

SoundHound AI AMELIA[®]

Custom UI Integration Guide 6.0.37

Table of Contents

- [Custom UI Integration Guide](#)
- [Chat overlay](#)
 - [Usage](#)
 - [Manual configuration](#)
 - [Closing chat overlay](#)
 - [Dispatching an event from parent window to chat-overlay](#)
 - [ameliaSay](#)
 - [endConversation](#)
 - [resetConversation](#)
 - [ameliaBpn](#)
 - [ameliaConversationAttributes](#)
 - [Receiving an event in parent window from chat-overlay](#)
 - [conversationNew](#)
 - [conversationJoined](#)
 - [agentJoined](#)
 - [missedMsg](#)
 - [errorMessage](#)
 - [Receiving an event message in parent window from integration message as part of a flow block](#)
- [Configuration](#)
 - [UI configuration](#)
 - [Actions](#)
 - [Colors](#)
 - [Translations](#)
 - [Custom CSS](#)
- [Chat notes](#)
 - [Text](#)
 - [ImageLarge](#)
 - [ImageSmall](#)
 - [Table](#)
 - [TableGraph](#)
 - [TimeSelector Pie](#)
 - [TimeSelector Metric](#)
 - [TimeSelector Metrics](#)
 - [TimeSelector Radar](#)
 - [TimeSelector Horizontal Bar](#)
 - [TimeSelector Vertical Bar](#)
 - [TimeSelector Timeline](#)
 - [AdaptiveCard](#)

- [Dividers](#)
- [Badge](#)
- [ActionItem](#)
- [Widgets](#)
 - [Utility property 'selectionUtteranceMetadata'](#)
 - [SingleSelectionField](#)
 - [SingleSelectionImageField](#)
 - [MultipleSelectionField](#)
 - [MultipleSelectionImageField](#)
 - [AutocompleteField](#)
 - [SurveyField](#)
 - [MultilineFormField](#)
 - [SingleDatePickerField](#)
 - [DateRangePickerField](#)
- [Amelia Overlay in IOS WebView](#)
 - [Sending messages from an app to Amelia](#)
 - [Receiving messages from Amelia in an app](#)
 - [Push notifications with Firebase Cloud Messaging](#)
 - [Preserving user session](#)
- [Amelia Overlay in Android WebView](#)
 - [Sending messages from an app to Amelia](#)
 - [Receiving messages from Amelia in an app](#)
 - [Push notifications with Firebase Cloud Messaging](#)

Custom UI Integration Guide

The custom UI bundle is a flexible and adaptable user interface solution that can be integrated into a variety of platforms, including web browsers and iOS/Android applications. It offers a range of customization options to ensure a seamless user experience that aligns with your specific needs.

This guide provides comprehensive instructions on how to configure, embed, and interact with the custom UI bundle from a parent window or iOS/Android application. The custom UI bundle is a highly customizable user interface that can be embedded into various platforms.

Chat Overlay

Embed Amelia Custom UI into a website.

Usage

To embed the Amelia chat into a website, either create a new UI bundle or open an existing one within the Amelia Administration module. Next, navigate to the "Embedding" tab and configure the IFrame URL in the following format:

```
https://<host>/Amelia/ui/<ui-bundle>/chat?embed=iframe&domain=<domain-code>
```

Moreover, adjust style properties (colors, sizes, positions, etc.) as necessary. Then, save your changes by clicking on "Create New Revision". Next, navigate to the "Revisions" tab, deploy the latest revision, and enable "Default for Bundle" for it. Lastly, insert the following code snippet into the `<body>` section of your website (before the closing `</body>` tag):

```
<script type="text/javascript" src="https://<host>/Amelia/ui/<ui-bundle>/assets/amelia.js"></script>
```

Make sure to replace `<host>` and `<ui-bundle>` parts respectively.

NOTE: If both the website and Amelia are hosted on the same domain, the `https://<host>` part can be omitted.

Mobile devices

To prevent overflow issue in fullscreen mode when the keyboard is open, add the `interactive-widget=resizes-content` property to the meta viewport element of your website:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, interactive-widget=resizes-content" />
```

Manual configuration




In scenarios where the "Embedding" tab configuration cannot be dynamically loaded from the Amelia instance (e.g. due to a CORS policy block), there is an option to manually provide the configuration using JavaScript. This can be done by inserting an additional script tag right before the `amelia.js` script and assigning the desired configuration to the `window.ameliaChatOverlayConfig` property.

```
<script type="text/javascript">
  window.ameliaChatOverlayConfig = {
    openOnLoad: true,
    mobileDeviceBreakpoint: "768px",
    paramsOverride: {
      domain: "my_domain",
      bpm_processName: "my_process",
    },
  };
</script>
```

```
</script>
<script type="text/javascript" src="https://<host>/Amelia/ui/<ui-bundle>/assets/amelia.js"></script>
```

Refer to the table below for available configuration options.

Name	Key	Type	Default	Description
Params override	paramsOverride	object	{}	Overrides Custom UI configuration parameters
Open on page load	openOnLoad	boolean	false	Automatically open chat when the page loads (use with 'Open delay' to control timing)
Open delay	openOnLoadDelay	number	0	Delay in milliseconds before opening chat on page load
Mobile device breakpoint	mobileDeviceBreakpoint	string	640px	The CSS media query breakpoint specifies the maximum screen width at which the layout of the chat overlay will be adjusted for mobile screen sizes.
Container position	containerPosition	list	bottom-right	Chat container position Allowed values: top-left, top-right, bottom-left, bottom-right
Container horizontal offset	containerHorizontalOffset	string	24px	Distance between left/right edge of the container element and left/right edge of the viewport depending on the Container position option
Container vertical offset	containerVerticalOffset	string	24px	Distance between top/bottom edge of the container element and top/bottom edge of the viewport depending on the Container position option
Container z-index	containerZIndex	string	1000	Chat container element z-index. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/z-index
Chat width	iframeWidth	string	390px	Chat element width in pixels
Chat height	iframeHeight	string	640px	Chat element height in pixels
Chat background color	iframeBackgroundColor	color	#ffffff	Background color of chat element
Chat container border radius	iframeContainerBorderRadius	string		Border radius of chat container element. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius
Chat container transition	iframeContainerTransition	transitionConfiguration	See below	Transition of chat container element. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/transition
Chat button width	floatingButtonWidth	string	48px	Chat button element width in pixels
Chat button height	floatingButtonHeight	string	48px	Chat button element height in pixels

Chat button background color	floatingButtonBackgroundColor	color	 #2074D9	Background color of chat button element
Chat button border	floatingButtonBorder	string		Border of chat button element. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/border
Chat button border radius	floatingButtonBorderRadius	string	100%	Border radius of chat button element. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius
Chat button box shadow	floatingButtonBoxShadow	string	0px 6px 18px 0px rgba(0, 0, 0, 0.08), 0px 2px 36px 0px rgba(0, 0, 0, 0.09)	Box shadow of chat button element. For more information, refer to https://developer.mozilla.org/en-US/docs/Web/CSS/box-shadow
Chat button icon	floatingButtonIcon	resource		Icon displayed for the chat button when the chat is closed
Chat button close icon	floatingButtonCloseIcon	resource		Icon displayed for the chat button when the chat is open
Unread messages badge text color	badgeTextColor	color	 #000000	Text color of unread messages badge element
Unread messages badge background color	badgeBackgroundColor	color	 #ADE3A8	Background color of unread messages badge element
Unread messages badge font size	badgeFontSize	string	0.625rem	Font size of unread messages badge element
Unread messages badge size	badgeSize	string	16px	Size (width and height) of unread messages badge element
Unread messages badge horizontal offset	badgeHorizontalOffset	string	0px	Distance between right edge of the badge element and right edge of the floating button element
Unread messages badge vertical offset	badgeVerticalOffset	string	0px	Distance between top edge of the badge element and top edge of the floating button element

Transition configuration payload

Key	Type	Default	Description
type	string	"none"	Allowed values: none, custom
css	string	""	CSS transition definition, see the example below

```
.amelia-chat-overlay-iframe-container {  
  opacity: 0;  
  transition: opacity 200ms ease-in-out;  
}  
.amelia-chat-overlay-iframe-container[data-open="true"] {  
  opacity: 1;  
}
```

Closing chat overlay

The ability to close the chat overlay is not enabled by default. To configure this functionality, the following steps must be completed:

1. Add an action of type "Close window" to the "Main header actions" section under the Actions tab of the UI bundle configuration.
2. For the close window icon, the default "close" icon can be used. Alternatively, any other icon may be selected through the Custom Icons tab if customization is desired.
3. The mechanism for closing the window relies on sending events from the child overlay (Amelia Chat) to the parent window (the customer's website). Therefore, ensure that both the "Parent Frame URL" and "Destination Frame URL" are correctly configured in "UI configuration" tab to facilitate this communication.

Dispatching an event from parent window to chat-overlay

In order to dispatch an event from **parent window to chat-overlay**, **"Parent frame URL"** property is required to be specified in chat's configuration.

Parent frame URL is a URL|Regex or list of URLs|Regexes from which an event can be dispatched.

ameliaSay

The ameliaSay message type is used to send a message as a user to the conversation flow.

Payload

Name	Type	Description
actionData	string	This is a string value that represents the message to be sent to the conversation flow.

Example

```
/** Parent window sample code */
iframe.contentWindow.postMessage({
  actionTypes: "ameliaSay",
  actionData: "I would like to get the weather forecast for New York",
}, iframeUrl.origin);
```

endConversation

The endConversation message type is used to end the current conversation.

Payload

Name	Type	Description
actionData	object	Object containing following fields: * 'skipPreCloseSurvey' - A boolean value that determines whether the pre-close survey should be skipped. This is an optional field. If not provided, the pre-close survey will be triggered. * 'closedBy' - A string value that determines the entity that closed the conversation. This is an optional field. If not provided, the conversation will be closed by the user. Possible values: user, agent, timeout.

Example

```
/** Parent window sample code */
iframe.contentWindow.postMessage({
  actionTypes: "endConversation",
  actionData: {
    skipPreCloseSurvey: true,
    closedBy: "user"
  },
}, iframeUrl.origin);
```

resetConversation

The resetConversation message type is used to end the current conversation and start a new one.

Payload

Name	Type	Description
actionData	object	Object containing following fields: * 'domainCode' - The domain code of the new conversation. This is an optional field. If not provided, the domain code of the current conversation will be used. * 'sessionLocale' - The locale of the new conversation. This is an optional field. If not provided, the locale of the current conversation will be used. * 'skipPreCloseSurvey' - A boolean value that determines whether the pre-close survey should be skipped. This is an optional field. If not provided, the pre-close survey will be triggered. * 'closedBy' - A string value that determines the entity that closed the conversation. This is an optional field. If not provided, the conversation will be closed by the user. Possible values: user, agent, timeout.

Example

```
/** Parent window sample code */
iframe.contentWindow.postMessage({
  actionTypes: "resetConversation",
  actionData: {
    domainCode: "WeatherDomain",
    sessionLocale: "en-US"
  },
}, iframeUrl.origin);
```

ameliaBpn

The ameliaBpn message type is a specific action used to execute a conversation flow within the Amelia platform.

Payload

Name	Type	Description
actionName	string	Name of the conversation flow to be executed. This should be a valid deployed conversation flow name.
actionData	object	<p>Object containing following fields:</p> <ul style="list-style-type: none">* 'args' field is used to pass variables to the conversation flow. The args field is an object that can store arbitrary key-value pairs. This object is flexible and can accommodate a variety of data types as values. However, it is important to note that this object only supports one level of nested depth. This means that while you can have an object as a value, that object itself cannot contain another object. If you need to pass more complex, nested objects, the values of those objects need to be stringified before they are passed in the args field. Remember, any attempt to add an object within an object without stringifying it will result in an error. Always stringify nested objects before adding them to the args field.* 'message' field is an arbitrary message to be sent to the conversation flow. This message can be used to provide additional context to the conversation flow.* 'submission' field is an object that has two fields 'shouldEcho' and 'staticUtterance'. 'staticUtterance' is an optional field, and has same functionality as 'message'. If both are specified, then 'message' is preferred. 'shouldEcho' is an optional boolean field that determines whether the utterance should be echoed back to the user.

Example

```
/** Parent window sample code */
iframe.contentWindow.postMessage({
  actionTypes: "ameliaBpn",
  actionName: "WeatherConversationFlow",
  actionData: {
    args: {
      topic: "Weather",
      location: "New York",
      details: "{ 'temperature': '20C', 'condition': 'Sunny' }",
    },
    message: "Starting weather conversation flow",
    submission: {
      shouldEcho: true,
      staticUtterance: "Initiating weather conversation"
    }
  },
}, iframeUrl.origin);
```

ameliaConversationAttributes

This message is used to set custom conversation attributes. These attributes can later be used in scripts, providing a flexible way to store and retrieve conversation-specific data.

Payload

Name	Type	Description
actionData	object	<p>Object that is designed to store arbitrary key-value pairs. This object is flexible and can accommodate a variety of data types as values. However, it is important to note that this object only supports one level of nested depth.</p> <p>This means that while you can have an object as a value, that object itself cannot contain another object. If you need to store more complex, nested objects, the values of those objects need to be stringified before they are stored in the ameliaConversationAttributes object.</p> <p>Remember, any attempt to add an object within an object without stringifying it will result in an error. Always stringify nested objects before adding them to the ameliaConversationAttributes field.</p>

Example

```
/** Parent window sample code */
iframe.contentWindow.postMessage({
  actionType: "ameliaConversationAttributes",
  actionData: {
    topic: "Weather",
    location: "New York",
    details: '{"temperature':'20C','condition':'Sunny'}',
  },
}, iframeUrl.origin);

/** Conversational flow script sample code */
// Set custom conversation attribute
ameliaConversationAttributes.put("topic", "Weather");

// Get a custom conversation attribute
def topic = execution.getCustomConversationAttribute("topic"); // Returns "Weather"

// Check if a custom conversation attribute has been set
def hasTopic = execution.containsCustomConversationAttribute("topic"); // Returns true

// Get all custom conversation attributes
def attributes = execution.getCustomConversationAttributesAsMap();
```

Receiving an event in parent window from chat-overlay

In order to receive an event message in **the parent window from chat frame**, **"Destination frame URL"** is required to be specified in chat's configuration;

Destination frame URL - specifies what the origin of this window must be for the event to be dispatched, either as the literal string "*" (indicating no preference) or as a URI. If at the time the event is scheduled to be dispatched the scheme, hostname, or port of this window's document does not match that provided in targetOrigin, the event will not be dispatched; only if all three match will the event be dispatched. This mechanism provides control over where messages are sent; for example, if postMessage() was used to transmit a password, it would be absolutely critical that this argument be a URI whose origin is the same as the intended receiver of the message containing the password, to prevent interception of the password by a malicious third party. Always provide a specific targetOrigin, not *, if you know where the other window's document should be located. Failing to provide a specific target could disclose the data you send to a malicious site. Source - <https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage#targetorigin>.

JS example:

```
function handleMessage(event: MessageEvent) {
  if (event.data.notificationType === "conversationNew") {
    // code to be executed
  }
}
```

```
window.addEventListener("message", handleMessage);
```

Message types:

conversationNew

The conversationNew message type is used to notify the parent window that a new conversation has been started.

Example

```
{
  notificationType: "conversationNew",
}
```

conversationJoined

The conversationJoined message type is used to notify the parent window that the user has joined the conversation.

Example

```
{
```

```
notificationType: "conversationJoined",
}
```

agentJoined

The agentJoined message type is used to notify the parent window that an agent has joined the conversation.

Example

```
{
  "notificationType": "agentJoined",
  "notificationMessage": "Agent has joined"
}
```

missedMsg

The missedMsg message type is used to notify the parent window that there is new unread message in the conversation.

Example

```
{
  "notificationType": "missedMsg",
  "notificationMessage": "(1) New Message"
}
```

errorMessage

The errorMessage message type is used to notify the parent window that an error has occurred while processing a message sent from the parent window.

Example

```
{
  actionType: "errorMessage",
  actionData: {
```

```
    message: "A conversation has not started"  
  }  
}
```

Receiving an event message in parent window from integration message as part of a flow block.

For parent window there is a possibility to receive an event from an integration message. Below is an example of how script block can be configured to facilitate this functionality.

Example

```
import groovy.json.JsonOutput;

def website_url = "https://www.example.com";
def jsonString = '''{
  "action": "sendChatOverlayEvent",
  "payload": {
    "componentType": "MessageIntegration",
    "subComponentType": "ChatOverlay",
    "formMessageType": "chatOverlay",
    "destinationUrl": "'" + website_url + "'",
    "data": {
      type: "custom_event"
      // other attributes
    }
  }
}'''
messageService.sendIntegrationMessage(JsonOutput.toJson(jsonString));
```

In the example provided above please note that destinationUrl in payload is used as "Destination frame URL" instead of "Destination frame URL" from the chat's configuration.

JS example:

```
function handleMessage(event: MessageEvent) {
  if (event.data.type === "custom_event") {
    // code to be executed
  }
}
window.addEventListener("message", handleMessage);
```


Configuration

Custom UI configuration is a valid JSON file. Configuration is split into 3 categories:

- UI
- Colors
- Translations

UI

The "ui" field in the JSON file is dedicated to configuring the user interface elements of an application. This section allows developers or designers to define the menu items, appearance, custom css scripts and JS. See more [here](#)

Actions

The "actions" field in the JSON file allows you to customize headers by adding buttons that can be linked to various actions. See more [here](#)

Colors

The "colors" field in the JSON file focuses on defining the color scheme used throughout the application or website. It provides a centralized location to set up color variables, allowing easy management and consistency across the entire UI. Developers and designers are allowed to change base colors, the rest of the colors (e.g. for hover) will be automatically calculated. See more [here](#)

Translations

The "translations" field in the JSON file handles the internationalization and localization aspects of the application. See more [here](#)

Example

```
{
  "ui": {
    "rejoin": false
  },
  "colors": {
    "brand": "#d8113f";
  },
  "translations": {
    "en": {
      "ok": "Good"
    }
  }
}
```

UI configuration

Name	Code	Type	Default	Description
Language	language	string	"en"	Default language for chat UI
Language selection policy	languageSelectionPolicy	string	"config,user"	Comma separated list of languages to be used in chat UI. The order determines language source priority. Available values: user, message, domain, config.
Max characters in chat input box	chatInputMaxLength	number	1024	If set, maximum chat input field characters will be applied
Show powered by Amelia in chat	showAmeliaFooterInChat	boolean	true	If powered by Amelia should be displayed under the chat input box
Submit suggested response when selected	suggestedResponsesSubmit	boolean	false	If selected option from an autocomplete should be submitted immediately
Submit autocomplete option value	bpnSelectDataValue	boolean	false	If selected option from an autocomplete should be submitted by value instead of name
Number of autocomplete suggestions	numberOfAutoCompleteSuggestions	number	5	This property is to configure the maximum number of auto complete suggestions to be displayed
Show input widgets in chat	showInputWidgetsInChat	boolean	false	If input widgets should be shown in chat
Show Amelia Chat browser tab title	browserTabTitle	string	"Amelia Chat"	Can customize browser tab title
Show avatar only for latest message	showAvatarOnlyForLatestMessage	boolean	false	If avatar should be displayed only for latest message
Avatar size	avatarSize	list	40	Size of the avatar (24px, 30px, 36px or 42px respectively) Allowed values: small, medium, large, extra-large

Group messages	groupMessages	list	"none"	If messages from the same sender should be grouped Allowed values: none, top, bottom
Show custom message time format	messageTimeFormat	string	"HH:mm:ss"	if empty - no timestamp will be displayed. Otherwise use provided format
Auto Close Timeout	autoCloseTimeout	number	0	Timeout in seconds after which the chat will be closed automatically
Auto close warning timeout	autoCloseTimeoutWarning	number	0	Timeout in seconds after which user will be notified that chat will be closed automatically
Show warning modal on conversation reset	warnOnReset	boolean	false	If true show warning modal that conversation will be reset
show icon in chat input, use for toggling	showOffTheRecord	boolean	false	when active, user can click on toggle icon to prevent record messages in the chat
Scroll to first unread message from Amelia	scrollToFirstOfAmeliaMessagesGroup	boolean	false	Deprecated: use 'Scroll on new messages' parameter instead. If true, chat will scroll to the first unread message from Amelia
Scroll on new messages	scrollOnNewMessages	list	"lastUser"	Determines how the chat should scroll when new messages are received. bottom - always scroll to the bottom, firstAmelia - first Amelia message in group remains visible, lastUser - last user message remains visible Allowed values: bottom, firstAmelia, lastUser
Show microphone input in chat	showMicrophoneInChat	boolean	true	If user can input text field in the chat via microphone
Show 'Send' text in send button	showSendButtonText	boolean	false	If send button text should be displayed
Stop speech record in chat timeout	stopSpeechRecognitionAfter	number	10000	This property helps to automatically stop voice recording when timeout expired

Display like/dislike on messages	showUserFeedbackButtons	boolean	false	This property enables display of like/dislike buttons in chat bubbles. Works if "Allow Amelia to collect the feedback from user" is enabled in domain settings.
Display like/dislike in chat input after conversation flow ends	showNpsCollection	boolean	true	If like/dislike buttons should be displayed above user chat input box after conversation flow ends
Start Amelia Voice Muted	startAmeliaVoiceMuted	boolean	false	Mutes Amelia Text to Speech on initial load
Show Start New Conversation for closed conversation	showStartNewConversationForClosed	boolean	false	Show Start New Conversation action button for closed conversation
Chat notes open on start	chatNotesOpenOnStart	list	"default"	Configures behaviour of chat notes on start. If 'default' - auto-open on desktop, do not auto-open on iframe/mobile. If 'opened' - auto-open on desktop and auto-open on iframe/mobile. If 'closed' - no auto-open on desktop and no auto-open on iframe/mobile Allowed values: default, opened, closed
Speech	speech	boolean	false	Enables Amelia Text to Speech
Hide Amelia joined message in chat during session change	hideSessionChangedToAmeliaMessage	boolean	false	When this is enabled and the joining agent is Amelia, OutboundAgentSessionChangedMessage will not be displayed in the chat
Show download button for HTML present messages	showDownloadButtonForPresentHtml	boolean	false	When enabled, download button will be shown for HTML present messages
Show copy button in chat	showCopyButton	boolean	false	If user can copy chat messages
Show search in selection widget threshold	showSearchInSelectionWidgetThreshold	number	10	Number of options beyond which search input should be shown in selection widget

Selection widget style	selectionWidgetStyle	list	"buttons"	Allowed values: buttons, list
Show Amelia's name in chat	showAmeliaMessageName	boolean	true	If Amelia's name on the Amelia utterance
Show agent's name in chat	showAgentMessageName	boolean	true	If agent's name on the agent utterance
Show user's name in chat	showUserMessageName	boolean	true	If user's name on the user utterance
Show user avatar in chat	showUserAvatar	boolean	false	If user avatar should be displayed in user chat bubble
Show Amelia avatar in chat	showAmeliaAvatar	boolean	true	If Amelia avatar should be displayed in Amelia chat bubble
Use Amelia avatar from domain	useDomainAmeliaAvatar	boolean	false	If active then domain avatar will be used for Amelia, if it not exist then fallback to default Amelia avatar
Custom amelia avatar in base64	customAmeliaAvatar	resource	""	if empty - default amelia avatar will be displayed. Otherwise use base64 image
Show Agent avatar in chat	showAgentAvatar	boolean	true	If Agent avatar should be displayed in Agent chat bubble
Custom user avatar in base64	customUserAvatar	resource	""	if empty - default user avatar will be displayed. Otherwise use base64 image
Custom user background in base64	customUserBackground	resource	""	If empty - default Amelia Background will be displayed. Otherwise use base64 image
Hide user background	hideUserBackground	boolean	false	If true background will be hidden
Blur chat background	blurChatBackground	boolean	true	If true chat background will be blurred
Live agent library	liveAgentLibrary	customResources	{}	List of random agents to have a chat with instead of Amelia

Show leave site prompt	showLeaveSitePrompt	boolean	true	Show warning message before leaving the chat. Only works if rejoin is disabled. Combination of disabled 'Show leave site prompt' and 'Rejoin' is going to close conversation before current window is about to be unloaded
Show agent joined notification	agentJoinNotificationDisabled	boolean	false	Show/hide the agent join browser growl notification (uses Notifications API)
User agent joined notification interval	agentJoinNotificationInterval	number	0	The user will be notified in the interval of this property until user has acknowledged by tabbing into Customer UI or clicking notification
Custom props of agent joined notification	agentJoinNotificationOptions	object	{}	This will pass the object of allowed Notifications API options refer to https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API/Using_the_Notifications_API
Show missed messages notification	missedMessagesNotificationEnabled	boolean	true	This notifies user using Notifications API of new messages when user is away from browser tab and Amelia or agent has sent new messages. If 'Destination frame URL' is present, event is sent to the parent iframe (reference to 'Receiving an event' section in docs)
Enable sound notification for missed messages	showMissedMessagesSoundNotification	boolean	true	This notifies user using sound of new messages when user is away from browser tab and Amelia or Agent has sent new messages. Functional only if the 'Show missed messages notification' is enabled.
Custom props of missed messages notification	missedMessagesNotificationOptions	object	{}	This will pass the object of allowed Notifications API options refer to https://developer.mozilla.org/en-US/docs/Web/API/Notification#instance_properties
Show privacy policy	showPrivacyPolicy	boolean	false	Configure whether to display privacy policy button on login screen
Privacy policy content	privacyPolicyContent	array	[]	Configure content to be displayed in privacy policy modal
Base font size	baseFontSize	number	16	Base font size in pixels
Set company logo position	companyLogoPosition	list	"left"	Set company logo position in the header or hide it Allowed values: hidden, left, right, center

Hide chat input when selection is only allowed via form	hideChatInputForFormOnly	boolean	false	Hide chat input when selection is only allowed via form
Set chat position on the screen	chatPosition	list	"center"	This property allows to set chat position on the screen Allowed values: left, center, right, full
Show the default chat input banner	showDefaultBanner	boolean	false	Doesn't show the default chat input banner on load to user if set to true
Bubble shape	bubbleShape	list	"squared"	This property allows to set the shape of the chat bubbles Allowed values: squared, rounded
Hide the chat input banner when user starts typing	hideBannerOnTyping	boolean	true	Hides the chat input banner when user starts typing if set to true
Hide the chat input banner when receiving user input in widget or right rail	hideBannerOnUserMessage	boolean	true	Hides the chat input banner when receiving user input in widget or right rail if set to true
Default color theme	colorTheme	list	"light"	Default set of colors to prefer. Colors can also be overridden by user configuration Allowed values: dark, light
Page header text	pageHeaderText	string	""	Configures text displayed in the page header
Chat header text	chatHeaderText	string	""	Configures text displayed in the chat header
Hide main header	hideMainHeader	boolean	false	If set main header will be completely hidden including left menu and logo
Hide chat header	hideChatHeader	boolean	false	If set chat header will be completely hidden
Hide main menu button	hideMainMenuButton	boolean	false	Deprecated: if set main menu button will be completely hidden

Chat notes panel width	chatNotesPanelWidth	number	360	Width of the chat notes panel in pixels
Custom browser tab title icon in base64	customFavicon	resource	""	if empty - default icon will be displayed. Otherwise use base64 image
Custom company logo in base64	customUserLogo	resource	""	if empty - default Amelia Logo will be displayed. Otherwise use base64 image
Custom company logo size - header and side menu	customCompanyLogoSizeHeader	number	16	Size (height) of the company logo in the main header and side menu in pixels
Custom company logo size - login screen	customCompanyLogoSizeLogin	number	16	Size (height) of the company logo on the login screen in pixels
Hide login fields	hideLogin	boolean	false	This property will hide/show the login fields (email, password, login button)
Hide login fields for mobile devices	hideLoginOnMobile	boolean	false	This property will hide/show the login fields (email, password, login button) for mobile devices
Hide logout button and url	hideLogout	boolean	false	This property allows to show/hide logout button and url
SSO Immediate Redirect	ssoImmediateRedirect	boolean	false	Immediately redirect to SSO login page when SSO is configured
Open SSO in a new tab	ssoNewTab	boolean	false	Open SSO login page in a new tab when SSO is configured
Redirect Path Variable Name	redirectPathVarName	string	"rp"	Name of URL query parameter that will be used to redirect user after login
Logout URL	logoutUrl	string	""	The URL to hit when the user clicks logout if set e.g. using SSO especially
Google API Key	googleApiKey	string	""	Google API Key

Geo Suggest Countries	geoSuggestCountries	string	""	List of countries. Should be comma separated string.
Allow unsolicited upload in chat	allowUnsolicitedUpload	boolean	false	If user can do unsolicited upload of files in chat
Automatic confirmation of unsolicited upload	unsolicitedUploadAutoConfirm	boolean	false	If unsolicited upload should be automatically confirmed, otherwise user will be able to discard it or choose another file
Allow pasting text into chat input as HTML	allowHtmlInput	boolean	false	This property allows or disallows pasting text into chat input as HTML
Upload file max size limit	uploadMaxSize	number	40000000	Maximum file size in bytes that can be uploaded. Function only if 'Upload file max size' is 'Custom'
Upload file max size	uploadMaxSizeOptions	list	"Max allowed"	If 'Max Allowed' then 'Upload file max size limit' will be ignored and file size will be checked on the server side Allowed values: Custom, Max allowed
Allow opening conversation by ID	allowOpeningConversationById	boolean	false	Allow opening particular conversation by passing the conversation ID as a URL query parameter "id"
Parent frame URL	parentFrameUrl	string	""	URL Regex or list of URLs Regexes from which an event can be dispatched. Used to enable communication from parent window to chat overlay
Destination frame URL	destinationFrameUrl	string	""	Specifies what the origin of this window must be for the event to be dispatched. Used to enable communication from chat-overlay window to parent window. can be either as the literal string "" (indicating no preference) or as a URI. If at the time the event is scheduled to be dispatched the scheme, hostname, or port of this window's document does not match that provided in targetOrigin, the event will not be dispatched; only if all three match will the event be dispatched. This mechanism provides control over where messages are sent; for example, if postMessage() was used to transmit a password, it would be absolutely critical that this argument be a URI whose origin is the same as the intended receiver of the message containing the password, to prevent interception of the password by a malicious third party. Always provide a specific targetOrigin, not *, if you know where the other window's document should be located. Failing to provide a specific target could disclose the data you send to a malicious site. Source - https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage#targetorigin
Custom fonts	fonts	fonts	{}	List of fonts to be integrated within the chat application. Each entry includes the font name, which is to be referenced in CSS styles, along with the corresponding URL to the font source.

Custom icons	icons	customResources	{}	Map of possible custom icons. Supports SVG only base64 format or string
Chat header actions	chatHeaderActions	actions	See Actions	This property enables action buttons in the chat header
Main header actions	mainHeaderActions	actions	See Actions	This property enables action buttons in the main header
Chat menu actions	chatMenuActions	actions	See Actions	This property enables action buttons in the chat menu header
Main menu actions	mainMenuActions	actions	See Actions	This property enables action buttons in the main menu header. If no actions are provided or do not meet the requirements, the main menu button will not be displayed.
Custom css stylesheet	customCSS	css	""	Custom css that will be run on application start
Custom js script	customJS	javascript	""	Custom js script that will be run on application start
Splash page HTML	splashPageHTML	html	""	HTML content displayed on the splash page that can be accessed using /splash path
Allowed href URI schemes	allowedHrefUriSchemes	array	[]	Additional URI schemes for 'href' attributes in chat messages, extending the default whitelist of safe schemes. For example, if 'custom://path' is considered safe, add 'custom:' to the list.
Render modal in document body	renderModalInBody	boolean	false	If enabled, modal component will be rendered in the document body instead of chat container

Actions

Usage

```
{
  "ui": {
    "mainHeaderActions": {
      "icon": "chatInfo",
      "label": "chatNotes",
      "type": "default",
      "payload": {
        "actionName": "chatNotes"
      },
      "pages": [
        "chat"
      ]
    }
  }
}
```

Available actions









Actions container	Description
chatHeaderActions	This property enables action buttons in the chat header
mainHeaderActions	This property enables action buttons in the main header
chatMenuActions	This property enables action buttons in the chat menu header
mainMenuActions	This property enables action buttons in the main menu header. If no actions are provided or do not meet the requirements, the main menu button will not be displayed.

Description

Prop	Description
icon	icon key that defined in Default icons or added in scope of custom user UI Configuration icons
label	translation key that defined in Translation Keys or added in scope of custom user Translations
type	custom user property, can be used for customization. Defined in Action Types
payload	custom data that will be sent after clicking on action button. Default actions can be called by passing default "actionName" prop, defined in Default Action Names

pages	list of pages where action button will be visible. If no set will be visible on any page.
-------	---

Default Icons

Icon key	Value	Icon key	Value	Icon key	Value	Icon key	Value
resetConversation		endConversation		changeDomain		alert	
chatInfo		search		user		password	

Action Types

Action Type	Description	Payload
default	Submit Default Action Names	See below
url	Navigate by defined url	See below
execute	Execute BPN action	See below
escalate	Escalate BPN action	See below

Default payload

```
{
  "type": "default",
  "payload": {
    "actionName": "resetConversation"
  }
}
```

Default Action Names

Action Name	Description
changeDomain	Navigate to domains page
logout	Navigate to logout page
login	Navigate to login page

chatNotes	Open chat notes if available
resetConversation	Reset current active conversation
endConversation	End current active conversation

URL payload

```
{
  "type": "url",
  "payload": {
    "url": "https://example.com"
  }
}
```

Execute payload

```
{
  "type": "execute",
  "payload": {
    "actionData": {
      "submission": {
        "shouldEcho": true,
        "staticUtterance": "static"
      },
      "args": {
        "data": "actual"
      }
    },
    "actionName": "delegate"
  }
}
```

















```
{
  "type": "execute",
  "payload": {
    "actionData": {
      "message": "refresh instances",
      "args": {
        "source": "legacy",
        "instance": "1356745"
      }
    }
  }
}
```

```
    }  
  },  
  "actionName": "delegate"  
}  
}
```

Escalate payload

```
{  
  "type": "escalate",  
  "payload": {  
    "processName": "revert"  
  }  
}
```

Colors

Color	Code	Default	Description
Brand color	brand	 #2074D9	Brand color that will be use as an accent color for backgrounds and borders
Brand text color	overlayBrand	 #FFFFFF	Color for text and icons that will are displayed on top of brand color
Primary background color	background	 #1F2224	Primary background color. Defines the main theme for the application
Primary text color	primary	 #FFFFFF	Color for text and icons that will are displayed on top of background color
Amelia/Agent message background	ameliaMessageBackground	#FFFFFF0F	Amelia/Agent message background color
Amelia/Agent message border	ameliaMessageBorder	#FFFFFF00	Amelia/Agent message border color
Amelia/Agent message color	ameliaMessageColor	#FFFFFFCC	Amelia/Agent message text color
Amelia/Agent message secondary color	ameliaMessageSecondaryColor	#FFFFFF7F	Amelia/Agent message secondary text color
User message background	userMessageBackground	 #2074D9	User message background color
User message border	userMessageBorder	#2074D900	User message border color
User message color	userMessageColor	#FFFFFF	User message text color
User message secondary color	userMessageSecondaryColor	#FFFFFF7F	User message secondary text color
Secondary background color	backgroundSecondary	 #2C2F32	Secondary background color. Used for secondary elements like modals, tables and chat notes
Input background color	backgroundInput	 #0000003D	Color for secondary inputs, like search and forms
Link color	link	 #5E9EFF	Color used for links in messages and chat notes
Success message color	success	 #ADE3A8	Color used for success messages
Error message color	error	 #FF8080	Color used for error messages
Alert message color	alert	 #FFDB99	Color used for alert messages
Info message color	info	 #96D0FB	Color used for info messages
Message text color	overlayText	 #000000	Text color for messages
Chart color 1	color1	 #1982E3	Chart color 1
Chart color 2	color2	 #FEAC2A	Chart color 2
Chart color 3	color3	 #EE5F68	Chart color 3

Translations

Supported languages

Language	Key
English	en
Spanish	es
Dutch	nl
German	de
Swedish	sv
Japanese	ja
Norwegian	no
Danish	da
French	fr

Translation Keys

Text	Key
OK	ok
Done	done
loading	loading
Preview	preview
Expand	expand
Like	like
File	file
Audio	audio
Camera	camera
Would you like to provide a comment?	provideCommentQuestion

No, Thank you	noThankYou
Dislike	dislike
Delete	delete
Add comment	addComment
Edit feedback comment	editFeedbackComment
Delete feedback comment	deleteFeedbackComment
Document	document
Open Menu	menuOpenTitleText
Close Menu	menuCloseTitleText
Close	close
Submit button	submitButtonTooltipText
Mic button	micButtonTooltipText
I changed my mind	cancelMessage
Feedback	feedback
Start new conversation	newConversation
Chat with our Virtual Assistant by typing your responses in the text box.	tooltip
Submit	submit
Send	send
Search	search
Confirm	confirm
Reset conversation	resetConversation
End conversation	endConversation
See Chat	seeChat
Use Voice	useVoice
Our session has ended.	timeoutConvoEndMessageTitle

It was a pleasure chatting with you today!	timeoutConvoEndMessageBody
You've been inactive in the chat	autoCloseWarningTitle
The conversation will automatically end	autoCloseWarningMessage
Yes, close the chat	autoCloseWarningConfirm
No, I'm here	autoCloseWarningReject
Change Domain	changeDomain
Agent	agent
User	user
Amelia	amelia
Unknown	unknown
Live Agent	agentHelp
Escalating to live agent.	escalationMessage
Active Conversations	activeConversations
Privacy Policy	privacyPolicy
Contact Us	mailTo
Mute	mute
Unmute	unmute
SoundHound AI Amelia	headerLogoAltText
Start Date	startDate
End Date	endDate
Next Month	nextMonth
Previous Month	previousMonth
Month	selectMonth
Year	selectYear
Clear Start Date	clearStartDate

Clear End Date	clearEndDate
Thumb Up	thumbUp
Thumb Down	thumbDown
Chat input	chatInput
Close chat notes	closeChatNotes
Progress will be lost	logoutConfirmation.header
Continue logging out?	logoutConfirmation.body
Leave Chat	logoutConfirmation.logoutButton
Resume	logoutConfirmation.closeModalButton
Current conversation will be lost	resetConfirmation.header
Continue resetting conversation?	resetConfirmation.body
Reset Conversation	resetConfirmation.resetButton
Stay in Current Conversation	resetConfirmation.closeModalButton
Agent has joined	agentHasJoinedMessage
The file exceeds 20MB: File is too big. Max file size: 20MB	errorMessages.fileExceedsLimit
Sorry, we aren't able to process your	uploadFailMessage.sorryCantUpload
file. Try again with a different file type.	uploadFailMessage.tryAgain
Drag & Drop file here	dragAndDrop.dropFileHere
or click to upload from computer	dragAndDrop.clickToUpload
Press Enter to upload a file	dragAndDrop.pressEnterToUpload
Tap to upload file	dragAndDrop.tapToUpload
Hi,	selectDomainGreetings.hi
	selectDomainGreetings.title
Select a domain	selectDomainGreetings.selectADomain
You have access to {{count}} domains	selectDomainGreetings.domainsCount

You have access to	selectDomainGreetings.access
domains	selectDomainGreetings.toDomains
Search	selectDomainGreetings.search
No matches found...	suggestedResponseNoMatchesFound
Enter your username or email	loginHintText.userEmail
Enter your password	loginHintText.password
Login	loginMessages.login
Login again	loginMessages.loginMessage
Logout	loginMessages.logout
Log In Failed	loginMessages.loginFail
Incorrect Username/Password	loginMessages.incorrectLoginError
Username	loginMessages.usernameLabel
Password	loginMessages.passwordLabel
Logging in...	loginMessages.loggingIn
We're processing your login request. Please wait.	loginMessages.processingRequest
Kindly proceed to complete the authentication within a newly opened browser tab	loginMessages.loginInNewTab
To facilitate Single Sign-On (SSO) login functionality, kindly enable the option for 'Pop-ups and redirects' within the site settings	loginMessages.enablePopup
No input	noMultiLineInput
EDITING IN CHAT	chatNoteEditTranslations.editingInChat
Loading	chatNoteEditTranslations.inProgress
Edit	chatNoteEditTranslations.active
Open chat notes	chatNotesTitleTextOpen
Close chat notes	chatNotesTitleTextClosed
Close tooltip	closeTooltipTitleText
Close tooltip	closeTooltipAltText

{{name}} Logo	logoAltText
Close input banner	closeInputBanner
(1) New Message	unreadSingleMessage
{{count}} New Messages	unreadMultipleMessage
Leave Site? Changes that you made may not be saved.	leaveSiteMessage
Typing now...	isTyping
Confirm Conversation Reset	reset.title
Current conversation will be lost	reset.header
This action will permanently end the current conversation. Are you sure you wish to proceed?	reset.body
Reset Conversation	reset.resetButton
Stay in Current Conversation	reset.closeModalButton
Amelia is not available	ameliaNotAvailable
A conversation has not started	conversationNotStarted
Conversation is being replayed	conversationIsReplayed
Agent has joined the conversation	agentHasJoined
Listening...	isListening
Type message here	typeMessage
Chat notes	chatNotes
Chat	chat
Not able to parse integration data	errorIntegrationDataParsing
Sub component type is not supported	subComponentTypeNotSupported
Action is not supported	actionNotSupported
File information	fileInformation
Open	open
Download	download

Section	section
Show more answers	showMoreAnswers
Anonymous	anonymous
Or	or
Please enter address	geoSuggestPlaceholder
Initializing...	initializing
Logging out...	loggingOut
SoundHound AI Amelia	footer.Amelia
Powered by	footer.poweredBy
conversations	conversations
Drag and Drop your file in this area or browse files	dragDropFile
File size exceeds the limit	fileSizeExceedsLimit
Unsupported format	fileTypeNotSupported
Score	score
Search for domain	searchForDomain
I invoked an action	invokedAction
The UiPath Assistant is required to complete this action.	uiPathAssistantRequired
Don't have the UiPath Assistant?	noUiPathAssistant
Get it now.	getItNow
Grant UiPath Robot Assistant Access	grandUiPathAccess
Assistant access code: {{code}}	assistanceAccessCode
Visitor	visitor
Send {{number}} item(s)	sendMultipleItems
Record audio message	recordAudioMessage
Start recording	startRecording

Stop recording	stopRecording
Play	play
Stop	stop
Recording...	recording
Upload audio file	uploadAudioFile
Error uploading file	errorUploadingFile
Click to record	clickToRecord
Permission denied. Please allow access to your microphone in settings	microphonePermissionDenied
Choose date	chooseDate
Cancel	cancel
Save	save
Required	required
Address	address
HTML Detected	richInputPasteModal.title
How would you like to paste the contents?	richInputPasteModal.body
HTML	richInputPasteModal.htmlMode
Text	richInputPasteModal.textMode
Retain the selected preference	richInputPasteModal.retainPreference
This conversation has ended	conversationHasEnded
Start a new conversation	startNewConversation
Something went wrong	somethingWentWrong
Try Again	tryAgain
Message size exceeded by {{size}}	messageSizeExceeded
Reload	reload
Starting new conversation...	startingConversation

Restarting conversation...	restartingConversation
Failed to load conversation	errorLoadingConversation
Reconnecting...	reconnecting
Copy to Clipboard	copyToClipboard
Toggle WYSIWYG Input	toggleWysiwygInput
Loading content...	loadingContent
No results found	noResultsFound
Open actions	openActions
Close actions	closeActions
Activate message recording	activateMessageRecording
Deactivate message recording	deactivateMessageRecording
Scroll left	scrollLeft
Scroll right	scrollRight
List Box	listbox
Group	group
Sources	sources
Link Blocked by Your Browser	linkBlockedByBrowser
It looks like your browser settings have blocked this link from opening automatically. To proceed, please click the button below to open the link manually.	linkBlockedByBrowserMessage
Failed	failed
Upload in progress	pendingUploadWarning.title
Please wait until the upload is complete...	pendingUploadWarning.text
Video cannot be played or not supported	video.notSupported
Video {{filename}} from {{bucketName}}	video.description
Please fill out this field.	field.required
The input is not valid. Please ensure it matches the required format: {{format}}.	field.invalidFormat

Custom CSS

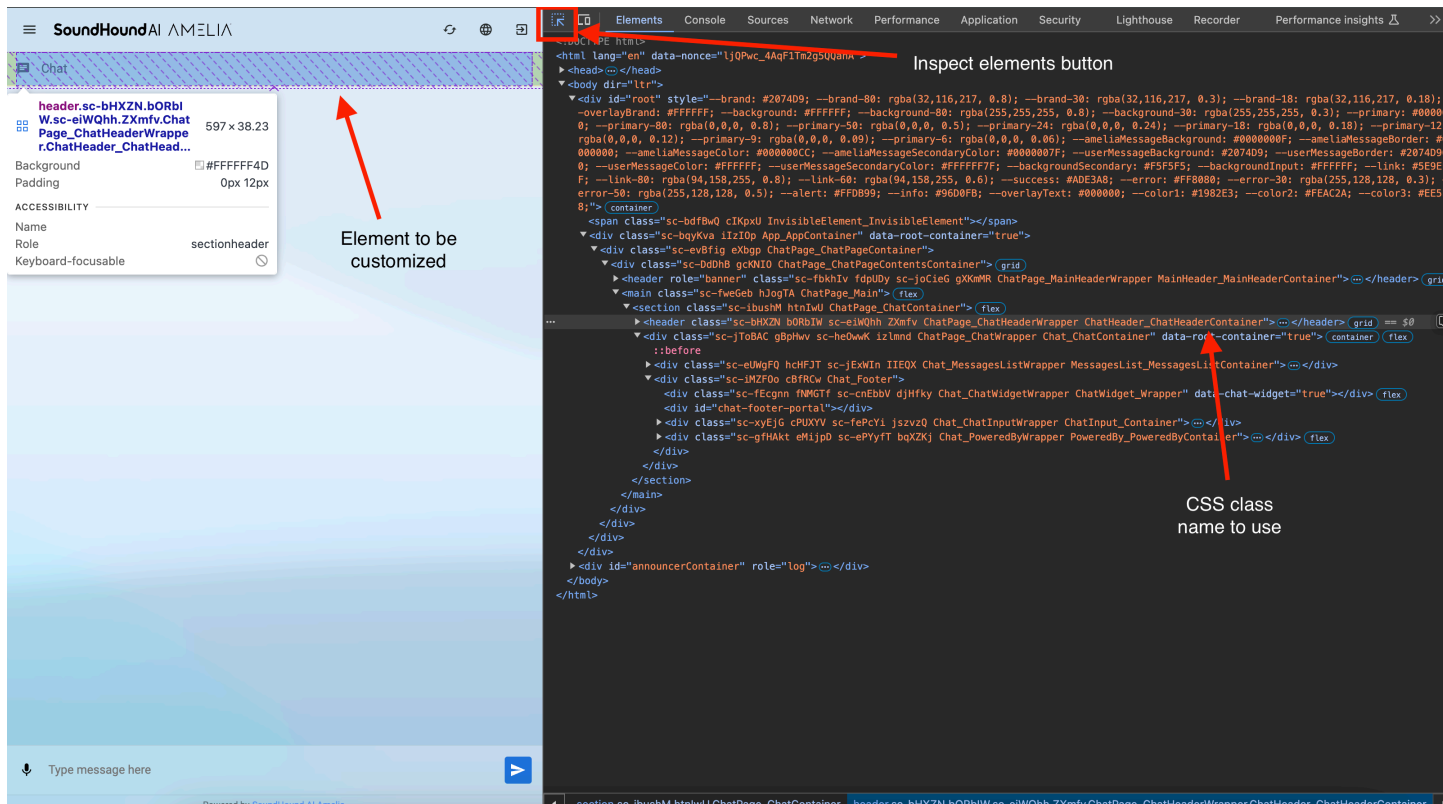
Custom CSS allows you to customize the look and feel of your UI Bundle. You can use CSS to style your UI Bundle to match your brand or to create a unique look and feel.

While custom CSS offers the flexibility to override default styles and create a unique look, **it should be used with caution** and generally as a last resort. **We recommend** that you first attempt to achieve your desired customization using the built-in **UI properties and color scheme options provided by the application**. These tools are designed to be stable and compatible across future updates, ensuring that your customizations remain intact.

The reason why Custom CSS should be used with caution has to do with code relying on the specific structure of HTML elements in the DOM (Document Object Model). As the application evolves, future updates may change the underlying DOM structure, potentially breaking your custom styles and causing unexpected issues.

How to use Custom CSS:

- Open developers console. In Chrome it would be Settings -> More tools -> Developer tools
- Go to the Elements tab
- Click on the "Inspect elements"
- Choose the element you want to customize
- Take the last class name from the element



- Chrome DevTools Elements panel showing a selected element and its CSS class names
- Apply the CSS code using the class name

- Save the changes

CSS example:

```
.ChatHeader_ChatHeaderContainer {  
    height: 72px;  
}
```

Chat notes

Utilize the `messageService.sendIntegrationMessage` method to ensure chat notes are displayed within the chat interface.

```
def payload =
  '''
    ...
  '''

messageService.sendIntegrationMessage(payload)
```

Utility properties

Property 'forceOpenState'

The `forceOpenState` property is utilized to control the automatic visibility of chat notes. This property ensures that chat notes are shown by default when a new chat session begins. Additionally, resetting the conversation triggers the chat notes to open again, maintaining their visibility for the user.

- Type: "open" | "closed"
- Default Value: "closed"

Property 'actionProcess'

The `'actionProcess'` property defines the actions that can be triggered by various components within the chat notes. This property enables interactive and dynamic functionalities, allowing users to initiate specific processes directly from the chat interface.

- Type: "object"

Properties:

- `actionLabel`: string - The label for the action, which is displayed to the user.
- `actionMessage`: string - The message that will be sent or displayed when the action is triggered.
- `actionName`: string - The name of the action process, used to identify the action.
- `actionArguments`: { [key: string]: string } - A dictionary of arguments that provide additional data for the action.
- `closeAfterAction`: "always" | "mobile" | "never" - Whether the chat notes side panel should close when an action item within the notes is clicked (always - always close chat notes panel, mobile - only close on mobile devices, never - never close chat notes panel). Defaults to "mobile".

Payload examples

Text

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "Text",
          "label": "Text Label",
          "value": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has
been the industry's standard dummy text ever since the 1500s,  when an unknown printer took a galley of type and
scrambled it to make a type specimen book."
        }
      ],
      "title": "First section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title



First section



Lorem Ipsum is simply dummy text of the *printing and typesetting* industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

ImageLarge

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "ImageLarge",
          "label": "Large Image",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasil-1977.jpg/revision/latest?cb=20160812013532",
          "description": "Description for Large Image"
        }
      ],
      "title": "First section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title



First section



Large Image

Description for Large Image



ImageSmall

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "ImageSmall",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasil-1977.jpg/revision/latest?cb=20160812013532",
          "label": "Small Image",
          "description": "It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English."
        }
      ],
      "title": "First section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title ✕

First section ⤴

Small Image

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English.



Table

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "Table",
          "label": "Table",
          "headers": [
            "Header 1",
            "Header 2",
            "Header 3",
            "Header 4",
            "Header 5"
          ],
          "rows": [
            [
              "cell 1",
              "cell 2",
              "cell 3",
              "cell 4",
              "cell 5"
            ],
            [
              "cell 1",
              "cell 2",
              "cell 3",
              "cell 4",
              "cell 5"
            ],
            [
              "cell 1",
              "cell 2",
              "cell 3",
              "cell 4",
              "cell 5"
            ]
          ]
        }
      ]
    }
  ]
}
```

Header title ✕

First section ^

Table ⌵			
HEADER 1 ⌵	HEADER 2 ⌵	HEADER 3 ⌵	HEA
cell 1	cell 2	cell 3	cel
cell 1	cell 2	cell 3	cel
cell 1	cell 2	cell 3	cel
cell 1	cell 2	cell 3	cel
cell 1	cell 2	cell 3	cel


```
    ],
    [
      "cell 1",
      "cell 2",
      "cell 3",
      "cell 4",
      "cell 5"
    ],
    [
      "cell 1",
      "cell 2",
      "cell 3",
      "cell 4",
      "cell 5"
    ]
  ]
}
],
"header": "Header title"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

TableGraph

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TableGraph",
          "label": "TableGraph",
          "src": {
            "title": "TableGraph title",
            "descr": "common graph description",
            "rows": [
              [
                {
                  "name": "00",
                  "value": "cell 1"
                },
                {
                  "name": "01",
                  "value": "cell 2"
                }
              ],
              [
                {
                  "name": "10",
                  "value": "cell 3"
                },
                {
                  "name": "11",
                  "value": "cell 4"
                }
              ]
            ]
          },
          "columns": [
            "Column 1",
            "Column 2"
          ]
        }
      ]
    }
  ]
}
```

Header title ✕

First section ^

TableGraph title ⌵

common graph description

COLUMN 1 ▾	COLUMN 2 ▾
00	01
10	11

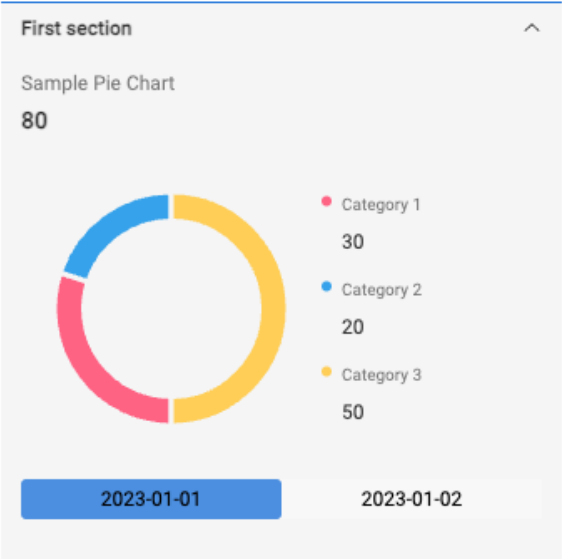
```
    ],  
    "sortOptions": {  
      "sortOrder": "asc",  
      "activeColumnIndex": "0"  
    }  
  }  
},  
  ],  
  "title": "First section"  
}  
],  
"action": "addChatNotesIntegration",  
"header": "Header title"  
}
```

TimeSelector Pie

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Pie",
          "markupProps": {
            "hasPieChart": "true",
            "dates": [
              "2023-01-01",
              "2023-01-02"
            ],
            "dateStrings": {
              "2023-01-01": "1st Jan 2023",
              "2023-01-02": "2nd Jan 2023"
            },
            "selected": "2023-01-01",
            "subComponents": [
              {
                "graphType": "pie",
                "size": "large",
                "primaryValue": "80",
                "fullCircleValue": 100,
                "title": "Sample Pie Chart",
                "dataPoints": [
                  {
                    "value": 30,
                    "color": "#FF6384",
                    "label": "Category 1",
                    "subLabel": "Details for Category 1"
                  },
                  {
                    "value": 20,
                    "color": "#36A2EB",

```

Header title



```
        "label": "Category 2",
        "subLabel": "Details for Category 2"
    },
    {
        "value": 50,
        "color": "#FFCE56",
        "label": "Category 3",
        "subLabel": "Details for Category 3"
    }
]
}
]
}
}
],
    "title": "First section"
}
],
    "action": "addChatNotesIntegration",
    "header": "Header title"
}
```

TimeSelector Metric

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Timeline",
          "markupProps": {
            "subComponents": [
              {
                "graphType": "metric",
                "title": "Account Balance",
                "bannerValue": "$19,020.22",
                "primary": "+2,000",
                "secondary": "10.5%",
                "timePeriod": "PAST MO"
              }
            ]
          }
        }
      ],
      "title": "First section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title



First section ^

Account Balance

\$19,020.22

TimeSelector Metrics

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Metrics",
          "markupProps": {
            "dates": [
              "2024-Q1",
              "2024-Q2",
              "2024-Q3",
              "2024-Q4"
            ],
            "dateStrings": {
              "2024-Q1": "Q1 2024",
              "2024-Q2": "Q2 2024",
              "2024-Q3": "Q3 2024",
              "2024-Q4": "Q4 2024"
            },
            "selected": "2024-Q2",
            "subComponents": [
              {
                "graphType": "metrics",
                "metricComponents": [
                  {
                    "title": "Revenue",
                    "bannerValue": "$1,200,000",
                    "dates": {
                      "2024-Q1": {
                        "primary": "$300,000",
                        "secondary": "Q1"
                      },
                      "2024-Q2": {
                        "primary": "$400,000",
```

Header title ×

First section ^

● Revenue

\$1,200,000

\$400,000 (Q2)

● Customer Satisfaction

89%

90% (Q2)

● New Customers

1,500

500 (Q2)

```
        "secondary": "Q2"
      },
      "2024-Q3": {
        "primary": "$250,000",
        "secondary": "Q3"
      },
      "2024-Q4": {
        "primary": "$250,000",
        "secondary": "Q4"
      }
    },
    "keyColor": "#4CAF50"
  },
  {
    "title": "Customer Satisfaction",
    "bannerValue": "89%",
    "dates": {
      "2024-Q1": {
        "primary": "85%",
        "secondary": "Q1"
      },
      "2024-Q2": {
        "primary": "90%",
        "secondary": "Q2"
      },
      "2024-Q3": {
        "primary": "88%",
        "secondary": "Q3"
      },
      "2024-Q4": {
        "primary": "89%",
        "secondary": "Q4"
      }
    },
    "keyColor": "#FFC107"
  },
  {
    "title": "New Customers",
    "bannerValue": "1,500",
    "dates": {
      "2024-Q1": {
        "primary": "400",
```



```
        "secondary": "Q1"
      },
      "2024-Q2": {
        "primary": "500",
        "secondary": "Q2"
      },
      "2024-Q3": {
        "primary": "300",
        "secondary": "Q3"
      },
      "2024-Q4": {
        "primary": "300",
        "secondary": "Q4"
      }
    },
    "keyColor": "#2196F3"
  }
]
}
]
}
],
  "title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

TimeSelector Radar

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Radar",
          "markupProps": {
            "inChatLog": "Performance Overview",
            "subComponents": [
              {
                "graphType": "radar",
                "dataPoints": [
                  {
                    "name": "Spider5",
                    "mileage": 5,
                    "price": 5,
                    "safety": 5,
                    "performance": 5,
                    "interior": 5,
                    "wood": 5,
                    "fill": "#f8f8f8",
                    "stroke": "#cccccc"
                  },
                  {
                    "name": "Spider4",
                    "mileage": 4,
                    "price": 4,
                    "safety": 4,
                    "performance": 4,
                    "interior": 4,
                    "wood": 4,
                    "fill": "white",
                    "stroke": "#cccccc"
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Header title ✕



```
{
  "name": "Spider3",
  "mileage": 3,
  "price": 3,
  "safety": 3,
  "performance": 3,
  "interior": 3,
  "wood": 3,
  "fill": "#f8f8f8",
  "stroke": "#cccccc"
},
{
  "name": "Spider2",
  "mileage": 2,
  "price": 2,
  "safety": 2,
  "performance": 2,
  "interior": 2,
  "wood": 2,
  "fill": "white",
  "stroke": "#cccccc"
},
{
  "name": "Spider1",
  "mileage": 1,
  "price": 1,
  "safety": 1,
  "performance": 1,
  "interior": 1,
  "wood": 1,
  "fill": "#f8f8f8",
  "stroke": "#cccccc"
},
{
  "name": "Spider0",
  "mileage": 0.1,
  "price": 0.1,
  "safety": 0.1,
  "performance": 0.1,
  "interior": 0.1,
  "wood": 0.1,
  "fill": "#f8f8f8",
```

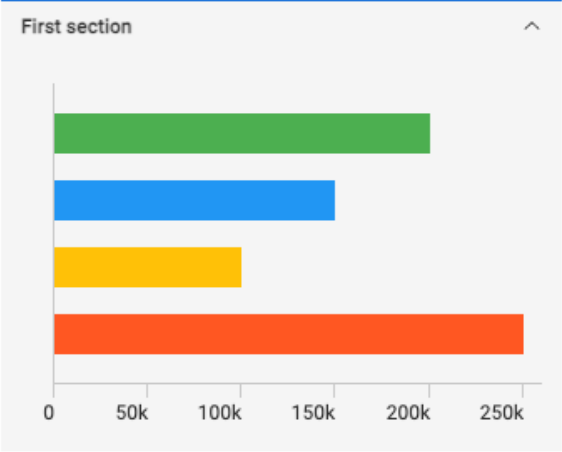
```
    "stroke": "#cccccc"
  },
  {
    "name": "Spider6",
    "mileage": 5,
    "price": 4,
    "safety": 5,
    "performance": 1.5,
    "interior": 4,
    "wood": 4.5,
    "fill": "rgba(114,172,240,0.5)",
    "stroke": "rgba(114,172,240,0.2)"
  }
],
"domains": [
  {
    "name": "mileage",
    "domain": [
      0,
      6
    ],
    "tickFormat": "round"
  },
  {
    "name": "price",
    "domain": [
      0,
      6
    ]
  },
  {
    "name": "safety",
    "domain": [
      0,
      6
    ]
  },
  {
    "name": "performance",
    "domain": [
      0,
      6
    ]
  }
]
```

```
    ]
  },
  {
    "name": "interior",
    "domain": [
      0,
      6
    ]
  },
  {
    "name": "wood",
    "domain": [
      0,
      6
    ]
  }
]
}
]
}
}
],
  "title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

TimeSelector Horizontal Bar

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Horizontal Bar",
          "markupProps": {
            "hasLargeBarGraph": "true",
            "dates": [
              "2024-Q1",
              "2024-Q2",
              "2024-Q3",
              "2024-Q4"
            ],
            "dateStrings": {
              "2024-Q1": "Q1 2024",
              "2024-Q2": "Q2 2024",
              "2024-Q3": "Q3 2024",
              "2024-Q4": "Q4 2024"
            },
            "selected": "2024-Q2",
            "subComponents": [
              {
                "graphType": "bar",
                "orientation": "horizontal",
                "colors": [
                  "#4CAF50",
                  "#2196F3",
                  "#FFC107",
                  "#FF5722"
                ],
                "dataPoints": [
                  {
                    "x": 200000,
```

Header title ✕

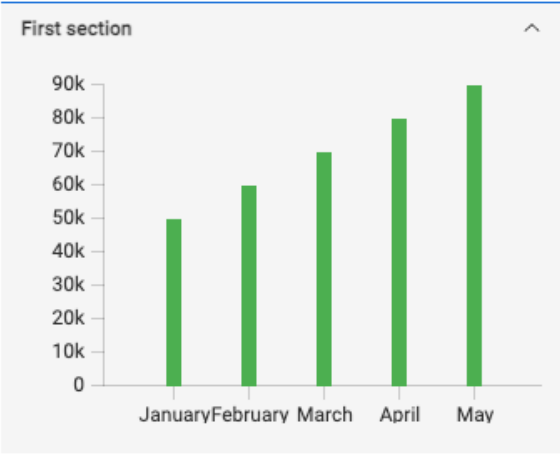


```
        "y": "Marketing",
        "color": "#4CAF50"
    },
    {
        "x": 150000,
        "y": "Sales",
        "color": "#2196F3"
    },
    {
        "x": 100000,
        "y": "HR",
        "color": "#FFC107"
    },
    {
        "x": 250000,
        "y": "IT",
        "color": "#FF5722"
    }
]
}
]
}
},
    "title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

TimeSelector Vertical Bar

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Vertical Bar",
          "markupProps": {
            "hasLargeBarGraph": "true",
            "dates": [
              "2024-01",
              "2024-02",
              "2024-03",
              "2024-04",
              "2024-05"
            ],
            "dateStrings": {
              "2024-01": "January 2024",
              "2024-02": "February 2024",
              "2024-03": "March 2024",
              "2024-04": "April 2024",
              "2024-05": "May 2024"
            },
            "selected": "2024-05",
            "subComponents": [
              {
                "graphType": "bar",
                "orientation": "vertical",
                "dataPoints": [
                  [
                    {
                      "x": "January",
                      "y": 50000,
                      "color": "#4CAF50"
                    }
                  ]
                ]
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Header title ✕

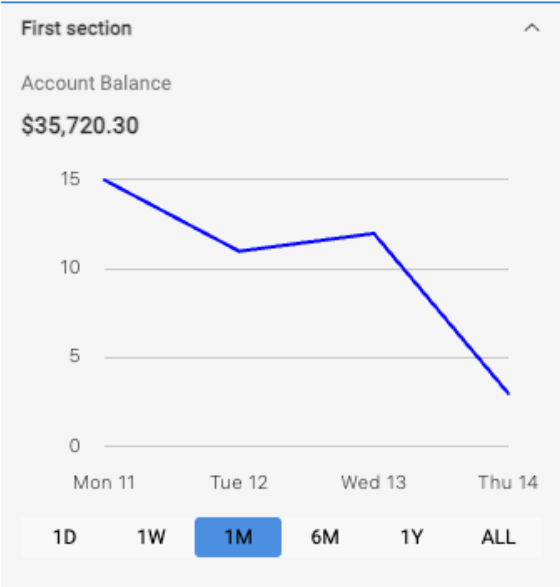



```
{
  {
    "x": "February",
    "y": 60000,
    "color": "#4CAF50"
  },
  {
    "x": "March",
    "y": 70000,
    "color": "#4CAF50"
  },
  {
    "x": "April",
    "y": 80000,
    "color": "#4CAF50"
  },
  {
    "x": "May",
    "y": 90000,
    "color": "#4CAF50"
  }
]
]
}
]
}
],
  "title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

TimeSelector Timeline

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "TimeSelector",
          "label": "Time Selector - Metric",
          "markupProps": {
            "selected": "1M",
            "dates": [
              "1D",
              "1W",
              "1M",
              "6M",
              "1Y",
              "ALL"
            ],
            "dateStrings": {
              "1D": "PAST DAY",
              "1W": "PAST WEEK",
              "1M": "PAST MO",
              "6M": "PAST 6MOS",
              "1Y": "PAST YR",
              "ALL": "ALL"
            },
            "subComponents": [
              {
                "dates": {
                  "1D": {
                    "primary": "+$300.00",
                    "secondary": "8%"
                  },
                  "1W": {
                    "primary": "+$1000.00",
                    "secondary": "25%"
                  }
                }
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Header title ✕



```
    },
    "1M": {
      "primary": "+$4,000.00",
      "secondary": "100%"
    },
    "6M": {
      "primary": "+$24,000.00",
      "secondary": "400%"
    },
    "1Y": {
      "primary": "+$100,000.00",
      "secondary": "1000%"
    },
    "ALL": {
      "primary": "-$100,000,000.00",
      "secondary": "10000%"
    }
  },
  "graphType": "timeline",
  "size": "large",
  "1D": {
    "sections": [
      {
        "dataPoints": [
          {
            "x": 1,
            "y": 3,
            "date": "Sept 11, 2017"
          },
          {
            "x": 2,
            "y": 9,
            "date": "Sept 13, 2017"
          }
        ]
      },
      {
        "keyColor": "blue",
        "bannerValue": "$19,020.22",
        "primary": "+$300.00",
        "secondary": "8%",
        "timePeriod": "PAST DAY",
        "title": "Account Balance"
      }
    ]
  }
}
```

```
]
},
"1W": {
  "sections": [
    {
      "dataPoints": [
        {
          "x": 5,
          "y": 12,
          "date": "Sept 11, 2017"
        },
        {
          "x": 6,
          "y": 3,
          "date": "Sept 13, 2017"
        }
      ],
      "keyColor": "blue",
      "bannerValue": "$9,000.00",
      "primary": "+$1000.00",
      "secondary": "25%",
      "timePeriod": "PAST WEEK",
      "title": "Account Balance"
    }
  ]
},
"1M": {
  "sections": [
    {
      "dataPoints": [
        {
          "x": 3,
          "y": 15,
          "date": "Sept 11, 2017"
        },
        {
          "x": 4,
          "y": 11,
          "date": "Sept 12, 2017"
        },
        {
          "x": 5,
```

```
        "y": 12,
        "date": "Sept 13, 2017"
    },
    {
        "x": 6,
        "y": 3,
        "date": "Sept 14, 2017"
    }
],
"keyColor": "blue",
"bannerValue": "$35,720.30",
"primary": "+$4,000.00",
"secondary": "100%",
"timePeriod": "PAST MO",
"title": "Account Balance"
}
]
},
"6M": {
    "sections": [
        {
            "dataPoints": [
                {
                    "x": 1,
                    "y": 10,
                    "date": "Sept 11, 2017"
                },
                {
                    "x": 2,
                    "y": 5,
                    "date": "Sept 12, 2017"
                },
                {
                    "x": 3,
                    "y": 15,
                    "date": "Sept 13, 2017"
                },
                {
                    "x": 4,
                    "y": 11,
                    "date": "Sept 14, 2017"
                }
            ],
```

```
{
  "x": 5,
  "y": 12,
  "date": "Sept 15, 2017"
},
{
  "x": 6,
  "y": 3,
  "date": "Sept 16, 2017"
}
],
"keyColor": "blue",
"bannerValue": "$19,020.22",
"primary": "+$24,000.00",
"secondary": "400%",
"timePeriod": "PAST 6MO",
"title": "Account Balance"
}
]
},
"1Y": {
  "sections": [
    {
      "dataPoints": [
        {
          "x": 1,
          "y": 10,
          "date": "Sept 11, 2017"
        },
        {
          "x": 2,
          "y": 5,
          "date": "Sept 12, 2017"
        },
        {
          "x": 3,
          "y": 15,
          "date": "Sept 13, 2017"
        },
        {
          "x": 4,
          "y": 11,
```

```
    "date": "Sept 14, 2017"
  },
  {
    "x": 5,
    "y": 12,
    "date": "Sept 15, 2017"
  },
  {
    "x": 6,
    "y": 3,
    "date": "Sept 16, 2017"
  },
  {
    "x": 7,
    "y": 5,
    "date": "Sept 17, 2017"
  },
  {
    "x": 8,
    "y": 6,
    "date": "Sept 18, 2017"
  },
  {
    "x": 9,
    "y": 11,
    "date": "Sept 19, 2017"
  },
  {
    "x": 10,
    "y": 1,
    "date": "Sept 20, 2017"
  },
  {
    "x": 11,
    "y": 3,
    "date": "Sept 21, 2017"
  },
  {
    "x": 12,
    "y": 5,
    "date": "Sept 22, 2017"
  },
  },
```

```
    {
      "x": 13,
      "y": 8,
      "date": "Sept 23, 2017"
    },
    {
      "x": 14,
      "y": 12,
      "date": "Sept 24, 2017"
    },
    {
      "x": 15,
      "y": 15,
      "date": "Sept 25, 2017"
    }
  ],
  "keyColor": "blue",
  "bannerValue": "$19,020.22",
  "primary": "+$100,000.00",
  "secondary": "1000%",
  "timePeriod": "PAST YR",
  "title": "Account Balance"
}
]
},
"ALL": {
  "sections": [
    {
      "dataPoints": [
        {
          "x": 1,
          "y": 10,
          "date": "Sept 11, 2017"
        },
        {
          "x": 2,
          "y": 5,
          "date": "Sept 12, 2017"
        },
        {
          "x": 3,
          "y": 15,
```



```
    "date": "Sept 13, 2017"
  },
  {
    "x": 4,
    "y": 11,
    "date": "Sept 14, 2017"
  },
  {
    "x": 5,
    "y": 12,
    "date": "Sept 15, 2017"
  },
  {
    "x": 6,
    "y": 3,
    "date": "Sept 16, 2017"
  },
  {
    "x": 7,
    "y": 5,
    "date": "Sept 17, 2017"
  },
  {
    "x": 8,
    "y": 6,
    "date": "Sept 18, 2017"
  },
  {
    "x": 9,
    "y": 11,
    "date": "Sept 19, 2017"
  },
  {
    "x": 10,
    "y": 1,
    "date": "Sept 20, 2017"
  },
  {
    "x": 11,
    "y": 3,
    "date": "Sept 21, 2017"
  },
  },
```

```
{
  "x": 12,
  "y": 5,
  "date": "Sept 22, 2017"
},
{
  "x": 13,
  "y": 8,
  "date": "Sept 23, 2017"
},
{
  "x": 14,
  "y": 12,
  "date": "Sept 24, 2017"
},
{
  "x": 15,
  "y": 8,
  "date": "Sept 25, 2017"
},
{
  "x": 16,
  "y": 12,
  "date": "Sept 26, 2017"
},
{
  "x": 17,
  "y": 9,
  "date": "Sept 27, 2017"
},
{
  "x": 18,
  "y": 5,
  "date": "Sept 28, 2017"
},
{
  "x": 19,
  "y": 12,
  "date": "Oct 03, 2017"
},
{
  "x": 19,
```

```
        "y": 14,
        "date": "Oct 04, 2017"
    },
    {
        "x": 20,
        "y": 18,
        "date": "Oct 05, 2017"
    },
    {
        "x": 21,
        "y": 21,
        "date": "Oct 08, 2017"
    }
],
"keyColor": "blue",
"bannerValue": "$19,020.22",
"primary": "-$100,000,000.00",
"secondary": "10000%",
"timePeriod": "ALL",
"title": "Account Balance"
}
]
}
]
}
]
}
],
"title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

AdaptiveCard

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "label": "Adaptive Card",
          "markupType": "AdaptiveCard",
          "markupProps": {
            "type": "AdaptiveCard",
            "version": "1.0",
            "body": [
              {
                "type": "TextBlock",
                "text": "Hello, this is an adaptive card!",
                "weight": "bolder",
                "size": "medium"
              },
              {
                "type": "Image",
                "url": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasilia-1977.jpg/revision/latest?cb=20160812013532",
                "altText": "a car",
                "size": "medium"
              },
              {
                "type": "TextBlock",
                "text": "Adaptive cards text block"
              }
            ],
            "actions": [
              {
                "type": "Action.OpenUrl",
                "title": "Learn More",
                "url": "https://adaptivecards.io"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Header title



First section



Hello, this is an adaptive card!



Adaptive cards text block

[Learn More](#)

[Execute Process](#)

```
{
  "type": "Action.Execute",
  "title": "Execute Process",
  "data": {
    "actionProcess": {
      "actionLabel": "Action Label",
      "actionMessage": "Action Message",
      "actionName": "Action Name",
      "actionArguments": {
        "first": "arg1",
        "second": "arg2"
      },
      "closeAfterAction": "always"
    }
  }
}
],
{
  "title": "First section"
}
],
"action": "addChatNotesIntegration",
"header": "Header title"
}
```

Dividers

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [],
      "title": "First section"
    },
    {
      "component": "Divider",
      "title": "Divider Uno"
    },
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [],
      "title": "Second section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title ✕

First section ^


Divider Uno

Second section ^

Badge

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "id": "unique-id-789",
      "component": "Badge",
      "title": "Sample Badge with Adaptive Card",
      "forceOpenState": "open",
      "aside": {
        "visitorPhoto": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasilia-1977.jpg/revision/latest?cb=20160812013532",
        "badgeItems": [
          {
            "text": "Important",
            "styles": {
              "color": "red",
              "fontSize": "12px"
            }
          },
          {
            "text": "New",
            "styles": {
              "color": "green",
              "fontSize": "14px"
            }
          }
        ]
      }
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

Header title



Important

New

ActionItem

```
{
  "componentType": "ChatNote",
  "payload": [
    {
      "component": "ChatNote",
      "forceOpenState": "open",
      "subcomponents": [
        {
          "subcomponent": "Markup",
          "markupType": "ActionItem",
          "label": "Action button name",
          "description": "lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum",
          "actionProcess": {
            "actionLabel": "Action Label",
            "actionMessage": "Action Message",
            "actionName": "Action Name",
            "actionArguments": {
              "first": "arg1",
              "second": "arg2"
            },
            "closeAfterAction": "mobile"
          }
        }
      ],
      "title": "First section"
    }
  ],
  "action": "addChatNotesIntegration",
  "header": "Header title"
}
```

A screenshot of a mobile application interface. At the top is a header bar with a title 'Header title' on the left and a close icon (an 'X') on the right. Below the header is a section titled 'First section' on the left and an upward-pointing chevron icon on the right. Centered within this section is a light gray rounded rectangle containing the text 'Action button name'. Below this rectangle is a paragraph of placeholder text: 'lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum lore ipsum'. The bottom of the image shows the top of a keyboard.

Widgets

There are two methods to ensure that widgets are displayed within the chat interface:

- 1. Recommended: Configure widgets through the user interface in Amelia, as part of the 'Amelia Asks' block within the DEB.
- 2. Alternative: Utilize the `messageService.sendIntegrationMessage` method within the 'Script' block.

Example of utilizing `messageService.sendIntegrationMessage` for alternative approach:

```
def payload =
    '''
        ...
    '''

messageService.sendIntegrationMessage(payload)
```

Utility property 'selectionUtteranceMetadata'

This property allows the display of selected item(s) in the chat with additional text elements such as a prefix, suffix, or conjunction, depending on the configuration.

- Type: "object"

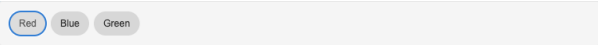
Properties:

- `prefixSingle`: string - A prefix that is added before the name of a single selected item.
- `suffixSingle`: string - A suffix that is added after the name of a single selected item.
- `prefixMultiple`: string - A prefix that is added before the names of multiple selected items..
- `suffixMultiple`: string - A suffix that is added after the names of multiple selected items.
- `conjunction`: string - A word or phrase (like "and" or "or") that is used to join the names of multiple selected items.

Payload examples for alternative approach

SingleSelectionField

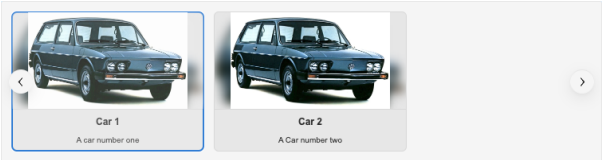
```
{
  "fields": [
    {
      "fieldType": "singleselection",
      "name": "preferred_color",
      "options": [
        {
          "name": "Red",
          "value": "red_color"
        }
      ]
    }
  ]
}
```



```
    },  
    {  
      "name": "Blue",  
      "value": "blue_color"  
    },  
    {  
      "name": "Green",  
      "value": "green_color"  
    }  
  ],  
  "selectionUtteranceMetadata": {  
    "prefixSingle": "You selected",  
    "suffixSingle": "as your preferred color."  
  }  
}  
]  
}
```

SingleSelectionImageField

```
{
  "fields": [
    {
      "fieldType": "singleselection",
      "name": "Cars",
      "options": [
        {
          "name": "Car 1",
          "value": "car1",
          "valueForDisplay": "Car one",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasil-1977.jpg/revision/latest?cb=20160812013532",
          "imageAltText": "A car number one"
        },
        {
          "name": "Car 2",
          "value": "car2",
          "valueForDisplay": "Car two",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasil-1977.jpg/revision/latest?cb=20160812013532",
          "imageAltText": "A Car number two"
        }
      ],
      "selectionUtteranceMetadata": {
        "prefixSingle": "You selected",
        "suffixSingle": "as your favorite car."
      }
    }
  ]
}
```



MultipleSelectionField

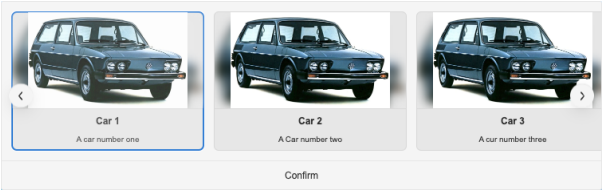
```
{
  "fields": [
    {
      "fieldType": "multipleselection",
      "name": "preferred_languages",
      "options": [
        {
          "value": "english",
          "name": "English"
        },
        {
          "value": "spanish",
          "name": "Spanish"
        },
        {
          "value": "french",
          "name": "French"
        }
      ],
      "selectionUtteranceMetadata": {
        "prefixMultiple": "You selected",
        "suffixMultiple": "as your preferred languages.",
        "conjunction": "and"
      }
    }
  ]
}
```

English Spanish French

Confirm

MultipleSelectionImageField

```
{
  "fields": [
    {
      "fieldType": "multipleselectionimage",
      "name": "Cars",
      "options": [
        {
          "name": "Car 1",
          "value": "car1",
          "valueForDisplay": "Car one",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasilia-1977.jpg/revision/latest?cb=20160812013532",
          "imageAltText": "A car number one"
        },
        {
          "name": "Car 2",
          "value": "car2",
          "valueForDisplay": "Car two",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasilia-1977.jpg/revision/latest?cb=20160812013532",
          "imageAltText": "A Car number two"
        },
        {
          "name": "Car 3",
          "value": "car3",
          "valueForDisplay": "Car three",
          "imageBase64": "http://vignette3.wikia.nocookie.net/hotwheels/images/a/ab/Vw-brasilia-1977.jpg/revision/latest?cb=20160812013532",
          "imageAltText": "A cur number three"
        }
      ],
      "selectionUtteranceMetadata": {
        "prefixMultiple": "You selected",
        "suffixMultiple": "as your favorite cars.",
        "conjunction": "and"
      }
    }
  ]
}
```



AutocompleteField

```
{
  "fields": [
    {
      "fieldType": "autocomplete",
      "name": "city",
      "isMultiselect": false,
      "options": [
        {
          "name": "New York",
          "value": "new_york",
          "selected": false,
          "disabled": false,
          "hidden": false
        },
        {
          "name": "Los Angeles",
          "value": "los_angeles",
          "selected": false,
          "disabled": true,
          "hidden": false
        },
        {
          "name": "Chicago",
          "value": "chicago",
          "selected": true,
          "disabled": false,
          "hidden": false
        }
      ],
      "selectionUtteranceMetadata": {
        "prefixSingle": "You selected",
        "suffixSingle": "as your city.",
        "prefixMultiple": "You selected",
        "suffixMultiple": "as your cities.",
        "conjunction": "and"
      }
    }
  ]
}
```

Q Search

New York

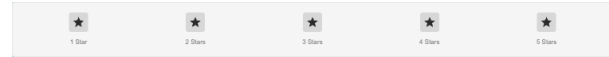
Los Angeles

Chicago

Confirm

SurveyField

```
{
  "fields": [
    {
      "fieldType": "survey",
      "data": {
        "submitOnItemClick": true,
        "cancelButton": true,
        "surveyType": "stars",
        "surveyData": [
          {
            "label": "1 Star",
            "value": 1
          },
          {
            "label": "2 Stars",
            "value": 2
          },
          {
            "label": "3 Stars",
            "value": 3
          },
          {
            "label": "4 Stars",
            "value": 4
          },
          {
            "label": "5 Stars",
            "value": 5
          }
        ]
      },
      "options": [],
      "selectionUtteranceMetadata": {
        "prefixSingle": "You rated this with",
        "suffixSingle": "stars."
      }
    }
  ]
}
```



MultilineFormField

```
{
  "fields": [
    {
      "name": "user_details",
      "fieldType": "multilineinput",
      "data": {
        "fields": [
          {
            "type": "text",
            "name": "First name",
            "value": "123",
            "regex": "^[A-Za-z]+$",
            "errorMessage": "Please enter a valid first name.",
            "required": true
          },
          {
            "type": "email",
            "name": "Email",
            "value": "456",
            "regex": "^[^@\\s]+@[^@\\s]+\\.\\.[^@\\s]+$",
            "errorMessage": "Please enter a valid email address.",
            "required": true
          },
          {
            "type": "select",
            "name": "Country",
            "value": "us",
            "regex": "789",
            "errorMessage": "",
            "required": true,
            "options": [
              {
                "value": "us",
                "text": "United States"
              },
              {
                "value": "ca",
                "text": "Canada"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

First name	<div>123</div> <div>Please enter a valid first name.</div>
Email	<div>456</div>
Country	<div>United States</div>
<div>Submit</div>	


```
    }  
  ]  
},  
"options": {},  
"selectionUtteranceMetadata": {  
  "prefixSingle": "You entered",  
  "suffixSingle": "as your details.",  
  "conjunction": "and"  
}  
}  
]  
}
```

SingleDatePickerField

```
{
  "fields": [
    {
      "fieldType": "datepicker",
      "name": "birthdate",
      "data": {
        "datePickerType": "dateSingle",
        "dateFormat": "MM/DD/YYYY",
        "dateRangeStart": "09/04/2024",
        "dontAllowDateBeforeToday": false
      }
    }
  ]
}
```

<

September ▾2024 ▾

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Submit

DateRangePickerField

```
{
  "fields": [
    {
      "fieldType": "datepicker",
      "name": "travel_dates",
      "data": {
        "datePickerType": "dateRange",
        "dateFormat": "MM/DD/YYYY",
        "dateRangeStart": "09/04/2024",
        "dateRangeEnd": "09/15/2024",
        "dontAllowDateBeforeToday": false
      }
    }
  ]
}
```

Start Date

9/4/2024

×

<

September

2024

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

End Date

9/15/2024

×

<

September

2024

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Submit

Amelia Overlay in IOS WebView

Sending messages from an app to Amelia

In case of using Amelia in WebView, `ui.parentFrameUrl` property in Amelia config should match Amelia origin. So if Amelia Custom UI loads from `https://subsomain.amelia.com/Amelia/ui/custom`, `ui.parentFrameUrl` should be `https://subdomain.amelia.com`.

One of the ways to send messages from an app to Amelia is to use `WebView.evaluateJavaScript` method. This method allows to execute JavaScript code in the context of the web view.

```
webView.evaluateJavaScript("window.postMessage({ actionTypes: \"resetConversation\", actionData: {} }, \"https://hypera.dev.amelia.com\") ")
```

Here we use `evaluateJavaScript` method to execute javascript code inside webView that calls `window.postMessage` method. Message format is the same as for Amelia Overlay embedded in an iframe.

Receiving messages from Amelia in an app

When Amelia wants to send a message to the app, it will try to call `window.webkit.messageHandlers.ameliaMessageHandler.postMessage` method. This method is not available by default, so we need to add it to the web view configuration. First, we need to add extension to our controller class that conforms to `WKScriptMessageHandler` protocol.

```
extension ViewController: WKScriptMessageHandler{
    func userContentController(_ userContentController: WKUserContentController, didReceive message: WKScriptMessage) {
        guard let dict = message.body as? [String : AnyObject] else {
            return
        }

        // process message
    }
}
```

Then we need to add a script message handler to the web view configuration.

```
let contentController = self.webView.configuration.userContentController
contentController.add(self, name: "ameliaMessageHandler")
```

Now, when Amelia sends a message to the app, `userContentController` method will be called. We can process the message in this method. Note that the name of the message handler should be "ameliaMessageHandler" as it is hardcoded in Amelia.

Push notifications with Firebase Cloud Messaging

To enable push notifications in Amelia, you need to add Firebase Cloud Messaging to your project. Follow the instructions in the [official documentation](#) to set up Firebase Cloud Messaging in your app.

To receive push notifications from Amelia, you need to send FCM token and FCM project id to Amelia as **firebaseDeviceToken** and **firebaseProjectId** initial conversation attributes.

Example

In AppDelegate.swift:

```
Messaging.messaging().delegate = self

// ...

extension AppDelegate: MessagingDelegate {
    func messaging(_ messaging: Messaging, didReceiveRegistrationToken fcmToken: String?) {
        print("Firebase registration token: \(String(describing: fcmToken))")

        let dataDict: [String: String] = ["token": fcmToken ?? ""]
        NotificationCenter.default.post(
            name: Notification.Name("FCMToken"),
            object: nil,
            userInfo: dataDict
        )
    }
}
```

In ViewController.swift:

```
override func viewDidLoad() {
    // ...

    NotificationCenter.default.addObserver(
        self,
        selector: #selector(loadWebView(notification:)),
        name: Notification.Name("FCMToken"),
        object: nil
    )
}

@objc func loadWebView(notification: NSNotification) {
    // ...
    guard let userInfo = notification.userInfo else { return }
    if let fcmToken = userInfo["token"] as? String, let projectId = FirebaseApp.app()?.options.projectID {
        guard let url = URL(string: "https://amelia-v5-currentstable-2.amelia.com/Amelia/ui/ssv/3/chat?domain=ssv&attrib_firebaseDeviceToken=\(fcmToken)&attrib_firebaseProjectId=\(projectId)") else {
            return
        }
    }
```

```
        self.webView.load(URLRequest(url: url))
    }
}
```

Here we subscribe to FCM token updates that usually happen once per app launch. When we receive a token, we produce a notification with the token as a payload. In the view controller, we listen to this notification and load the web view with the token and Firebase Project ID as query parameters in the URL.

Preserving user session

To make sure amelia chat in Web View connects to the same conversation after the app is restarted, you can save cookies from the web view and restore them when the app is launched again.

Example for IOS 11 and above

ViewController.swift

```
@objc func loadWebView(notification: NSNotification) {

    loadCookie()
    // ...
}

func saveCookie() {
    self.webView.configuration.websiteDataStore.httpCookieStore.getAllCookies { cookies in
        let data: Data = NSKeyedArchiver.archivedData(withRootObject: cookies)
        UserDefaults.standard.set(data, forKey: "cookie")
    }
}

func loadCookie() {
    let data: Data? = UserDefaults.standard.object(forKey: "cookie") as? Data
    if let cookie = data {
        let datas: NSArray? = NSKeyedUnarchiver.unarchiveObject(with: cookie) as? NSArray
        if let cookies = datas {
            for c in cookies {
                if let cookieObject = c as? HTTPCookie {
                    self.webView.configuration.websiteDataStore.httpCookieStore.setCookie(cookieObject)
                }
            }
        }
    }
}

extension ViewController: WKScriptMessageHandler{
```

```
func userContentController(_ userContentController: WKUserContentController, didReceive message: WKScriptMessage) {  
  
    saveCookie()  
  
    // ... the rest of the code  
}  
}
```

`saveCookie` method saves all cookies from the web view to the user defaults. When the app is launched again, we call `loadCookie` method to restore cookies from the user defaults to the web view.

The tricky part is when the `saveCookie` method should be called. Session cookie is not set until the chat has been initialized. Therefore, we can not call `saveCookie` in `didFinishNavigation` callback. In the example above we call `saveCookie` method every time we receive a message from Amelia application in the ios application. In order for this approach to work, we need to make sure that Amelia application is able to send messages to the app. Refer to the [Receiving messages from Amelia in an app](#) section for more details.

Another approach would be to set up a timer that will call `saveCookie` method every few seconds.

Amelia Overlay in Android WebView

Sending messages from an app to Amelia

In case of using Amelia in WebView, `ui.parentFrameUrl` property in Amelia config should match Amelia origin. So if Amelia Custom UI loads from `https://subsomain.amelia.com/Amelia/ui/custom`, `ui.parentFrameUrl` should be `https://subdomain.amelia.com`.

One of the ways to send messages from an app to Amelia is to use `WebView.evaluateJavascript` method. This method allows to execute JavaScript code in the context of the web view.

```
_webView?.evaluateJavascript(
    "window.postMessage({ actionType: \"resetConversation\", actionData: {} }, '*');"
    null
);
```

Here we use `evaluateJavascript` method to execute javascript code inside webView that calls `window.postMessage` method. Message format is the same as for Amelia Overlay embedded in an iframe.

Receiving messages from Amelia in an app

When Amelia wants to send a message to the app, it will try to call `window.ameliaMessageHandler.postMessage` method. This method is not available by default, so we need to add it to the web view configuration. First, create a class that will process messages from Amelia.

```
class WebViewMessageTarget {
    @JavascriptInterface
    fun postMessage(json: String) {
        // process message
    }
}
```

Then we inject an object of this class to the web view so it can be accessed from JavaScript.

```
webView.addJavascriptInterface(WebViewMessageTarget(), "ameliaMessageHandler")
```

Note that the name of the message handler should be "ameliaMessageHandler" as it is hardcoded in Amelia.

Push notifications with Firebase Cloud Messaging

To enable push notifications in Amelia, you need to add Firebase Cloud Messaging to your project. Follow the instructions in the [official documentation](#) to set up Firebase Cloud Messaging in your app.

To receive push notifications from Amelia, you need to send FCM token to Amelia as **firebaseDeviceToken** initial conversation attribute.

Example:

```
FirebaseMessaging.getInstance().token.addOnCompleteListener(OnCompleteListener { task ->
    if (!task.isSuccessful) {
        Log.w(TAG, "Fetching FCM registration token failed", task.exception)
        return@OnCompleteListener
    }
    _webView?.loadUrl(String.format("https://hypera.dev.amelia.com/Amelia/ui/ssv/21/chat?domain=ssv&attrib_firebaseDeviceToken=%s", task.result))
})
```

Here we subscribe to FCM token updates and send the token to Amelia as a query parameter in the URL.