DHEEYANTRA
RESEARCH LABS

Dhee.AI Rest API Integration

The document is a developer reference defining the process and entities involved to integrate Dhee.AI with any external application.

Dhee can be integrated to any chat application using REST APIs which act as webhooks. This document defines the request and response protocols for these webhooks.

Note that the document uses the Test deployment of Dhee.AI as an example in most cases. The URLs described would be different in a production deployment.

How ?

The integration process flow is broadly explained in 3 steps:

- Initiate conversation with Dhee using an API call to get a unique conversation ID.
- Any utterance of the end user is packaged as an “Utterance” object and sent to Dhee.AI’s webhook.
- Any utterance of Dhee is packaged as an “Utterance” object and sent to the Client's webhook.
- The conversation ID attribute of the Utterance object will be used by the Client application to route the utterances to the respective conversing user.

The ping pong ball - Utterance objects

The messages are carried to and fro in the “Utterance” object serialized as JSON.

This is the sample structure of Utterance object in its most simple form:

```
{
   "text":"Hello !",
   "conversationId":"070089d0-9c85-11e9-a809-e144d46c0876",
   "clientId":18,
   "userId":672,
   "incoming":true
}
```

There’s an authentication

Dhee.AI team will be providing the credentials to access the conversation APIs.

They will be in the form of a username password pair. The authentication mechanism will Basic to keep the API calls simple.

Starting a Conversation !

To start off a conversation, we need to let Dhee know of the user’s arrival. This is done by invoking a chat initiation API call.

Get Chat initialization Information

Send a POST request to https://qa.dhee.net.in/social/app/init-chat
Header:
Authorization = "Basic " + [username + password hash]
Params:

Name	Type	Description	Mandatory
contactNumber	String	The contact number of the end user. In cases where this is not available please provide “0000000000” (ten zeroes)	Yes

language	String	The language of conversation. Should be one among : ● ENGLISH ● HINDI ● BANGLA ● TAMIL ● MARATHI ● GUJARATI ● TELUGU ● KANNADA ● MALAYALAM *The language should also be available in your subscription plan with Dhee.	Yes
deviceModel	String	The model of the smart phone or device	No
locale	String	User’s locale. Eg: en_US	No
timeZone	String	Users Time Zone Eg: Asia/Calcutta	No
latitude	String	The latitude of the geolocation of user	No
longitude	String	The longitude of the geo location of the user.	No

Sample Response :

```
{
  “Success” : true,
  “Result” : {
              "clientId":1,
              "conversationId":"8f493220-9cc5-11e9-8e3c-a5780068a364"
           }
}
```

Save this object as chat initialization information (*chatInitInfo*).

If there is no free bot of Dhee to chat with, the response will be:

```
{
 “Success” : false,
 “Result” : {
            "clientId":1,
            "userId":-1,
            "chatInitError":"NoFreeHandle"
          }
}
```

Receiving messages from Dhee

The client application should expose a REST API to receive messages from Dhee.

The rest API should expect a JSON string in the request body, which can be serialized into an Utterance object.

Sample structure of incoming message (and the Utterance object) :

```
{
  "id":"97c6d4d0-9cd8-11e9-9e4e-d312a1dbee0b",
  "conversationId":"98ad0b90-9cd7-11e9-aabc-59b9e89540b1",
  "text":"Good bye.",
  "semanticClass":null,
  "sourceJID":null,
  "targetJID":"g0_of_1@dheeyantra.com",
  "userId":0,
  "clientId":1,
  "domainId":0,
  "incoming":false,
  "time":1562078968477,
  "replyTo":null,
  "escalated":false,
  "escalationSupportUserJid":null,
  "escalationEndUserJid":null,
  "escalationStimulusId":null,
  "allowMultiple":false,
  "preferredLanguage":"ENGLISH",
  "commandMessage":"CLOSE_NOW",
  "currentContext":null,
  "semanticClasses":[

  ],
  "responderType":null,
  "categoryTag":null,
  "missedQuery":false,
  "customCommand":null
}
```

Of the above, these are important properties for the client :

Name	Description
conversationId	This UUID should be used to route the utterance to the end user.
text	The message to be displayed to the user. The text might sometimes contain extended display commands for printing buttons, interactive lists etc. A detailed explanation of the display types is appended in one of the following tables.
commandMessage	Actionable commands from Dhee Server to the client. Eg, CLOSE_NOW = close the current chat window. A detailed explanation of each command message is appended in one of the following tables.
currentContext	This is a key-value map which may have important information to help a custom command. Examples: If taking a payment, the key value pairs will hold information about payment amount, payment gateway details etc. If setting an alarm, the key value pairs will hold information about the time of alarm, whether to repeat etc.

Sending Messages To Dhee

To send a message to Dhee, JSON representing utterances of the end users need to be posted to Dhee.AI’s webhook.

Send a POST request to https://qa.dhee.net.in/social/app/chat-utterance

Header:
Authorization = "Basic " + [username + password hash]

```
Body : {
             "text":"<the-text-end-user-entered>",
             "conversationId":"<chatInitInfo.conversationId>",
          }
```

A Note on painting responses from Dhee

When painting responses from Dhee, we need to understand how Dhee currently presents the Lists, buttons, image carousels etc. These are detailed in the below section.

Extended Display Commands

These are encoded text that can appear within the “text” property of an incoming utterance object.

An example text having such display type will be as given below:

An OTP has been generated and sent to your mobile number. Please enter it in the box below. [[EXT:CUSTOMTEXT|OTP here]][[EXT:BUTTON|*Resend OTP^Resend OTP]]

Below table explains the extended display commands (that are included for formatting and better user experience as well)

Type	Format of appearance in Utterance.xtet	Description
EXT:BUTTON	[[EXT:BUTTON|<button1-label>^<button1-post-back-text>|<button2-label>^<button2-post-back-text>| ... |<buttonN-label>^<buttonN-post-back-text>]]	Label is what is displayed to the user as button text. Post back text is what is sent back to Dhee, when the end user clicks on the button.
EXT:INTERACTIVE_LIST	[[EXT:INTERACTIVE_LIST|<item1-label>^<item1-postback-text>| … |<itemM-label>^<itemN-postback-text>]]	Same as for buttons, but the display is expected to be List like, one item below the other.
EXT:UNORDERED_LIST	[[EXT:UNORDERED_LIST|<item1-text>|<item2-text|...|<itemN-text]]	This command is to display items as read only lists. I.e, they are only formatting and are not interactive.

Please note that for all of the above commands, it is at the discretion of the client programmer to choose and use the most matching client side UI component to match the expectation of the display type..

Get things done via Custom commands

Command Messages

Command messages appear in the commandMessage parameter of the Utterance object. They are directed from Dhee.AI to the client (or vice versa) to signal control events of a conversation. The below table explains the command messages in detail:

Name	Direction	Description
CLOSE_NOW	server-to-client	Command for the client to consider the conversation as ended.
CUSTOM	server-to-client	Any custom command which is set in **Utterance.customCommand**, defined per client's needs. Eg. command to show a notification on a smartphone.
CLOSE	client-to-server	Signal the server that the end user wishes to close the conversation. Eg, the user clicking the close button of the chat window should send this command message to the server.

[***
There are other command messages used to synchronize escalation to human support users, taking payments etc, which are the specialized use cases and won't be needed to run a general automated conversation.]

Get Help !

The Dhee.AI team is always around to answer to your queries or concerns. Please find our contact details below

Name	Email	Designation
Mallikarjun Govindaraju	mallik.shetty@dheeyantra.com	Senior Cognitive Engineer
Soumyadip Mallick	soumyadip@dheeyantra.com	Lead R&D Engineer
Sachin Kaushik	sachin.kaushik@dheeyantra.com	Director of Operations
Sreekumar Jairaj	sree@dheeyantra.com	CTO

Let’s make marvels together !