Integrate Oracle Digital Assistant Web SDK in Visual Builder Cloud projects

Rohit Dhamija, Nov 2019

With Oracle Digital Assistant 19.10 and later, Oracle provides a new Web SDK, the Oracle Web SDK, for integrating Oracle Digital Assistant to Web applications and web sites. This article will guide you through the steps required to integrate the Oracle Digital Assistant web SDK in a Visual Builder Cloud Service (VBCS) project.
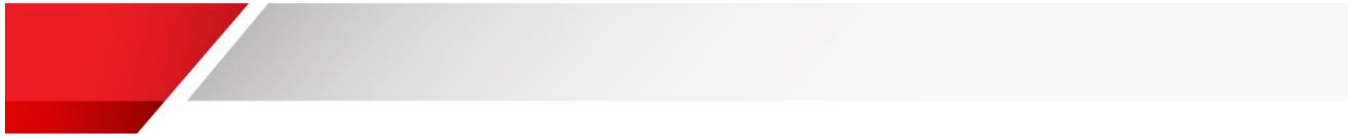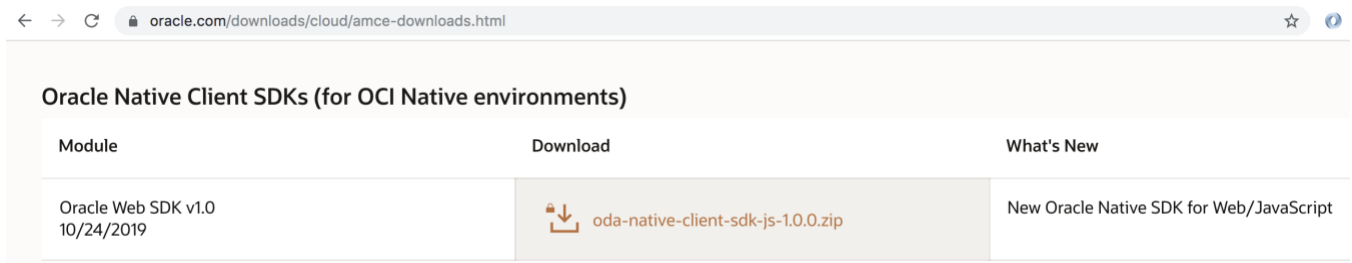
Pre-requisites

Before going further:

1. Validate the skill functionality in the embedded conversation tester and ensure that it is working properly.

2. Set up Oracle Web SDK channel in your Digital Assistant (DA) instance and associated it with a skill. The steps are documented in the Oracle TechExchange article "Overview of the new Oracle Web SDK and its customization features in Oracle Digital Assistant 19.10 and later".

3. Please take a note of following Oracle Web channel parameter settings

   a. **URI**, i.e. your chat server URL

   b. **Channel ID**, i.e. Web channel Id

4. Download the Oracle Web SDK 1.0: https://www.oracle.com/downloads/cloud/amce-downloads.html

Figure 1 Downloads page

5. Have an Oracle Visual Builder Cloud Service instance up and running

Getting Started

In this section, you'll go through the steps to invoke Oracle web chat-widget in a sample VBCS application.

1. Create a VBCS project.

2. Import Oracle digital assistant web SDK in your project.

3. Add code to configure and invoke the web chat widget.

4. Run the project having oracle digital assistant chat widget integrated.

Oracle Digital Assistant - TechExchange

Step 1: Create a VBCS project

In this section, you will create a VBCS web application

1. Login into your VBCS instance and click on **New** to create an application

2. Provide application name, optionally description and click **Finish,** as shown below

Figure 2 Create application

3. Next, select **Web Apps**, since you plan to build VBCS based web application

Start building your application

Connect to Data

Service Connections
Create Service Connections to consume existing REST endpoints provided by a service and use them in your applications

Business Objects
Create Business Objects to define custom REST endpoints for your database based on the needs of your applications

Create Apps

Mobile Apps
Create native mobile applications that make use of features on mobile devices

Web Apps
Create modern web applications that run in browsers on desktop and mobile devices

Add Artifacts

Components
Get additional components for use in your applications

Process Cloud
Integrate your Process Cloud Service business processes into your applications

Figure 3 Choose the type of VBCS application

4. You don't have any web application defined yet, so click on **+ Web Application**, provide an identifier of the web application, i.e. index and click **Create**

5. Once created, you should see your project structure like image below:
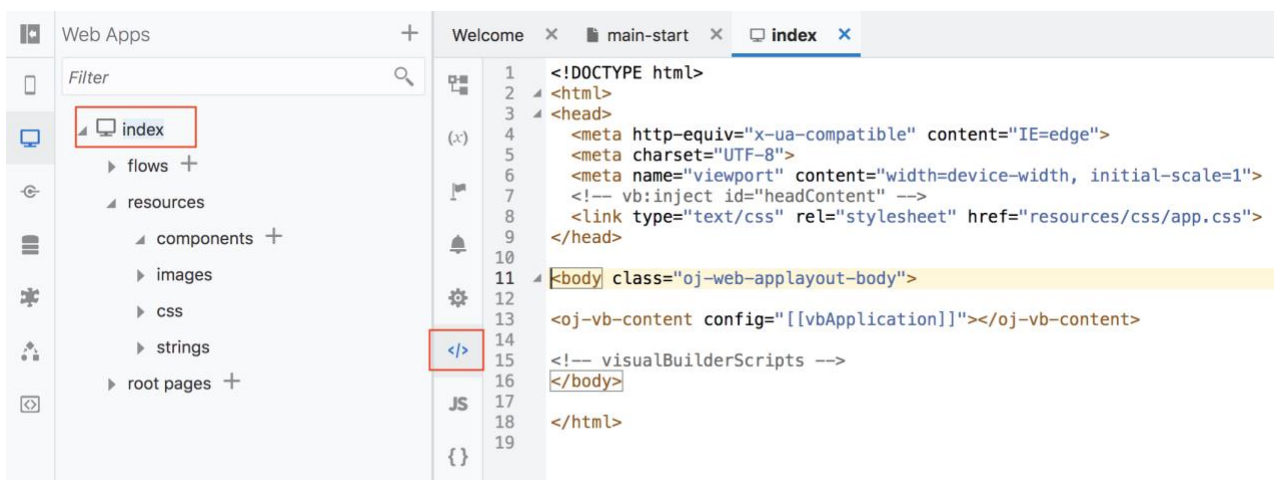
Figure 4 VBCS project structure

6. Click on run button to run your basic VBCS web application, you should see the output in a separate tab like below:
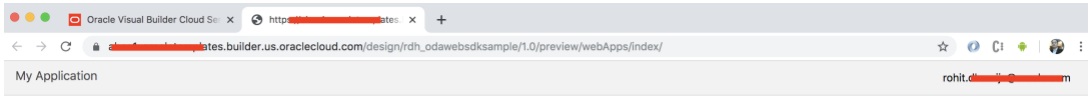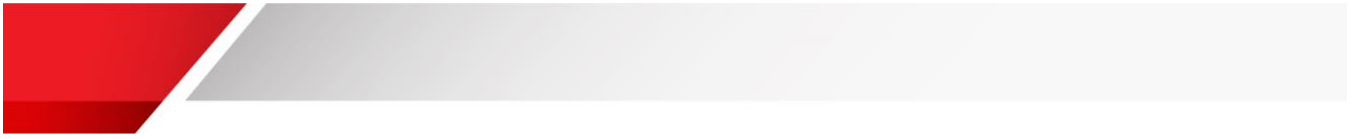
Figure 5 VBCS project output

With this, you completed the VBCS starter web application needed in the next section.

Step 2: Import Oracle Web SDK in VBCS project

Following the instructions in this section, you add the Oracle Digital Assistant Web SDK to your VBCS project

1. Extract oda-native-client-sdk-js-1.0.0.zip
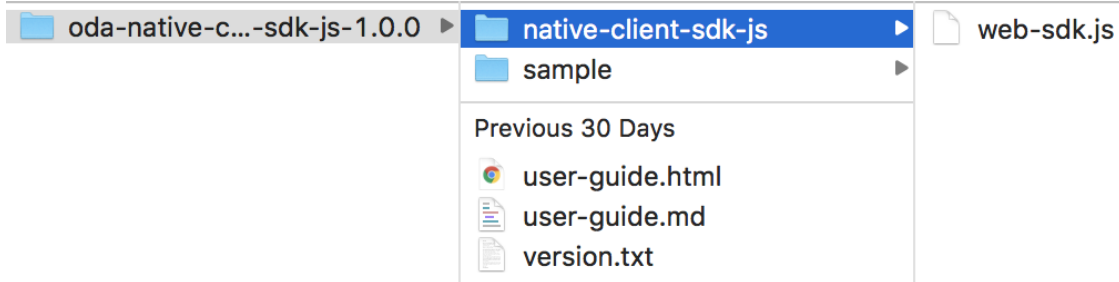
Figure 6 Web SDK folder structure

2. Compress **native-client-sdk-js** folder having web-sdk.js file

3. In your VBCS project, right-click on **resources** and click **Import,** as shown below:
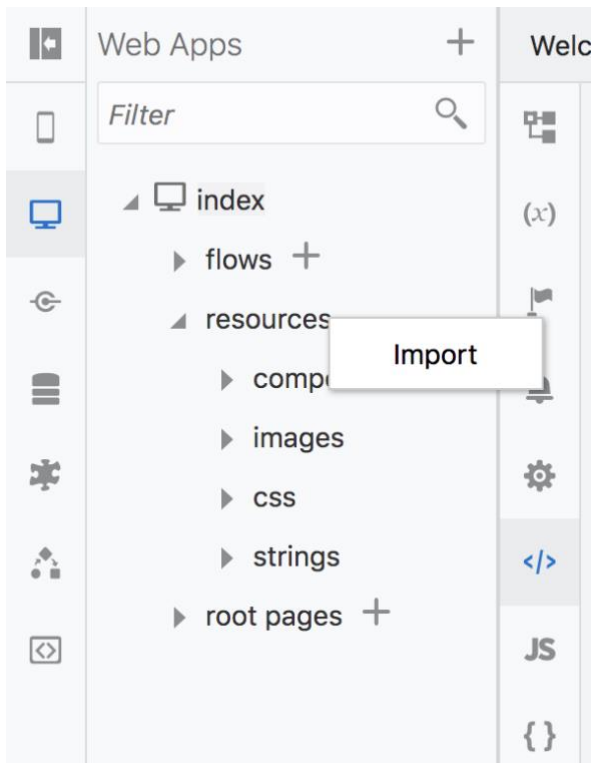
Figure 7 Import resources

4. Import the compressed **native-client-sdk-js.zip** and click on Import, as shown below

Figure 8 Imported resources

Your imported files will appear under resources branch as shown below:
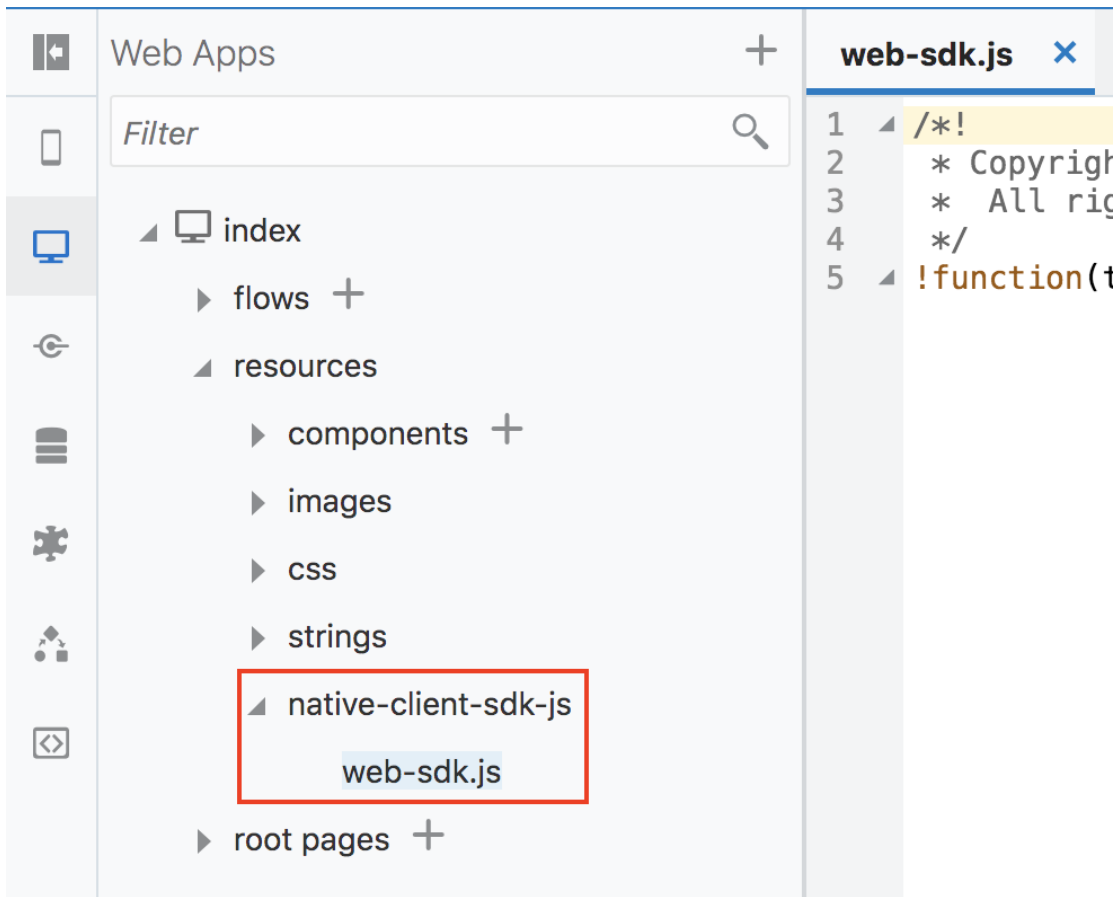
Figure 9 Updated project

At this point, you have successfully added Oracle web SDK into your VBCS project.

Step 3: Configure the Web Messenger on Your VBCS Page

In this section, you will invoke the oracle web SDK to connect and invoke the web-widget.

1. Click on index and select HTML as shown below:
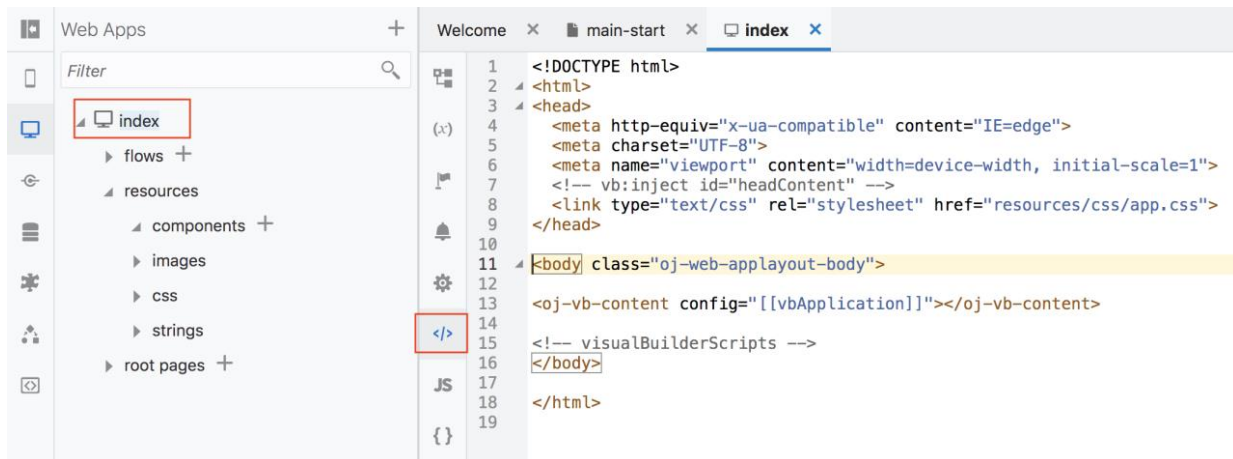
Figure 10 Index HTML code

2. Add following code before closing the **</head>** tag

```
<script src="resources/native-client-sdk-js/web-sdk.js"></script>
<script>

    var chatWidgetSettings = {

      URI: 'YOUR_URI',

      channelId: 'YOUR_CHANNELID'

  };

      setTimeout(() => {

          window.Bots = new WebSDK(chatWidgetSettings);

          Bots.connect().then(() => {

              console.log("Connection Successful");

          }, (reason) => {

              console.log("Connection failed");

              console.log(reason);

          });

      }, 2000);

</script>
```

Your index.html should look something like this as shown below:

Oracle Digital Assistant - TechExchange

```
Welcome  ✕   🖥 index  ✕

    1    <!DOCTYPE html>
    2  ◢ <html>
    3  ◢ <head>
    4      <meta http-equiv="x-ua-compatible" content="IE=edge">
    5      <meta charset="UTF-8">
    6      <meta name="viewport" content="width=device-width, initial-scale=1">
    7      <!-- vb:inject id="headContent" -->
    8      <link type="text/css" rel="stylesheet" href="resources/css/app.css">
    9      <script src="resources/native-client-sdk-js/web-sdk.js"></script>
   10      <script>
   11          var chatWidgetSettings = {
   12              URI: 'idcs-oda-                                    tant.oci.oc-test.com/',
   13              channelId: '79a6de          ece69b0'
   14          };
   15          setTimeout(() => {
   16              window.Bots = new WebSDK(chatWidgetSettings);
   17              Bots.connect().then(() => {
   18                  console.log("Connection Successful");
   19              }, (reason) => {
   20                  console.log("Connection failed");
   21                  console.log(reason);
   22              });
   23          }, 2000);
   24      </script>
   25    </head>
   26
   27  ◢ <body class="oj-web-applayout-body">
   28
   29    <oj-vb-content config="[[vbApplication]]"></oj-vb-content>
   30
   31    <!-- visualBuilderScripts -->
   32    </body>
   33    |
   34    </html>
   35
```
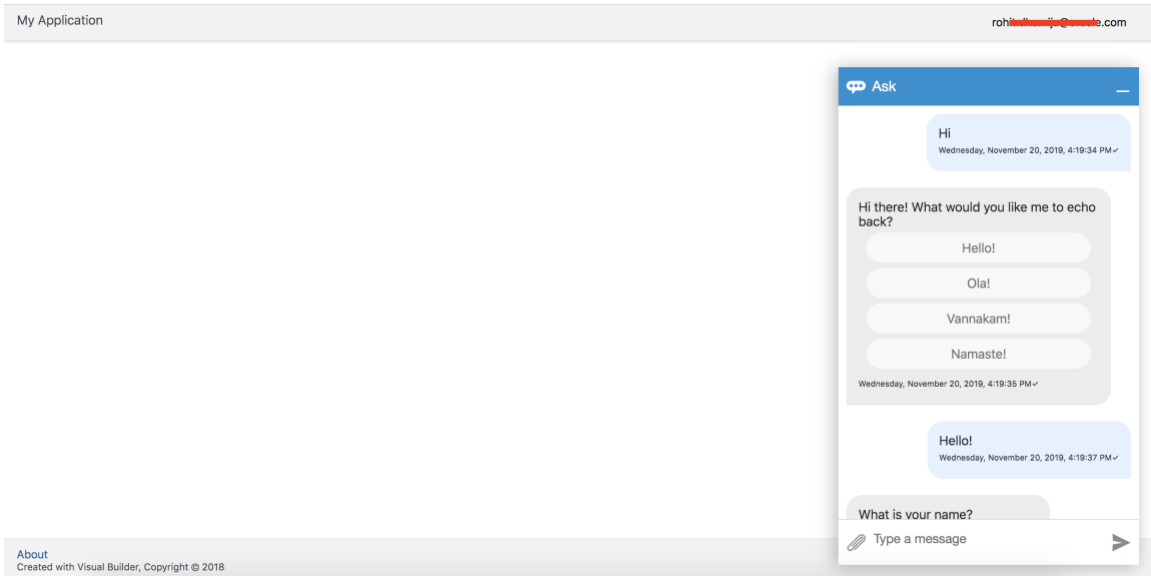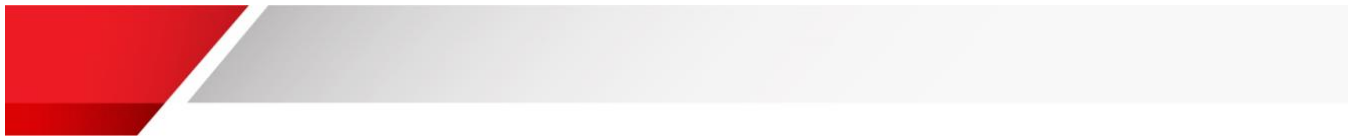
Figure 11 Updated code having bots configuration and connection

About the added code

1. At Line 9 , you are pointing to the web-sdk.js file stored in resources/native-client-sdk-js folder

2. Line 11-14, you have set the URI and channelID , which will be passed as a parameter

3. Line 15, you call the functionality after 2 seconds. This is done to ensure that the page get's loaded.

4. Line 16, you initialized the library with the configuration

5. Line 17, establishes the connection with the server

6. Line 23, defines the delay between the rendering of a page and the display of the messenger icon or widget. The default setting is 2 seconds.

Step 4: Run the project
Finally, you can run and test the integration:

1. Click on **Run**, the web application will start running in a separate tab. Observe the bottom right of the window, you will observe the chat widget icon.

2. Click on widget and start communicating with your bot

ORACLE®   Oracle Digital Assistant - TechExchange

Figure 12 VBCS project with chat-widget

Conclusion

This article shows the steps required to add the Oracle Web SDK to a Visual Builder Cloud Service page.