

Customization of WebMessenger

Customization of WebMessenger	1
Default mode	2
Embedded mode (OPTIONAL)	2
Strings customization	3
Display Style	3
JobPal.init	4
Button Size	4
Fixed Intro	5
Colors	5
Business profile	7
Background Image	8
Sound notification	9
Browser storage	9
Menu items	10

Default mode

In the default mode we operate an auto-rendering mechanism. Within the scope of that code you can adjust to enable a fully expanded view after timeout (implement before `</script>`). Keep in mind that would mean that every user entering the page will be considered as an active user for the chatbot.

If you want to open the web view please call `JobPal.open()`; , eg open after the page load `"setTimeout(() => JobPal.open(), 2000);"`

Embedded mode (*OPTIONAL*)

To embed the widget in your existing markup, you need to pass `embedded: true` when calling `JobPal.init`. By doing so, you are disabling the auto-rendering mechanism and you will need to call `JobPal.render` manually. This method accepts a DOM element which will be used as the container where the widget will be rendered.

```
JobPal.init({
  appId: '<app-id>',
  embedded: true
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);

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

The embedded widget will take full width and height of the container. You must give it a height, otherwise, the widget will collapse.

Instead of `JobPal.open()` and `JobPal.close()`, you will have to render `WebMessenger` into your container and control the visibility of the container itself, e.g.:

```
var chatContainer = document.getElementById("chat");

JobPal.render(chatContainer);

JobPal.startConversation();
```

Strings customization

The Web Messenger lets you customize any strings it displays by overwriting its keys. Simply add the `customText` key in your `JobPal.init` call and specify new values for the keys used in it. You can find all available keys [here](#). If some text is between `{}`, or if there is an html tag such as `<a>`, it needs to stay in your customized text. For example:

```
JobPal.init({
  appId: '<app-id>',
  customText: {
    headerText: 'How can we help?',
    inputPlaceholder: 'Type a message...',
    sendButtonText: 'Send'
  }
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

Display Style

The Web Messenger can be displayed as a button or as a tab. The default style is the `button` mode.

When the display style is a button, 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.

JobPal.init

Specify the `displayStyle` and `buttonIconUrl` in the call to `JobPal.init`.

```
JobPal.init({
  appId: '<app-id>',
  // ...
  displayStyle: 'button',
  buttonIconUrl: 'https://myimage.png'
  // ...
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

Button Size

You can customize the size of the button by setting a `buttonWidth` and `buttonHeight`. When not provided, the button will have a default size of 58 x 58 pixels. Specify the `buttonWidth` and `buttonHeight` in the call to `JobPal.init`.

```
JobPal.init({
  appId: '<app-id>',
  // ...
  buttonWidth: '90',
  buttonHeight: '90'
  // ...
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
```

```

    // Something went wrong during initialization
  }
);

```

Fixed Intro

You can set the introduction pane to fixed mode by setting `fixedIntroPane` to `true`.

When set, the pane will be pinned at the top of the conversation instead of scrolling with it. The default value is `false`.

```

JobPal.init({
  appId: '<app-id>',
  // ...
  fixedIntroPane: true
  // ...
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);

```

Colors

The supported color customizations are:

- The **Brand Color** customizes the color of the messenger header. It is also used for the color of the button or tab in idle state, as well as the color of the default app icon. If no color is specified, the brand color will default to `#65758e`.
- The **Conversation Color** customizes the color of customer messages and actions in the footer. If no color is specified, the conversation color will default to `#0099ff`.

- The **Action Color** changes the appearance of links and buttons in your messages. It is also used for the 'Send' button when it is in active state. If no color is specified, the action color will default to **#0099ff**.

Brand Color:

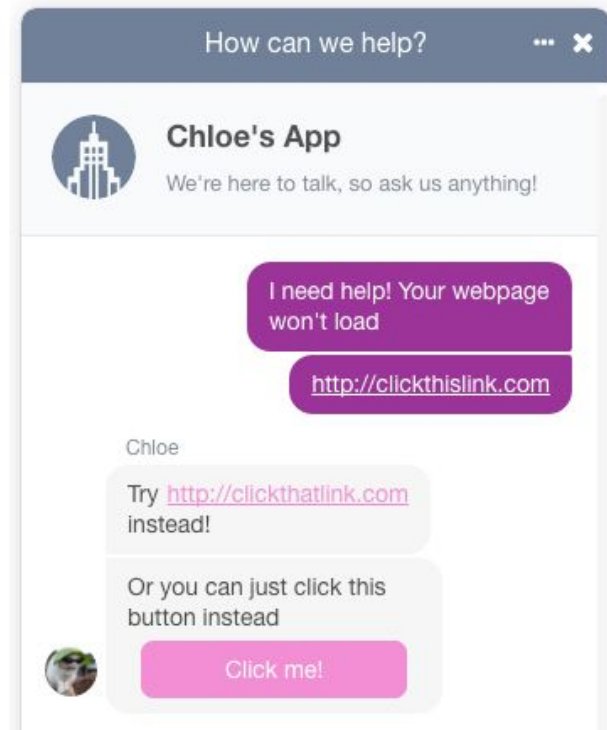
#65758e

Conversation Color:

#912d8d

Action Color:

#f183ce



```
JobPal.init({
  appId: '<app-id>',
  // ...
  customColors: {
    brandColor: "7af442",
    conversationColor: "b7b7b5"
    actionColor: "191919"
  }
  // ...
})
.then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

Business profile

You can customize your business branding with the `businessName` and `businessIconUrl` settings. For the `businessIconUrl` setting, the image must be at least 200 x 200 pixels and must be in either JPG, PNG, or GIF format.



If the branding settings are not set, they fall back to the app's settings. The app name is used as the `businessName`, and the app icon is used as the `businessIconUrl`.

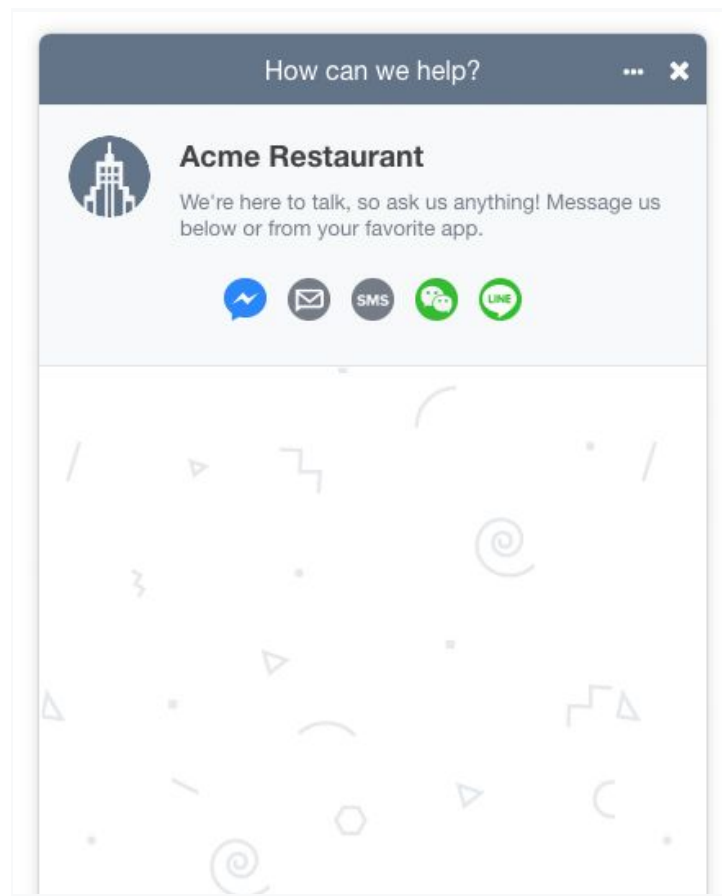
```
JobPal.init({
  appId: '<app-id>',
  // ...
  businessName: 'Acme Corporation',
  businessIconUrl:
    'https://is2-ssl.mzstatic.com/image/thumb/Purple127/v4/67/a4/5b/67a45b68-c821-842f-22f5-9fa83b605ce3/mzl.evidoiah.jpg/1200x630bb.jpg'
  // ...
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

Background Image

You can customize the background image in your SDK with the `backgroundImageUrl` setting. The image will be displayed at its full size, and tiled if it is not large enough to fill the conversation.

For reference, the following CSS will be used to display the background:

```
{  
  background-image: url('https://a-nice-texture.png');  
}
```




```

JobPal.init({
  appId: '<app-id>',
  // ...
  backgroundImageUrl: 'https://a-nice-texture.png'
  // ...
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);

```

Sound notification

By default, a sound notification will be played when a new message comes in and the window is not in focus. To disable this feature, you need add the `soundNotificationEnabled` option to the `JobPal.init` call, like this:

```

JobPal.init({
  appId: '<app-id>',
  soundNotificationEnabled: false // Add this line to your 'JobPal.init'
  call
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);

```

Browser storage

By default, the Web Messenger will store the identity of anonymous users in the `localStorage` of the browser. Using the `localStorage` will persist the user identity

throughout browser sessions (including page reloads and browser restarts). To clear the user identity once the browser is closed, use `sessionStorage` instead.

```
JobPal.init({
  appId: '<app-id>',
  browserStorage: 'sessionStorage' // Add this line to your 'JobPal.init'
call
}).then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

Menu items

The Web Messenger features a menu that allows the user to send various message types. The types displayed in this menu can be customized, or the menu can be hidden altogether. If you want to control this menu, add the `menuItems` option to the

`JobPal.init` call:

```
JobPal.init({
  appId: <app-id>,
  menuItems: {
    imageUpload: true,
    fileUpload: true,
    shareLocation: true
  },
  // ...
})
.then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```

To hide the menu completely, override the `menuItems` option as follows:

```
JobPal.init({
  appId: <app-id>,
  menuItems: {},
  // ...
})
.then(
  function() {
    // Your code after init is complete
  },
  function(err) {
    // Something went wrong during initialization
  }
);
```