

LiveConnect Web Widget Configuration

Architecture Diagram



Strings customization

The list of localizable strings that can be modified. If an option is not given a custom string, the default value will be used.

Option	Default value
actionPostBackError	An error occurred while processing your action. Please try again.
clickToRetry	Message not delivered. Click to retry.
clickToRetryForm	Form not submitted. Click anywhere on the form to retry.
connectNotificationText	Give us a way to reach out:
connectNotificationSingleText	Be notified when you get a reply.
connectNotificationSingleButtonText	Turn on <name> notifications
connectNotificationOthersText	others
contactUsButtonText	Contact Us
conversationTimestampHeaderFormat	MMMM D YYYY, h:mm A
couldNotConnect	Couldn't connect. You can {retry}.
couldNotConnectInner	retry connecting now
couldNotConnectWithRetry	Couldn't connect. We'll keep retrying, or you can {retry}.
couldNotConnectWithRetryInner	try now
emailChangeAddress	Change my email
emailDescription	To be notified by email when you get a reply, enter your email address.
emailFieldLabel	Your email
emailFieldPlaceholder	Your email address
emailFormButton	Continue
fetchHistory	Load more
fetchingHistory	Retrieving history...
fileTooLargeError	Max file size limit exceeded ({size})
fileTypeError	Unsupported file type.
formErrorInvalidEmail	Email is invalid
formErrorNoLongerThan	Must contain no more than ({characters}) characters
formErrorNoShorterThan	Must contain at least ({characters}) characters
formErrorUnknown	This doesn't look quite right
formFieldSelectPlaceholderFallback	Choose one...

frontendEmailChannelDescription	To talk to us using email just send a message to our email address and we'll reply shortly:
headerMessage	Start chatting live with an associate, or reach out to customer support using the link below.
headerText	How can we help?
imageClickToReload	Click to reload image.
imageClickToView	Click to view {size} image.
imagePreviewNotAvailable	Preview not available.
inputPlaceholder	Type a message...
inputPlaceholderBlocked	Complete the form above...
introAppText	Message us below or from your favourite app.
introductionText	We're here to talk, so ask us anything!
linkError	An error occurred when attempting to generate a link for this channel. Please try again.
messageError	An error occurred while sending your message. Please try again.
messageIndicatorTitlePlural	{{count}} New messages
messageIndicatorTitleSingular	{{count}} New message
messageRelativeTimeDay	{value}d ago
messageRelativeTimeHour	{value}h ago
messageRelativeTimeJustNow	Just now
messageRelativeTimeMinute	{value}m ago
messageTimestampFormat	h:mm A
messageSending	Sending...
messageDelivered	Delivered
messengerChannelDescription	Connect your Facebook Messenger account to be notified when you get a reply and continue the conversation on Facebook Messenger.
notificationSettingsChannelsDescription	You can also talk to us from your favourite app or service.
notificationSettingsChannelsTitle	Other Channels
notificationSettingsConnected	Connected
notificationSettingsConnectedAs	Connected as {username}
prechatCaptureGreetingText	Please fill out the form to get connected with an associate.
prechatCaptureNameLabel	Name
prechatCaptureNamePlaceholder	Enter your name
prechatCaptureEmailLabel	Email
prechatCaptureEmailPlaceholder	name@company.com
prechatCaptureConfirmationText	Thanks! We are attempting to find an associate for you.
prechatCaptureMailgunLinkingConfirmation	You'll be notified here and by email at {email} once we reply.
sendButtonText	Send
settingsHeaderText	Settings
shareLocation	Share location
tapToRetry	Message not delivered. Tap to retry.

tapToRetryForm	Form not submitted. Tap anywhere on the form to retry.
unsupportedMessageType	Unsupported message type.
unsupportedActionType	Unsupported action type.
uploadInvalidError	Invalid file.
uploadPhoto	Upload photo.
uploadVirusError	A virus was detected in your file and it has been rejected

Example:

```
const customText = {
  'en-US': {
    prechatCaptureGreetingText: 'Please fill out the form to get
connected with a fashion advisor.',
    prechatCaptureConfirmationText: 'Thanks for that! We are
attempting to find a fashion advisor for you.',
    headerText: '<Header Text>'
  }
};
```

Configuration options

`init(options)`

Initializes the Tulip Live Chat widget in the web page using the specified options. It returns a promise that will resolve when Tulip Live Chat is ready.

Option	Optional?	Default value	Description
locale	Yes	en-US	Locale used for date formatting using the <language>-<COUNTRY> format. Language codes can be found here and country codes here . <ul style="list-style-type: none"> *Note 1 : ** The country part is optional, and if a country is either not recognized or supported, it will fallback to using the generic language. If the language isn't supported, it will fallback to en-US. Note 2: this is <i>only</i> used for date formatting and doesn't provide built-in translations for Tulip Live Chat. Translations need to be passes as customText instead.
imageUploadEnabled	Yes	true	Enables the image upload feature. (deprecated: use <code>menuItems.imageUpload</code> ; if this option is <code>false</code> , it will disable <code>menuItems.imageUpload</code> and <code>menuItems.fileUpload</code>)
fixedIntroPane	Yes	false	When enabled, the introduction pane will be pinned at the top of the conversation instead of scrolling with it.
embedded	Yes	False	Tells the widget it will be embedded. (see Embedded section below)
displayStyle	Yes	button	Choose how the widget will appear on your website. Must be either <code>button</code> or <code>tab</code> . When <code>tab</code> selected, it'll appear as a pane at the bottom.
buttonIconUrl	Yes		When the <code>displayStyle</code> is <code>button</code> , you have the option of selecting your own button icon. The image must be at least 200 x 200 pixels and must be in either JPG, PNG, or GIF format.
buttonWidth	Yes	58px	When the <code>displayStyle</code> is <code>button</code> , you have the option of specifying the button width.

buttonHeight	Yes	58px	When the <code>displayStyle</code> is <code>button</code> , you have the option of specifying the button height.
businessName	Yes		A custom business name.
businessIconUrl	Yes		A custom business icon url. The image must be at least 200 x 200 pixels and must be in either JPG, PNG, or GIF format.
contactUsButtonUrl	Yes		Link to a Contact Us page
customColors	Yes	See below.	Colors used in the Tulip Live Chat UI.
customText	Yes	See the example below	Strings used in the Tulip Live Chat UI. You can use these to either customize the text or translate it. <i>Note:</i> Some strings include variables (surrounded by <code>{ }</code>) which must remain in your customized text.
browserStorage	Yes	<code>localStorage</code>	Choose the storage type to use for storing user identity in the browser. Must be either <code>localStorage</code> or <code>sessionStorage</code> . See below
prechatCapture	Yes	See below	Enables automatically capturing a user's name and email via a form before the start of a conversation. Disables the chat input until the form has been submitted.

customColors

Option	Optional?	Default value	Description
brandColor	Yes	ee3d50	This color will be used in the messenger header and the button or tab in idle state. Must be a 3 or 6-character hexadecimal color.
conversationColor	Yes	151043	This color will be used for customer messages, quick replies and actions in the footer. Must be a 3 or 6-character hexadecimal color.
actionColor	Yes	333333	This color will be used for call-to-actions inside your messages. Must be a 3 or 6-character hexadecimal color.

Prechat Capture

Prechat Capture is an add-on built by which allows you to capture a user's data like name and email before the start of a conversation.

When enabled, the chat input is disabled and the user is greeted by a welcome message and a form asking for their name. Once the form is successfully submitted, a thank you message is sent to the user and the conversation is then initialized, connecting the user to your business.

Option	Description
prechatCaptureGreetingText	Text for the initial greeting message.
prechatCaptureNameLabel	Label displayed for the default form's first field.
prechatCaptureNamePlaceholder	Placeholder for the default form's first field.
prechatCaptureEmailLabel	Label displayed for the default form's second field.
prechatCaptureEmailPlaceholder	Placeholder for the default form's second field.
prechatCaptureConfirmationText	Text for the confirmation message sent when the form is completed.
prechatCaptureMailgunLinkingConfirmation	Text for the notification message when a user has linked their email address.

widgetConfiguration.prechatCapture

Option	Optional?	Default value	Description
enabled	Yes	<code>true</code>	Enables the prechat capture experience.
fields	Yes		Overrides the default Prechat Capture fields.

widgetConfiguration.prechatCapture.fields

Fields are included in the `fields` array of a `form` type message to specify the fields to be completed.

Arguments	Optional?	Description
<code>type</code>	No	Specifies the type of field. Options are <code>text</code> , <code>email</code> , <code>select</code> .
<code>name</code>	No	Specifies the name of the field. Each field name must be unique per form. Maximum length of 128 characters.
<code>label</code>	No	Specifies what label the field will be displayed with. Maximum length of 128 characters.
<code>placeholder</code>	Yes	Specifies the placeholder text of the field that will be rendered. Maximum length of 128 characters.
<code>minSize</code>	Yes	<code>Text</code> field only. Specifies the minimum possible length of the response. Defaults to 1 if not specified.
<code>maxSize</code>	Yes	<code>Text</code> field only. Specifies the maximum possible length of the response. Defaults to 128 if not specified.
<code>options</code>	Yes	<code>Select</code> field only. Array of option objects (see below) that can be selected. The array is limited to 20 options.

Option

Options are included in the `options` array of a `select` type field to specify which options are available for selection.

Arguments	Optional?	Description
<code>name</code>	No	String specifying the name of the option. Field option name must be unique per field. Maximum length of 128 characters.
<code>label</code>	No	String specifying the label that the option will be displayed with on Tulip Live Chat. Maximum length of 128 characters.

Example:

```

widgetConfiguration.prechatCapture: {
  enabled: true,
  enableEmailLinking: true,
  fields: [
    {
      type: 'email',
      name: 'email',
      label: 'Email',
      placeholder: 'your@email.com',
    },
    {
      type: 'text',
      name: 'name',
      label: 'Name',
      placeholder: 'Your name',
    },
    {
      type: 'select',
      name: 'country',
      label: 'Country',
      placeholder: 'Choose your country of residence',
      options: [
        {
          name: 'us',
          label: 'United States',
        },
        {
          name: 'ca',
          label: 'Canada',
        },
      ],
    },
  ],
}

```

Browser storage

By default, Tulip Live Chat will store the identity of anonymous users in the `localStorage` of the browser.

Using the `localStorage` will persist the user identity through out browser sessions (including page reloads and browser restarts). To clear the user identity once the browser is closed, use `sessionStorage` instead. [Learn more](#)

Embedded mode

As describe above, to activate the embedded mode, you need to pass `embedded: true` when calling `TulipLiveChat.init`. By doing so, you are disabling the auto-rendering mechanism and you will need to call `TulipLiveChat.render` manually. This method accepts a DOM element which will be used as the container where the widget will be rendered.

```
TulipLiveChat.init({
  integrationId: '<integration-id>',
  embedded: true,
});

TulipLiveChat.render(document.getElementById('chat-container'));
```

The embedded widget will take full width and height of the container.