?**

Conversational AI is when we apply AI in conversational interaction with human.

* A subset of Artificial Intelligence
* Focuses on a conversational interaction with a human.
* Various Forms are Chatbots, Digital assistants and Virtual Assistants.
* Uses subdomains of AI, such as NLP, ML and DL

**Basic Model Concepts**

**Intent:**

**E.g :** Find a slot for zoom meeting with Bob on Friday at 10 am

**Intent**: User wants to set up a meeting.

**What is an Intent?**

* An intent maps to a known use case. It's the entry point for a conversation.
* An intent is resolved for a set of related messages that a machine has previously seen during training or sees for the first time.
* In natural language processing, intents are mapped to conversations a user wants to have.
* Identifying the intent is the first step in conversation AI.

**What are utterances?**

Different ways to set up a meeting:

*I want to meet Bob from 9 am to 2am to discuss sales forecast.*

*For January 12 2024 at 10am for 2 hours create an invite for a zoom.*

*Send a calendar Invite for Wednesday 11am to 12pm for a meeting.*

**What are utterances?**

* Utterances are sample messages of how users would address an intent.
* Represent real user messages
* Must be curated to avoid bias or blurring
* Intents should be neither undertrained nor over trained.

**What is an entity?**

Find a slot for zoom meeting with Bob on Friday at 10 am

**LivePerson Platform:**

Quick Start Guide: <https://knowledge.liveperson.com/getting-started-quick-start-guides-conversation-builder-quick-start.html/>

The **Getting Started with Bot Building - Messaging** tutorial series includes five tutorials that build on one another:

* [**Dialogs & Patterns**](https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-dialogs-patterns.html): Learn how to get started with a basic Messaging bot. This tutorial explores the concept of dialogs (the different conversation flows a bot can have) and patterns, which allow the bot to match user input and respond intelligently. You are guided through the creation of a simple "Hello World" bot.

Link : <https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-dialogs-patterns.html>

* [**Intents**](https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-intents.html): Dive deeper into user input matching and look at the more advanced concepts of intents and entities. Intents allow you to use our Natural Language Understanding (NLU) engine to enable your bot to more accurately match user input, while entities help with storing variable-like parameters for quick and easy data access.

Link : <https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-intents.html>

* [**Integrations**](https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-integrations.html): Explore integrations, which allow your bot to query external APIs, bringing powerful new options to your conversations. Integrations let you programmatically access catalogs, databases, and other web services to super power your bot responses.

Link: <https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-integrations.html>

* [**Deploy the Bot**](https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-deploy-the-bot.html): Bring it all together by deploying your Messaging bot to the Conversational Cloud environment and testing it out.

Link: <https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-deploy-the-bot.html>

* [**Escalate to an Agent**](https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-escalate-to-an-agent.html): Ensure users have all their needs met by providing a way for them to contact a human agent. Test this functionality by accepting incoming messages from the Conversational Cloud.

Link: <https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-escalate-to-an-agent.html>

**Videos Tutorial:**

Link: <https://developers.liveperson.com/videos.html>

**Bot Grouping**: You will create a Bot and as per your requirement then you will push required bots into grouped and ungrouped bots.

Link for Doc: [C:\Users\Rakshitha.H\Documents](file:///C:\Users\Rakshitha.H\Documents)

**Conversation Assist:** Below document includes on how to create Knowledge AI, users & skills, Generative AI feature.

Link for doc: <https://developers.liveperson.com/tutorials-using-conversation-assist-prerequisite-steps.html>

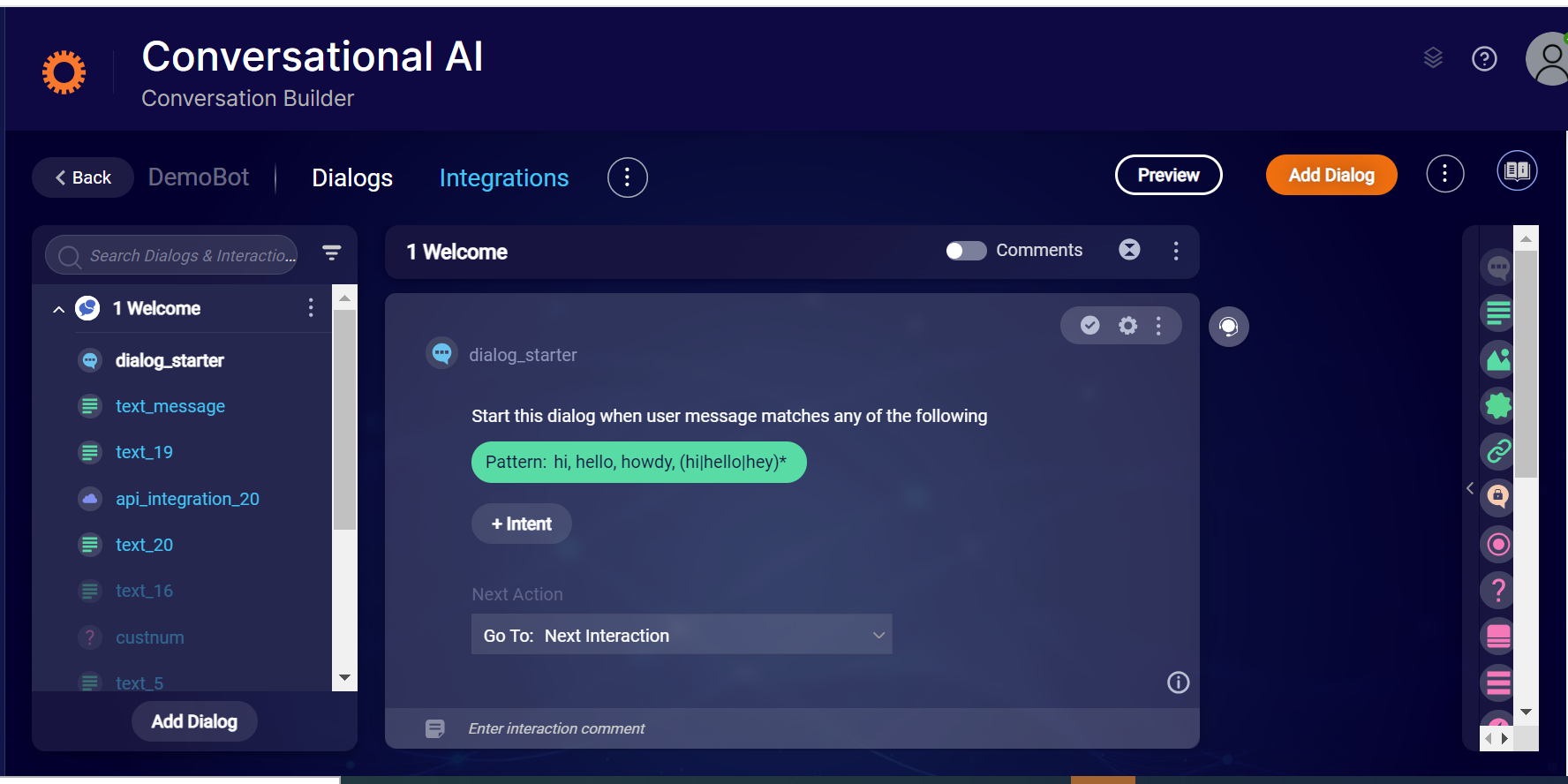
**Custom Code:**

* Custom JavaScript code should be compatible with ES5, i.e., ECMAScript 5 (2009). And there is an exception: The toLocaleString method isn't supported.

* Custom JavaScript code must complete within 5 seconds. Otherwise, it times out, and the execution flow continues on as per the [order of operations](https://developers.liveperson.com/conversation-builder-interactions-interaction-basics.html#order-of-operations). In general, the code shouldn’t do heavy calculations and should be fast.

**Bot Workspace:**

LivePerson Conversation Builder is the primary workspace for building a bot.



A bot consists of one or more dialogs that each contain one or more interactions. In the image above of the Dialogs view in Conversation Builder, note the following:

The open "DemoBot" bot has 3 dialogs. These are listed in the dialogs panel on the left.

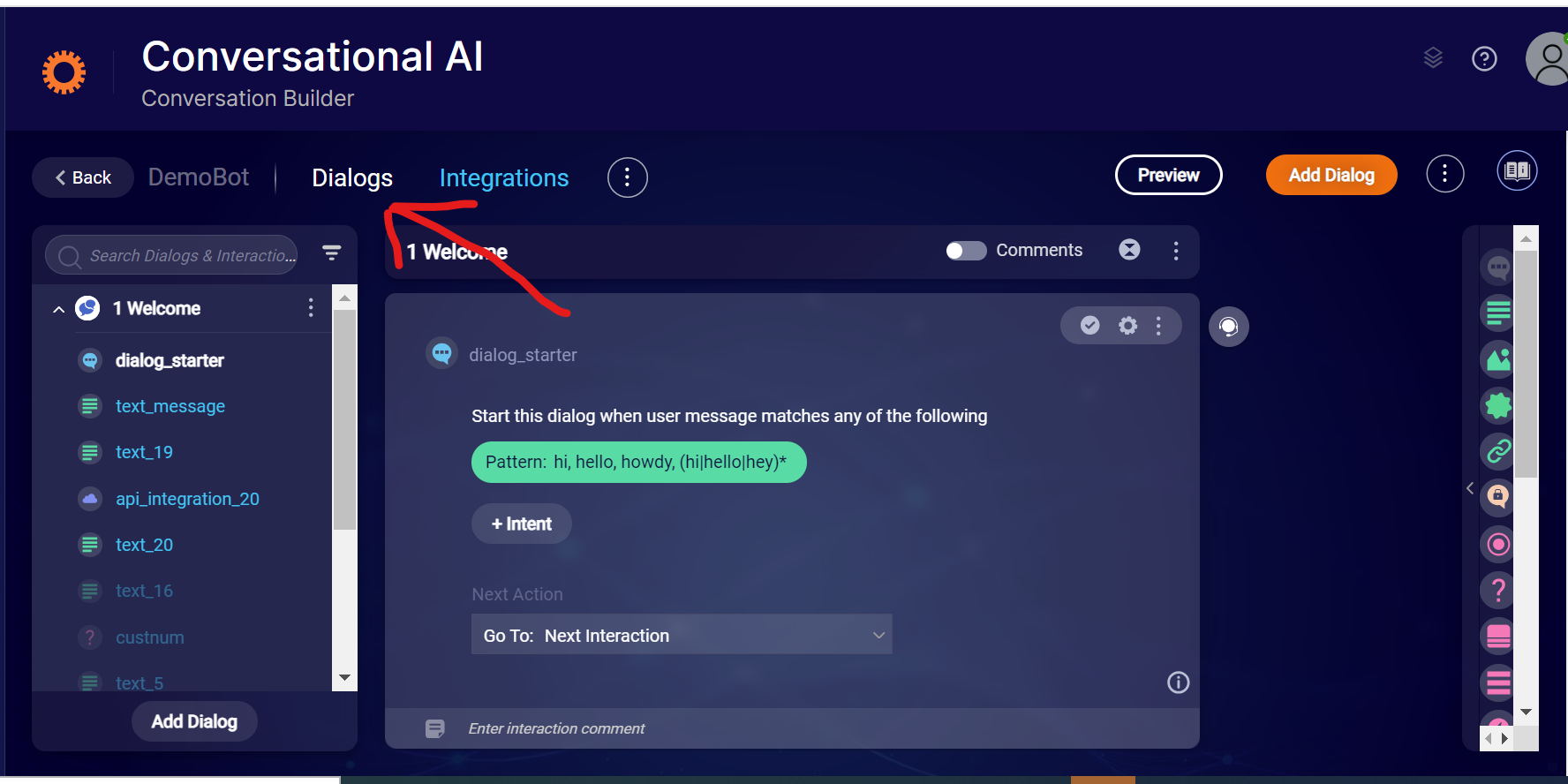
The Welcome dialog contains 7 interactions. These are also listed in the dialogs panel on the left.

The Welcome dialog is currently displayed in the dialog editor in the center of the screen. As such, its name appears in bold in the dialogs panel.

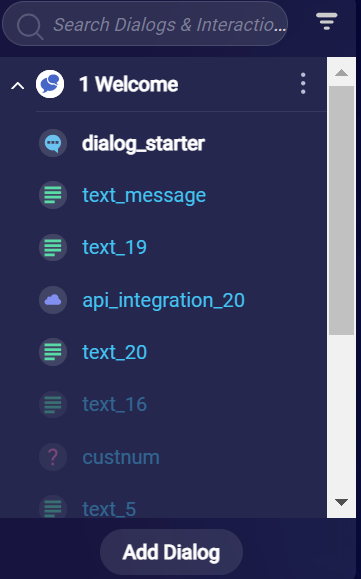
Note: Conversation Builder supports the creation of bots for all the Conversational Cloud channels. Because a bot simply listens to a piece of text sent from a user, any channel that provides text to Conversational Cloud can potentially trigger a bot. However, there are channel-specific formats, features and conventions that might not be available for all channels. Please contact your LivePerson account team for information on formats supported for specific channels.

**The Dialog View**

When you first open a bot, you are taken to the **Dialogs** view. Use this view to make changes to the bot's dialogs and the interactions within them.

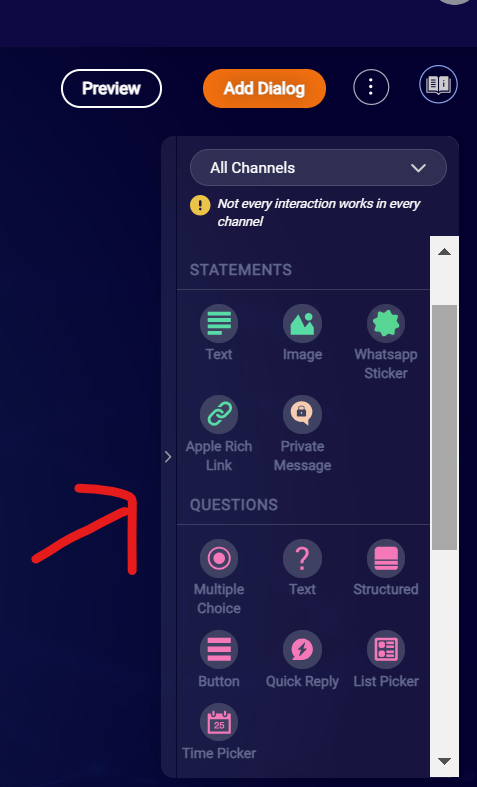


In the left panel in the Dialogs view, there's an expandable list of the dialogs in the open bot. The dialogs are displayed vertically. You can use the search box to search by dialog name, interaction name, or interaction content.



**The interaction toolbar:**

The interactions toolbar is located on the right side in the Dialogs view.



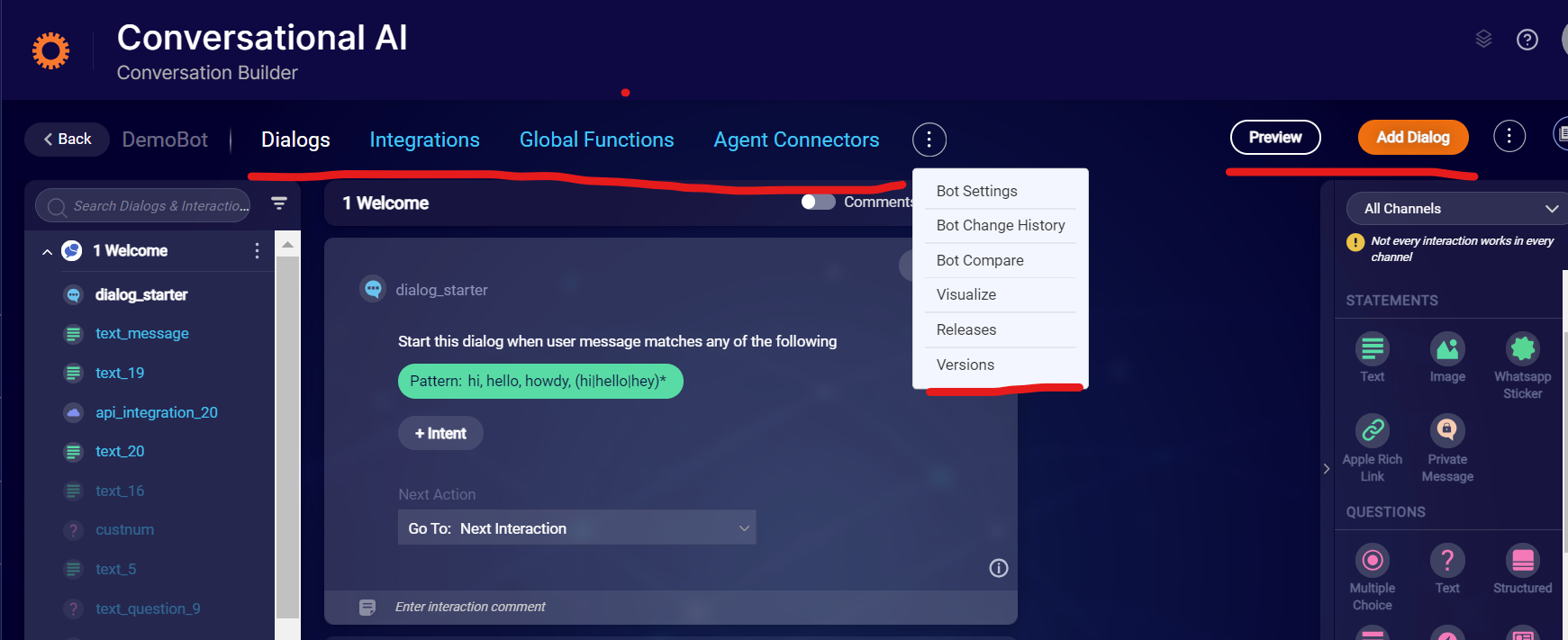
The interactions toolbar contains tools for adding different types of interactions to the dialog that's currently open in the dialog editor in the center of the screen.

In the image above, we're working on a Messaging bot, so the toolbar smartly contains only the interactions that can be used in Messaging bots. You can use a different set of interactions in Voicebots.

Select an interaction to add it to the open dialog. You can then start configuring the interaction. For example, if you add a simple, text-based question, you'll be able to enter the question that you want the bot to ask the user.

**The Menu Bar**

The menu bar is displayed in the upper-left corner and contains the following options:

****

**Dialogs**: Create and edit dialogs in this view.

**Integrations**: Create and edit integrations in this view.

**Global Functions**: Write code snippets for use within dialogs here. See scripting functions for built-in methods.

**Agent Connectors**: Deploy the bot to a Conversational Cloud environment.

The Three-dot icon makes available more options:

**Bot Settings** (see this section)

**Bot Change History**

**Bot Compare**

**Visualize**

**Releases**

**Versions**

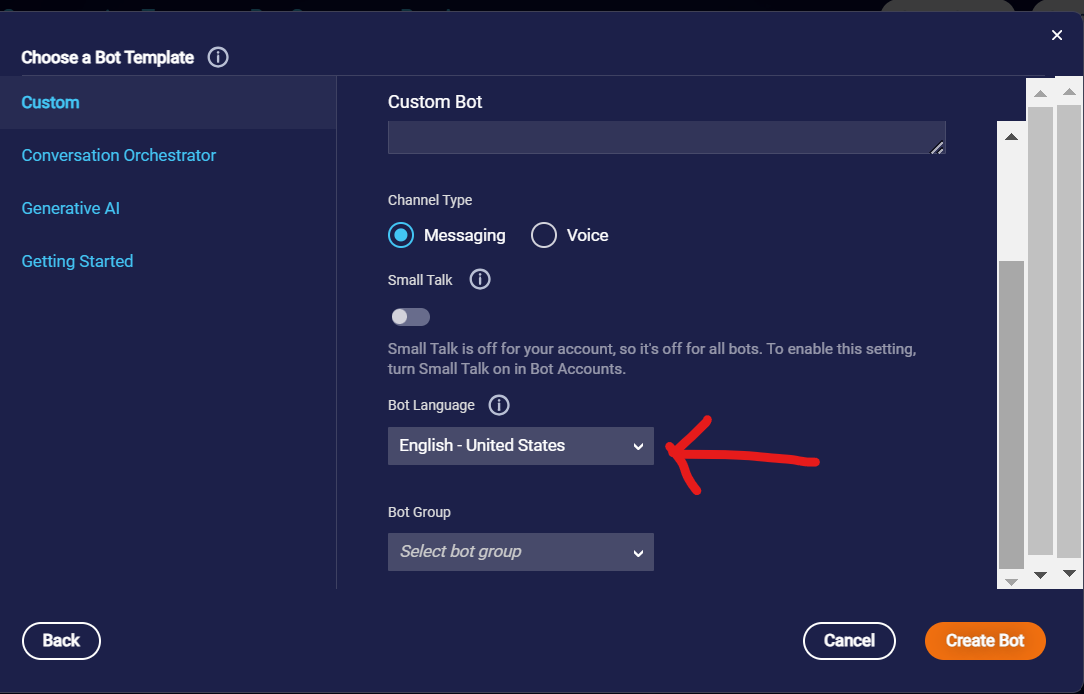
In the upper-right corner, there are a few more options:

**Preview**: Test the open bot by previewing a conversation. From here you can also access the bot's logs for debugging purposes.

**Add Dialog**: Add a dialog to the open bot.

**Bot Basics**

When you create a LivePerson Conversation Builder bot, you specify its bot language:



Select the language based on how you'll train the bot: If you will use an intent domain, you must select the language of the domain that will be associated with the bot. For example, if the intent domain uses English, select "English" for the bot language. If these languages don't match, errors during NLU processing will occur. Alternatively, if the bot will use only pattern matching, you can select or enter any language for the bot language.

In general, specify the same language for the bot and its integrations, i.e., the intent domains and the knowledge bases that the bot uses, if any.

**Messaging bots**

You can select any of the following for the bot language:

* Arabic
* Chinese (Cantonese) — Hong Kong
* Chinese (Simplified) — China
* Chinese (Traditional) — Taiwan
* Danish
* Dutch — Netherlands
* English
* English — Australia
* English — Canada
* English — Great Britain
* English — India
* English — United States
* French
* French — Canada
* French — France
* German — Germany
* Hebrew
* Hindi
* Indonesian
* Italian — Italy
* Japanese — Japan
* Korean — Korea
* Norwegian
* Polish
* Portuguese — Brazil
* Portuguese — Portugal
* Romanian
* Russian
* Sinhalese
* Spanish
* Spanish — Latin America
* Spanish — Mexico
* Spanish — Spain
* Swedish
* Thai
* Turkish — Turkey
* Ukranian
* Urdu
* Vietnamese
* Other

**Bot Basics**: <https://developers.liveperson.com/conversation-builder-bots-bot-basics.html>

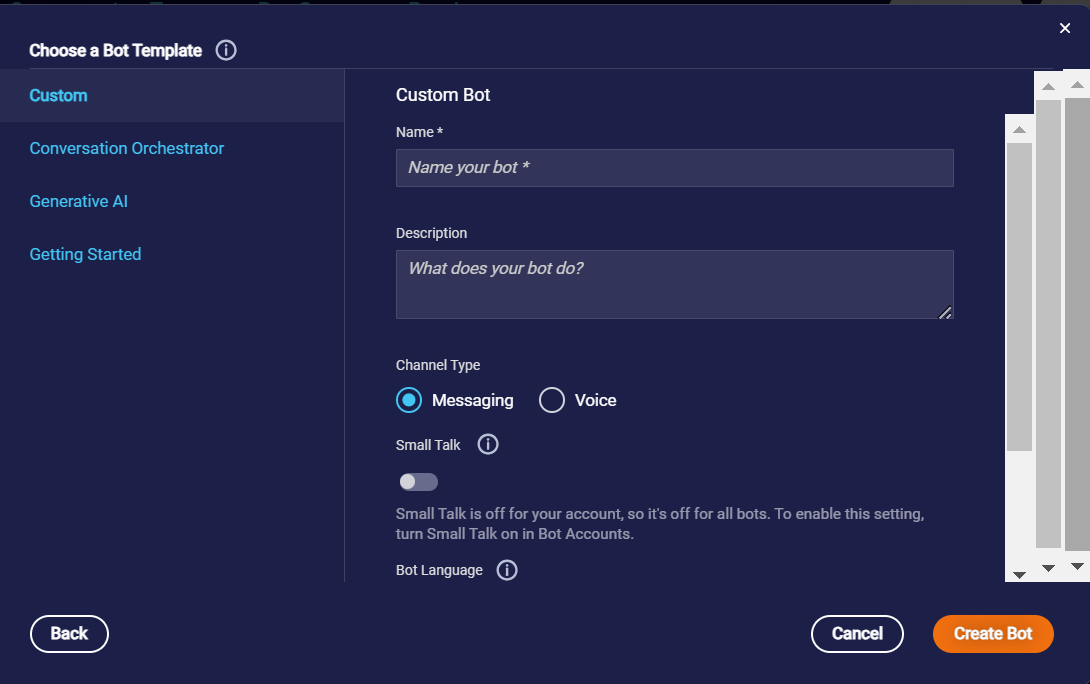
**Bot Templates:**

LivePerson Conversation Builder bot that you create is based on a bot template. There are two general types of bot templates:

* Custom
* Industry-specific

**Custom Bot Template:**

Use a Custom bot template when you want to create a bot from scratch.



There are two templates available:

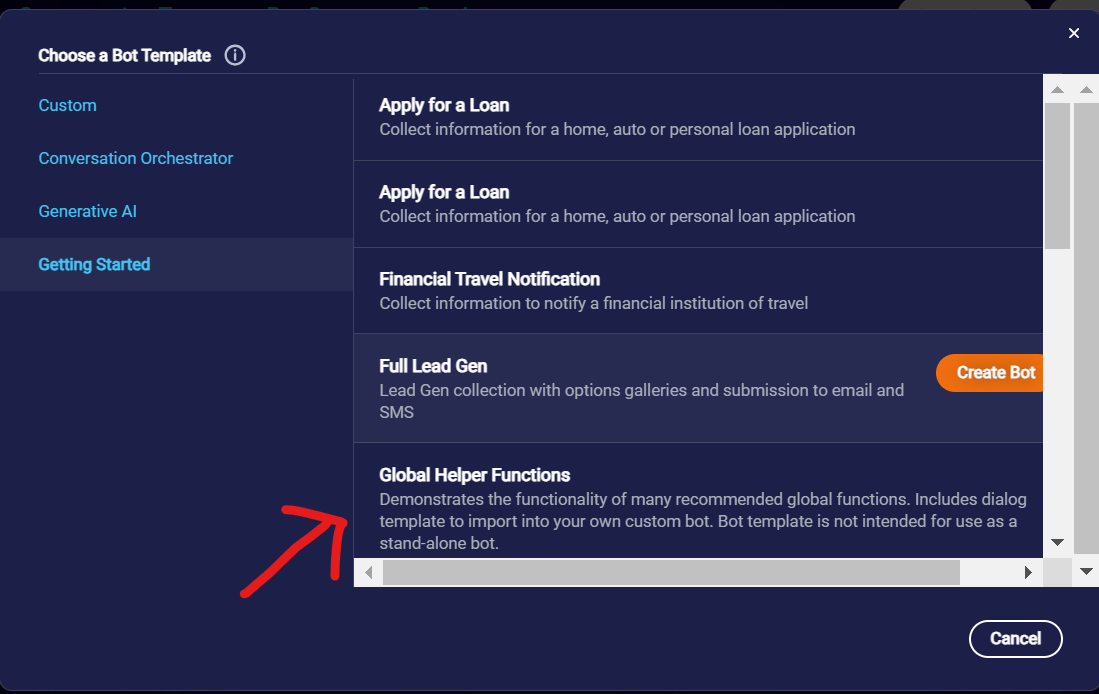
Custom Bot: Use this template to create a bot from scratch that accomplishes a business function, such as fetching a consumer’s order status or creating an account for the consumer. When you create a bot based on this template, the bot is configured minimally. For example, it includes just a Welcome dialog and a Fallback dialog.

Survey Bot: Use this template to create a post-conversation survey bot from scratch. Here again, the bot is configured minimally. These bots include special interactions and functionality designed to support a post-conversation survey experience. For example, you can include predefined interactions for capturing standard survey metrics like Customer Satisfication (CSAT) and Net Promoter Score (NPS).

**Industry-specific bot templates**

To support common use cases, Conversation Builder offers a set of predefined, industry-specific bot templates, and bot templates for general use cases like routing or FAQs. These enable rapid adoption of automation. Each template can contain:

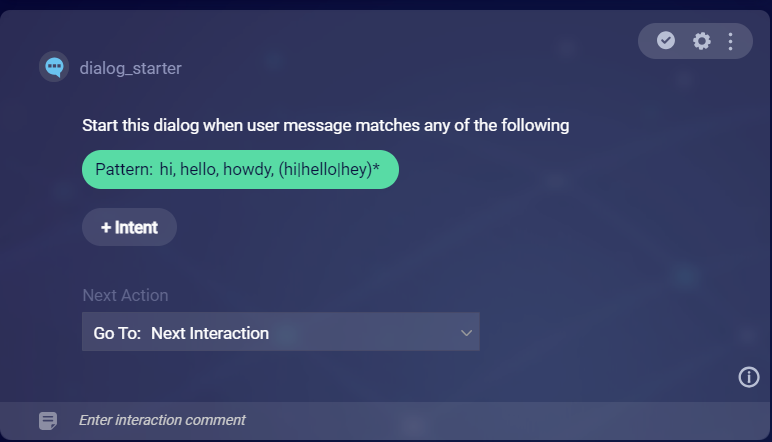
* The intents associated with the use case
* The conversational dialogs with appropriate interactions for the respective channels or end points (Web, Apple Messages for Business, SMS, etc.)
* The API integrations with industry-leading services (e.g., Shopify in Retail) and brand-specific services
* Create a bot from an industry-specific template
* From the dashboard that lists your bots, click New Bot in the upper-right corner.
* Select the bot template.
* Review the template’s description, and click Create Bot.



**Dialog Basics:** <https://developers.liveperson.com/conversation-builder-dialogs-dialog-basics.html>

**Interaction**

* **Dialog Starter:** Dialog Starter interactions are how dialogs are initially triggered, so most dialogs start with a Dialog Starter interaction. (Fallback dialogs work differently.) A dialog can have only one Dialog Starter interaction, and it must start the dialog.
* In a Messaging bot, it’s the consumer that starts things off by supplying some input: a message or a question. In response, the bot tries to match the input with either a pattern or an intent in a Dialog Starter interaction in one of its dialogs. If a match is found, that dialog is triggered, and its flow begins.

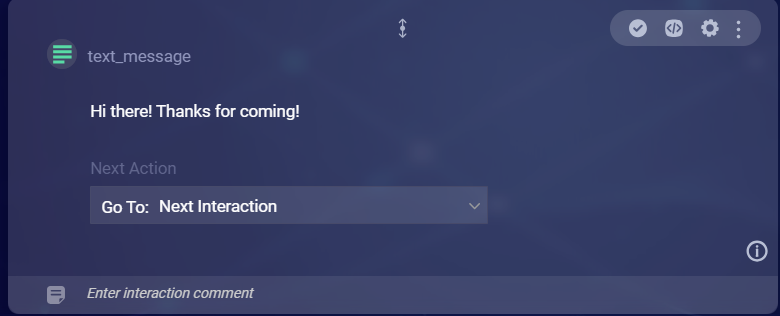


In this dialog starter to match user input you can add pattern or intent matching.

**Statements Interactions**

Statement interactions simply display information and then execute the next action. They don’t expect or wait for a user response.

**Text:** Text statements send the message provided, for example:

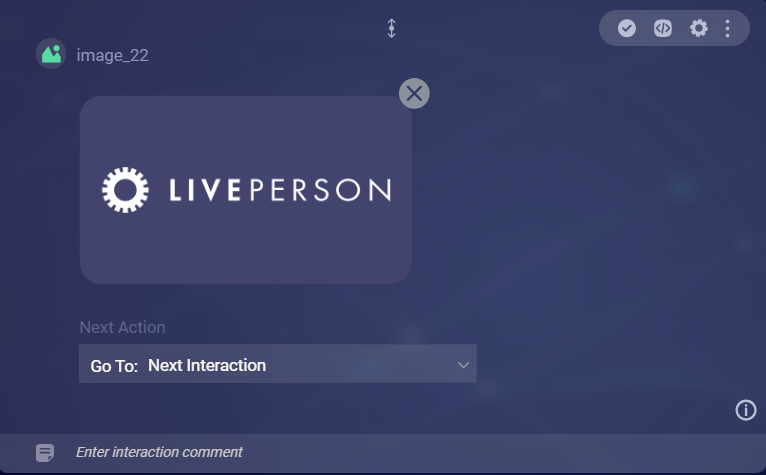
****

On the face of the interaction, enter the message you would like to send.

Text statements can display dynamic values through the use of variables.

**Image:**

Image statements send a single image.

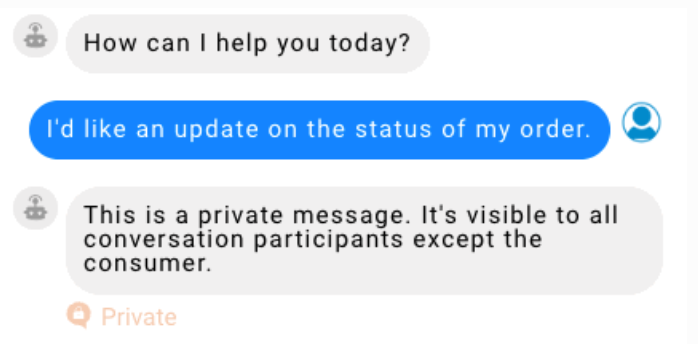


**Private message:**

Private messages are messages that are visible to all conversation participants except the consumer(user).

Human agents can send private messages within a conversation. This allows agents to privately seek guidance from managers. In turn, it also allows managers to provide that guidance privately. More on this in our Knowledge Center.

Bots can send private messages too. You can design a bot to send a private message at any time in the conversation flow. Typically, though, you'll want to do this just before a transfer to a human agent.



More about private message: <https://developers.liveperson.com/conversation-builder-interactions-statements.html>

**Questions:**

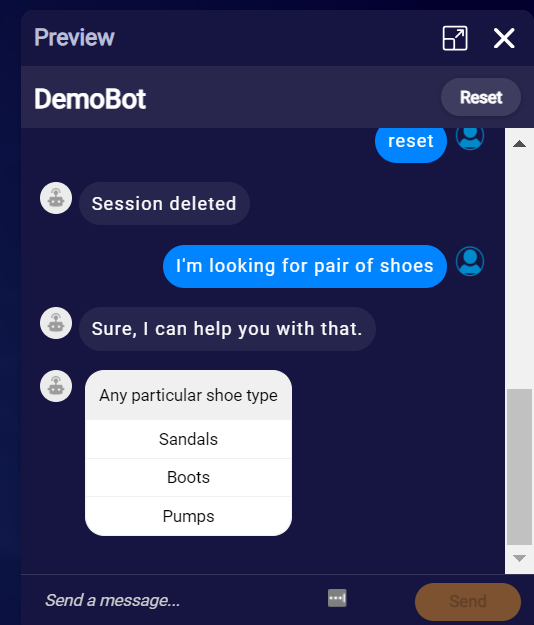
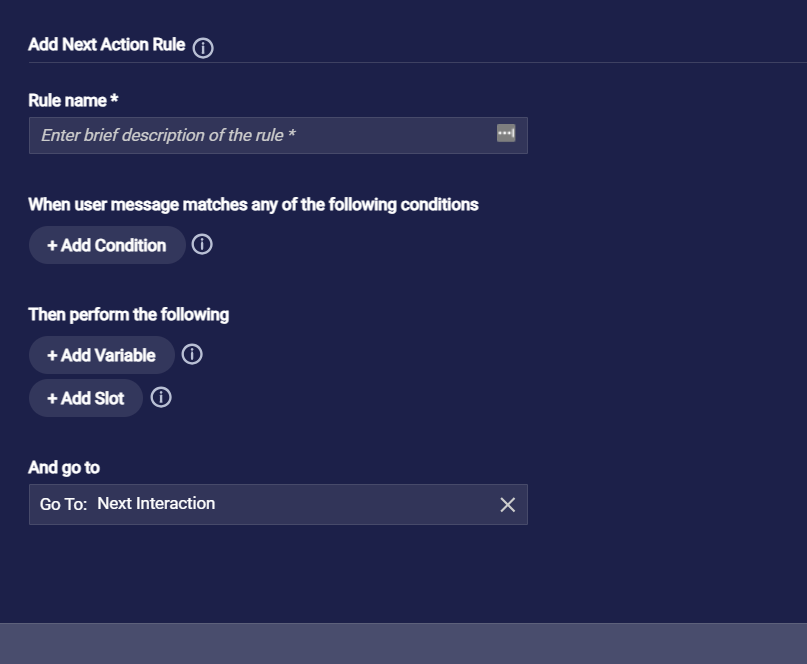
Questions present information to the user (a question that expects a reply of some kind, a list of things to pick from, etc.), and they expect and wait for a user response before executing the next action.

With a question, you can take the user’s response, evaluate it against a condition (i.e., does it match a pattern, an intent, a regular expression, or an exact value?), and then act accordingly. For example, if you ask the user for a 7-digit account number, you’ll likely want to perform a check that the user entered exactly 7 numbers. If the user did, you can then safely pass that value into an API call or perform some other action with it

User responses to question interactions can be saved in variables, making them available for future use.

**Multiple Choice:**

Multiple choice questions let the consumer select an answer from a list of choices.

** **

In Custom rule you can match condition like Intent, exact value, options and regular expression matching.

You can also store the user response in variable under the add variable section using variable syntax.

You can also set up a next action.

**Text:**

Text questions expect and wait for a text-based response from the consumer.

Enter the question to send. The maximum character length is 255.

More question interaction check here: <https://developers.liveperson.com/conversation-builder-interactions-questions.html>

**Integration:** We can do Knowledge AI, File upload, API Integration, Agent Transfer and many more on LivePerson.

Detailed document: [https://developers.liveperson.com/conversation-builder-interactions-integrations.html](%20https:/developers.liveperson.com/conversation-builder-interactions-integrations.html)

**Variables:**

You can store data that you’ve collected throughout a bot conversation with a consumer in variables. Variables are more commonly used than slots. Use a variable when you simply want to save a value.

**System variables**

There are several system variables that store information that's commonly needed in use cases. You can use these variables in your interactions:

{$chatBotId}: Returns the ID of the bot.

{$chatBotUserId}: Returns the ID of the consumer.

{$chatBotUserPlatformId}: Returns the ID of the bot user agent. This is provided by Conversational Cloud.

{$conversationId}: Returns the ID of the current conversation. This is provided by Conversational Cloud.

{$firstname}: Returns the first name of the bot user agent. This is provided by Conversational Cloud.

{$quickReplyPayload}: Returns the quick reply payload for the current interaction.

{$userMessage}: Returns the current user message.

**Store the consumer's response**

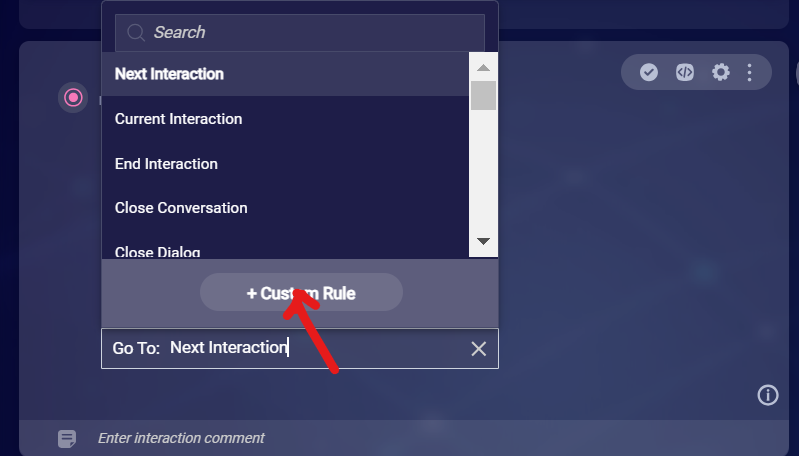
The most common use case for variables is storing consumer responses to questions. Frequently, you’ll want to capture what the consumer just said as the value of a variable. You can use {$userMessage} to do this, for example:



You can also use {$query} in the same way; it works like {$userMessage}.

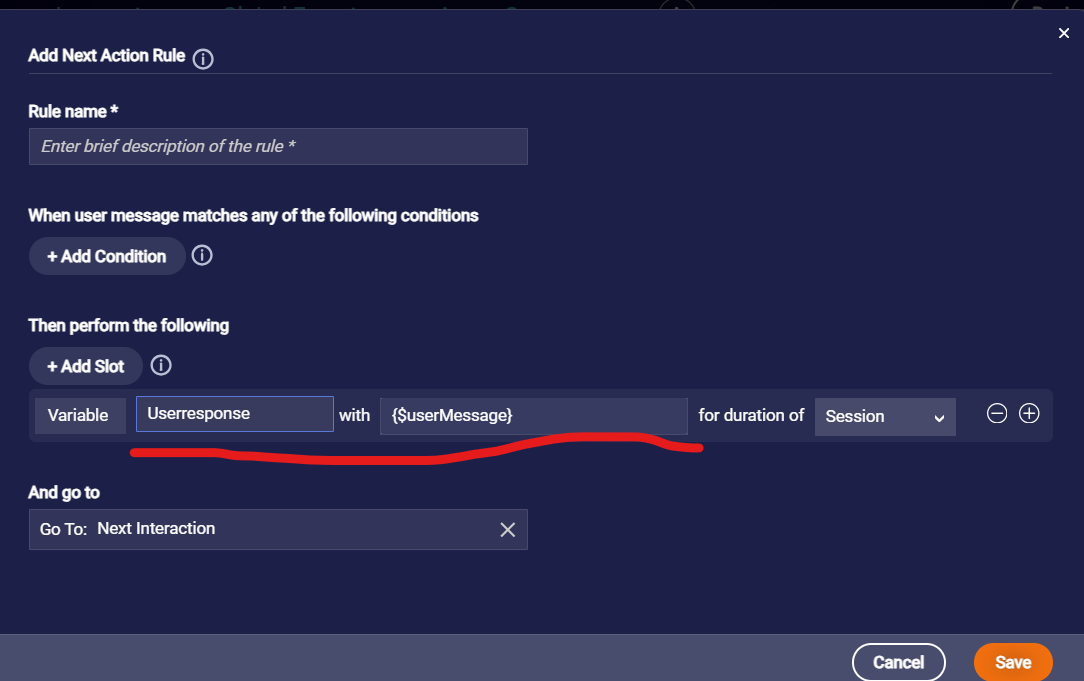
**Add a variable**

In the interaction, click Next Action, and then click + Custom Rule.



The Custom Rule button that appears at the bottom of the dropdown when you click Next Action

In the window that appears, define the custom rule. To add the variable, click + Add Variable, and then define the variable using the fields provided.



For the variable's value, you can specify:

A system variable, for example: {$userMessage} or {$firstname}. See the list of available system variables farther above on this page.

Another botContext variable, for example: variableName. Don't enter {$botContext.variableName} here. That's the syntax for referencing botContext variables elsewhere. A string, for example, the number “1” or the phrase “vip.”

If you like to access bot variable in anywhere across the flow you can use below syntax: {botContext.variablename}

**Get and set bot variable:**

Use the following built-in functions to get and set session data.

Use the setBotVariable function to set the value of a bot variable, so you can subsequently use it in the bot flow. By default, the function stores the value in request scope, but you can specify the scope as a parameter.

Use the getBotVariable function to get the value of a bot variable. Bot variables that are not set return NULL.

Note: “Is the data stored in its original type or as a string?” The answer is, “It depends.” That is, if you set a variable and then retrieve it within the same message scope, the data is stored and retrieved in its original type.

**Example**

In the example below, we're using getBotVariable to retrieve a string. Then we're converting it to an integer before using it as an integer.

// retrieving an integer

var count = botContext.getBotVariable('Howmanyitems');

count = count\*1;

if (count > 10) {

botContext.sendMessage('You have more than 10 items!');

} else {

botContext.setBotVariable('Howmanyitems',0,true,false);

botContext.sendMessage('Sorry, you do not have any items with you… ');

}



**Get current user message**

getCurrentUserMessage is used for getting the most recent message from the user, whether typed or tapped (buttons or quick replies).

Check- here for in-built functions list: [https://developers.liveperson.com/conversation-builder-scripting-functions-get-set-session-data.html#get-and-set-bot-variable](https://developers.liveperson.com/conversation-builder-scripting-functions-get-set-session-data.html%23get-and-set-bot-variable)

<https://developers.liveperson.com/conversation-builder-scripting-functions-functions-list.html>

**Global Functions:**

**Global functions are functions that you define in the Global Functions area of a bot.**

**Global functions have two key benefits:**

Global functions are functions that you define in the Global Functions area of a bot.

Global functions have two key benefits:

* Code reuse
* Initialization of the conversation

First, any function that you define in Global Functions can be called from anywhere else in the bot, i.e., from any interaction in any dialog. This allows you to easily reuse code.

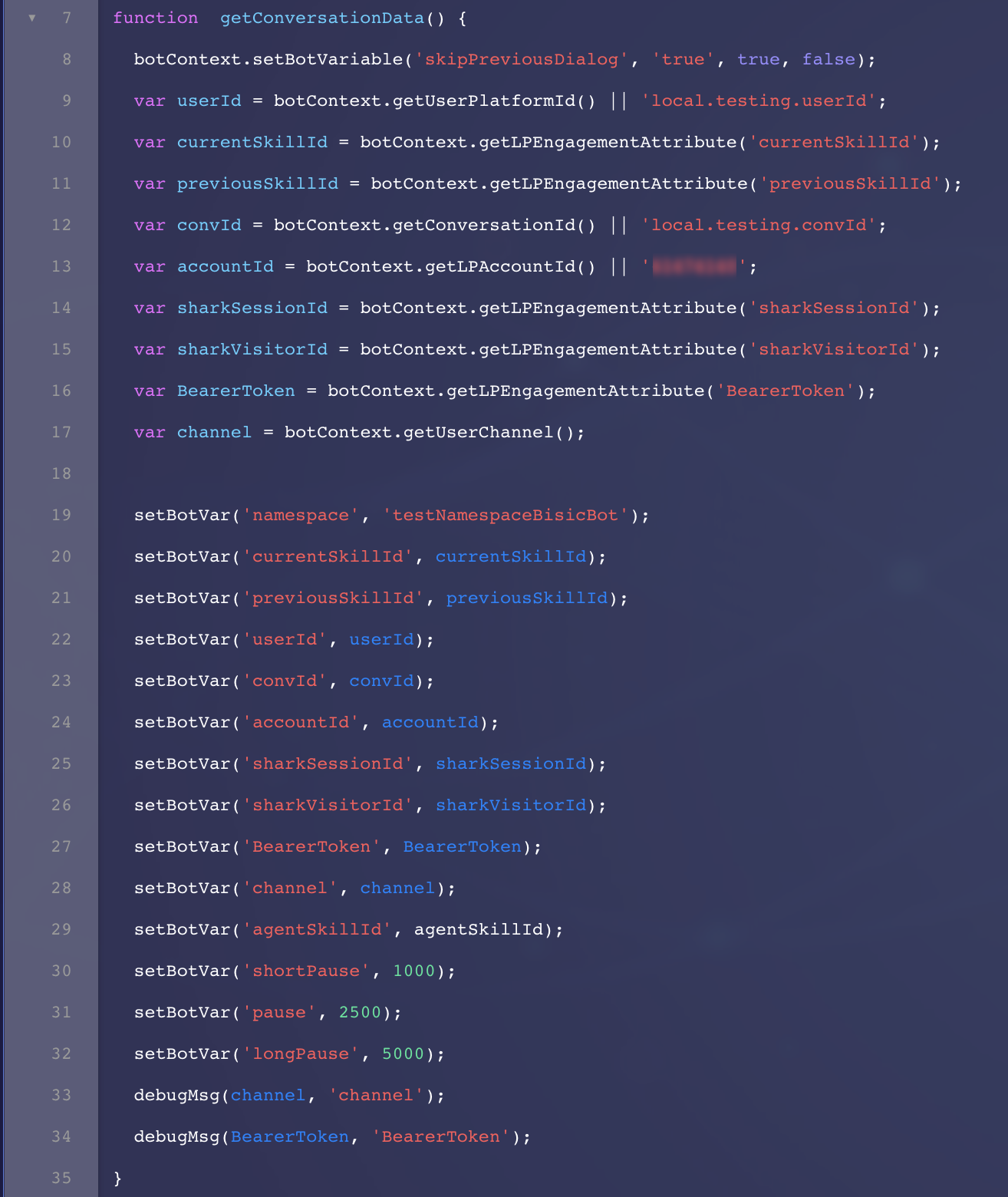
Second, a bot’s Global Functions includes an empty initConversation function by default. initConversation is run immediately when the conversation begins, so it’s a good place to do anything that needs to be done at that time, for example, to run a function or to initialize variables that you’ll need later.

Often, bot developers try to initialize variables just in time, which can sometimes be too late in the execution flow. For example, it’s recommended that you obtain the bearer token needed for an API call at the conversation’s start instead of in the Pre-Process code of the Integration interaction. Using initConversation for this kind of work ensures the variable is available when you need it.

Take care when writing JavaScript for global functions. Errors might not be thrown in all, true error cases. So, when you experience erroneous bot behavior, sometimes it can be difficult to diagnose a problematic global function as the root cause.

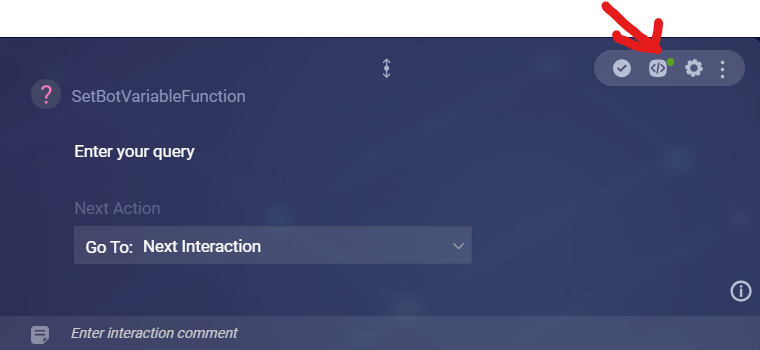
Store functions in Global Functions, so you can access them from anywhere in the bot.

Get and save system data (user data and session data) in the initConversation function, so it can be accessed from anywhere in the bot. Keep in mind the requirements and best practices for custom code.



**Configuration – Custom Code:**

Custom code indicator: An interaction that contains custom code displays a green dot.



Check for this to understand at a glance whether there's code in the Pre-Process Code, Process User Response code, or Post-Process Code in the interaction.

You'll also see this green dot next to the name of any code tab that contains custom code:



**Access the Custom Code panel**

Select the interaction.In the interaction's upper-right corner, click Custom Code icon (Custom Code icon).

This displays the Custom Code panel.

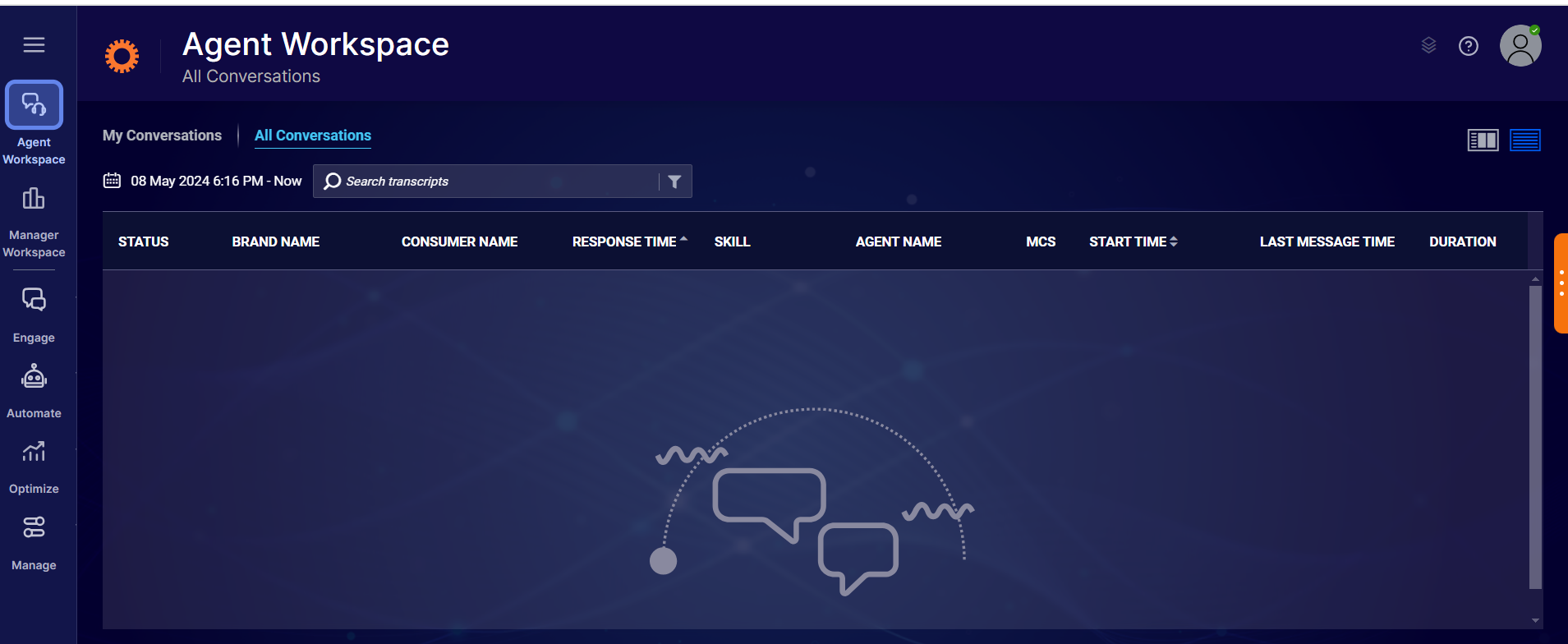
The Process User Response panel for entering custom code

There are three tabs for adding code:

* Pre-Process Code
* Post-Process Code
* Process User Response

Use the desired tab to add the code. You can use the built-in scripting functions to access variables and manipulate data.

**Agent Workspace:**

****

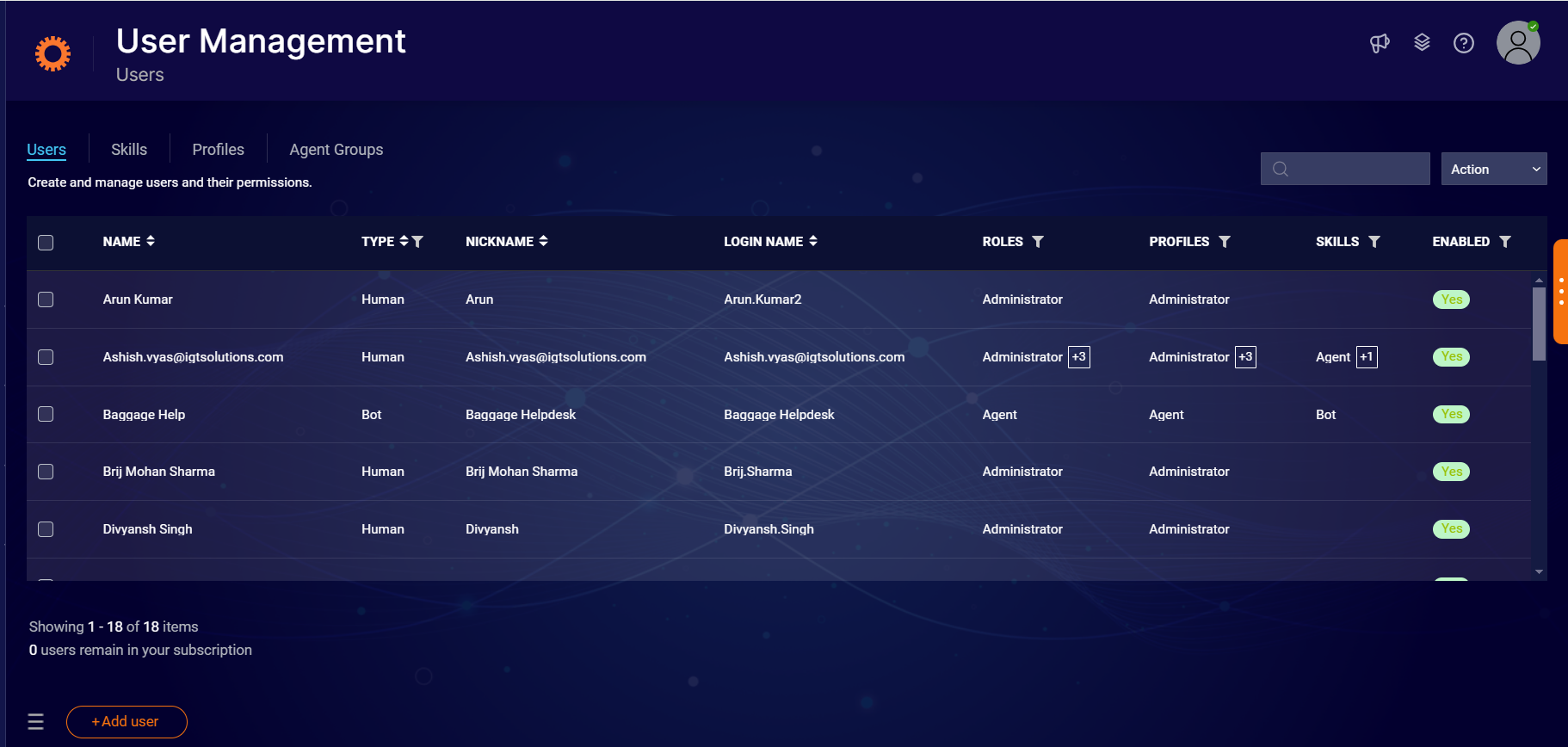
In this section you can filter out present/past conversation on using specific date, conversation ID which are triggered by user/consumer and check out the visitor’s information and overall conversation happen between about bot and users.

For Live chats agent will notification about user wants to connect with agent based on skill we added on the flow it’s triggered or connected to specific agent.

We are able to see the skill name and few more details on this conversation logs under agent workspace.

**Manager Workspace:**

**Users and skills**

****

At this section you can create user profile to specify as different role like agent, agent manager etc..

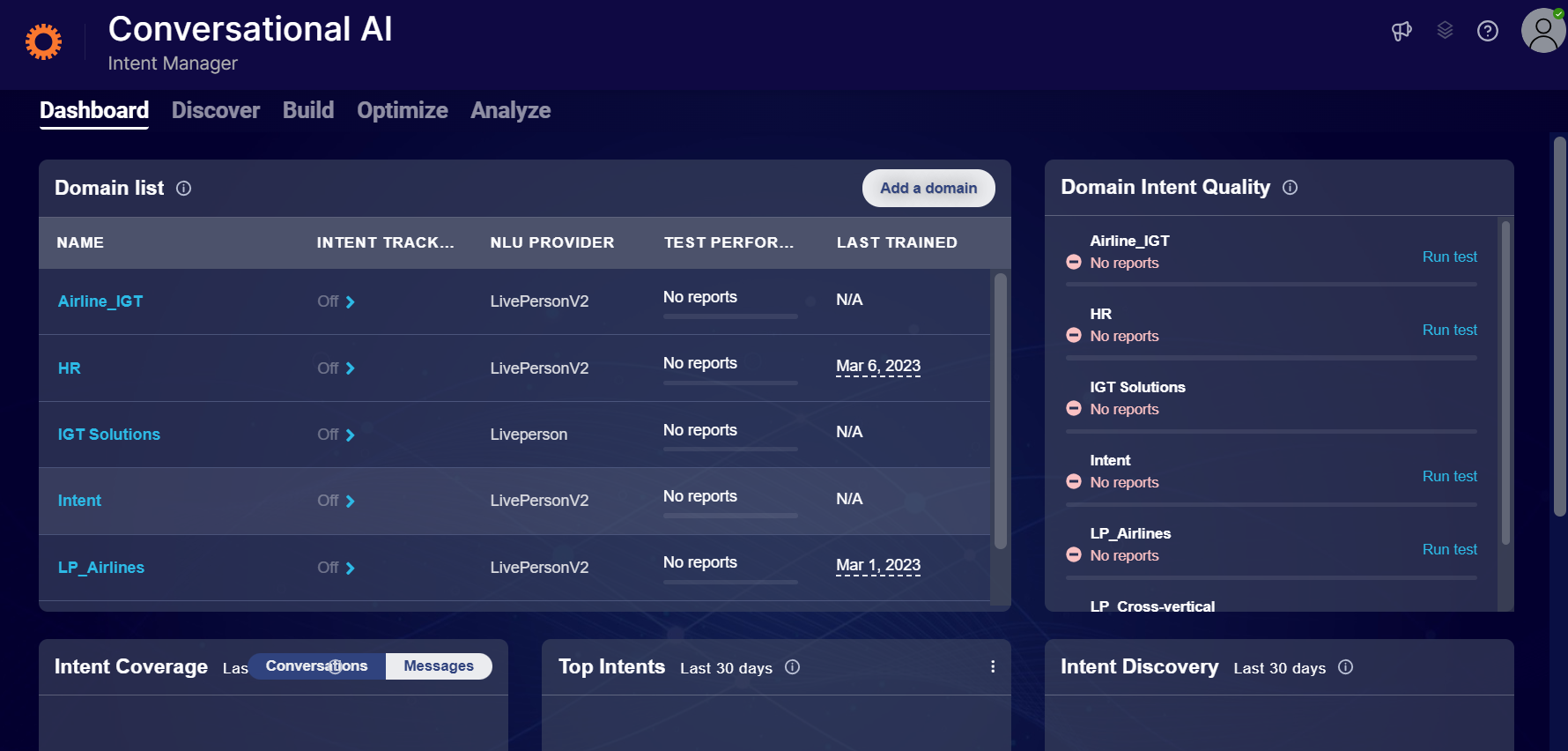
Skill: you can create specific skill according to your flow requirements.

Check this doc on users and skill creation:

<https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-deploy-the-bot.html>

<https://developers.liveperson.com/tutorials-getting-started-with-bot-building-messaging-escalate-to-an-agent.html>

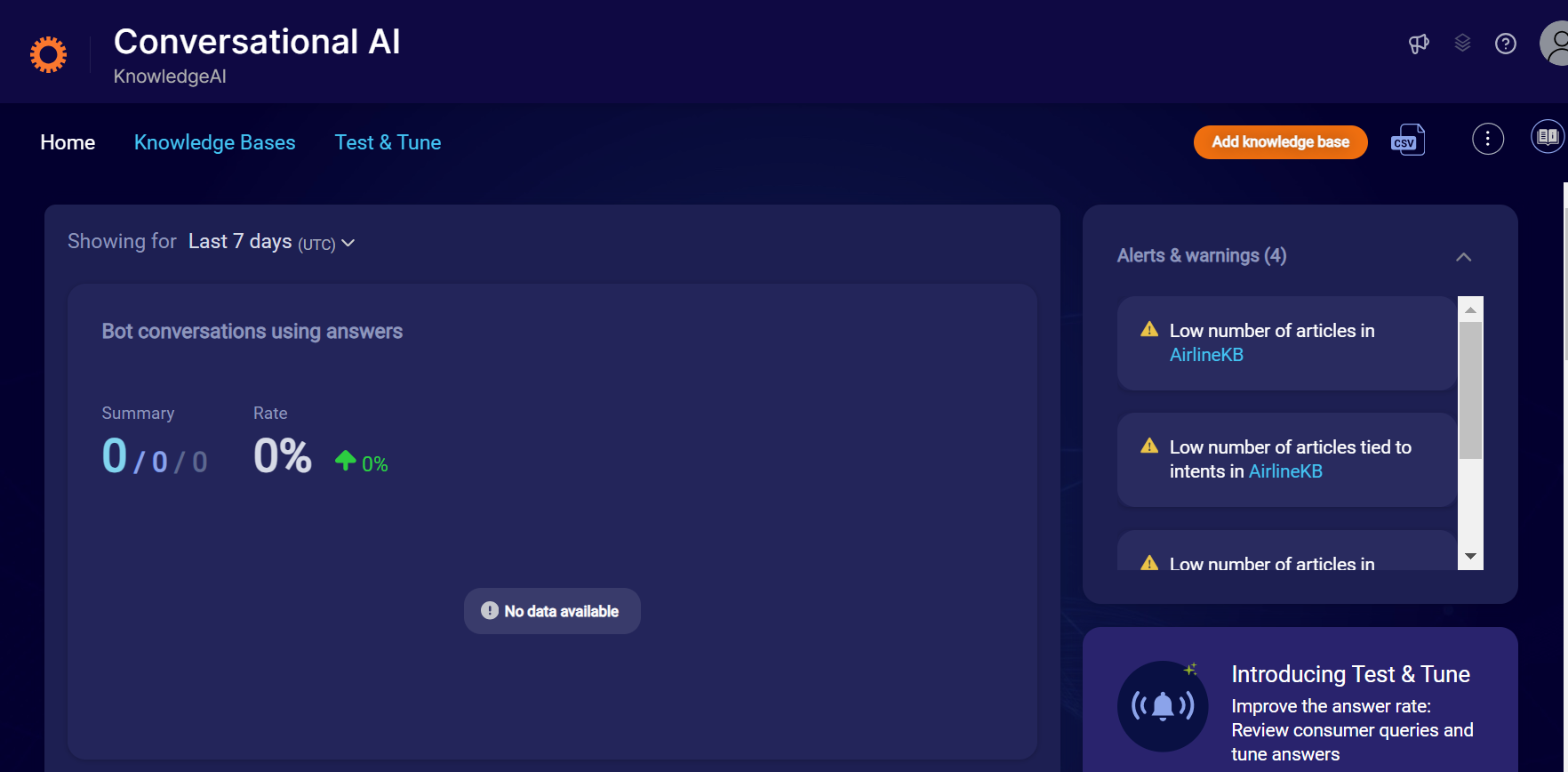
**Intent Manager:**



In this section you can create Intent, Entity and train your model for your bot.

Also you can Analyze your model and test the Intent and Entity.

**Knowledge AI:**

****

In this section you can create FAQ bot and make use of Integration to connect your knowledge AI to your bot.

You can add or create articles here and later test it whether it’s giving write answers what you uploaded under article section.

For more info you can check out this doc: <https://developers.liveperson.com/conversation-builder-integrations-knowledgeai-integrations.html>