The below are the steps to host a chatbot using web channel.

**Folder and Files Description:**

This folder will be provided by chatbot team.

Custom\_UI – It contains customizations to the existing webchat provided by Azure.

• Custom.css – Contains customized css styles.

• Custom.js – contains customized javascript and jquery.

Botchat-fullwindow.css – Contains card css styles for other screen resolution.

Botchat.css – contains css styles for the whole chatbot.

Botchat-es5.js – contains entire JS code for html elements and functions.

Images folder – contains images which will be used in user interface.

**Assumptions:**

1. Server space will be provisioned to place all the Chatbot files.
2. NodeJS will be installed on the server space.
3. Public IP will be enabled for the port where we placed our files and the users request to botframework will be made.
4. Chatbot will be hosted on customers webpage(intranet).

**Step-1:**

Extract all the SDK files into your local machine.

**Step-2:**

Open index.html file and update the highlighted path with the actual path in server where we will place our files.

<!-- add the below codes in client server -->

<link href=" **Server\_PATH** /botchat.css" rel="stylesheet" />

<link href=" **Server\_PATH** /botchat-fullwindow.css" rel="stylesheet" />

<link href=" **Server\_PATH** /jquery-min-ui.css" rel="stylesheet" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<script src=" **Server\_PATH** /botchat-es5.js" type="text/javascript"></script>

<script src=" **Server\_PATH** /custom.js" type="text/javascript"></script>

<link href=" **Server\_PATH** /custom.css" rel="stylesheet"></link>

<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>

<script src="https://code.jquery.com/jquery.min.js"></script>

<script src="https://code.jquery.com/ui/1.9.2/jquery-ui.js"></script>

**Step-3:**

Include the below code to the webpage code(home page) where we want to show the chatbot.

<!-- Chatbot triggering icon -->

<div class="minimized"><span class="messages"></span></div>

<div class="minimizedClose"><span class="closeChat"></span></div>

<div id="BotChatElement"></div>

<!-- Chatbot configuration -->

<script>

var model = {

"userId": "You",

"userName": "You",

"botId": "LEM\_Chatbot\_V1\_dev",

"botIconUrl": "https://docs.botframework.com/static/devportal/client/images/bot-framework-default.png",

"botName": "LEM\_Chatbot\_V1\_dev",

"secret": "siWkdZtNdGc.QF9LViLg73zNexdfHofdQFMqXtF-U3BenUcGKfby4vc",

"iconUrl": "https://docs.botframework.com/static/devportal/client/images/bot-framework-default.png",

"directLineUrl": "https://webchat.botframework.com/v3/directline",

"webSocketEnabled": "true",

"speechTokenEndpoint": "https://api.botframework.com/v3/speechtoken",

"useLatestWebChat": false

};

botConnection = new BotChat.DirectLine({

secret: model.secret,

token: model.token,

domain: model.directLineUrl,

webSocket: true

});

BotChat.App({

botConnection: botConnection,

sendTyping: true,

user: { id: model.userId, name: model.userName, hostName: window.location.href },

bot: { id: model.botId, name: model.botName },

resize: 'window',

locale: 'en'

}, document.getElementById("BotChatElement"));

</script>

**Step-4:**

Now place all the SDK files along with modified files on the server and try to load the home screen (WEB Channel page).

* We (Chat bot team) will provide all the websdk files with customized user interface for chatbot.