

ORACLE

Oracle Web Channel

Program agenda

- 1 Oracle Web SDK
- 2 Web messenger customization
- 3 Programming APIs
- 4 Web messenger security
- 5 Voice support

Program agenda

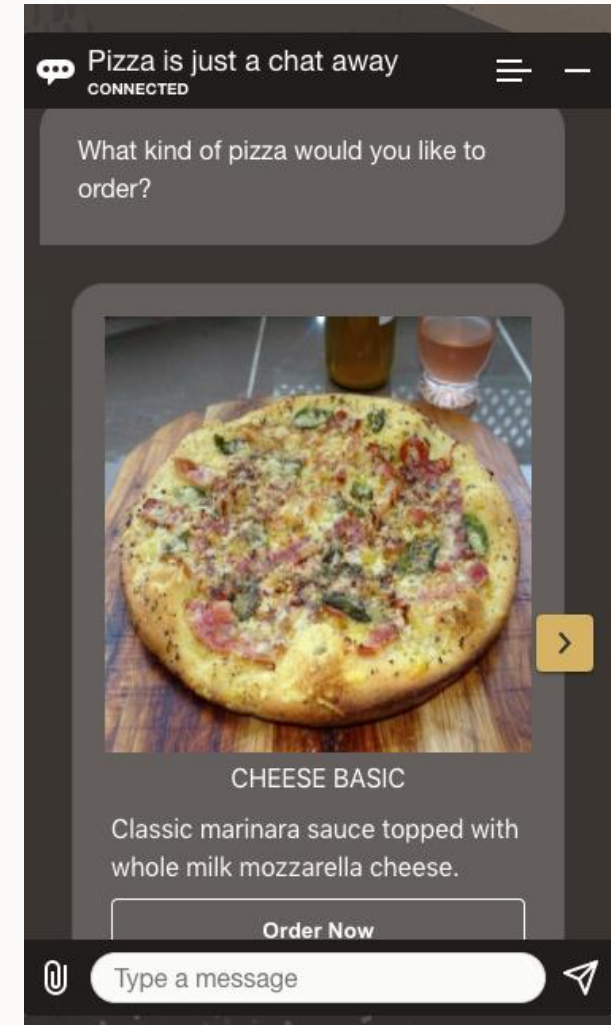
- 1 **Oracle Web SDK**
- 2 Web messenger customization
- 3 Programming APIs
- 4 Web messenger security
- 5 Voice support

Oracle Digital Assistant web integration

The integration of Oracle Digital Assistant in the Web is the most common customer use case

Oracle Digital Assistant web integration support

- Web SDK provides a widget to put on a website or application
- Oracle Web channel exposes a digital assistant or skill to web SDK



Oracle Web SDK

Web messenger widget

- Ready-made messenger
 - Use as popup window or embedded in page
- Declarative look & feel customization
- Voice support

JavaScript API

- Support for programmatic access (JavaScript)
 - Send messages
 - Intercept user messages and bot responses

Communicates with chat server in Oracle Cloud

Attachment server to upload documents

- Customize the Chat Widget

Network Configuration

Feature Flags

Functionality

Layout

Custom Icons

Custom Colors

Custom Text

Customize CSS Classes

- Features

Autocomplete

+ Delegation

Embedded Mode

Headless SDK

Long Polling

Voice Recognition

Response Narration



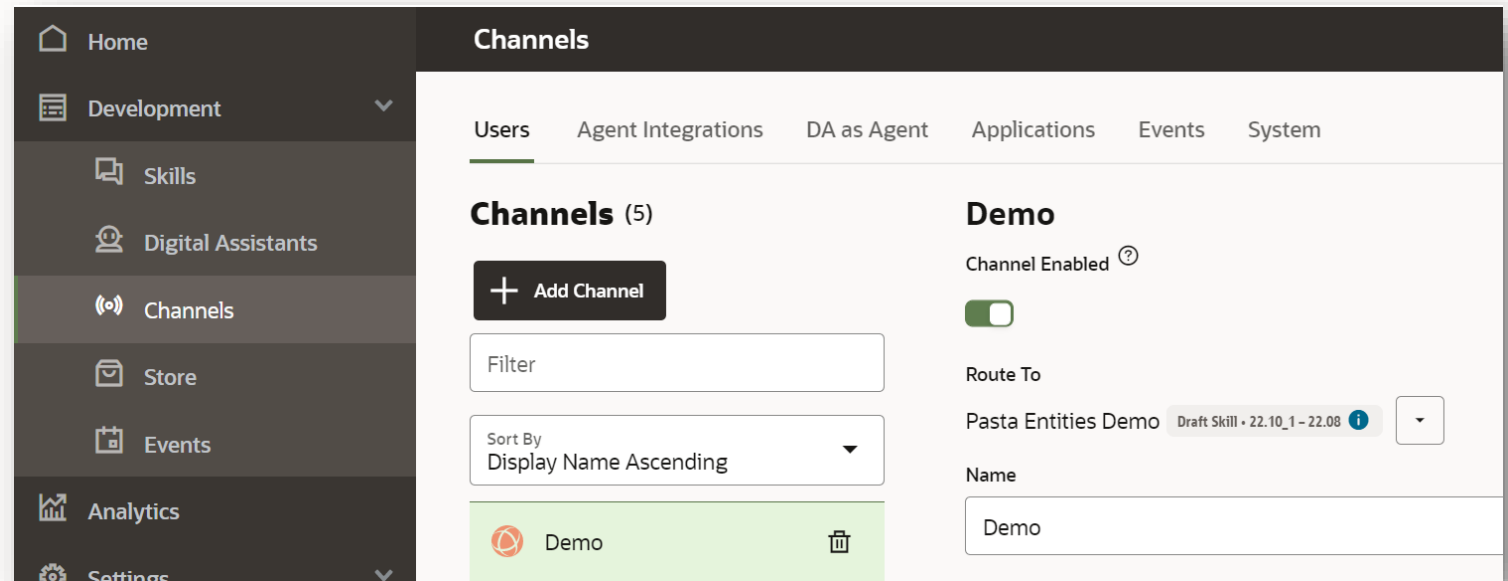
Oracle web channel

Exposes digital assistant or skill to web

- Declarative configuration
- Zero-downtime configuration

Access control

- Domain whitelisting
- Optional JWT token
 - Uses secret key



Oracle web channel configuration

Channel Enabled allows to take channel offline

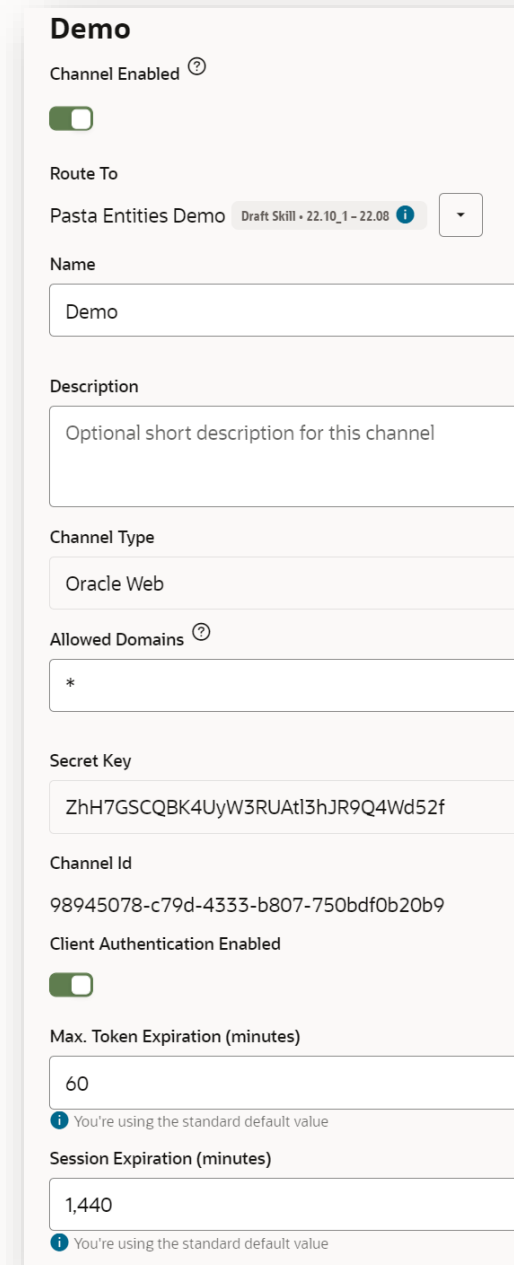
Route To binds channel to digital assistant or skill

Allowed Domains defined https://... domains that can access channel through web SDK

Secret Key needed if client authentication is set to enabled

Channel Id required for web messenger configuration

Client Authentication Enabled requires client to pass secret key or JWT token



The screenshot shows the configuration page for a channel named "Demo". The "Channel Enabled" toggle is turned on. The "Route To" dropdown is set to "Pasta Entities Demo" with a sub-label "Draft Skill - 22.10.1 - 22.08". The "Name" field contains "Demo". The "Description" field has a placeholder "Optional short description for this channel". The "Channel Type" is "Oracle Web". The "Allowed Domains" field contains an asterisk (*). The "Secret Key" field contains "ZhH7GSCQBK4UyW3RUAtI3hJR9Q4Wd52f". The "Channel Id" is "98945078-c79d-4333-b807-750bdf0b20b9". The "Client Authentication Enabled" toggle is turned on. The "Max. Token Expiration (minutes)" field is set to "60" with a note "You're using the standard default value". The "Session Expiration (minutes)" field is set to "1,440" with a note "You're using the standard default value".



Downloading the Oracle Web SDK

bit.ly/ODADoc

Oracle Digital Assistant

SDK Downloads

[Node.js SDK for the Bots Component Service](#)
[Client SDKs](#)

Oracle Native Client

These SDKs work with Oracle Digital Assis

[Android](#)
[iOS](#)
[Web](#)

Web SDK	
Module	Download
Oracle Web SDK 22.10 - Latest 11/08/2022	↓ Oracle Web SDK 22.10
Oracle Web SDK 22.08 08/30/2022	↓ Oracle Web SDK 22.08



Adding the web widget to a web application or website

Start with web sample in web sdk download

- Copy web-sdk.js and settings.js to your web application or site
- Edit settings.js with your channel details



Add JavaScript reference to website or application

- In page header

```
<head>
  ...
  <script src="scripts/settings.js"></script>
  <script src="scripts/web-sdk.js" onload="initSdk('Bots')"></script>
</head>
```

Web SDK reference

Includes:

- Custom colors/icons/text
- Security
- Feature flags

API examples for:

- Intercepting user and bot messages
- Setting user profile variables
- Sending messages
 - Hidden messages
 - Text messages
 - Postback messages



The screenshot shows a web browser window with the address bar displaying `docs.oracle.com/en/cloud/paas/digital-assistant/sdk-js/index.html`. The main heading of the page is "Oracle Digital Assistant Native Client SDK for Web". Below the heading, it indicates "Release 22.08" and "F26105-25". A brief introductory text states: "Use Oracle Digital Assistant Native Client SDK for Web to add live messaging to your website or web app." A table of contents follows, listing various sections such as "Installation", "Browser Support", "SDK Security", "Settings", and "Customizing CSS Classes".

docs.oracle.com/en/cloud/paas/digital-assistant/sdk-js/index.html

Oracle Digital Assistant Native Client SDK for Web

Release 22.08

F26105-25

Use Oracle Digital Assistant Native Client SDK for Web to add live messaging to your website or web app.

- Installation
 - HTML Script Tag
 - Import the SDK Library In JavaScript
 - Import the Library Using the Asynchronous Module Definition API
- Browser Support
- SDK Security
 - Channel with Client Auth Disabled
 - Channel with Client Auth Enabled
 - JWT Token
- Settings
 - Network Configuration
 - Feature Flags
 - Functionality Configuration
 - Layout Modification
 - Custom Colors
 - Custom Icons
 - Custom Text
 - Customizing CSS Classes



Program agenda

- 1 Oracle Web SDK
- 2 **Web messenger customization**
- 3 Programming APIs
- 4 Web messenger security
- 5 Voice support

”

The easiest way to configure and customize the Web messenger is to use an external configuration file like the one used in the Oracle Web SDK sample

Feature flags

Feature flags are properties

- Declaratively set
- Optionally, use JavaScript to set values dynamically upon messenger start

Enable or disable messenger features

- Enable "clear message" button
- Headless use of web sdk
- Open chat window on page load
- Enable speech recognition
- etc.

Property name	Optional	Default Value	Description
displayActionsAsPills	Yes	false	Display action buttons as pills
enableAttachment	Yes	true	Flag to configure attachment sharing in the chat widget
enableAutocomplete	Yes	false	Flag to enable autocomplete feature when user types a message to the skill
enableBotAudioResponse	Yes	false	Flag to configure how the skill's responses are uttered. If true, the skill's responses are spoken using the Web API.
enableClearMessage	Yes	false	Flag to enable button to clear messages on the chat widget header
enableHeadless	Yes	false	Flag to use the SDK without its UI. Allows developers to develop their own UI for chat.
enableLocalConversationHistory	Yes	false	Flag to enable loading previous conversations for given userId on the given browser when widget is initialized
enableLongPolling	Yes	false	Use http requests to connect to server if WebSocket fails to connect
enableSecureConnection	Yes	true	Flag to configure secure communication - https vs http and wss vs ws
enableSpeech	Yes	false	Flag to use the speech recognition service to convert the user voice to text messages
enableSpeechAutoSend	Yes	true	Flag to enable or disable automatically sending voice recognized text to chat server
enableTimestamp	Yes	true	Flag to enable read timestamp for messages
openChatOnLoad	Yes	false	Flag to expand chat widget when page is loaded
openLinksInNewWindow	Yes	false	If true, when a link in a message is clicked, the target is opened in a new window instead of following the user's browser preference
showConnectionStatus	Yes	false	Flag to enable display of connection status in header
showPrevConvStatus	Yes	true	Flag to configure display of status message at the end of older messages from previous conversations
showTypingIndicator	Yes	true	Flag to display chat bubble when waiting for skill's response
disablePastActions	Yes	'all'	Field to disable button clicks on messages that user has interacted with. Allowed values <code>all</code> , <code>none</code> , and <code>postback</code>



Displaying an initial welcome message

A Common customer requirement

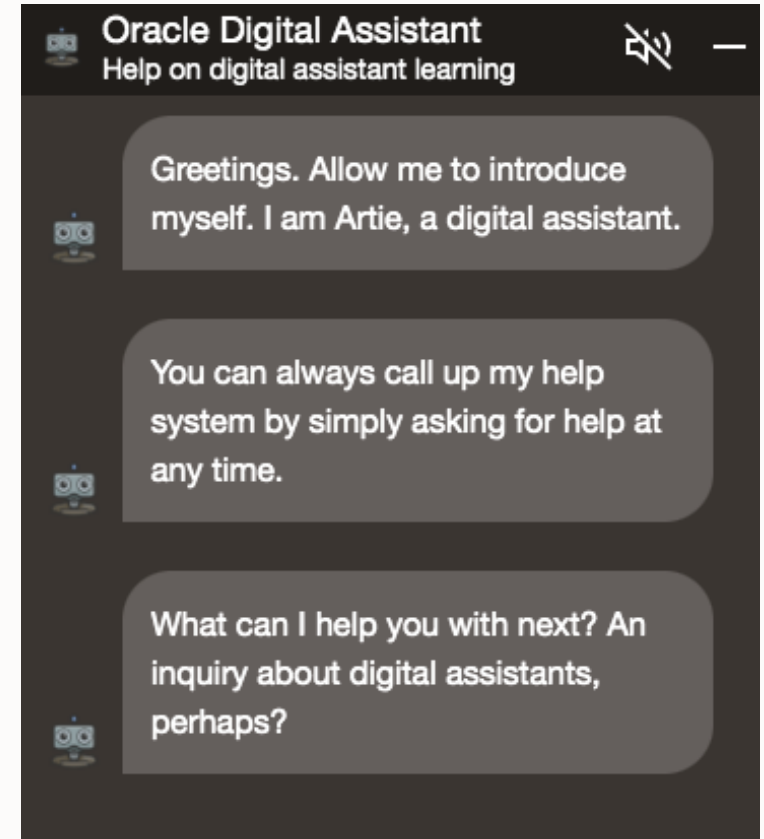
- Messenger displays message from bot when opened

Hidden message sent to bot upon messenger start

- Message interpreted by bot as if user has sent a request
 - Implicit and explicit routing, or postback message
 - User only sees bot response
 - Appears as if bot called the user

Two options

- `initUserHiddenMessage` configuration property
- `Bots.sendMessage('message', { hidden: true })` function call



Look & feel customization

Change colors & fonts

- 3 provided themes
 - *redwood-dark*
 - *dark blue theme*
 - *light blue theme*

- Custom using CSS

Move element positions

Customize icons

CCS class customization

Property name	Optional	Default Value	Description
badgePosition	Yes	<code>{"top": "0", "right": "0"}</code>	Position of the notification badge icon with respect to the skill button
colors	Yes	<code>{"branding": "#1B8FD2", "text": "#212121", "textLight": "#737373"}</code>	Colors used in the chat widget. See the Custom Colors section for more details.
conversationBeginPosition	Yes	<code>'top'</code>	Starting position of the conversation in the widget. If set to <code>'top'</code> , the first messages appear on the top of widget. If <code>'bottom'</code> , they appear on the bottom.
font	Yes	<code>'16px "Helvetica Neue", Helvetica, Arial, sans-serif'</code>	Font for chat widget
height	Yes		
messagePadding	Yes		
position	Yes		
theme	Yes		
width	Yes		

Property name	Description
<code>botButtonIcon</code>	The chat bot button that's displayed as the minimized chat button
<code>logoIcon</code>	The chat logo icon that's displayed at the header of the chat widget
<code>agentAvatar</code>	The Avatar icon that's displayed alongside the response messages coming from a live agent. If not provided, and <code>botIcon</code> is provided, then the <code>botIcon</code> is displayed. If neither <code>agentAvatar</code> nor <code>botIcon</code> is provided, then no icon is displayed.
<code>botIcon</code>	the icon alongside skill response messages. If not provided, no icon is displayed.
<code>personIcon</code>	The icon that displays alongside user messages. If not provided, no icon is displayed
<code>chatBubbleIcon</code>	The loading chat bubble icon
<code>clearMessageIcon</code>	The clear message button icon at widget header
<code>audioResponseOffIcon</code>	The icon of toggle button when audio responses are turned off
<code>audioResponseOnIcon</code>	The icon of toggle button when audio responses are turned on
<code>closeIcon</code>	The icon for the button to minimize chat widget at widget header
<code>attachmentIcon</code>	The attachment upload button icon
<code>sendIcon</code>	The send message button icon
<code>keyboardIcon</code>	The keyboard icon, displayed in button to switch from voice to keyboard mode
<code>micIcon</code>	The mic icon, displayed in button to switch from keyboard to voice mode
<code>audioIcon</code>	The audio attachment icon, displayed when attachment source URL is not reachable
<code>fileIcon</code>	The file attachment icon
<code>imageIcon</code>	The image attachment icon, displayed when attachment source URL is not reachable
<code>videoIcon</code>	The video attachment icon, displayed when attachment source URL is not reachable
<code>errorIcon</code>	The error icon. Displayed with error messages or information
<code>downloadIcon</code>	The icon for download button for skill attachment messages
<code>expandImageIcon</code>	The icon for image expand button for image attachments



Label and message customization

Declaratively change button labels and messenger strings and labels

- E.g. match bot personality in the message strings and button labels

Key	Default Value	Description
agent	'Agent'	The text used for the Agent.
agentMessage	'{0} says'	The skill message indicator for screen readers. It is spoken by the screen readers before the skill responses. The text ({0}) is replaced by the agent name.
attachment_audio	'Audio attachment'	The text used for the TTS utterance of an audio attachment.
attachment_file	'File attachment'	The text used for the TTS utterance of a file attachment.
attachment_image	'Image attachment'	The text used for the TTS utterance of an image attachment.
attachment_video	'Video attachment'	The text used for the TTS utterance of a video attachment.
attachmentAudioFallback	'Your browser does not support embedded audio. However you can {0}download it{/0}.'	The fallback message that is displayed in place of an audio attachment if the audio can not be rendered by the client. The text between {0} and {/0} is set to a link to download the file.
attachmentVideoFallback	'Your browser does not support embedded video. However you can {0}download it{/0}.'	The fallback message that's displayed in place of a video attachment if the video can not be rendered by the client. The text between {0} and {/0} is set to a link to download the file.
audioResponseOff	'Turn audio response on'	The tooltip that appears when hovering over the audio utterance on the button in the header.
audioResponseOn	'Turn audio response off'	The tooltip that appears when hovering over the audio utterance off the button in the header.
avatarAgent	'Agent icon'	The alternative text used for the agent icon that's displayed alongside the agent messages.
avatarBot	'Bot icon'	The alternative text used for the skill icon that's displayed alongside the skill messages.
avatarUser	'User icon'	The alternative text used for the user icon that's displayed alongside the user messages.
card	'Card {0}'	CRC Card identifier. The text ({0}) is replaced by the card index.

Language support

Messenger widget labels and strings can be translated

- Browser locale used to determine language
 - Locale cannot be changed using JavaScript
- Default language used if no translation found

```
"i18n": {  
  "fr": {  
    "chatTitle": "Soutien"  
  },  
  "en": {  
    "chatTitle": "Support"  
  },  
  "es": {  
    "chatTitle": "Apoyo"  
  },  
  "zh-cn": {  
    "chatTitle": "支持"  
  }  
}
```

Show partial messages

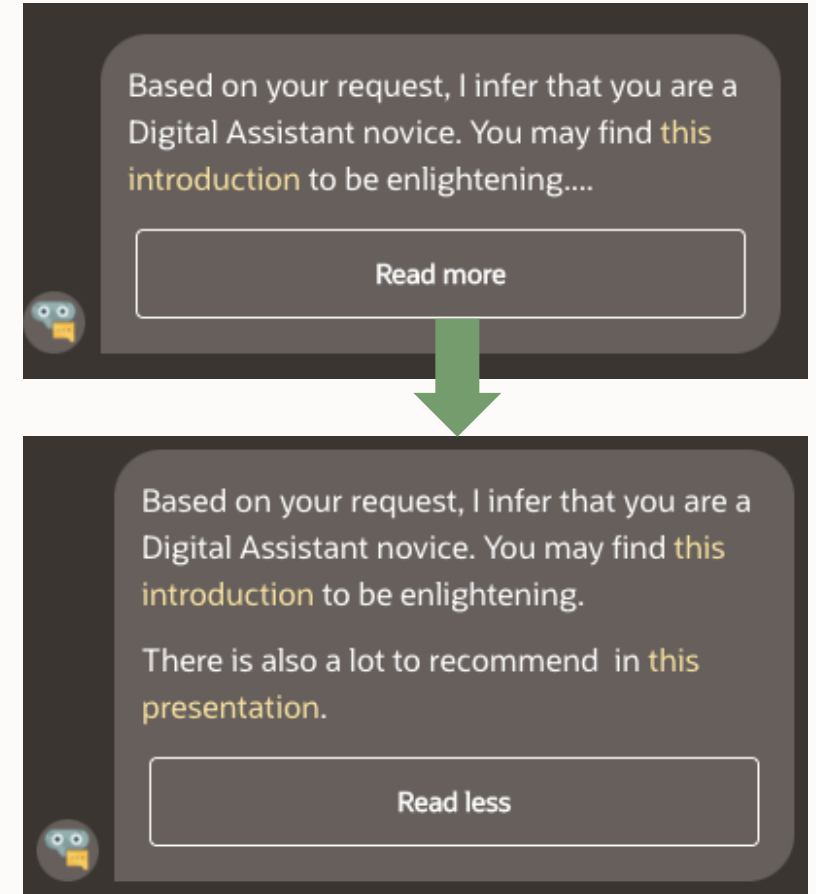
Don't make users scroll too long

Web SDK allows to split messages by a known delimiter

- Show first paragraph, hide the others
- User can expand message to see all or just continue with chat

Implementation

- Use ICU message bundle for channel specific message
- Add delimiter to message in resource bundle
- Use delegate object to filter message by delimiter



Examples of real Web SDK implementations

Hi !
This is Paddy, your personal digital assistant

TENSION-FREE IRONING
WITH INDIA'S NO 1st IRON BRAND

How can I help you today?

- Product Features
- Repair
- Demo
- Installation
- Shop.Bajaj Order Query
- Track Service Request
- Demo Video
- Locate Service Center
- Product Registration
- Consumer Care
- Shop Us
- Blog

Type a message

Zaki The Bot
Your Personal Virtual Assistant

Locators

Choose

If you are not 100% sure of what you are looking for, I will help you discover it. Just click one of the three options below to start our journey

Wednesday, October 12, 2022 at 11:01 AM ✓

Deposit & Save Money

Choose

You can just give me a quick hint and I'll bring it up

Wednesday, October 12, 2022 at 11:01 AM ✓

Type a message

Restart Conversation إعادة المحادثة

LineaMadrid

¿En qué le puedo ayudar? 😊

- Chat con un agente 🙋
- Criterios de acceso Distrito Centro 📍
- Calles en Distrito Centro 🚗
- Invitaciones para un día 🗓️
- Dar de baja un permiso 🚧
- Buscar parkings 🅇
- Distrito Centro i
- Distrito Centro - Distintivos ambientales 🟢
- Plaza Elíptica i

Escriba aquí

Hello!
CONNECTED

Hi I'm happy to help you! You can select from the options below or ask me a question. To get additional assistance please SIGN IN

Orders & Refunds

Order & refund information

- Recent order query
- Cancel a subscription
- Request a refund

Type a message...

Program agenda

- 1 Oracle Web SDK
- 2 Web messenger customization
- 3 **Programming APIs**
- 4 Web messenger security
- 5 Voice support

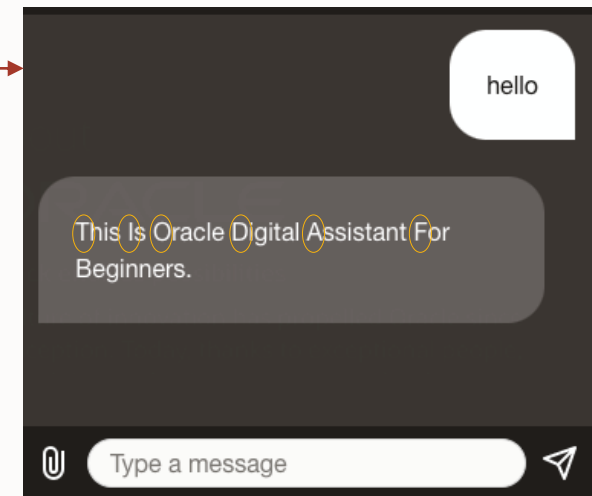
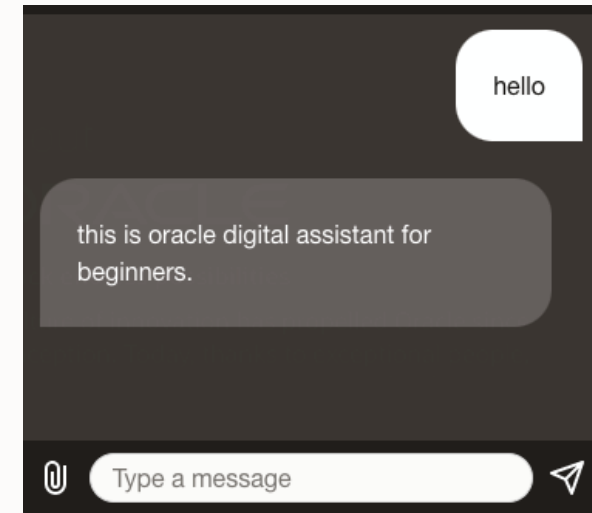
Interacting with user and bot messages

Delegate object

- JavaScript functions to interact with user and bot messages
- Messages can be read, changed and hidden

Example: Capitalize text messages

```
Bots.setDelegate({  
  beforeDisplay(message) {  
    if(message.messagePayload.type == 'text'){  
      let text = message.messagePayload.text.replace(  
        /\w\S*/g, (w) => (w.replace(/^\\w/, (c) => c.toUpperCase())));  
  
      message.messagePayload.text = text;  
    }  
    return message;  
  },  
  beforeSend(message) {return message;},  
  beforePostBackSend(postback) {return postback;}  
});
```



Sending user profile information

Oracle Digital Assistant has no direct access to the messenger

User profile information are passed from messenger as key-value pairs

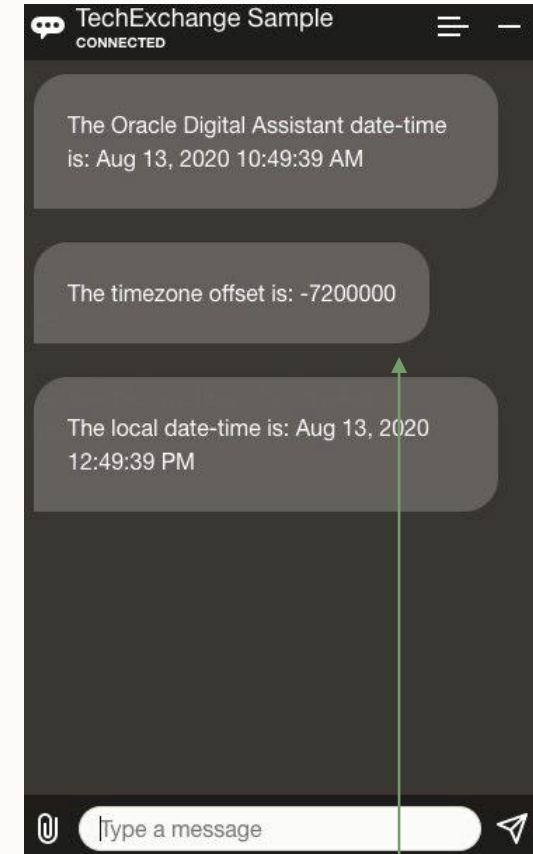
- E.g. firstName, lastName, timezoneOffset
- Accessible from BotML: `${profile.<attribute name>}`

Custom profile information can be passed using Web SDK

- `initUserProfile` configuration property
- `Bots.updateUser()` function call

Profile variables are also accessible from custom components

- `conversation.variable('profile.<attribute_name>');`



```
printTimeOffset:  
  component: "System.Output"  
  properties:  
    text: "The timezone offset is: ${profile.timezoneOffset}"  
    keepTurn: true  
  transitions:  
    next: "printLocalDate"
```

Sending user profile information: *initUserProfile*

settings.js

```
initUserProfile: {profile: {  
  firstName: 'Jane', lastName: 'Smith', email: 'me@oracle.com',  
  properties: {  
    justGotUpdated: true  
  }  
}}
```

`${profile.firstName}`

`${profile.lastName}`

`${profile.email}`

`${profile.properties.value.justGotUpdated}`

Sending messages programmatically

Message can be sent from JavaScript

- Text messages
- Postback messages
- Hidden messages

`Bots.sendMessage(<string>)`

`Bots.sendMessage(<json_object>)`

`Bots.sendMessage(<[string | object]>,[hidden: true])`

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

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


Program agenda

- 1 Oracle Web SDK
- 2 Web messenger customization
- 3 Programming APIs
- 4 **Web messenger security**
- 5 Voice support

Oracle Web SDK security mechanism

Communication uses secure web sockets (wss)

- https for document uploads

Domain whitelisting

- Restricts web channel access to websites from registered domains
 - Oracle Web SDK messenger widget embedded in web page or -app

Enforcement of https as the protocol of the hosting website

- https:// as a protocol of the surrounding website is enforced by default
- Can optionally (not recommended) be disabled

JSON Web Token (JWT) authorization

- Channel secret generated on Oracle Web channel

Domain Whitelisting

Limits access to allowed domains

- Web SDK messenger included in website downloaded from allowed domain

Domain whitelisting options

- `https://<name>.<extension>` for single domain whitelisting
- Comma separated list of domains for multiple domain whitelisting
- `*.<name>.<extension>` for using wildcard in domain listing
- Single `*` for general unrestricted access

Channel Type	Oracle Web
? * Allowed Domains	<input type="text" value="https://oracle.com"/>
Secret Key	TCLxtzH6BPLgdvRoCgjG2c3tvSulOyBg
Channel Id	a0c7ffcc-b0f2-456f-99e5-291be9b239d0

Channel Type	Oracle Web
? * Allowed Domains	<input type="text" value="https://oracle.com,*acme.com,https://acme2.com"/>
Secret Key	TCLxtzH6BPLgdvRoCgjG2c3tvSulOyBg
Channel Id	a0c7ffcc-b0f2-456f-99e5-291be9b239d0

Channel Type	Oracle Web
? * Allowed Domains	<input type="text" value="*.oracle.com"/>
Secret Key	TCLxtzH6BPLgdvRoCgjG2c3tvSulOyBg
Channel Id	a0c7ffcc-b0f2-456f-99e5-291be9b239d0

Channel Type	Oracle Web
? * Allowed Domains	<input type="text" value="*"/>
Secret Key	TCLxtzH6BPLgdvRoCgjG2c3tvSulOyBg
Channel Id	a0c7ffcc-b0f2-456f-99e5-291be9b23



JSON Web Token (JWT)

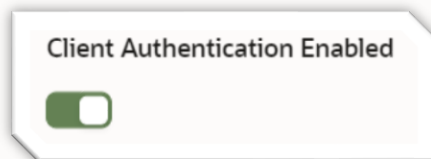
JWT token

- Contains claim (secret key) and username
- Web SDK calls JWT token generator upon web messenger start

JWT token should be generated on remote server

- Generating JWT on web page is not secure

Requires *Client Authentication Enabled* setting in Oracle web channel to be active



Tutorial <https://docs.oracle.com/en/cloud/paas/digital-assistant/tutorial-web-jwt/>

Program agenda

- 1 Oracle Web SDK
- 2 Web messenger customization
- 3 Programming APIs
- 4 Web messenger security
- 5 **Voice support**

Voice support in Oracle Web SDK

Oracle Web channel has integrated speech recognition

- Allows users to talk directly to skills and digital assistants

Voice messages are sent to chat server

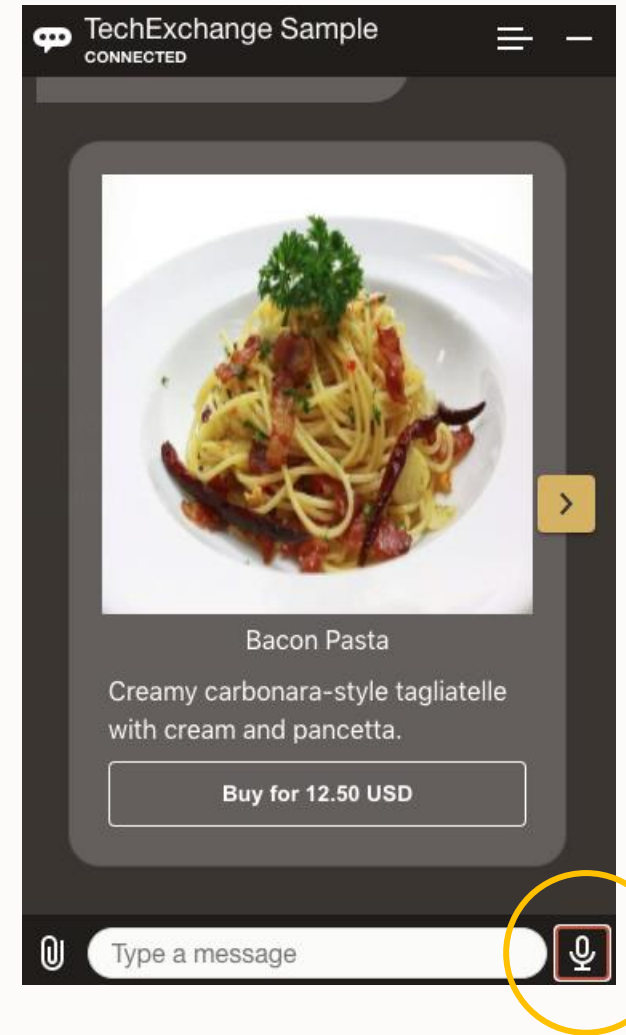
- Transforms speech to text
- Text sent to Oracle web channel

Set *enableSpeech* configuration property to true

- enables the microphone button
- displays in place of the send button

Speech recording can be edited before sending to the bot

- Disabled by default



Bot response narration

SDK narrates responses using the device's speech synthesis APIs

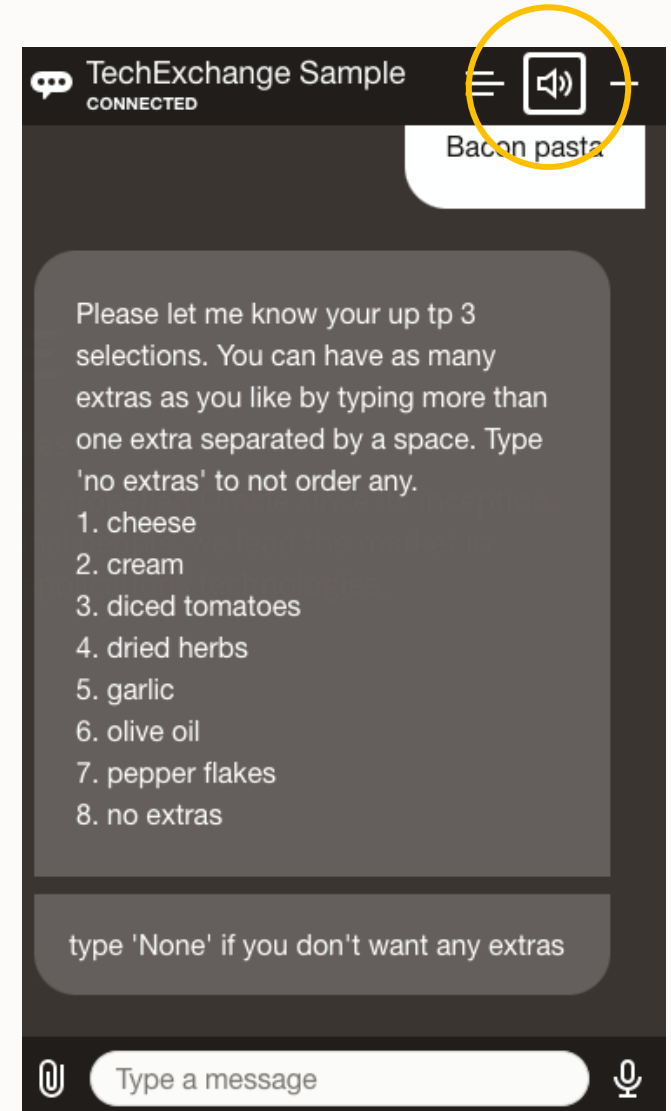
- Voice that reads the skill or digital assistant messages can be configured

Set *enableBotAudioResponse* configuration to enable narration

- Voice can be defined per locale
- Uses default voice if not voice configured for a locale

If you want to use voice narration, keep bot responses simple

- Voice reads all displayed content
- Keep bot responses short if using voice



Determine the choice of voices for a browser

Voice support depends on device

- Use script to detect browser voices
- <https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis/getVoices>

```
Bots.on(WebSDK.EVENT.WIDGET_OPENED, () => {
  console.log('Widget is opened');

  function populateVoiceList() {
    if(typeof speechSynthesis === 'undefined') {
      return;
    }

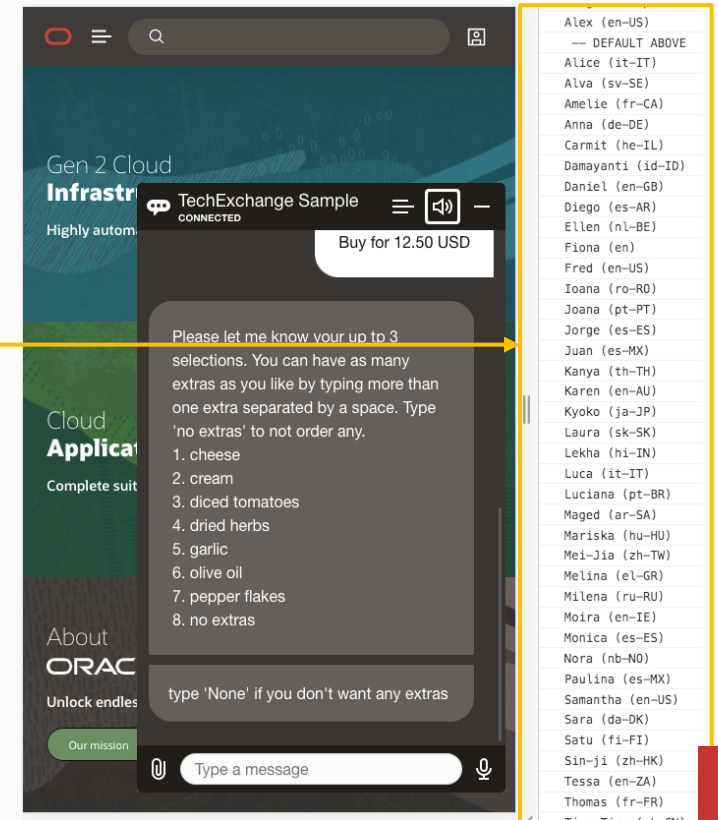
    var voices = speechSynthesis.getVoices();

    for(var i = 0; i < voices.length; i++) {
      console.log( voices[i].name + ' (' + voices[i].lang + ')');

      if(voices[i].default) {
        console.log( ' -- DEFAULT ABOVE');
      }
    }
  }

  populateVoiceList();

  if (typeof speechSynthesis !== 'undefined' && speechSynthesis.onvoiceschanged !== undefined) {
    speechSynthesis.onvoiceschanged = populateVoiceList;
  }
});
```



ORACLE