The New Oracle Web SDK: A Feature Overview
Rohit Dhamija, November 2019

With Oracle Digital Assistant 19.10 and later, Oracle provides a new web SDK for integrating Oracle Digital Assistant to web applications and web sites.

The new Oracle Web SDK provides all features available in the current version of the Oracle JS Client SDK, but also goes beyond in regards to security features and new messenger functionalities.

This article explores some of the new features in the Oracle Web SDK.
THIS ARTICLE EXPLORES SOME OF THE NEW FEATURES IN THE ORACLE WEB SDK. ........................................... 1
BEFORE YOU BEGIN ............................................................................................................................................. 3
INTEGRATING ORACLE DIGITAL ASSISTANT IN A SAMPLE WEB-PAGE ............................................................. 3
  CREATING AN ORACLE WEB CHANNEL ........................................................................................................... 3
  ADDING THE ORACLE WEB SDK TO THE WEB PAGE ....................................................................................... 6
  SAMPLE PROJECT (BASIC_SAMPLE.HTML) ....................................................................................................... 7
EMBEDDING CHAT WIDGET IN WEB-PAGE ........................................................................................................ 10
  SAMPLE PROJECT (EMBEDDED_SAMPLE.HTML) .................................................................................................. 10
CONFIGURING CHAT WIDGET UI .......................................................................................................................... 13
  LAYOUT .............................................................................................................................................................. 13
  ICONS ................................................................................................................................................................. 13
  COLORS ............................................................................................................................................................ 13
  TEXT ................................................................................................................................................................ 13
  SAMPLE PROJECT (CUSTOMUI_SAMPLE.HTML) .................................................................................................. 15
MODIFY FEATURE AND FUNCTIONALITIES .......................................................................................................... 17
  FEATURE ......................................................................................................................................................... 17
  FUNCTIONALITY ............................................................................................................................................... 17
  SAMPLE PROJECT (CUSTOMFEATURE_SAMPLE.HTML) ....................................................................................... 18
UPDATING AND DISPLAYING USER INFORMATION .............................................................................................. 21
  SAMPLE PROJECT (USERPROFILE_SAMPLE.HTML) .......................................................................................... 21
TRIGGERING BOT-INITIATED CONVERSATIONS .................................................................................................. 24
  SAMPLE PROJECT (HIDDENMSG_SAMPLE.HTML) .............................................................................................. 24
CONCLUSION ............................................................................................................................................................ 25
Before You Begin

The Oracle Web SDK requires Oracle Digital Assistant 19.10 or later. So to work with it you need to

- ensure that you have access to an instance of Oracle Digital Assistant version 19.10 or later.
- have Node.js and the http-server npm package installed on your machine. To install Node the http-server package, run `npm install -g http-server` from a command line or terminal. If you are on a MAC, ensure you run `sudo npm install -g http-server`
- Download the latest Oracle Digital Assistant Oracle Web SDK
- Download and extract the code provided with this TechExchange article to your local machine

Integrating Oracle Digital Assistant in a Sample Web-Page

In this section, you will learn to host and run Oracle Digital Assistant chat widget in a sample web-page.

Creating an Oracle Web Channel

1. Open Digital assistant click on the top left hamburger menu and navigate to Channels menu
2. Under Users tab, click on + Channel button, as shown below

![Figure 1 Creating Channel](image)
3. Click the **Create Channel** button and set the following options
   a. Provide a name to the channel, e.g. RDH_OracleWebChannel
   b. Select Channel Type as **Oracle Web**
   c. Under "Allowed Domains" enter *
   d. **Disable** the client authentication enabled option

Click on **Create** button

![Create Channel](image-url)

*Figure 2 Oracle Web Channel*
4. Now set **Route To** a skill or digital assistant to display on the web channel. For example, as shown in the image below, select the CrcDemoBot and toggle the **Channel Enabled switch to show enabled**.

![Configure oracle web channel](image)

**Figure 3 Configure oracle web channel**

5. Next, take a note of the **Channel Id**, as well as the **Oracle Digital Assistant instance URI** as shown in the image below. You need both information to configure the chat widget.

![Copy URI from address bar](image)

**Figure 4 Copy URI from address bar**
Adding the Oracle Web SDK to the Web Page

1. Locate settings.js file in the code\scripts folder of the extracted code_v1.zip file you downloaded for this article. The image below shows the extracted folder structure.

![Code folder structure]

2. Open settings.js and update values for **URI** and **channelId** as shown below:

   ```javascript
   1 const YOUR_URI = "idcs-oda-4be50dd1d3ad43e00128037572ff-e9e-d0.data.digitalassistant.oci.oc-test.com";
   2 const YOUR_CHANNELID = "ca6e40f-361e-1036-b67-c960897e4545";
   ```

![URI and channelId]

3. Save the settings.js file and close it.
Sample Project (basic_sample.html)

Let's now start with running ODA chat-widget within a sample HTML page. Please follow the below steps:

1. Open the `basic_sample.html` and observe the code for invoking the ODA chat-widget inside a sample HTML file

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Oracle Web SDK Sample</title>
  <script src="Scripts/settings.js"></script>
  <script>
    var chatWidgetSettings = {
      URI: YOUR_URI,
      channelId: YOUR_CHANNELID
    };  
    !function (globalObj, doc, library, name) {
      function initiateSDK() {
        setTimeout(function () {
          globalObj[name] = Object.assign(WebSDK);
          globalObj[name] = new globalObj[name](chatWidgetSettings);
          globalObj[name].connect().then(() => {
            (reason) => {
              }, 2000);
            try {
              var scripts = doc.getElementsByTagName("script")[0],
              lib = doc.createElement("script");
              lib.async = true;  
              lib.src = library;
              lib.onreadystatechange = initiateSDK();
              scripts.parentNode.insertBefore(lib, scripts);
            } catch (e) {
              })(window, document, "scripts/web-sdk.js", "Bots");
        </script>
    </head>

    <body>
    <h3 align="center">This is a sample page having ODA web widget built using Oracle Web SDK</h3>

    </body>
</html>
```
1. Next, on a command line or terminal, navigate into the "code" root folder of the extracted ZIP file.

2. Run the following command **http-server -p 3000**

   ![Command output](image)

   **Figure 7** Host web application

3. Enter following URL in browser and hit enter: **http://localhost:3000/basic_sample.html**

   ![Browser output](image)

   **Figure 8** ODA Chat widget

4. A chat window will appear as shown below
This is a sample page having ODA web widget built using Oracle Web SDK.

Figure 9 Sample page running ODA chat-widget

NOTE: KEEP THE HTTP-SERVER UP AND RUNNING. YOU NEED IT FOR THE OTHER SAMPLES IN THIS ARTICLE.
Embedding Chat Widget in Web-Page

In this section, you will learn how easy it is to embed a chat widget inside a webpage. To activate the embedded mode, you need to add `embedded: true` to the settings of the web widget. In addition, you need to add the “targetElement” property with the ID of a DOM element (HTML component). The DOM element will be used as a container to render the messenger widget in. The embedded widget will occupy the full width and height of the container. Note that you must specify a height for the containing DOM element as otherwise the messenger widget does not show.

Sample project (embedded_sample.html)

To try the embedded mode

1. Open embedded_sample.html
2. Observe following changes to open the widget in embedded mode
   a. The `embedded` property is set to true
   b. The `targetElement` points to the div markup element
   c. div element with ID container-div added to page body
3. **To test this, enter following URL in browser and hit enter:** [http://localhost:3000/embedded_space.html](http://localhost:3000/embedded_space.html)
This is a sample page having ODA web widget built using Oracle Web SDK

Figure 11 Embedded chat-widget inside webpage
Configuring Chat Widget UI
Oracle Web SDK provides lots of UI customization options for layout, icons, colors, text and many more. Before starting with a customizing sample, let's first see the overview of customization options

Layout
Oracle Web SDK provides flexibility to modify the chat widget layout. Some of the items that can be modified are

1. Position of notification badge icon with respect to bot button
2. Starting position of conversation in widget.
3. Font for chat widget
4. Width and height of chat widget.
5. Padding around messages in chat widget
6. Field to specifically place the chat widget in browser window

Icons
You can customize following icon settings:

1. Image URL for chat bot button, chat logo icon, icon depicting person alongside user response, icon depicting bot alongside bot response
2. Image URL for loading chat bubble icon
3. Height and width of loading chat bubble icon
4. Image URL for file type, audio type, image type, video type, send button

Colors
Following colors can be modified to provide a custom look for the widget:

1. The primary color for branding the widget. The color is used as the header background, and as hover color on footer buttons.
2. The text color of messages in chat widget.
3. The text color of secondary text in messages, e.g. card descriptions in chat widget.

Text
Following strings can be customized to provide localized text:

1. Tooltip that appears on hovering over audio utterance on button in header.
2. Text placeholder that is displayed while card image is fetched and loaded.
3. Title of the chat widget that is displayed on header.
4. Tooltip that appears on hovering over clear, close button in header.
5. Status text when connection between chat widget and server is closing, established, connecting or closed
6. Placeholder text that appears in user input field.
7. Status text that is displayed at the end of older messages.
8. Text that is displayed while user location is requested, request denied, message not sent to the server
9. Tooltip that appears on hovering over send or upload button in footer.
10. Error text displayed when upload file size is too heavy, unsupported file type or upload failure
Sample project (customui_sample.html)
In this project, you will learn how to customize icons, layout, colors using Oracle web SDK properties.

You need to just follow the below steps

1. Open customui_sample.html and observe chatWidgetSettings variable, as shown in screen-shot below.

```
var chatWidgetSettings = {
  URI: YOUR_URI,
  channelId: YOUR_CHANNELID,
  font: '14px Helvetica Neue', Helvetica, Arial, sans-serif', // layout modification property
  height: '60vh', // layout modification property
  width: '20vw', // layout modification property
  colors: { // custom colors property
    branding: '#01538A',
    text: '#B20637'
  },
  botButtonIcon: 'images/Robot.png', // custom icons property
  logoIcon: 'images/logo.png', // custom icons property
  botIcon: 'images/Robot.png', // custom icons property
  personIcon: 'images/userIcon.png', // custom icons property
  locale: 'en-us',
  i18n: {
    'en-us': {
      chatTitle: 'Support' // custom text property
    }
  }
};
```

Figure 12 Layout, custom color, icon and text property

2. Observe the layout, colors, icons and text property in the chatWidgetSettings variable.

3. Enter following URL in browser and hit enter: http://localhost:3000/customui_sample.html

4. You will observe the customized layout as shown below.
Figure 13 Customized chat-widget for icons and text
Modify Feature and Functionalities

Feature
1. Configure attachment sharing in chat widget
2. Enable clean message button on chat widget header
3. Enable loading previous conversations for given userId on the given browser when widget is initialized
4. Configure secure communication - https vs http and wss vs ws
5. Enable read timestamp for messages
6. Expand chat widget when page is loaded
7. Enable display of connection status in header
8. Display chat bubble when waiting for bot's response
9. Field to disable button clicks on messages that user has interacted with, allowed values all, none, postback

Functionality
1. Add additional elements in chat widget header.
2. Embed the chat widget
3. Enable embedding YouTube video link present in text message
4. Add stickie content at top or bottom of chat.
5. Add scrolling content at top or bottom of chat.
6. Enable debug mode
7. A user text message to send once the chat widget is ready without displaying in chat widget, used to initiate conversation
8. Object that contains locale fields; each locale maintains i18n key-value pairs for various text used in widget. See custom text section for the keys that can be customized.
9. Default locale of chat bot in which to display text
10. Maximum number of messages to be saved in local Storage at a time
Sample project (customfeature_sample.html)
In this project, you will learn how to customize functionalities like adding a clear button, enabling local conversation history, showing connection status, showing typing indicator, adding a custom header at the top and much more using Oracle web SDK feature properties.

Let us now see this in action, follow the below steps

1. Open customfeature_sample.html and observe chatWidgetSettings variable, as shown in screen-shot below

```javascript
var chatWidgetSettings = {
  URI: YOUR_URI,
  channelId: YOUR_CHANNELID,
  enableClearMessage: true,
  enableLocalConversationHistory: true,
  enableTimestamp: true,
  showConnectionStatus: true,
  showTypingIndicator: true,
  disablePastActions: 'all',
  customHeaderElementId: "customHeaderElement",
  embedTopStickyId: "topStickyElement",
  isDebugMode: true,
  locale: 'en-us',
  "i18n": {
    "en-us": {
      "chatTitle": "Pizza King"
    }
  }
};
```

Figure 14 Feature properties

2. Enter following URL in browser and hit enter: http://localhost:3000/customfeature_sample.html

3. You will observe the customized layout as shown below
Figure 15 Customized chat-widget

There’s no better feeling in the world than a warm pizza box on your lap - Kevin James

hello
Friday, October 18, 2019, 2:40:27 PM

Previous conversations

order pizza
Friday, October 18, 2019, 2:42:31 PM

Order a pizza
Also, you will notice a typing indicator a chat bubble which appears while waiting for bot's response. This is automatically achieved with `showTypingIndicator` property which is set to true by default. Below screen shot shows how a typing indicator appears while waiting for the bot’s response.

![Typing Indicator](image)

Figure 16 Typing Indicator
Updating and Displaying User Information

In this section, you will see how you can update the user profile information and text customization while displaying the user information!

Sample Project (userprofile_sample.html)

This sample project will consist of a sample skill which will be used to print profile information. Also this skill shows how we can incorporate HTML tags to customize the text message.

1. Use the skill dashboard and import helloBot(1.0).zip provided with this article

2. Open the skill and navigate to the dialog flow, observe the start state it should look something like below screenshot

   ```javascript
   start:
     component: "System.Output"
     properties:
       text: "Hi $profile.givenName $profile.surname $profile.lastOrderedItems";
       would you like to repeat your last ordered items: $profile.properties.value.lastOrderedItems"
     transitions:
       return: "done"
   ```

3. Validate and run the skill in system tester, and type hi

4. You will just get the output as shown below. The output just prints the text as is.

   ![System tester window](image)

5. Next, open the userprofile_sample.html from the extracted userprofile folder and observe the initiateSDK function

Oracle Digital Assistant - TechExchange
6. Observe that we have used two functions in this sample, first one is `Bots.sendMessage`. This function sends a message on the user behalf. In the code above, a message is sent right after the Oracle Web SDK widgets connects to Oracle Digital Assistant.

You can use any one of the format as shown below

```javascript
Bots.sendMessage({
    type: 'text',
    text: 'hello'
});
// OR
Bots.sendMessage('hello');
```

7. Next, observe `Bots.updateUser` function which is used for updating the user profile information. The example updates the following user profile information: first name, surname, email and user's last ordered items.

8. Enter following URL in browser and hit enter: `http://localhost:3000/userprofile_sample.html`
Figure 19 Send Message and User Profile demo

PLEASE RESTART THE BROWSER OR CLEAR THE BROWSER CACHE INCASE YOU FIND ISSUES IN RUNNING THIS SAMPLE
Triggering Bot-Initiated Conversations
In this section, you learn how a bot can trigger a bot conversation when a user enters a web page. All you need to do is to set `initUserHiddenMessage`.

Sample Project (hiddenmsg_sample.html)

1. Open `hiddenmsg_sample.html` and observe that the `initUserHiddenMessage` property in the `chatWidgetSettings` variable is set to `hello`.

   ```javascript
   var chatWidgetSettings = {
     URI: YOUR_URI,
     _channelId: YOUR_CHANNELID,
     userId: 'rdh',
     initUserHiddenMessage: "Hello!"
   };
   
   function (globalObj, doc, library, name) {
     function initiateSDK() {
       setTimeout(function () {
         globalObj[name] = Object.assign(WindowsSDK);
         globalObj[name] = new globalObj[name](chatWidgetSettings);
         globalObj[name].connect().then(() => {
         
         Bots.updateUser({
           profile: {
             givenName: 'Rohit',
             surname: 'Dhamija',
             email: 'rohit.dhamija@oracle.com',
             properties: {
               lastOrderedItems: "1 Medium Veg Farmhouse"
             }
           }
         });
       }, 2000);
     
     }, (reason) => {
       console.log("Connection failed");
       console.log(reason);
     });
   
   Figure 20 initUserHiddenMessage property
   ```

2. Enter following URL in browser and hit enter: [http://localhost:3000/hiddenmsg_sample.html](http://localhost:3000/hiddenmsg_sample.html)

3. Observe that the **hello** message that’s sent from the settings is NOT displayed. However, we do get a response from the bot as expected.
You are done testing with all the sample projects, you may now stop the http-server running in the terminal.

**Conclusion**

You have seen some outstanding customizations features of Oracle Web SDK. You have seen how to host Digital Assistant in a sample web page, embed the chat widget in a web page, configure the widget UI, add features to it like a clear button, enable local conversation history, show connection status, show the typing indicator, add a custom header at the top, update and display user-profile information, and implement a bot-initiated conversation.