CHATBOT IMPLEMENTATION : R4

1. Pre-Requisites:

To use chatbots in Salesforce, the user should have the below licenses:

1. Chat User
2. Service Cloud User

The chatbot and other setup used in the implementation are accessed through ‘Service Setup’.

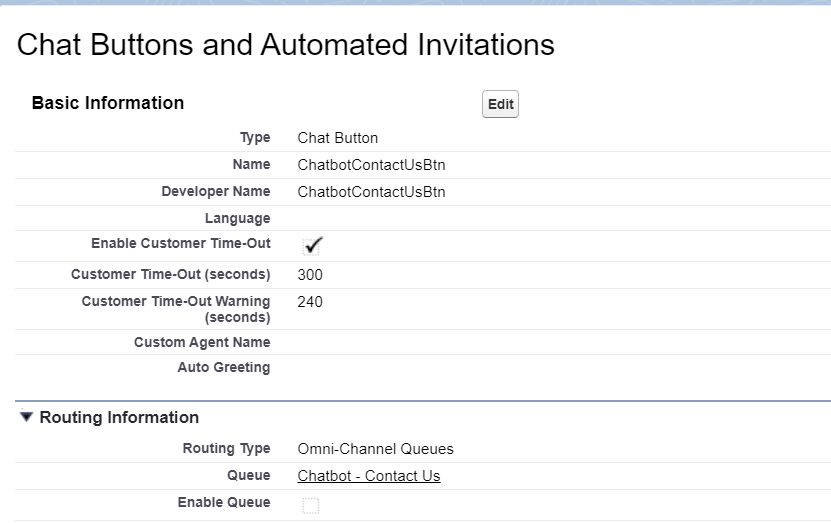
1. Omni Channel Setup:

I have attached another document which explains the Omni Channel setup in detail. Also, enable “Chat” apart from Omni Channel in this setp.

1. Chat Button:

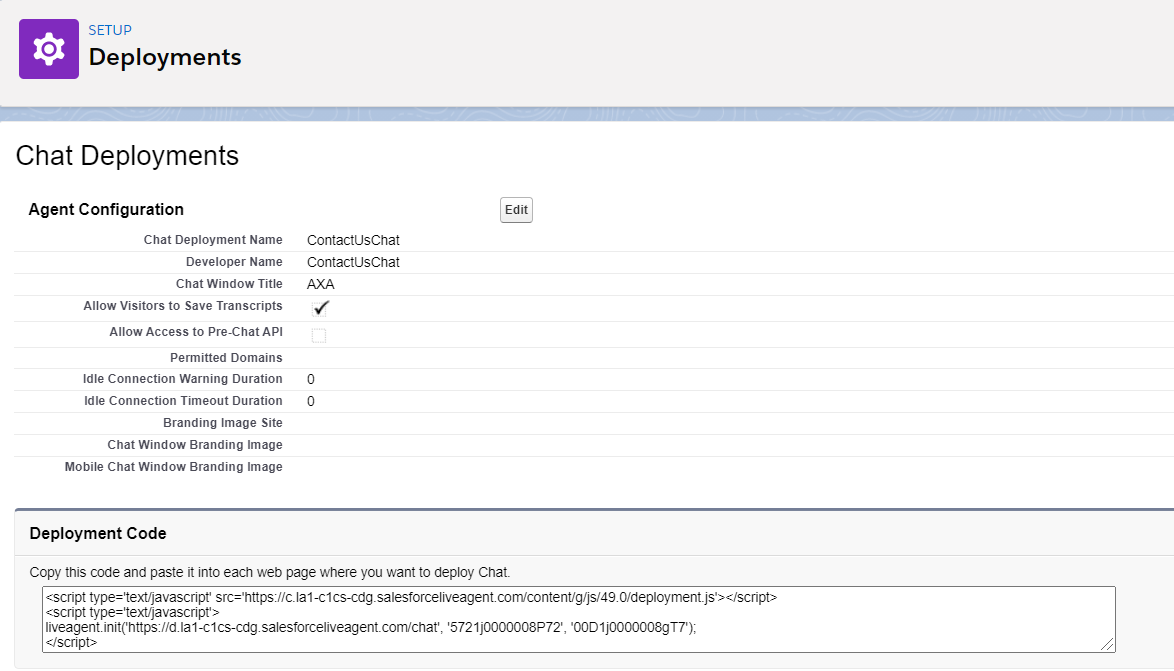
Navigation : Service Setup < Chat Buttons < Select the chat button created as part of Omni Channel Setup.

In the chat button, we enable the customer time out and give the duration for warning message to appear and the duration after which the chat session ends. ‘Customer Time-Out’ should always be greater than the warning time. A default warning message will be shown after the warning time duration if the chatbot is left idle.



1. Create chat deployment:

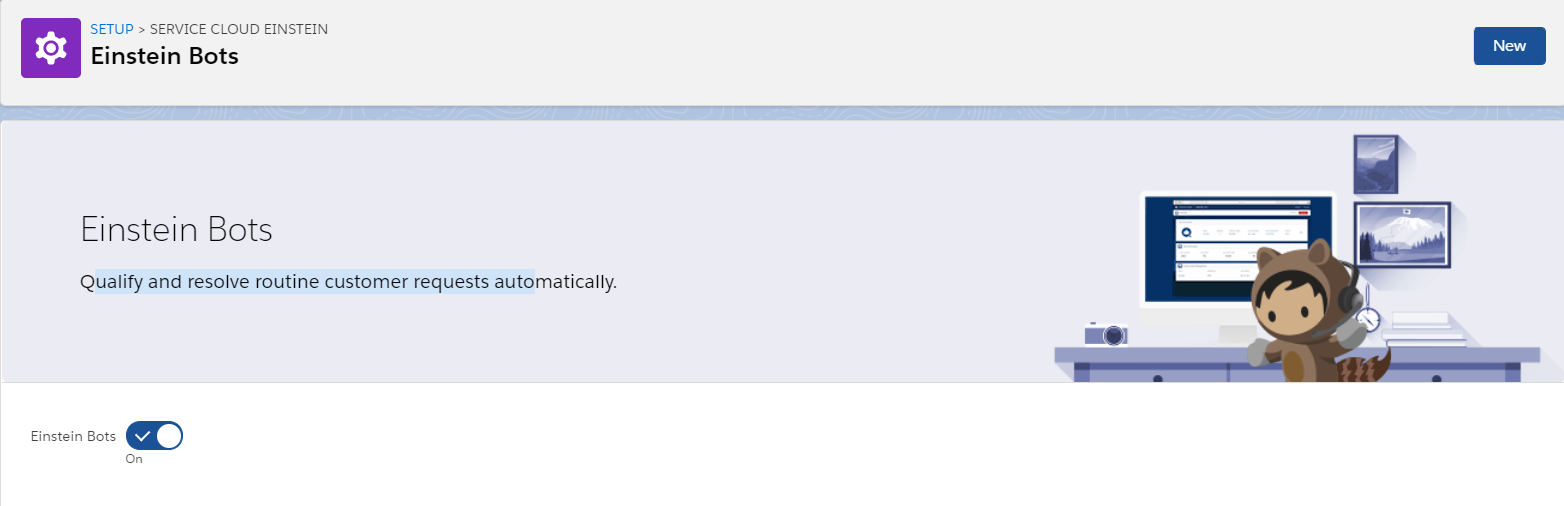
In this step, ensure to check “Allow Visitors to Save Transcripts”. The chat users can down a transcript of their bot conversation.



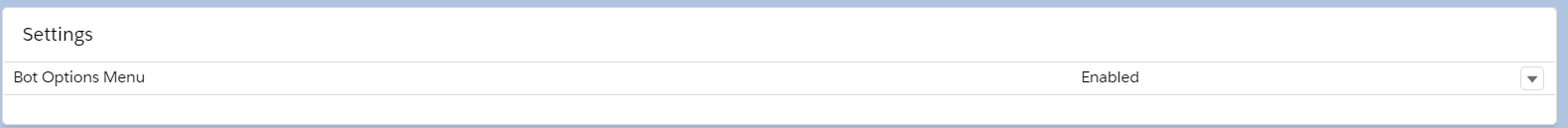
1. Enable Einstein Bot

Navigation : Service Setup < Einstein Bots

Toggle the Einstein Bots option.



Under Settings in the same page, enable “Bot Options Menu”

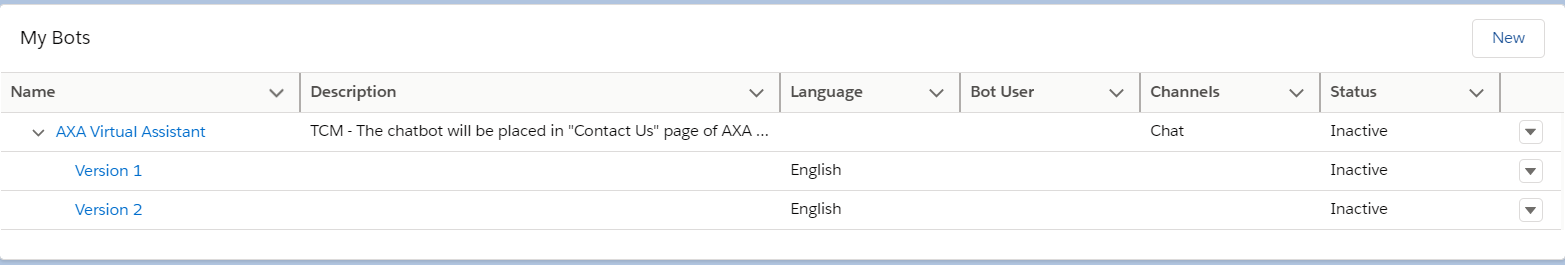


1. My Bots

In the same page as above, the existing bots will be displayed, also, we can choose to create a new Bot. If we must create a new bot, ensure that we redo all the steps again. Each bot / Live agent should have a dedicated “Embedded Service Channel”

Any changes to the existing bot, it is recommended to create a new version of existing bot.

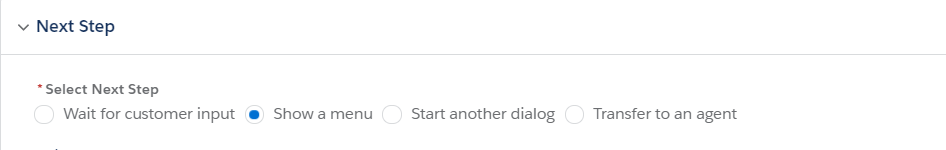
*When we create a new Version of existing bot, it will inherit the bot training (artificial intelligence) from the previous versions. The bot learning can’t be undone. If we want to get rid of bot learning (if behavior of the bot is messy) it is recommended to create a new Bot instead of a new version.*



1. Bots – Dialogs

Based on the chatbot flow charts, the dialogs and messages are created. *(The API names should not contain underscores for any part of the bot implementation as per AXA standards)*

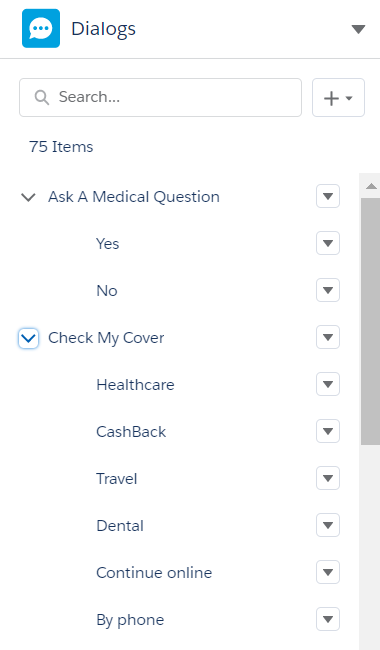
1. At the end of each dialog we could choose to do any of the below actions:



Through the entire implementation of chatbot, we have only used the bot to navigate to relevant contacts / URLs. To should a multiple dialogs after the intended dialog, choose the option “Show a menu”. If we have only one dialog, choose “Start Another Dialog”.

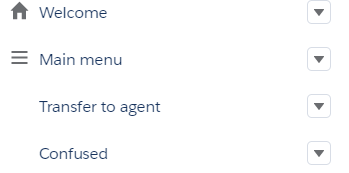
If we have reached end of flow for any new further implementation, place “End Chat” dialog after that.

1. Always place the dialogs in dialog groups. There are multiple dialogs with same name (Ex: ‘Yes’, ‘No’). To avoid any confusion, place the dialogs in appropriate groups.



1. Standard Dialogs:

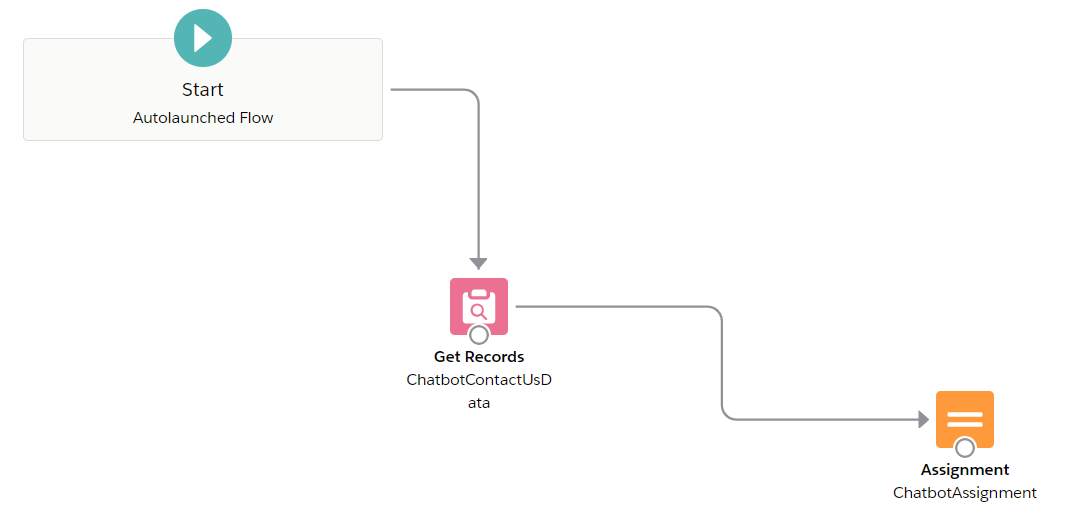
The below displayed dialogs are Standard dialogs and are created as and when the bot is created. We can’t delete these dialogs, but we can customize them.



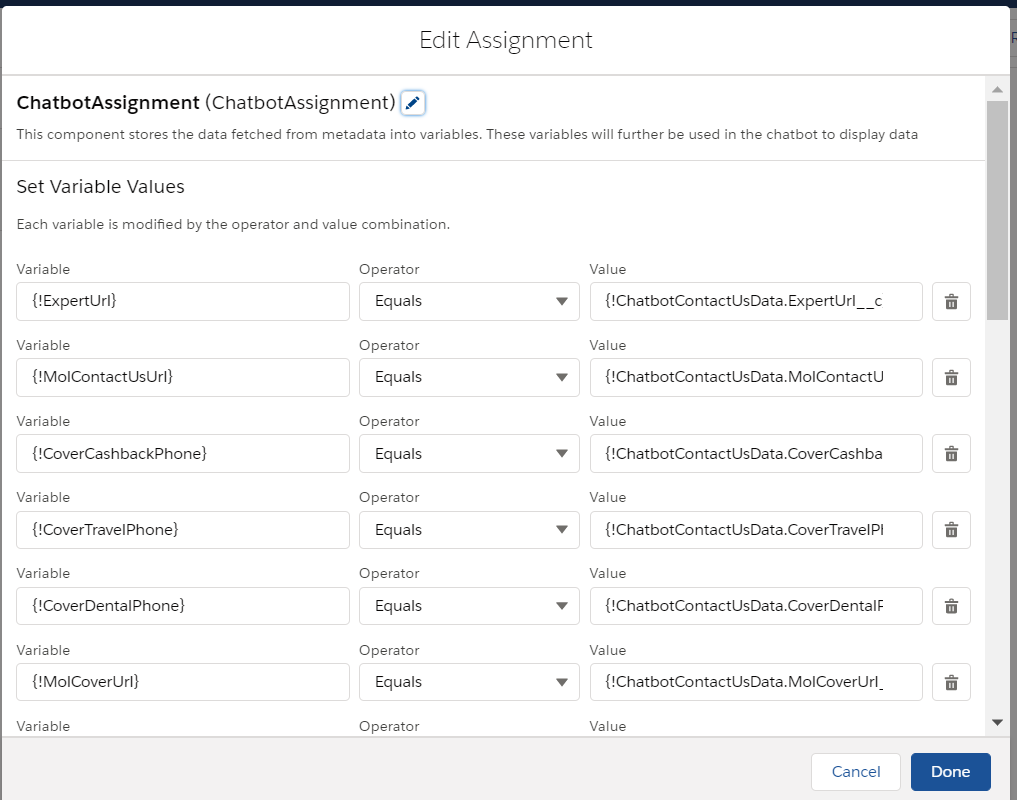
1. Welcome : every time the bot is launched, this dialog is fired.
2. Confused: When the user types in random text and it doesn’t fall into any of the Utterances of intent sets, this dialog will be displayed. The default behavior of this dialog is to display the customized / standard message and then show the previous menu options which the user has accessed. This behavior cant be over ridden.
3. Main Menu : The dialogs which the bot should display as and when the bot is launched should be placed here. The main menu can now also be used in a different menu as we have enabled it in step 5.
4. Dynamic Data for Phone numbers and URLs

The contact numbers and URLS are not used directly in the chatbot messages, but are referred from custom metadata through flow.

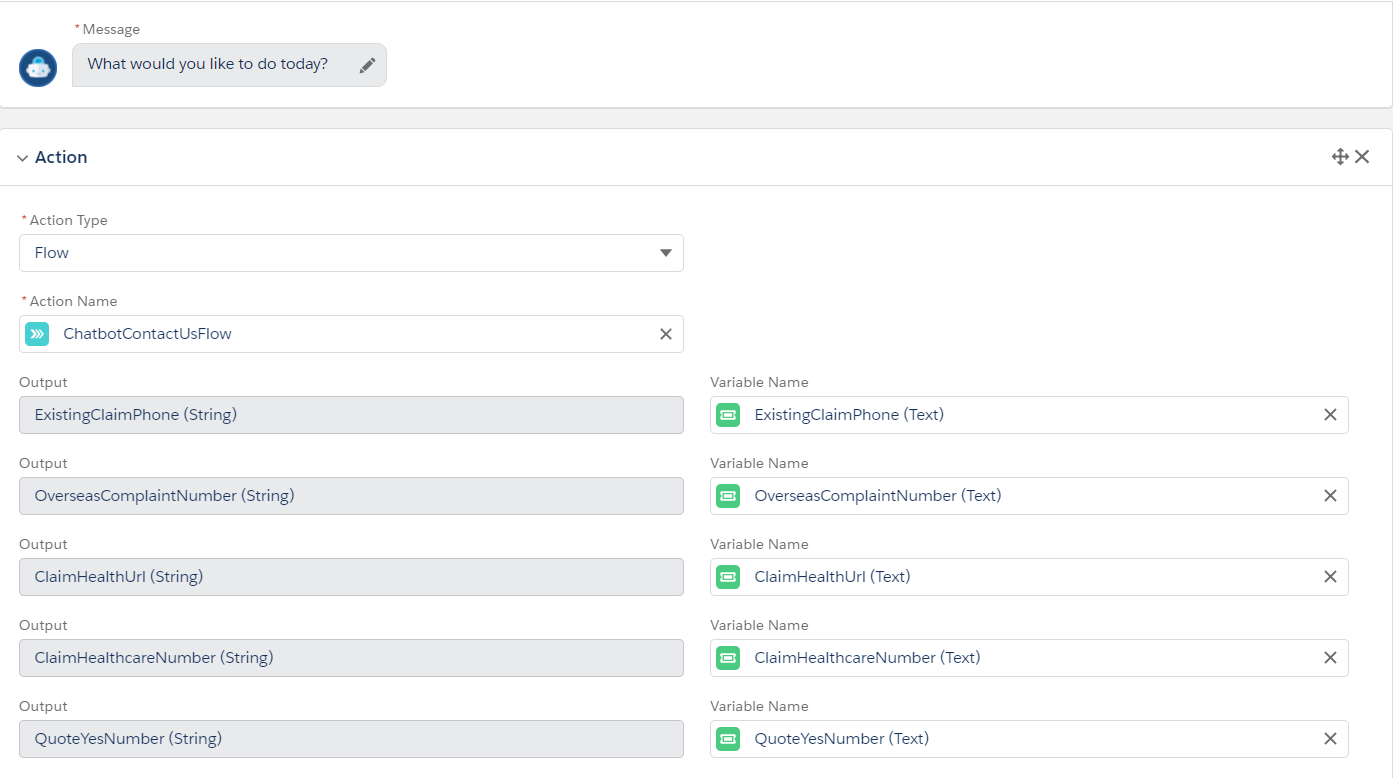
In the flow “[ChatbotContactUsFlow](javascript:void(0);)” we have fetched the record from custom metadata and stored all the values from the fields in distinct variables.



Assignment :

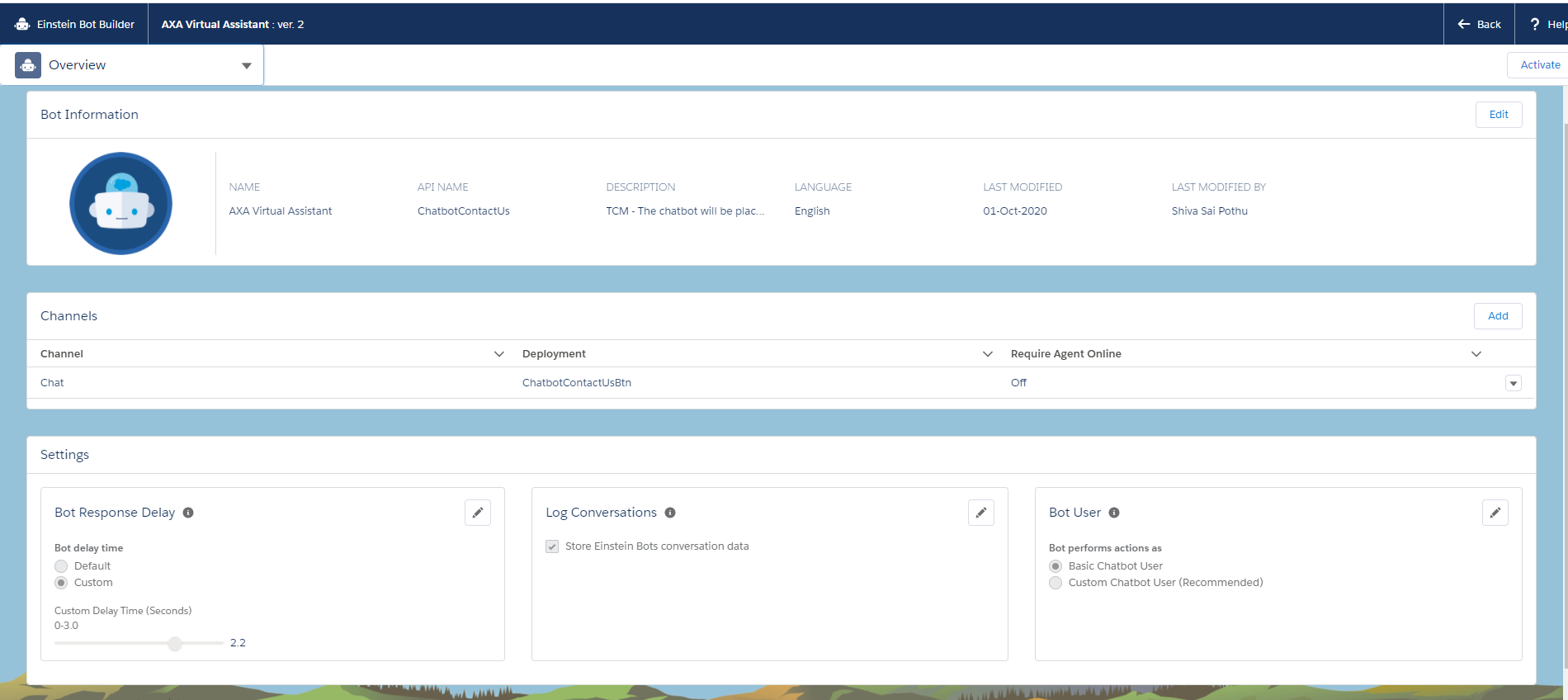


In the chat bot, we have created new variables to hold the data coming from flow. After the greeting message is displayed, we call the above flow and store all the values in fields in the bot variables.



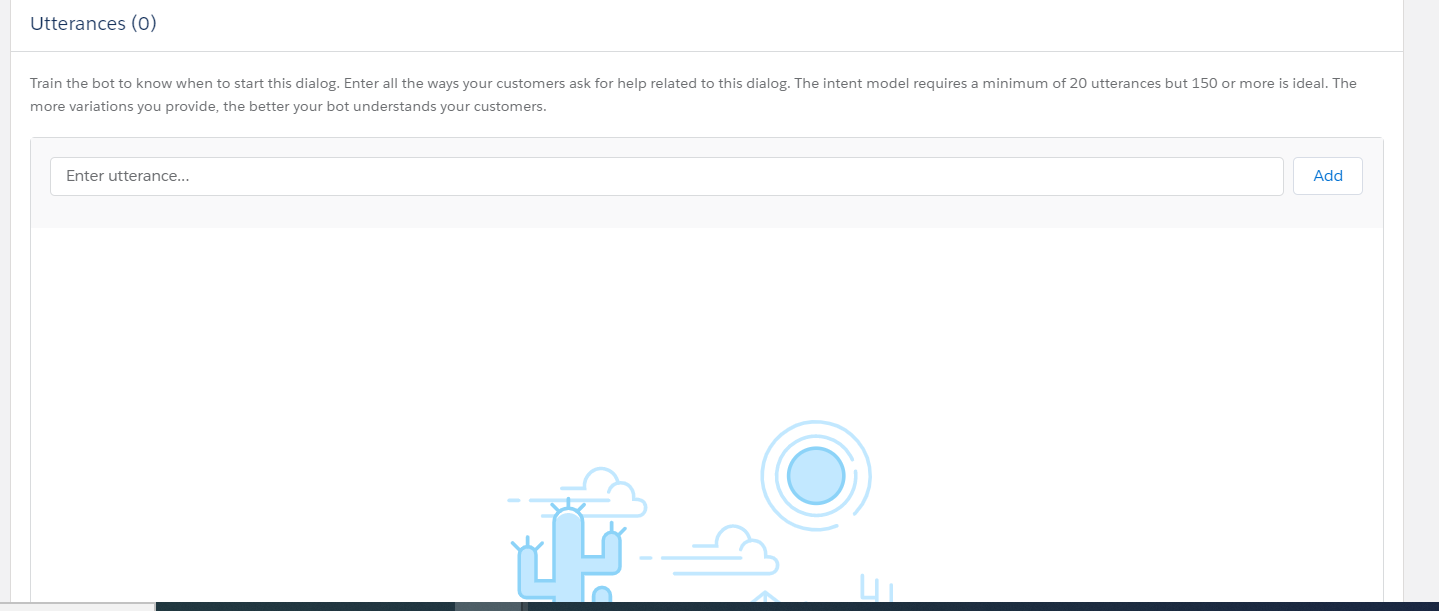
1. Chatbot “Overview”

We can navigate to the chatbot “Overview” Page, through the drop menu from top left of the page.



1. Channels : In the channels section, we can choose to change to a different “Embedded Service Channel” the channel which we populate during the bot creation will be displayed.
2. Bot Response Delay : This would allow us to change the bot response time. We have customized it to respond with the delay of one second than its default behavior. This setting is found in the bottom left of the page.
3. Log Conversations : Enable this option to store the bot conversations into Chat Transcripts.
4. Bot User: The default option used is “Basic Chatbot User”. When we opt for “Custom Chatbot User”, we need to create a user with all required licenses and populate the same user. This would allow us to use the bot in applications like “Field Service Lightning” which is not possible through the other option. Also, it would cost us the additional licenses used for that user.
5. Utterances :

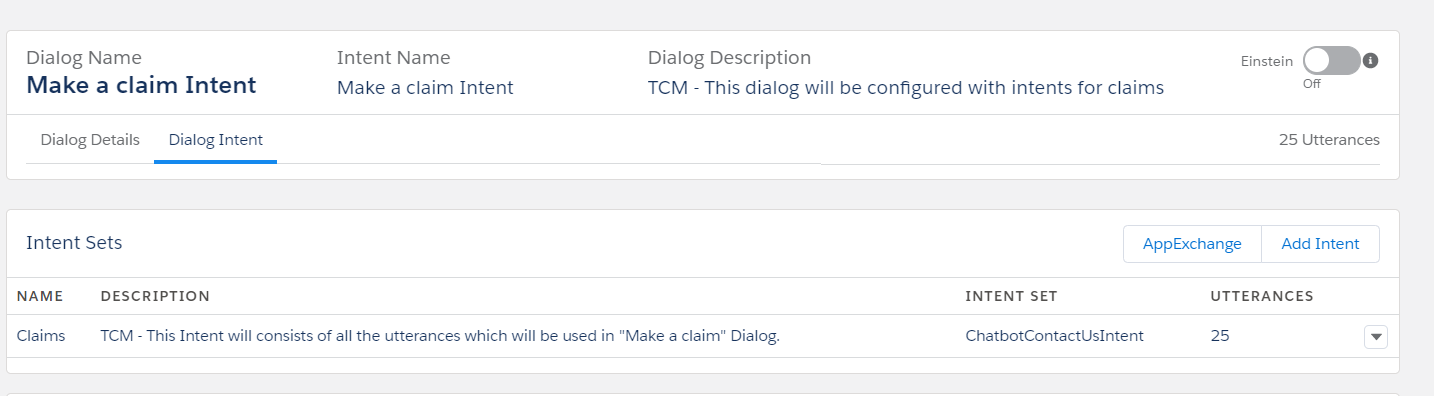
As of now, the user is capable of navigating through the bot only by clicking on the dialogs. When we use utterances, we can enable to user to use free text to some extent. When the user input matches to any of the utterances (exact match), the bot would display the Dialog relevant to the utterance. We can feed the utterances under the dialog intents:



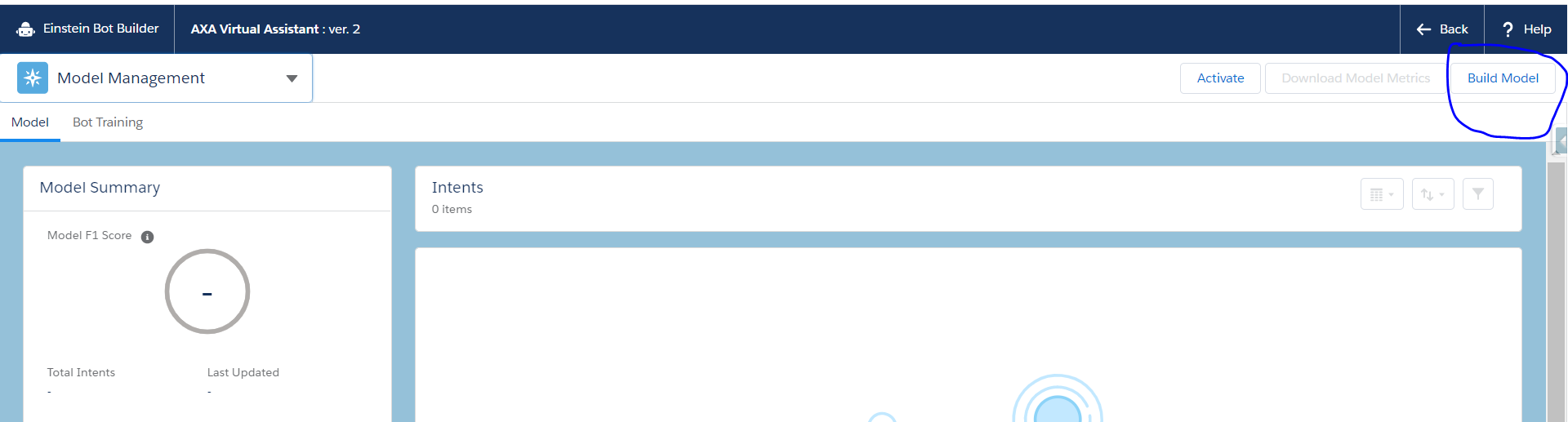
1. Intent Sets :

Navigation : Service Setup < Intent Sets

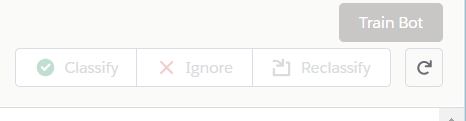
1. If we have to use the artificial intelligence of the Einstein bots, we should opt to use “Intent Sets” we should place atleast 150 utterances in the intent sets and assign the same to the relevant dialogs.



1. Enable the toggle (Einstein) in the dialog Intent page.
2. Top right Drop Down Menu < Model Management : Here, click the option “Build Model” after all the intent sets are populated. It builds an data model based on the intent sets. The processing could take upto 5 mins.



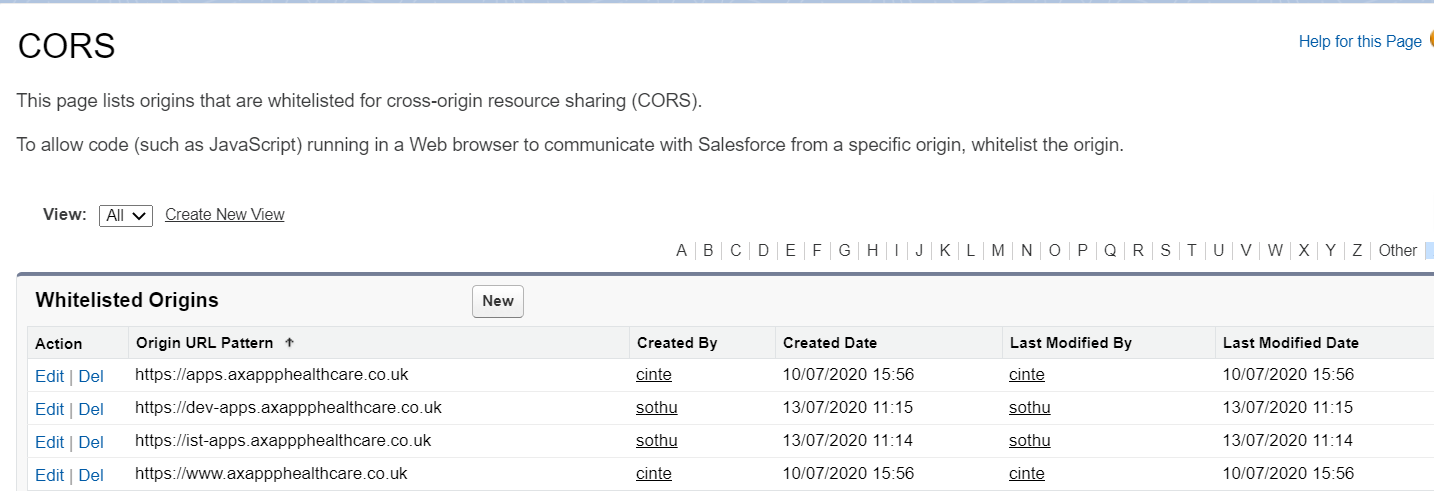
1. Bot Training: Ensure to train the bot regularly for better bot behavior. Bot training is available in the same page as above. As part of training, the page will display the user inputs and the dialog the bot redirected that specific user. We have to mark it as correct if it’s the expected behavior of change it to relevant dialog.



1. CORS

To use the chatbot in any website, the website has to be whitelisted in CORS:

Navigation : Service SetUp < CORS:



1. Chatbot Branding :

Another Document is attached in the chatbot branding

1. Embedded Code Snippet :

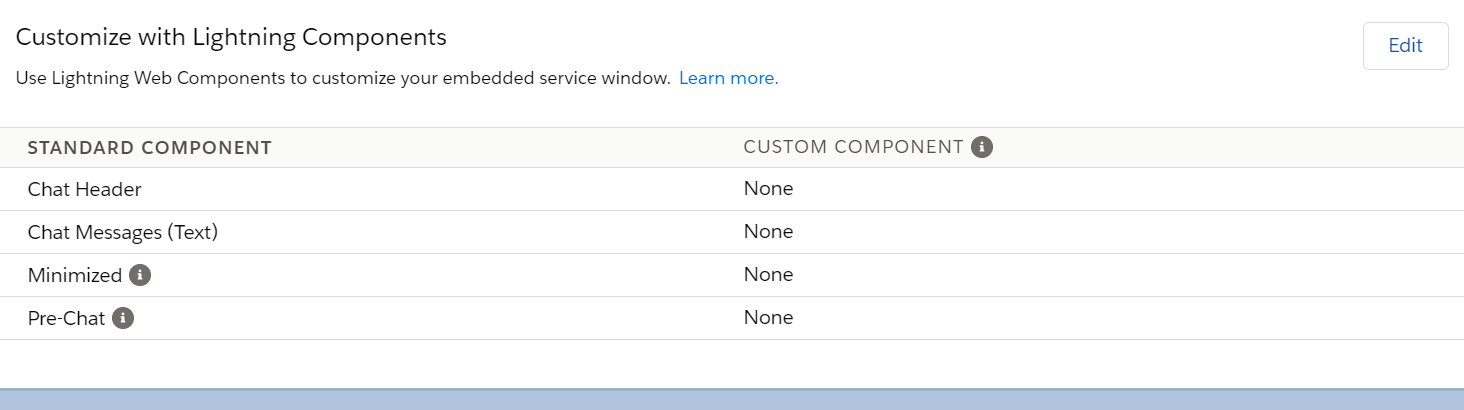
Once the bot is built and ready to be used, we need to place the ‘Embedded code snippet’ in the desired website.

Navigation: Service Setup < Embedded Service Deployments < Choose the appropriate deployment (click ‘View’ from right end drop down) < “Get Code” from Embedded Service Code Snippets

Lot of additional styling are added by the AXA website team to ensure that the chatbot is aligned to the design of the website. We should compare the code from our code snippet are that of the existing one in prod and then make our changes.

None of the changes in the chatbot will affect the code snippet. Only when any of the changes below are done, the code snippet will be changed:

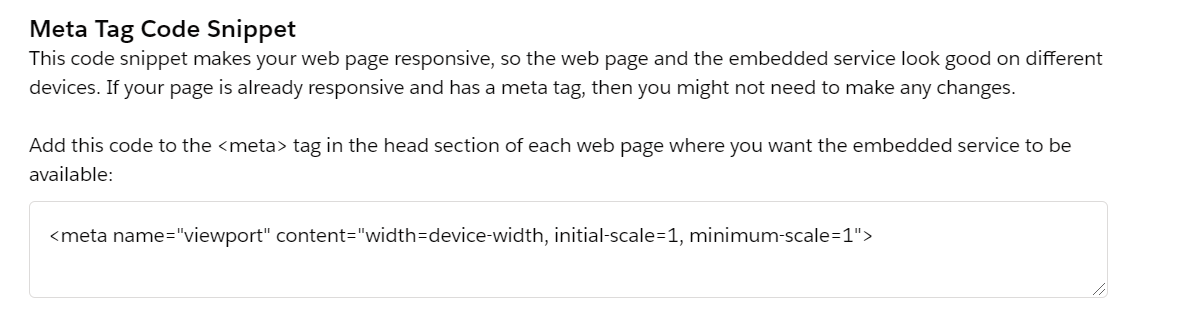
1. Replacing old components with new ones (Embedded service deployment, chat button, chat deployment, etc)
2. Changes in branding (done through Embedded service deployments ex: width and height of chat window, colors used, images placed, change of standard labels and messages)
3. Using new custom components (lwc) to style chatbot:



Navigation for above picture : Service setup < Embedded Service Deployments << Choose the appropriate deployment (click ‘View’ from right end drop down) < Chat settings

1. Meta Code Snippet :

When the embedded code snippet is provided to the website team, we have also asked them to place the below meta code snippet in the header tag of the website:



<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1">

This piece of code snippet has made the chatbot compatible with mobile as well as desktop devices. Another document for the same POC is attached.

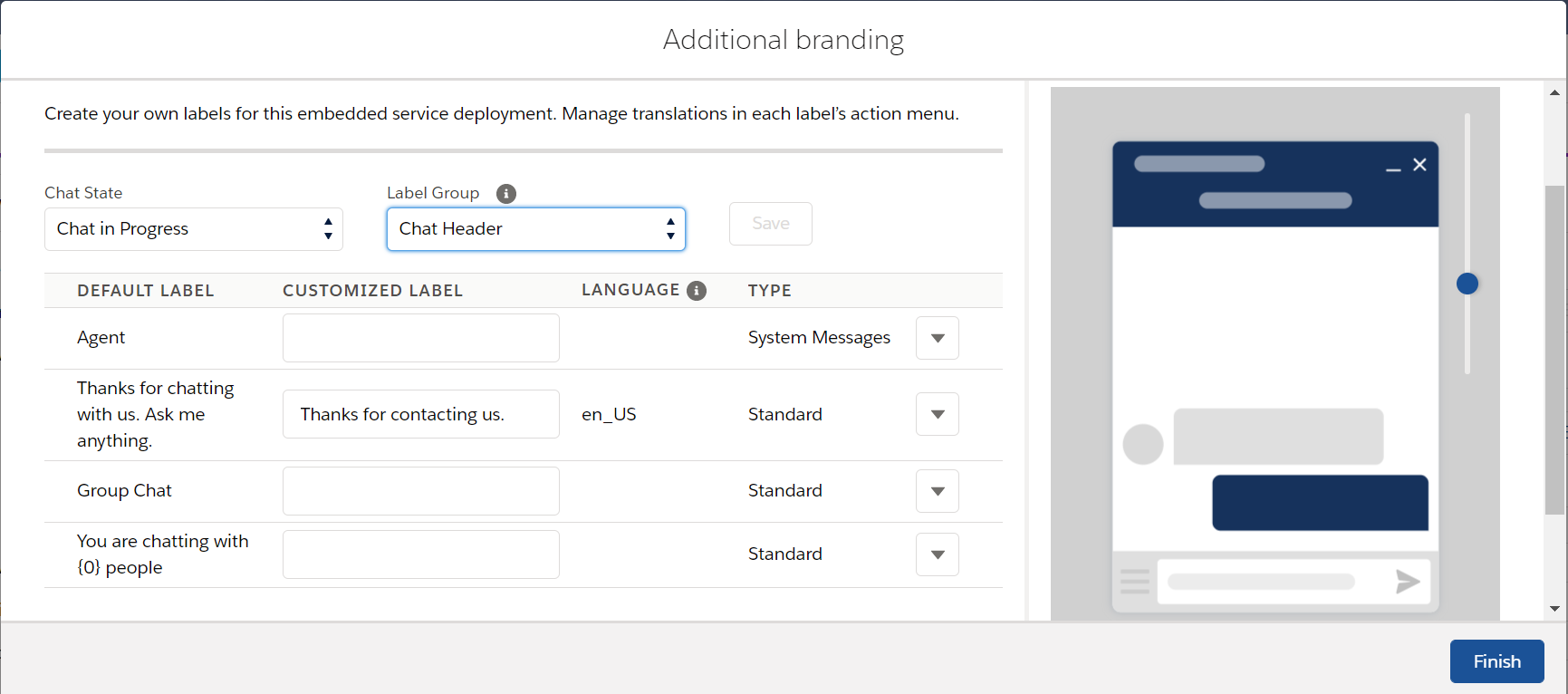
1. Static resources as images:

We can use static resources as images for chat logo and avatar. It can be placed directly in the additional branding of Embedded Service deployment or can also be placed in the code snippet. For chatbot, it didn’t work if we placed the urls of static resources in additional branding, hence we opted to use it in code snippet.

When placing the URLs for static resources, ensure to remove the “?” at the end of url if you find. The “?” at the end of url doesn’t allow the static resource to be accessed outside salesforce.

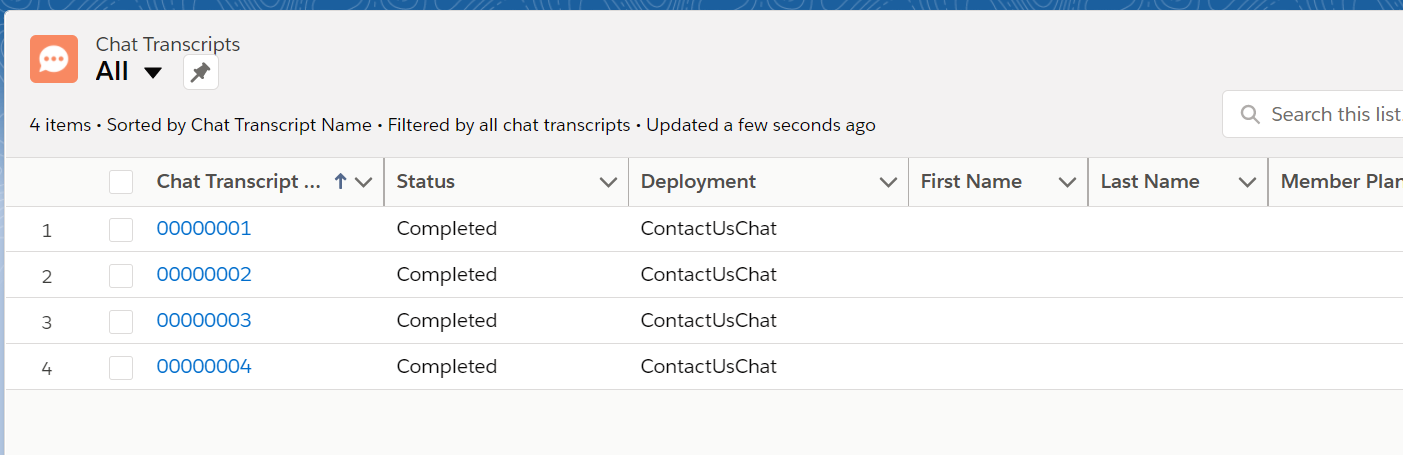
1. Customizing labels and messages:

All the labels and standard messages which we see on chatbot can be customized in the embedded service deployment (In additional branding).



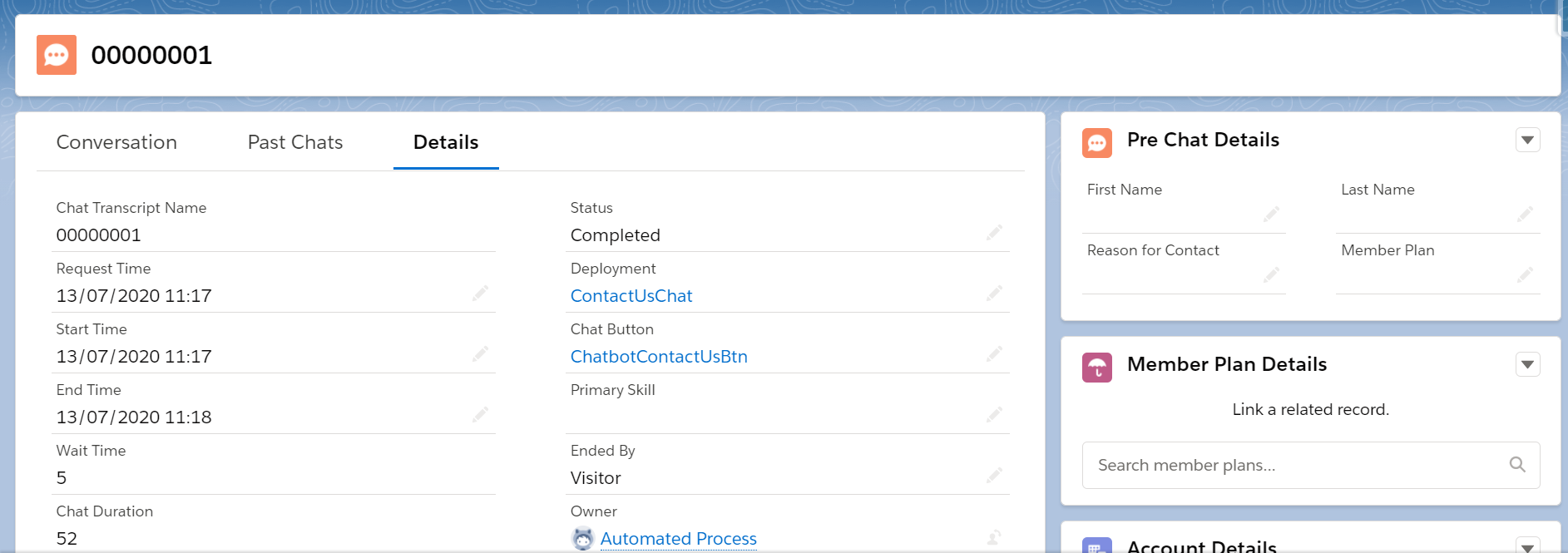
When we write a new message to over ride the existing one, a custom label will be auto created. Ensure to deploy the custom label as well in case of any such changes.

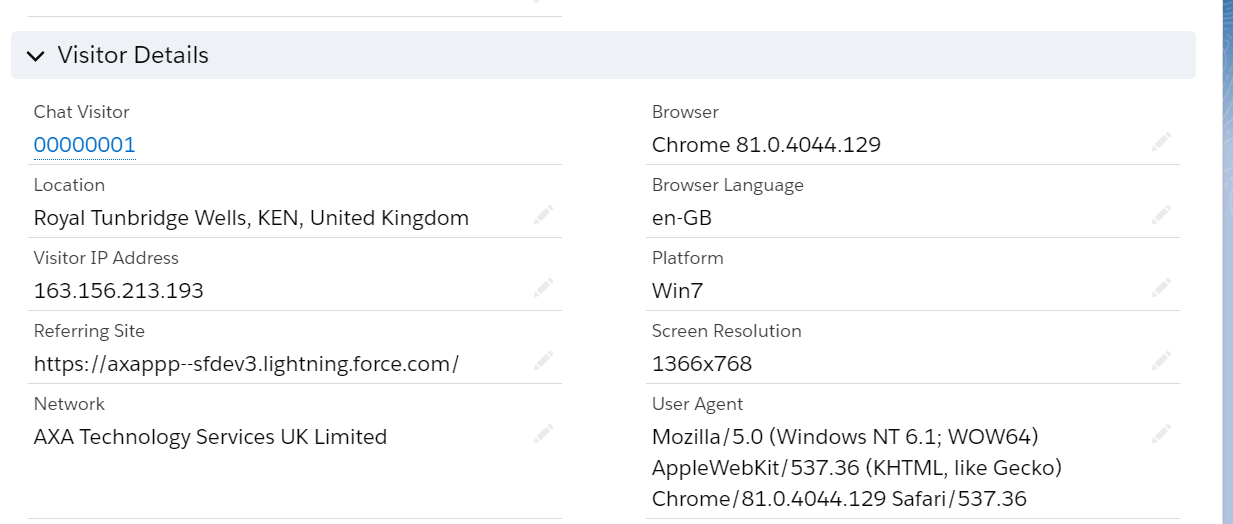
1. All the chat conversations will be stored in Chat Transcripts object:



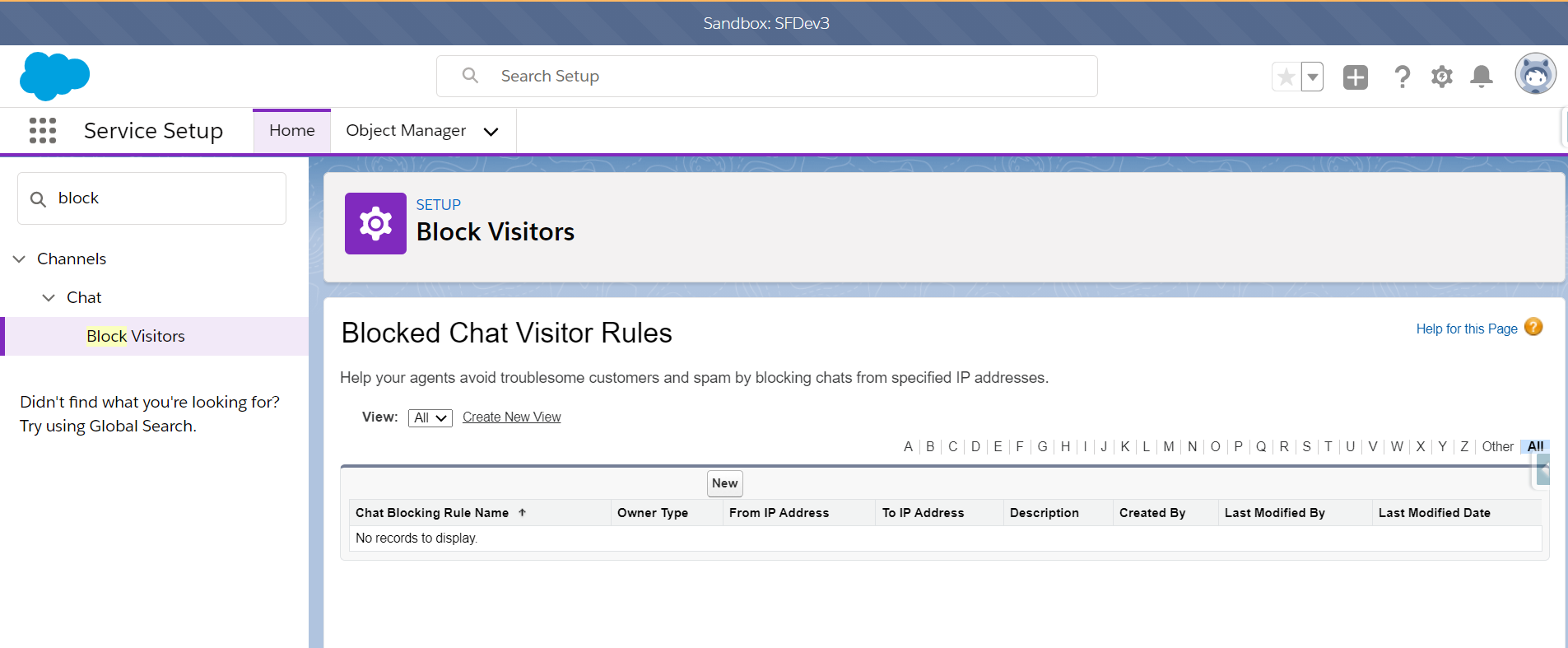
1. Blocking the user:

If the team receives any spam message complaints, that particular IP address can be blocked. The IP address is to be extracted from the chat transcripts: (Chat Transcripts < Visitor Details)





We can place this IP address in “Block Visitors”



The given IP address can no longer access the chat bot.

1. To Redact any sensitive information like credit card number, “Sensitive Data Rules” can be used:

