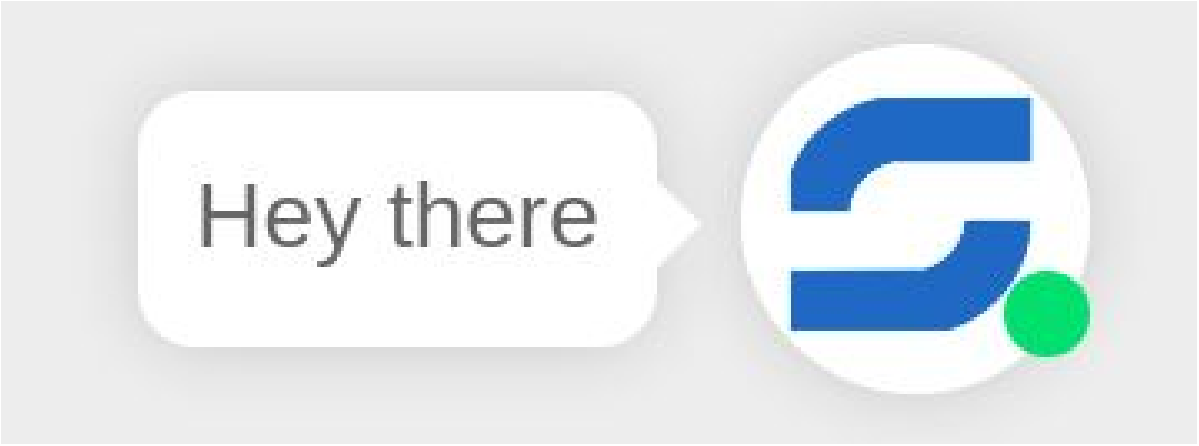


Adding feature to Preview Chatbot

What is a Chatbot ?

A Chatbot is a software that can have a conversation with a human. For example a user could ask the bot a question or give it an instruction and the bot could respond or perform an action as appropriate.

Chatbot in SUSI

You can access all features of SUSI on https://chat.susi.ai. But this restricts the usage of SUSI to this site only. To solve this issue, we can use a Chatbot which can be embedded on any website and then can be used by the people who visit that website, hence generalizing the usage of SUSI.

How to embed SUSI Chatbot on any website ?

To embed SUSI Chatbot on any website, go to https://skills.susi.ai and login and then check out the Botbuilder section. There you can find a deploy button which will give you a code which looks something like this :

```
<script type="text/javascript" id="susi-bot-script" data-token="sMpeVATZ7r3gJGvo5DlSsVOchIYZRE"
src="https://skills.susi.ai/susi-chatbot.js"></script>
```

As can be seen from the code above, it is simply a script tag with an id and a custom attribute *data-token* (which is the unique access token of the User for that session). The source file for this script is *susi-chatbot.js*. This is the file that is responsible for displaying the Chatbot on the screen.

But, let's say that you configured your Chatbot in the entire Botbuilder process. So, we need to have a Preview button which could show us a preview of the Chatbot with the chosen configurations, otherwise we would have to test it out by embedding it on our website, which would be tiresome.

How does the Preview Button work ?

The code for Botbuilder page is in the Botbuilder.js file.

The code for the Preview button is as follows :

```
{!this.state.chatbotOpen?(<RaisedButton
      label='Preview'
      style={{ width: '148px' }}
      onClick={this.injectJS}
  />):(<RaisedButton
      label='Close Preview'
      style={{ width: '148px' }}
      onClick={this.handleChatbotClose}
/>)}
```

It can be seen that on clicking the 'Preview' button, *injectJS()* function is called, and on clicking the 'Close Preview' button, *handleChatbotClose()* function is called.

```
injectJS = () => {
    const myscript = document.createElement('script');
    myscript.type = 'text/javascript';
    myscript.id = 'susi-bot-script';
    myscript.setAttribute('data-token',cookies.get('loggedIn'));
    myscript.src = '/susi-chatbot.js';
    myscript.async = true;
    document.body.appendChild(myscript);
    // some code
};
```

```
handleChatbotClose = () =>{
 // some code
 $('#susi-launcher-container').remove();
 $('#susi-frame-container').remove();
 $('#susi-launcher-close').remove();
 $('#susi-bot-script').remove();
}
```

The above 2 functions are the key functions in understanding how the Preview button is previewing the Chatbot on the screen.

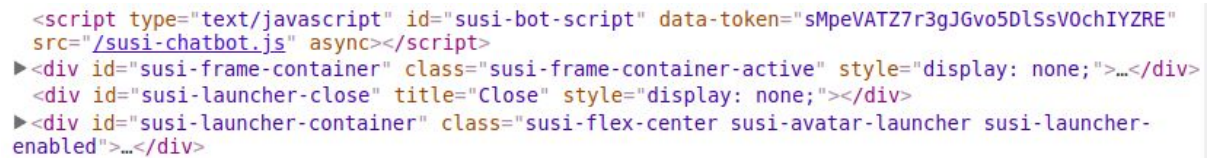
Let us see the functioning of *injectJS()* function first. Firstly, the createElement() method creates an element node of type 'script'. We store this element node in a variable *myscript*. We then add attribute values to the script tag by using *myscript.attribute_name = attribute_value.* We can add 'type', 'id', 'src' and 'async' attributes in this way because they are predefined attributes.

However, we need to also add a custom attribute to store the access_token of the User, because we would need to pass down this information to detect the owner of the Chatbot. To add a custom attribute, we use the following code :

```
myscript.setAttribute('data-token',cookies.get('loggedIn'));
```

This add a custom attribute ‘data-token’ and sets its value to the access token of the session, which would later be used to identify the User.

This script is then appended to the body of the document. As soon as the script is appended to the page, 3 new tags are created and added to the page’s HTML. These 3 tags contain the code for the UI and functionality of the Chatbot itself.



Next, let us look at the functioning of *handleChatbotClose()* function. This function’s basic objective is to close the Chatbot and remove the 3 components that were rendered as a result of the *injectJS()* function, and also remove the script itself. As can be seen from the code of the function, we are removing the 4 components which can be seen in the above image of the developer console.

So, this is how the Preview functionality has been added to the Chatbot. Visit https://skills.susi.ai and login to check out the Botbuilder section to try it out.

Resources

- Injecting JavaScript in React
- Removing HTML elements
- Adding custom attributes to an element