



## Initiating bot conversations from the Oracle Digital Assistant Client SDK for JavaScript web messenger widget on startup

Abhay Bhavsar, March 2019

The *Oracle Digital Assistant Client SDK*<sup>1</sup> for JavaScript provides functionality to communicate with a Oracle Digital Assistant from the web. Part of the Client SDK for JavaScript functionality is a JavaScript based web widget that can be added to your web applications or web sites to launch a message window.

In a typical chat conversation, users initiate the conversation by typing a message similar to shown below, assuming the receiving bot has an intent defined so it responds to the request.

Hi

Hello

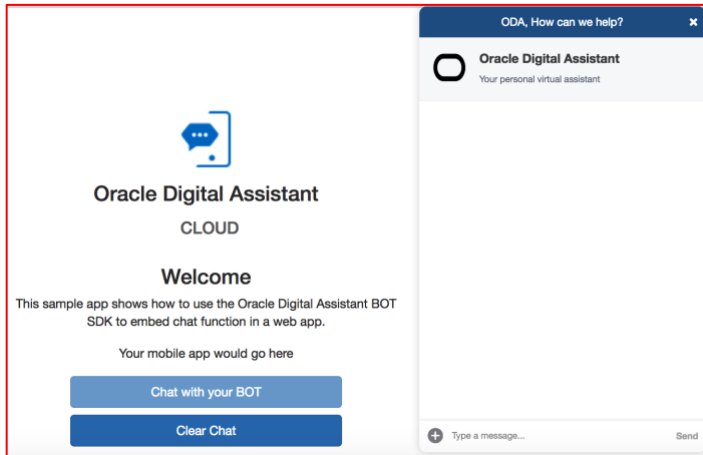
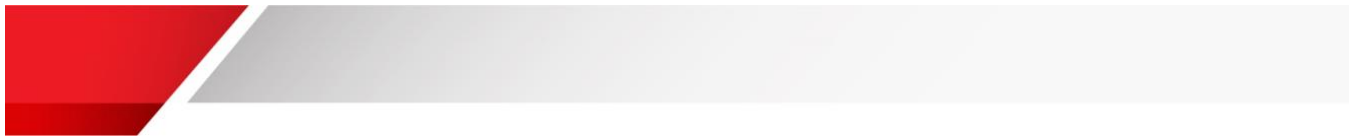
I would like to make a complaint

The bot replies with a text message or rich UI response (e.g. card layouts or a list) that then is understood by the web client and rendered for the user to make a selection or type a next message.

A problem reported by users is that the chat window initially is empty (see image below), leaving the bot users stuning of how to start a conversation. So ideally, the chat windows open with the Oracle Digital Assistant bot to welcome a user with a clear message or menu to choose from.

---

<sup>1</sup> <https://www.oracle.com/technetwork/topics/cloud/downloads/amce-downloads-4478270.html>



**Note:** Oracle also provides a sample implementation '*ODA Client samples for JavaScript*' that you can download and run to test the web channel or to learn how to build your own custom integration. You can download the sample implementation from here:

<https://www.oracle.com/technetwork/topics/cloud/downloads/amce-downloads-4478270.html>

## Oracle Digital Assistant Client SDK for JavaScript setup and web client

The *Oracle Digital Assistant Client SDK* for JavaScript provides a readme file that contains detailed steps on how to use it and how to set it up.

**Note:** The *Oracle Digital Assistant Client SDK* is also available as a library for Android and iOS to integrate a bot conversation in a native mobile application<sup>2</sup>.

Using the provided sample application of the *Oracle Digital Assistant Client SDK for JavaScript*, you can quickly test any chatbot for which you created a web channel in Oracle Digital Assistant. When creating the web channel, you get an application ID that you need to copy and paste into the start screen of the sample app.

Assuming you have Node installed, as well as the Node Package manager (npm), follow the instructions below to start the web chat client (a node server provided with the sample) and begin a bot conversation

1. Open a terminal window and navigate to the *chat-sample-web* folder of the downloaded and extracted sample
2. Run `npm install` to install all Node.js dependencies

```
Abhays-MacBook-Pro:chat-sample-web abhaybhavsar$ pwd
/Users/abhaybhavsar/PM/Product/SDK/bots-client-sdk-js-samples-18.4.3.0(1)/chat-sample-web
Abhays-MacBook-Pro:chat-sample-web abhaybhavsar$ npm install
npm WARN chat-sample-web@1.0.0 No repository field.
npm WARN chat-sample-web@1.0.0 No license field.

added 48 packages from 47 contributors and audited 119 packages in 4.341s
found 0 vulnerabilities
```

3. Launch the web client server by issuing `npm start`

```
Abhays-MacBook-Pro:chat-sample-web abhaybhavsar$ npm start

> chat-sample-web@1.0.0 start /Users/abhaybhavsar/PM/Product/SDK/bots-client-sdk-js-samples-18.4.3.0(1)/chat-sample-web
> node server.js

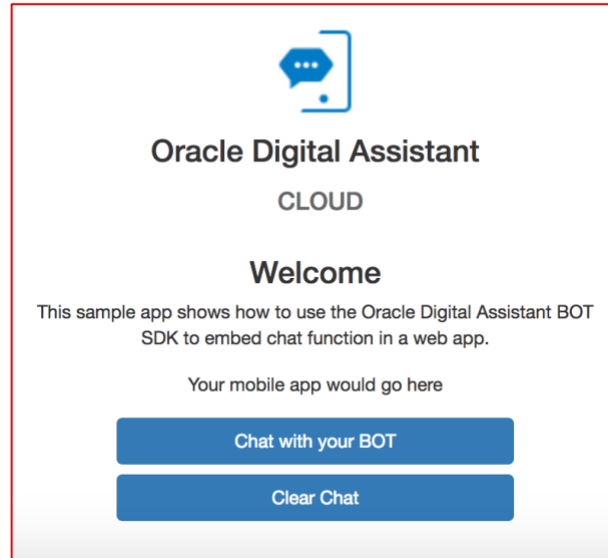
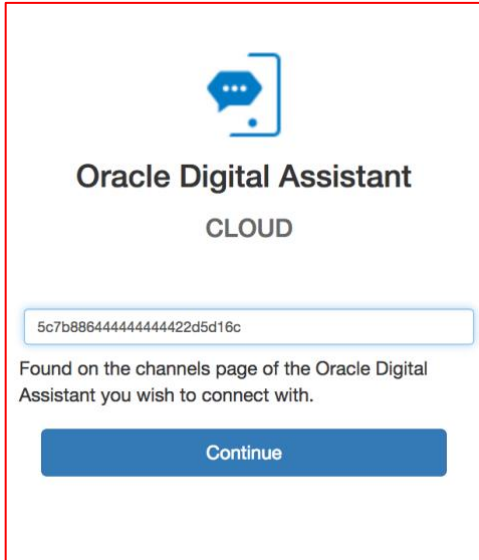
listening on port 3000
```

4. Open a web browser and type the following URL into the URL field  
<http://localhost:3000>
5. Enter your application id that you obtained when creating the web channel in Oracle Digital Assistant and press **Continue**

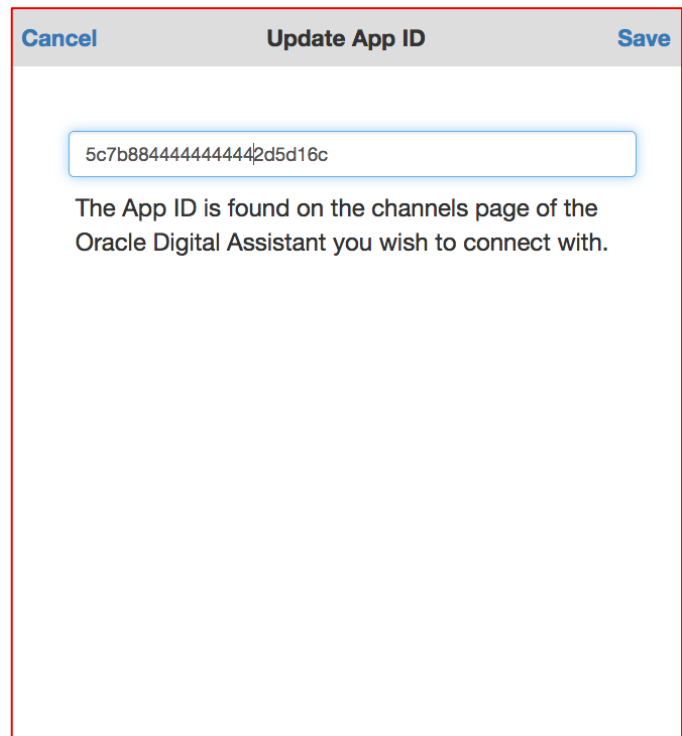
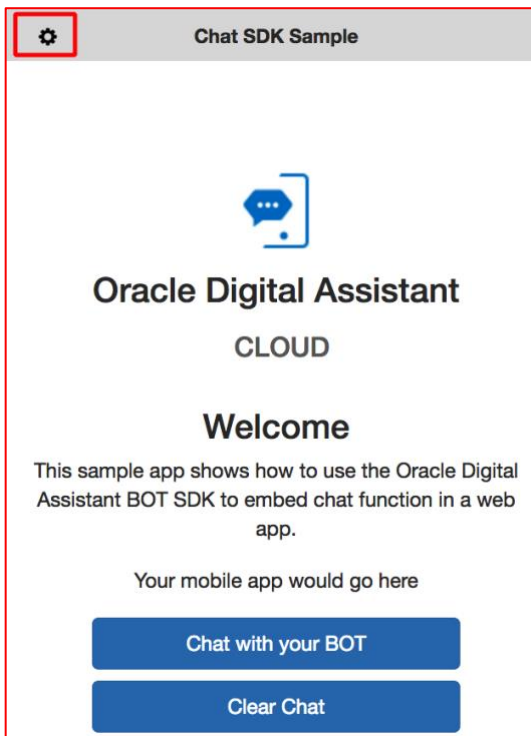
---

<sup>2</sup> <https://www.oracle.com/technetwork/topics/cloud/downloads/amce-downloads-4478270.html>

6. Notice that, when starting the conversation by clicking the **Chat with your Bot** button, the message window initially is empty. This is what you are going to change next



7. Click on Continue and then on the next screen click on the tool icon on the left top and save your app id.



## Initiate a conversation on launch

The Oracle Digital Assistant Client SDK for JavaScript, as the name indicates, is a JavaScript implementation, which means the values for that most settings can be provided dynamically. The sample client actually uses a base configuration in a separate file, the app.js file, which is located in <bots-client-sdk-js-samples folder>/chat-sample-web/app/scripts folder.

Use the app.js folder to change the messenger client launch behavior to send an initial message to the bot when client starts. For this you need to modify the **initBots' Bots.on(...)** function located in the file.

Note that the initial message should be such that the bot responds with something reasonable, ideally a welcome message explaining how it wants to be used.

The code sample below sends 'hi', which by default is handled by the Oracle Digital Assistant welcome intent.

```
// Initiate the conversation
// This event triggers when the chat widget is opened
Bots.on('widget:opened', function(){
  if (Bots.getConversation().messages != null
    && Bots.getConversation().messages.length < 1) {
    Bots.sendMessage("Hi");
  }
});
```

The following code shows the complete function inside app.js for your reference.

```
function initBots(appld){
  return Bots.init({
    appld: appld,
    // This code has been removed for simplicity.
    // .....
  })
}.then(function (res){
  Bots.updateUser(
    {
      "givenName":"John",
      "surname":"Snow",
    }
  )
})
```

```

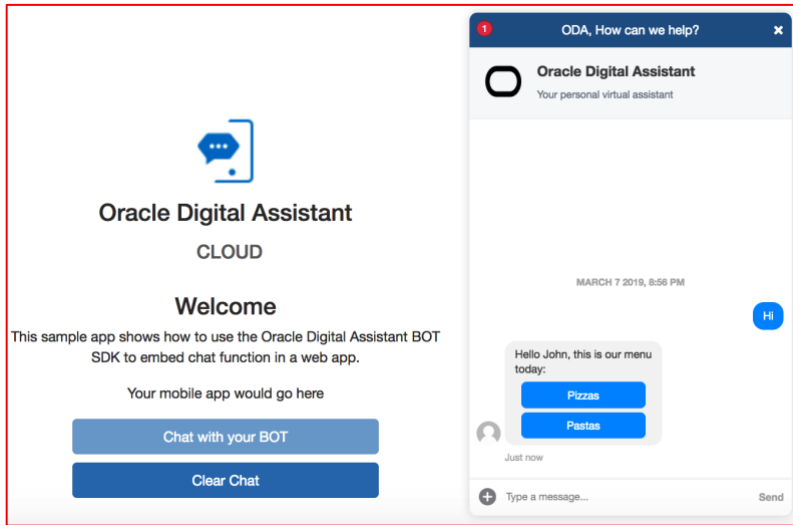
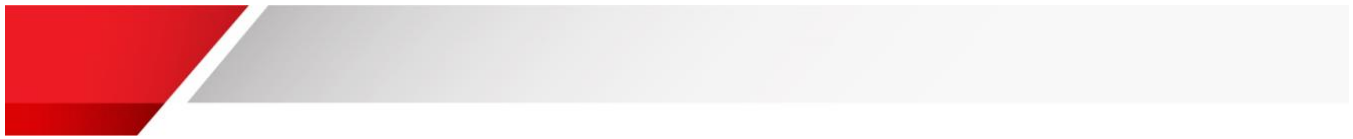
    "email": "john.snow@winterfell.com",
    "properties": {
      "smoochCustomVariable1":"Lord",
      "smoochCustomVariable2":"Commander"
    }
  }
}

).catch(function (err) {
  console.error(err);
});

// Initiate the conversation
// This event triggers when the chat widget is opened
Bots.on('widget:opened', function(){
  if (Bots.getConversation().messages != null
    && Bots.getConversation().messages.length < 1) {
    Bots.sendMessage("Hi");
  }
});
});
}

```

So, next time the messenger client starts, a "Hi" message is sent as an initial user message to the bot. You can also train the bot to respond to other messages like "..." or "begin", etc..if you don't want to use 'Hi' as the initial message.



## Conclusion

This article shows you how you can initiate a conversation with a bot by having the SDK send a message on behalf of the user. It also shows how an application can initiate a conversation and how these two differ from the greeting messages shown by Facebook messenger on launch of the bot.