

Visitor UI JavaScript API

boot

The boot method is responsible for bootstrapping Velaro chat with necessary configurations and mounting the chat window to the DOM. This call will be included in the code copied from the chat designer, but is documented here for more information on configuration options.

```

var config = {
  // siteId will be included in your deployment script.
  // This value should not be changed.
  siteId: [your siteId],

  // When groupId is zero, it will send all chats through
  // the default group. To get a specific groupId, change
  // groups in the chat designer and view the deployment
  // script.
  groupId: 0,

  // This option disables inline chat. You can set this to
  // true if you only want to use popout chat or use other
  // api methods without displaying inline chat.
  disableInline: false,

  // This option disables the chat launcher. If you want to
  // customize the behavior or appearance of the chat launcher
  // beyond the capabilities of the chat designer, set this
  // to true and use other API methods to display and hide the
  // chat window and launcher.
  disableLauncher: false,

  // If a chat session should persist across subdomains, this
  // value should be set. For example, if a chat is started on
  // foo.example.com, and the user navigates to bar.example.com,
  // cookieDomain should be set to '.example.com'
  cookieDomain: '',

  // Custom vars allow you to persist custom data with a chat.
  // This unlocks the potential to design custom routing rules
  // and invitation rules and is also useful for certain CRM
  // integrations.
  customVars: {
    customerId: "s9d8fh9s8dgfsdf",
    cartValue: 53.99
  }
}

var callback = function() {
  // this callback is optional
  console.log("fires after the chat window has been rendered to the
DOM.")
}

Velaro('boot', config, callback)

```

destroy

Destroys the inline chat window.

```
Velaro('destroy')
```

expand

Expands the inline chat window.

```
Velaro('expand')
```

collapse

Collapses the inline chat window.

```
Velaro('collapse')
```

isExpanded

Returns true if the window is expanded and false if it is not.

```
var isExpanded = Velaro('isExpanded')

if(isExpanded) {
  console.log('chat is expanded')
} else {
  console.log('chat is collapsed')
}
```

popout

Pops the chat window out into a new window.

```
Velaro('popout')
```

onExpand

Fires a callback when the chat window has been expanded.

```
Velaro('onExpand', function(){ console.log('expanded') })
```

onCollapse

Fires a callback when the chat window has been collapsed.

```
Velaro('onCollapse', function(){ console.log('collapsed') })
```

setLauncherVisibility

Hides or shows the chat launcher.

```
// hide the launcher
Velaro('setLauncherVisibility', 'hidden')

// show the launcher
Velaro('setLauncherVisibility', 'visible')
```

isChatAvailable

Check agent, group, and scheduling status to determine if an agent is available to chat.

```
Velaro('isChatAvailable', function(isAvailable) {
  if(isAvailable) {
    console.log('agent is available to chat')
  } else {
    console.log('agent is not available')
  }
})
```

This method can also be used to show or hide an image based on agent availability.

```
Velaro('isChatAvailable', function(isAvailable) {
  if(isAvailable) {
    $("#myImage").show();
  } else {
    $("#myImage").hide();
  }
})
```

onQueueEntered

Fires a callback when the visitor enters a queue.

```
Velaro('onQueueEntered', function(){
    console.log('visitor entered queue')
})
```

onAgentAssigned

Fires a callback when the visitor is assigned to an agent.

```
Velaro('onAgentAssigned', function(){
    console.log('visitor assigned to agent')
})
```

onReceiveMessage

Fires a callback whenever the visitor receives a message during a chat containing the message. (For both visitor or agent sent messages.)

```
Velaro('onReceiveMessage', function(message){
    if (message.lineCreator === 4){
        console.log('visitor sent a message')
    } else if (message.lineCreator === 1){
        console.log('agent sent a message')
    }
})
```

onAgentTyping

Fires a callback containing an array of all agents currently typing whenever an agent begins or stops typing.

```
Velaro("onAgentTyping", function(agentsTyping) {
    console.log("currently typing agents: ", agentsTyping);
});
```

onChatClosed

Fires a callback when the visitor closes the chat window.

```
Velaro("onChatClosed", function() {  
    console.log("chat window closed");  
});
```

onAgentClosed

Fires a callback when the agent closes the chat.

```
Velaro("onAgentClosed", function() {  
    console.log("agent closed chat");  
});
```

onShowPrechatSurvey

Fires a callback when the visitor is shown a pre-chat survey.

```
Velaro("onShowPrechatSurvey", function() {  
    console.log("pre-chat survey shown");  
});
```

onShowPostchatSurvey

Fires a callback when the visitor is shown a post-chat survey.

```
Velaro("onShowPostchatSurvey", function() {  
    console.log("post-chat survey shown");  
});
```

onShowUnavailableSurvey

Fires a callback when the visitor is shown an unavailable survey.

```
Velaro("onShowUnavailableSurvey", function() {  
    console.log("unavailable survey shown");  
});
```

onShowUnavailableHtml

Fires a callback when the visitor is shown unavailable html

```
Velaro("onShowUnavailableHtml", function() {
    console.log("unavailable html shown");
});
```

onShowUnavailableUrl

Fires a callback when the visitor is shown an unavailable url

```
Velaro("onShowUnavailableUrl", function() {
    console.log("unavailable url shown");
});
```

mountDynamicButton

This method will mount a chat button within the provided DOM element. This button will automatically update as agent availability changes. See <https://app.velaro.com/#admin/design/chat-designer?groupId=0§ion=popout-button> to configure the appearance of the button.

```
Velaro('mountDynamicButton', {
    el: document.getElementById('your-button-container')
})
```

addConversion

Creates a conversion record

```
Velaro('addConversion', {
    // the id of your conversion can be found at
    // https://app.velaro.com/#/admin/visitors/conversions
    conversionId: [your conversion id],

    dollarAmount: 49.99,

    // optional parameters to persist with the conversion
    customData: {
        customerId: '9283gf9823gf98g'
    }
})
```



Velaro_Visitor U...vaScript+API.pdf