**Birlasoft Chatbot**

Birlasoft chatbot is a chatbot application which is integrated with SAP Fiori to help end users to interact with SAP through a chatbot.

1. Technologies and platforms involved.
   1. SAP Conversational AI / Recast AI platform
   2. Node JS
   3. SAP Cloud Foundry
   4. SAP UI5 and Fiori
2. Architecture.

SAP Cloud Foundry or Local System

SAP web dispatcher

SAP Conversational AI Platform

Browser/ SAP Fiori Launchpad

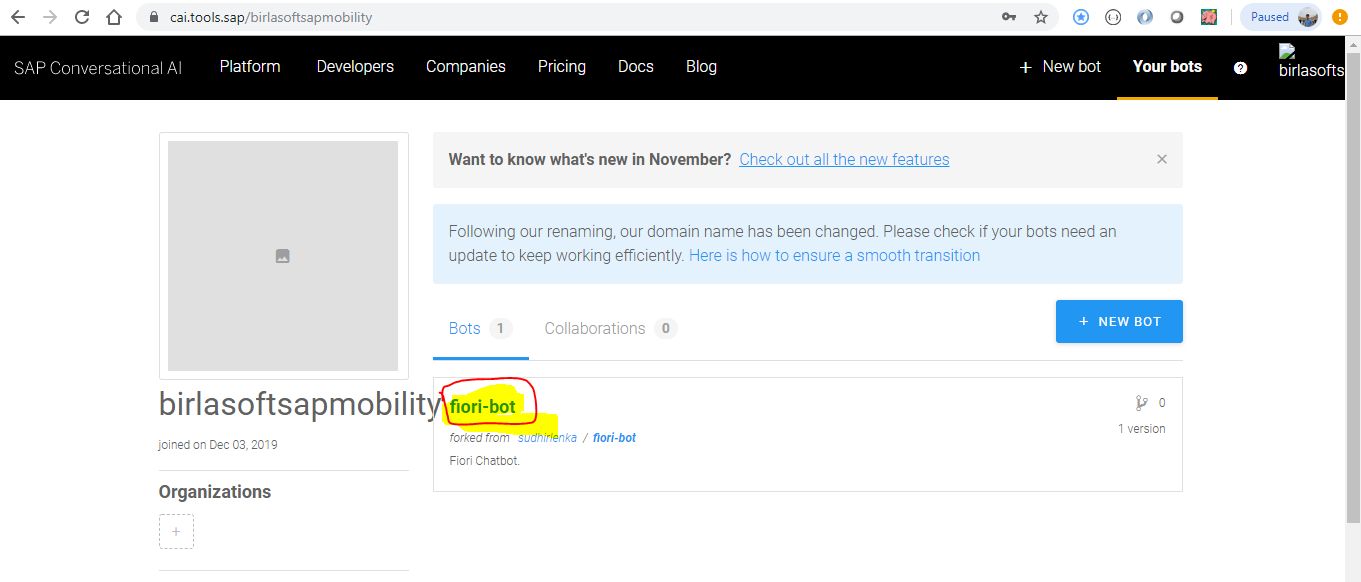
http

**SAP**

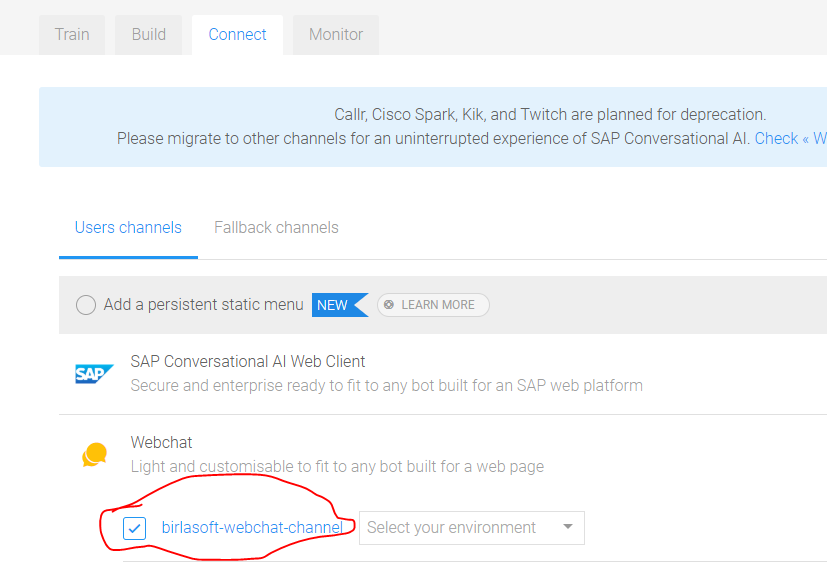
https

OData

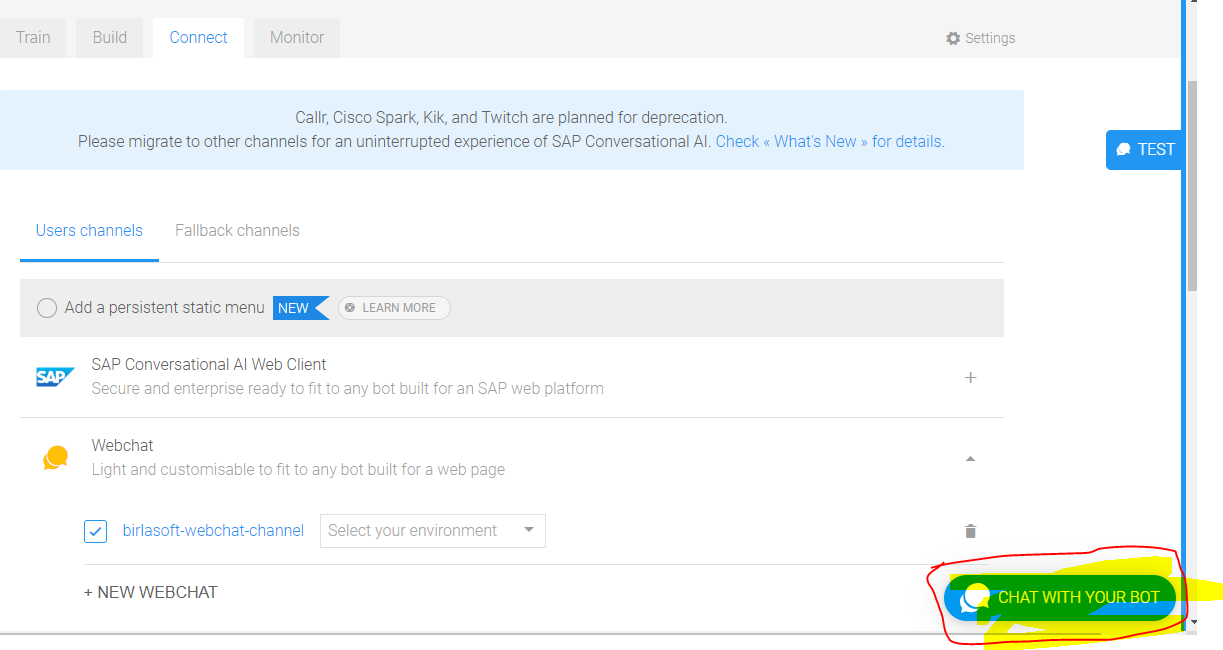
1. SAP Conversational AI Platform/ Recast AI Platform.
   1. Login to SAP conversational AI platform(<https://cai.tools.sap/>).
   2. Verify if the bot is already there with the name **fiori-bot**.

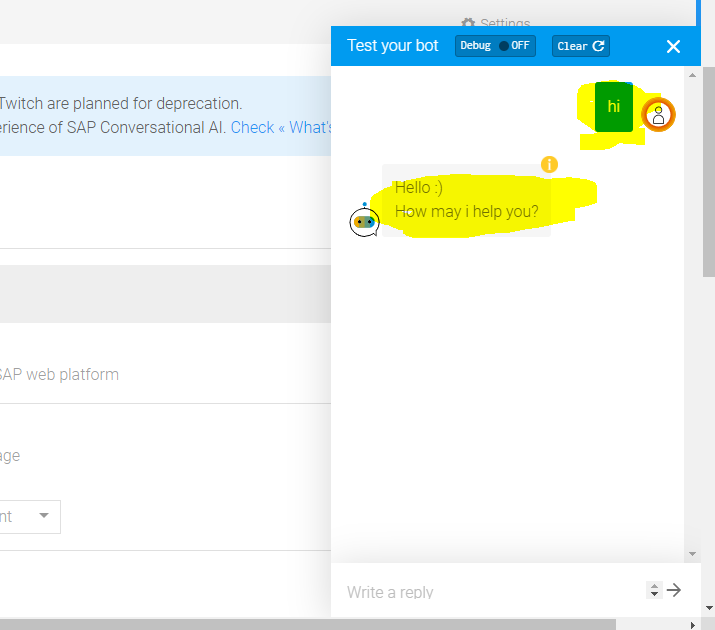


* 1. If you want to copy the bot from one account to another account, you can **fork** it.
  2. Go into the connect settings in the bot and verify if the webchat channel is there or not.

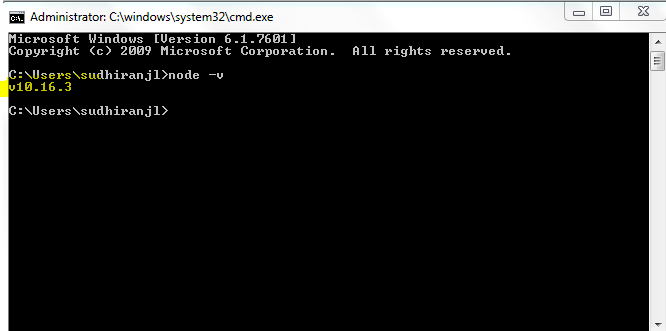


* 1. You need to create it if the bot is newly created or forked somewhere else’s account.
  2. If everything is fine, then you should be able to test your bot in the SAP Conversational AI platform itself.
  3. To test the bot, click on the **Chat with your bot** button displayed at bottom right corner.

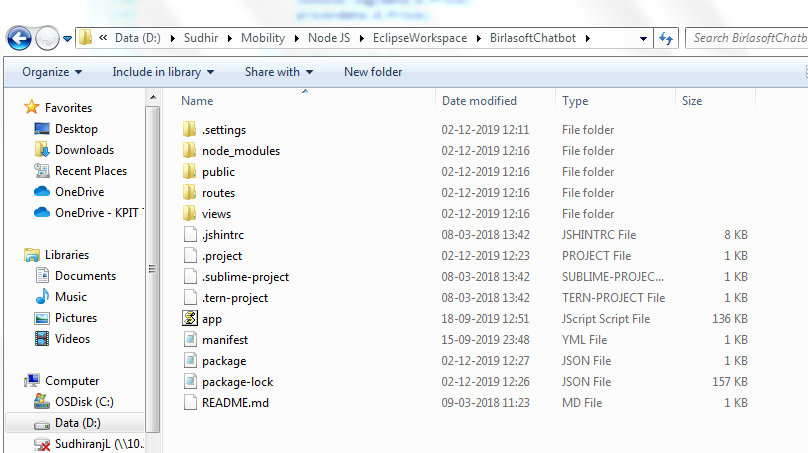


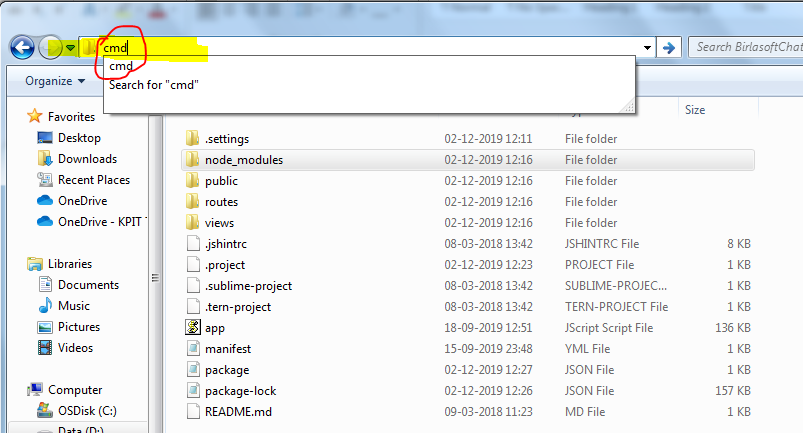


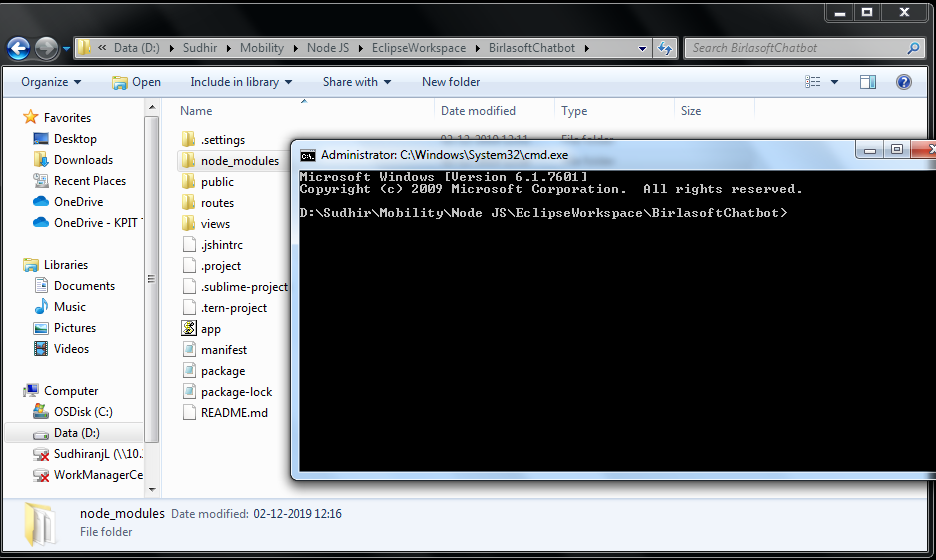
1. Technical Terms in SAP Conversational AI Platform.
   1. Train: Training to the bot. Your bot should be able to handle all the possible questions.
      1. Intents: Grouping of similar question. Which will serve a particular purpose.
      2. Entities:
   2. Build : It decides how the questions will be answered.
      1. Skill: it groups the intents.
         1. Triggers: It triggers the skill if the certain condition matches in the question.
         2. Requirement:
         3. Actions: The actions are triggered to prepare and send the response to the bot. It’s triggered if all the requirements match. Actions makes a POST call to Node JS if the response need to be fetched from the back-end.
   3. Connect: It allows our bot to connect with different applications. In our case we use Webchat to connect our bot with the Fiori. Here we can do different settings for our bot.
   4. Monitor: This helps us to monitor the bot at the runtime.
   5. Settings: This hold different settings for our bot.
      1. Options: Bot connector in the options holds the URL of Bot connector.
2. Node JS
   1. Node JS will act as a middle ware to fetch the data from SAP.
   2. For Birlasoft Chatbot the Node JS code can be found in the Git Hub.
      1. Git hub Repository : <https://github.com/birlasoftsapmobility/Chatbot.git>
   3. Node JS App will be deployed somewhere in the public network and will be triggered from the SAP Conversational AI platform.
      1. In Local system.
      2. In Cloud Foundry.
3. Steps to deploy and run Chatbot Node JS application in local system.
   1. Download and Install Node JS in your system.
   2. Once it’s installed, you should be able to see the version of the Node JS on the command prompt if it’s successfully installed.



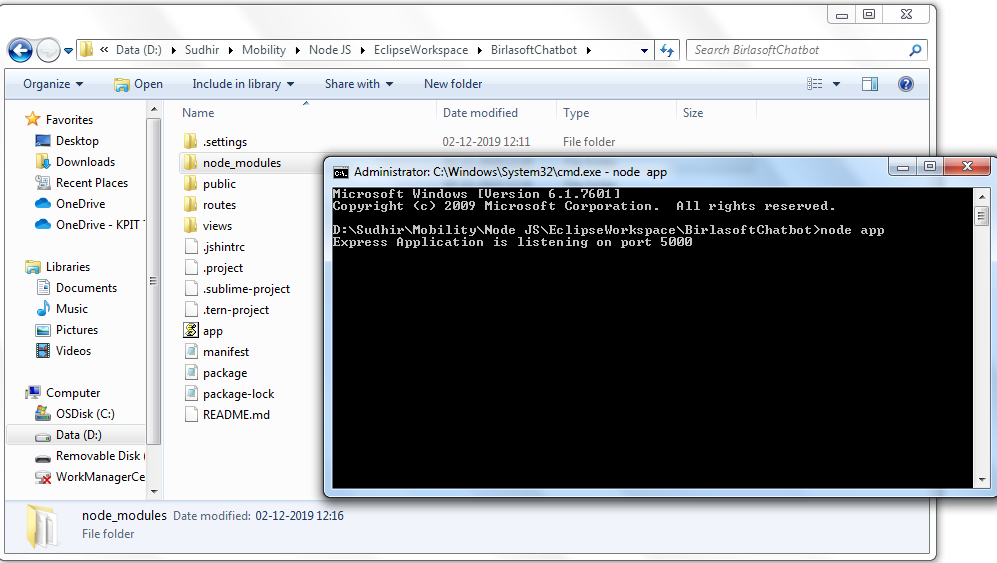
* 1. Go to the folder where Node JS application is placed and open the command prompt from the root directory of the App.



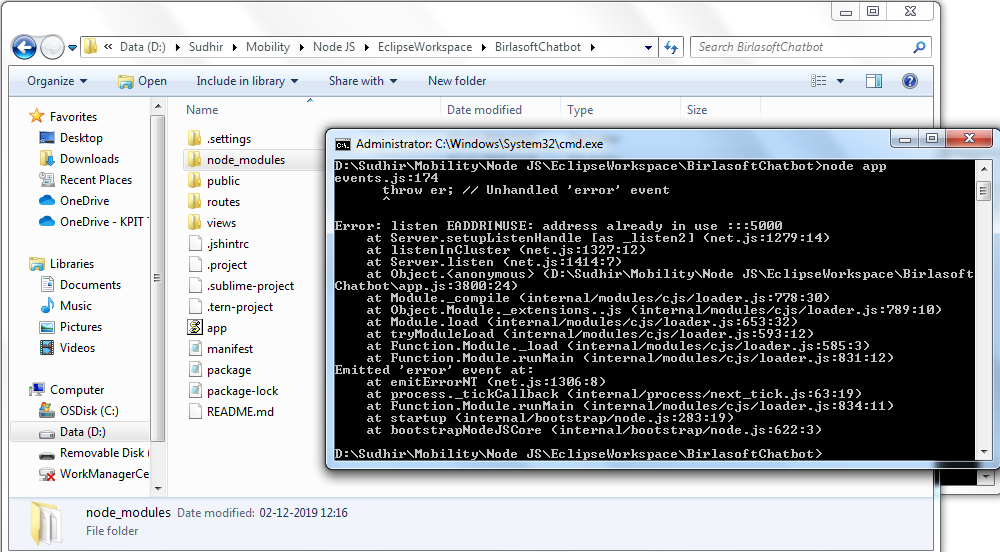




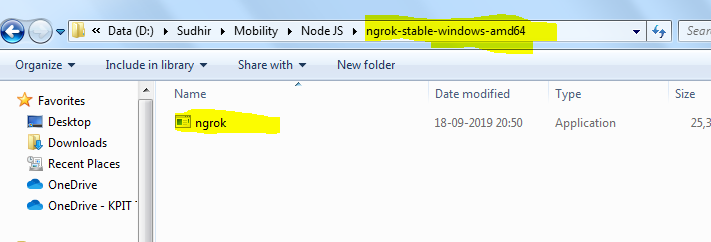
* 1. Write the command “**node app**” on the command prompt to run the Chatbot Node JS application.
  2. If everything goes fine, then App should be running fine and you should be able to see the successfully message on the command prompt.



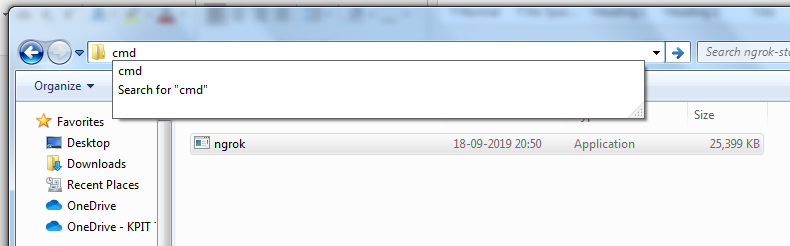
* 1. Please make sure the port 5000 is not being used by any other application. If it’s being used by some other application, then command prompt will display the following error message.

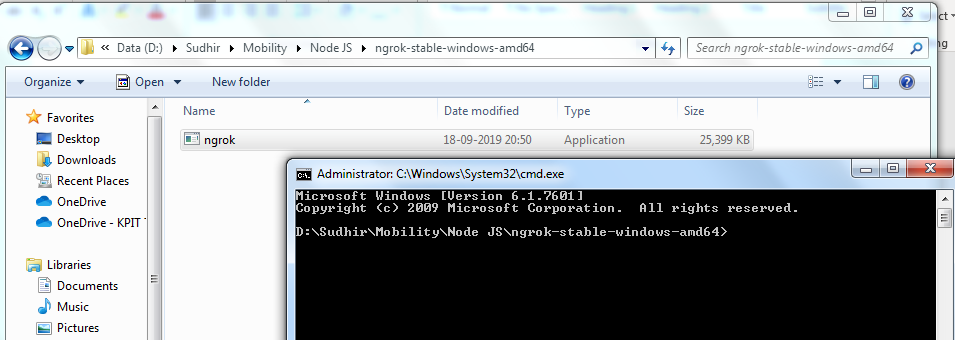


* 1. Now the Node JS app is up and running fine but it’s running on local system, so it will not be accessible from the public network. To integrate it with our chatbot it should be accessible from the public network, so that it can be triggered from the SAP Conversational AI platform.
  2. We need **NGROK** application our local system, which can expose our Node JS app to the public network.
  3. Download and extract the zip file onto your system.

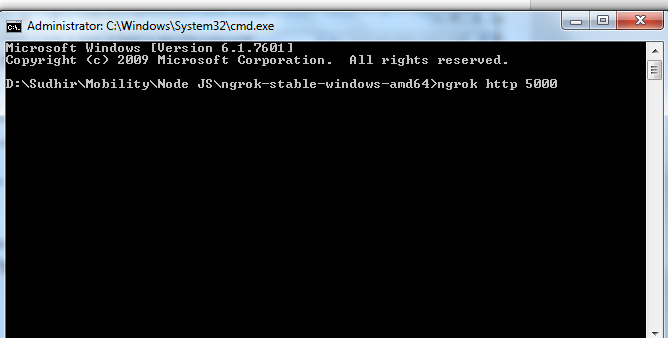


* 1. Go to the folder where ngork.exe is placed after extraction and launch the command prompt.

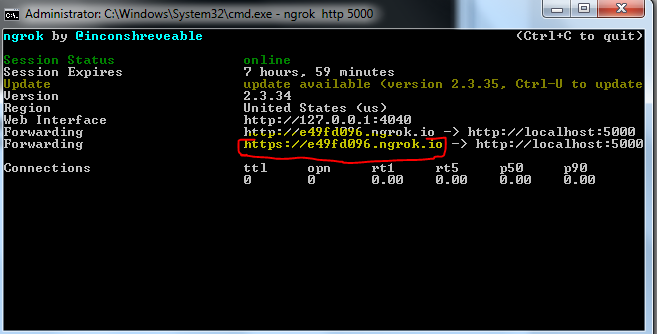




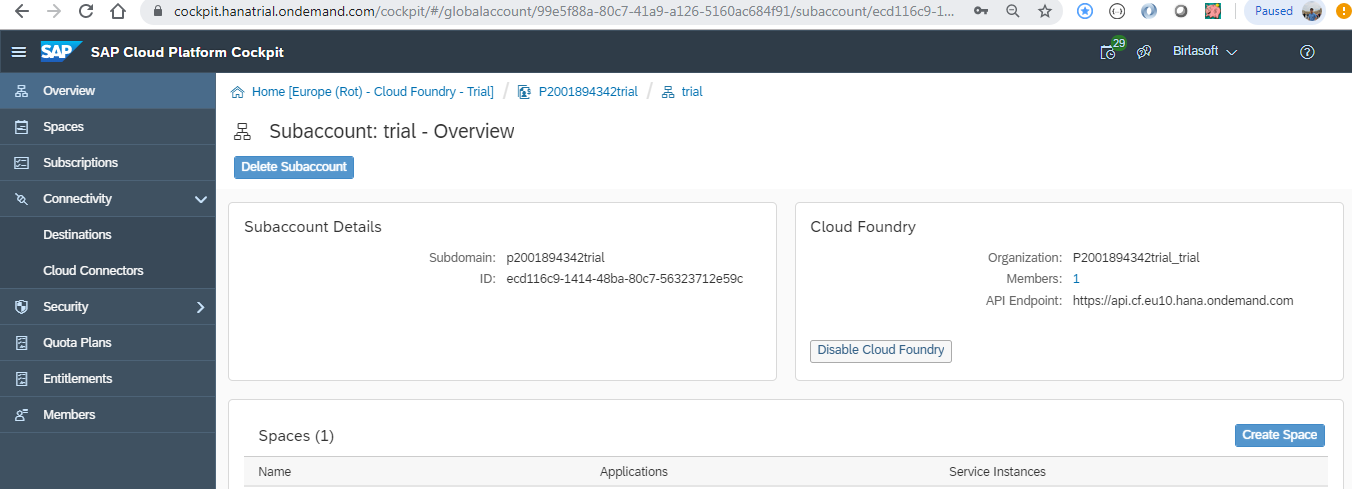
* 1. Now launch the NGROK from the newly opened command prompt by using the command “ngrok http 5000”.



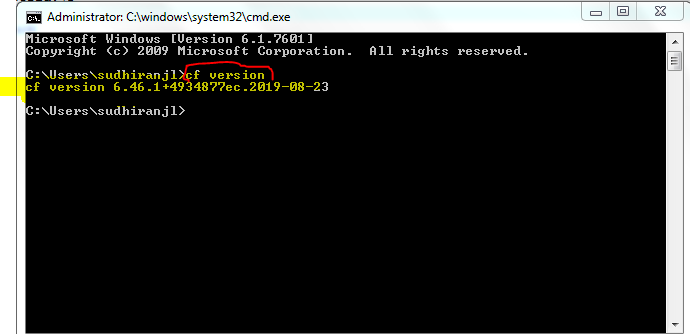
* 1. If everything goes fine, then Node JS App should be exposed publicly with a new URL and this URL can be used in the SAP Conversational AI platform.



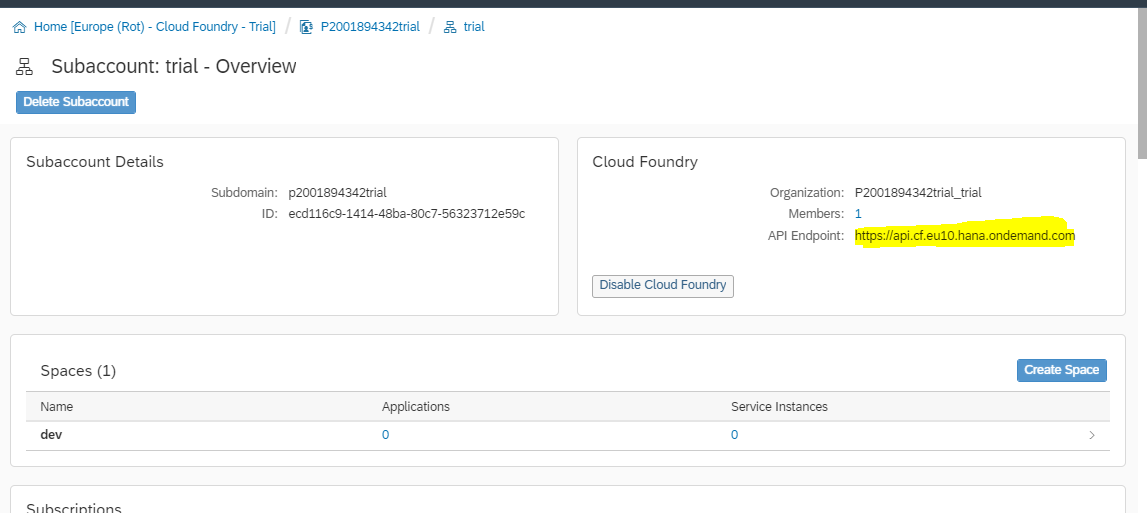
1. Steps to deploy and run Node JS application in the SAP Cloud Foundry.
   1. Login to SAP Cloud Foundry and setup the environment if it’s not setup already.
      1. <https://cockpit.hanatrial.ondemand.com/cockpit/#/home/trial>



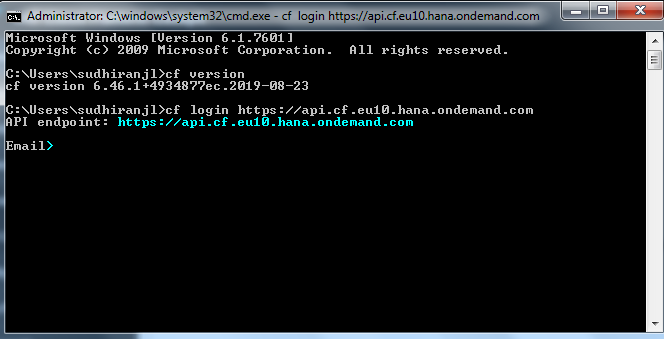
* 1. Download and install SAP Cloud Foundry Command line interface(CLI).
     1. <https://developers.sap.com/tutorials/cp-cf-download-cli.html>
  2. If the CLI is installed successfully then you should be able to see the version of the CLI in the command prompt. To check the version of CLI please use the command “**cf version**”.



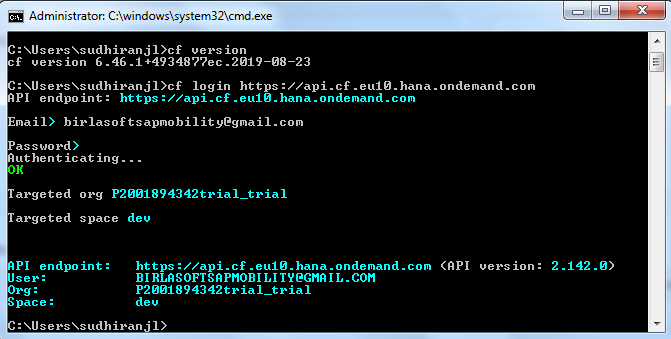
* 1. Connect your local system with Cloud Foundry account by using the following commands.
     1. Copy the CF end point URL from your Cloud Foundry account.



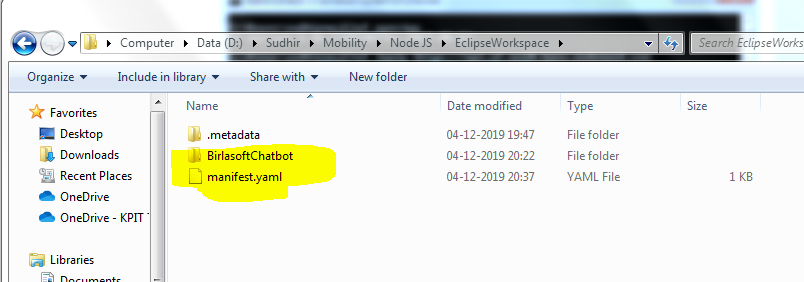
* + 1. Use the command “Cf login <CF end point URL>” on the command prompt to connect the CF account. And enter all the details as prompted.

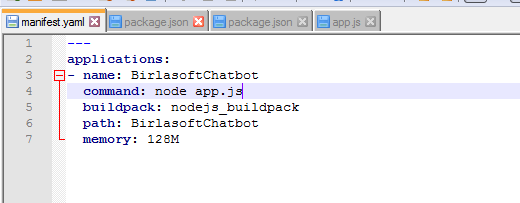


* + 1. If everything is fine then you should see the Org and Space of your account.

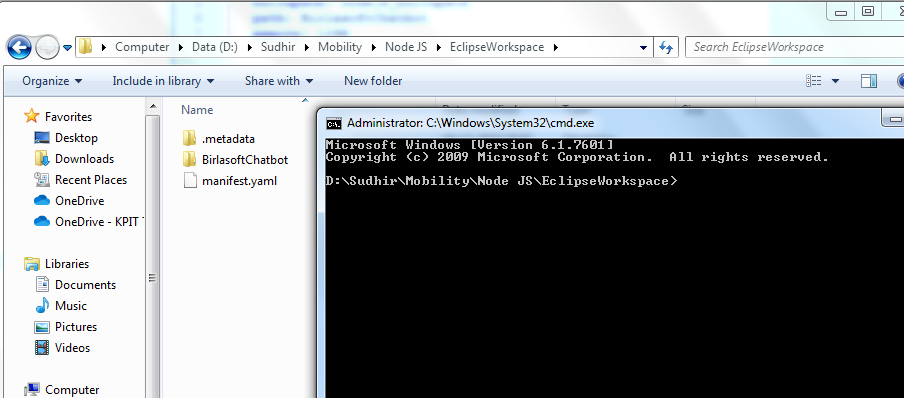


* 1. Now your local system is connected with your SAP Cloud Foundry account and we can deploy the Node JS application on the Cloud Foundry.
  2. To deploy any code to Cloud Foundry(CF) we need to create a deployment descriptor (**manifest.yaml**).
  3. Go to the folder where Node JS project is placed and create a manifest.yaml file, which can be downloaded from the Github.
     1. <https://github.com/birlasoftsapmobility/Chatbot/blob/master/manifest.yaml>

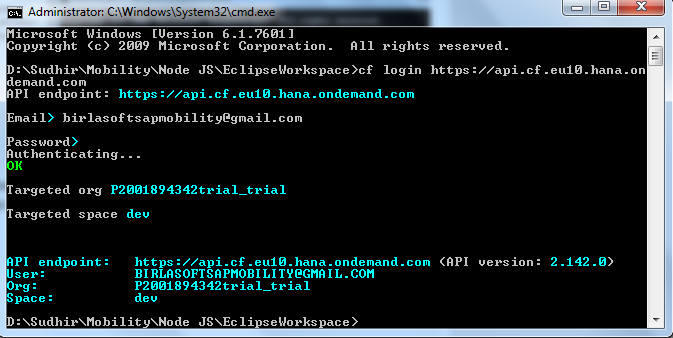




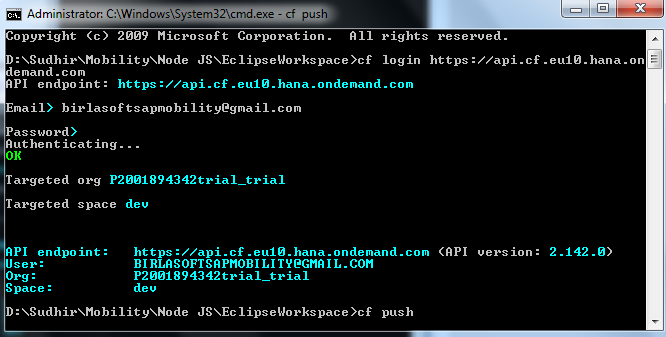
* 1. Open the command prompt from the current path where the Node JS project is placed.



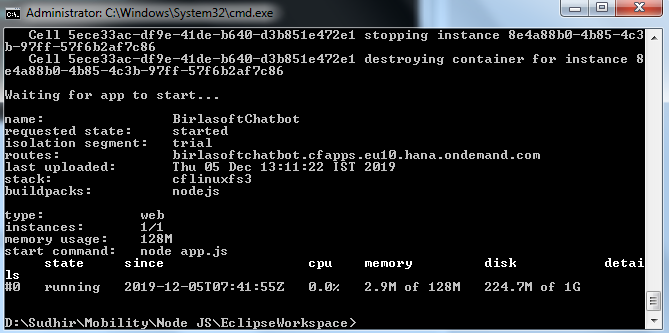
* 1. Perform the steps mentioned earlier to connect your local system to Cloud Foundry.
     1. cf login <https://api.cf.eu10.hana.ondemand.com>

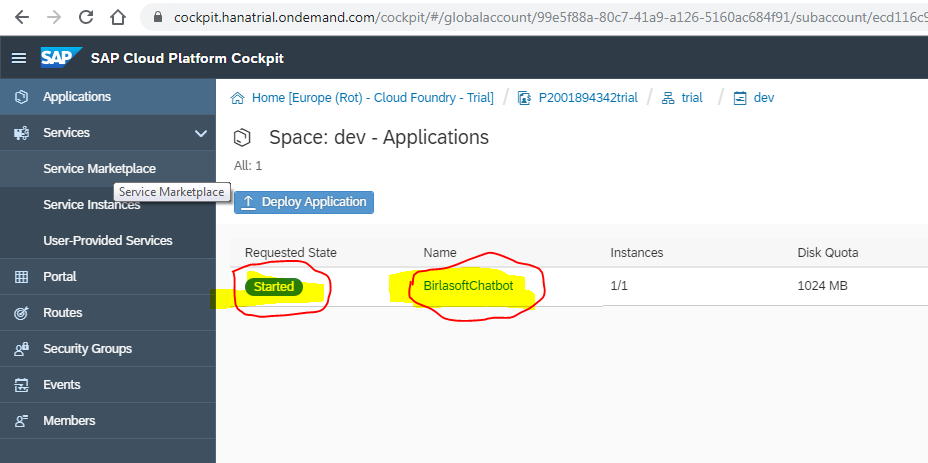


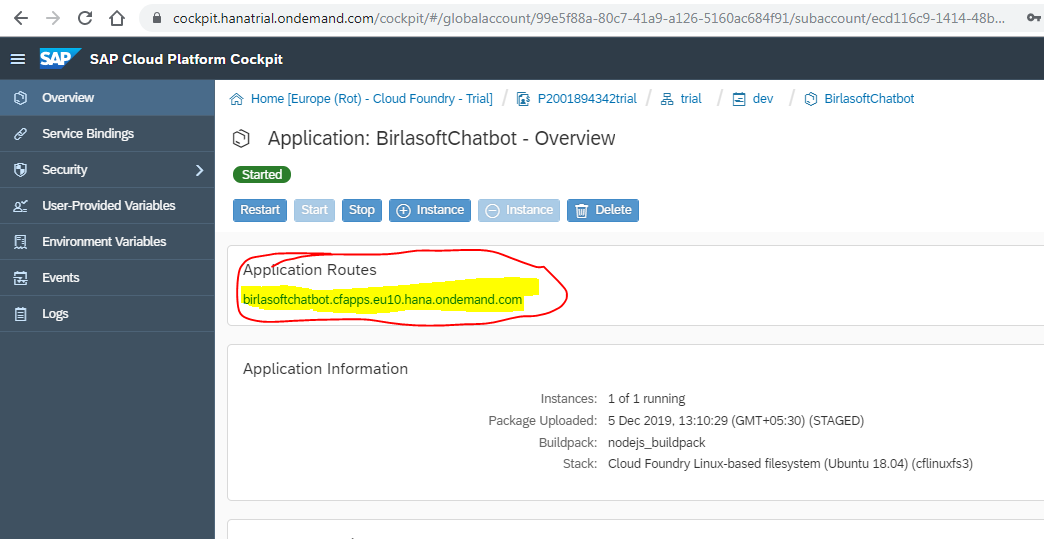
* + 1. To push the Node JS code to Cloud Foundry use the command “cf push”.



* + 1. If everything is fine, then it will be deployed successfully, and the application should be up and running fine.

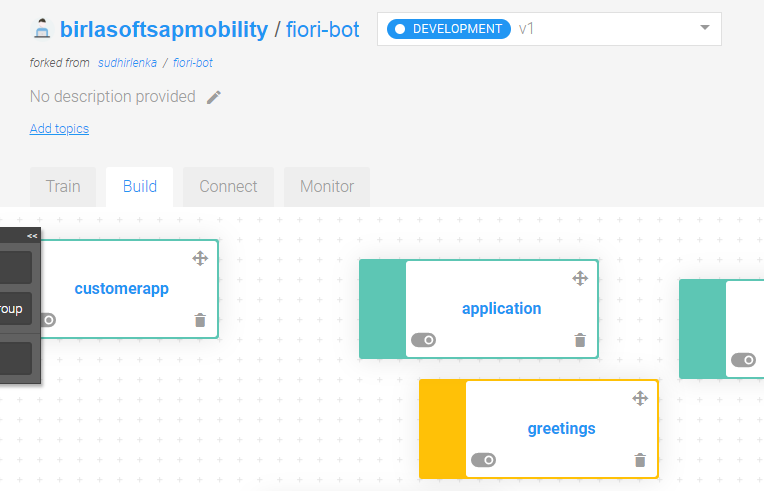




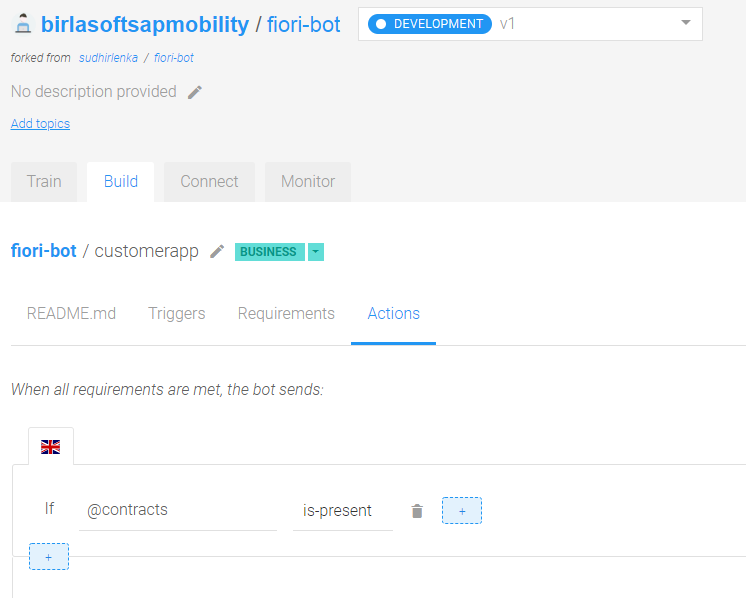


* + 1. Now the URL of the Node JS application can be used in the conversational AI platform.
       1. <https://birlasoftchatbot.cfapps.eu10.hana.ondemand.com/>

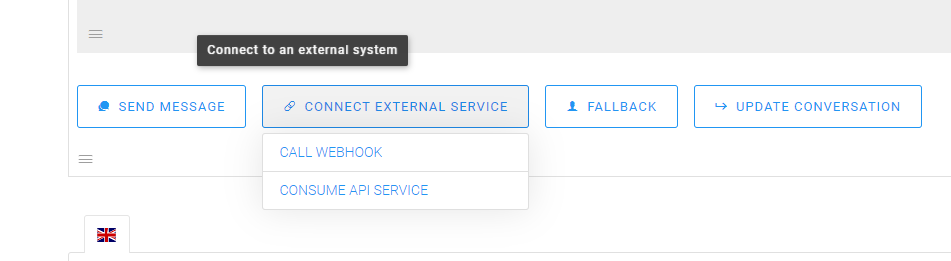
1. Configure Node JS application URL in the SAP Conversational AI platform.
   1. Go to the bot and then go to build to select any skill.



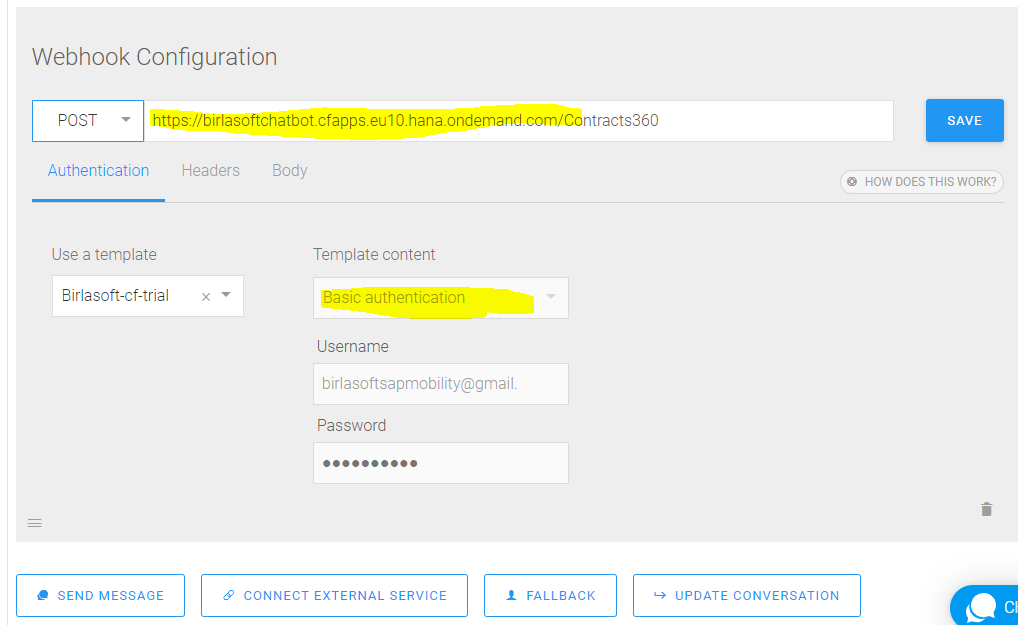
* 1. Select any skill and go to Actions tab.



* 1. Select **Connect External Service** and then select **Call Webhook**.

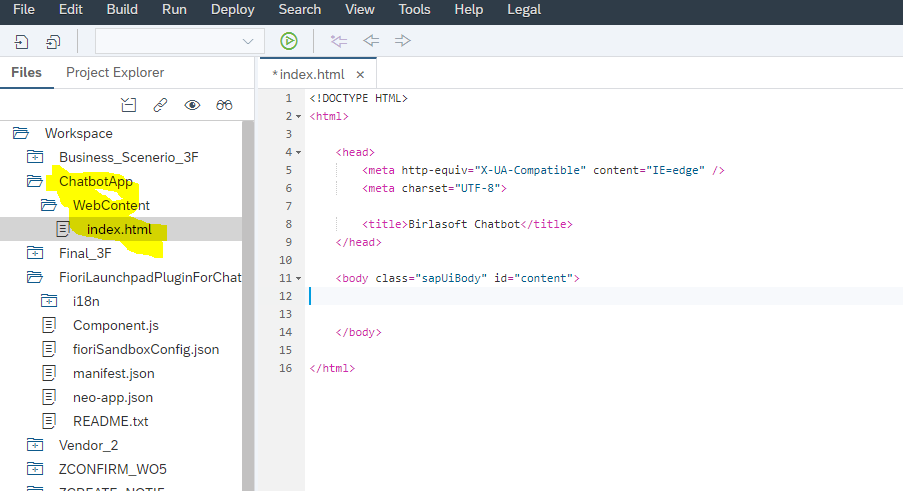


* 1. Now maintain the URL of the Node JS app along with the route and the credentials.

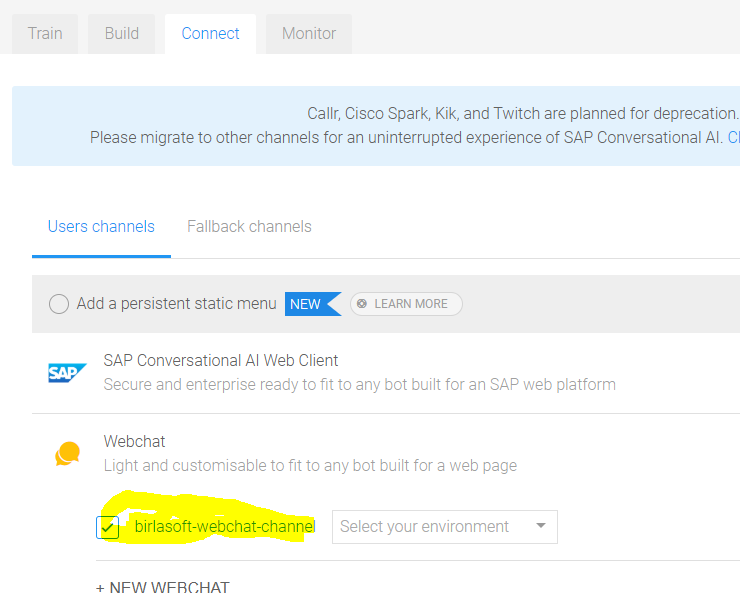


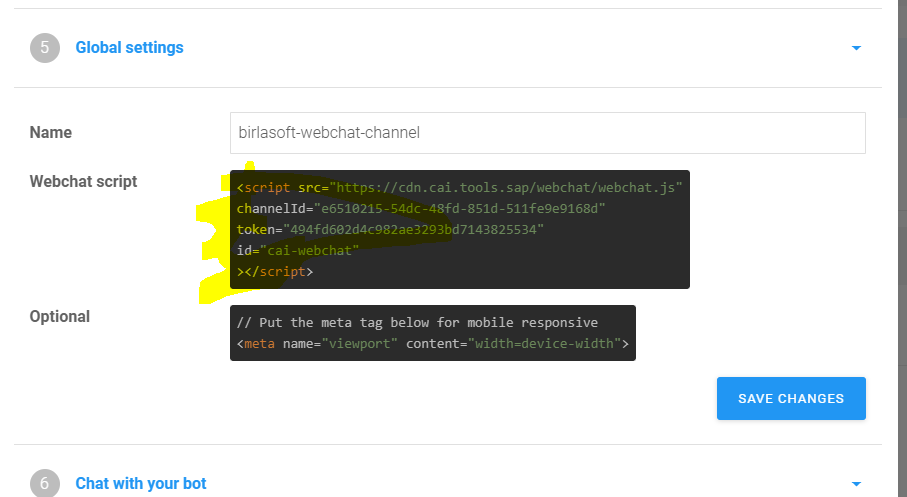
* 1. Perform the similar action for all the skills.
  2. Now the bot is live and it can be used in the Fiori.

1. Integration of Chatbot with SAP Fiori.
   1. Now we need to create two projects to integrate Chatbot with Fiori.
      1. Fiori Plugin: it’s used to show a chat icon in the tool bar of Fiori launchpad, user can launch the chatbot by clicking the icon.
      2. HTML Project : it holds a simple html page which connects to chatbot created in SAP Conversational AI platform. This HTML page will be used in the Fiori plugin project.
         * + Both the above projects can be found in the Web IDE.
   2. Create a simple project and create a simple HTML file in the WebContent.

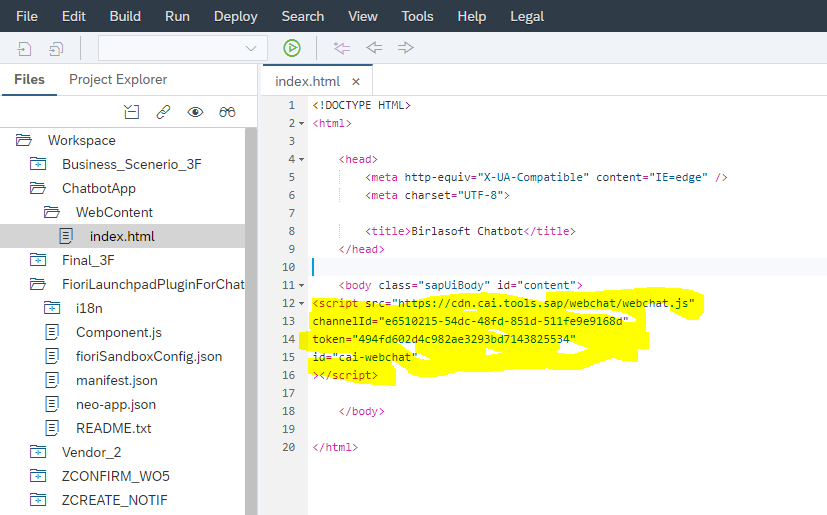


* 1. Now go to SAP Conversational AI platform and copy the path of the generated script.

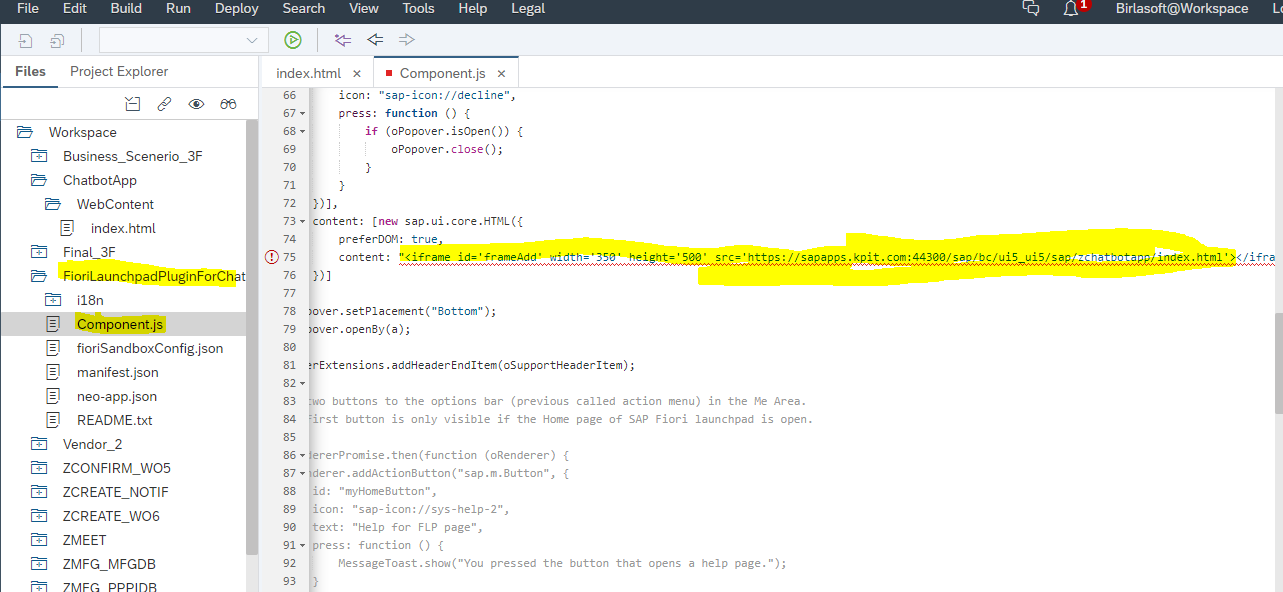




* 1. Copy the script and put it in the HTML page.

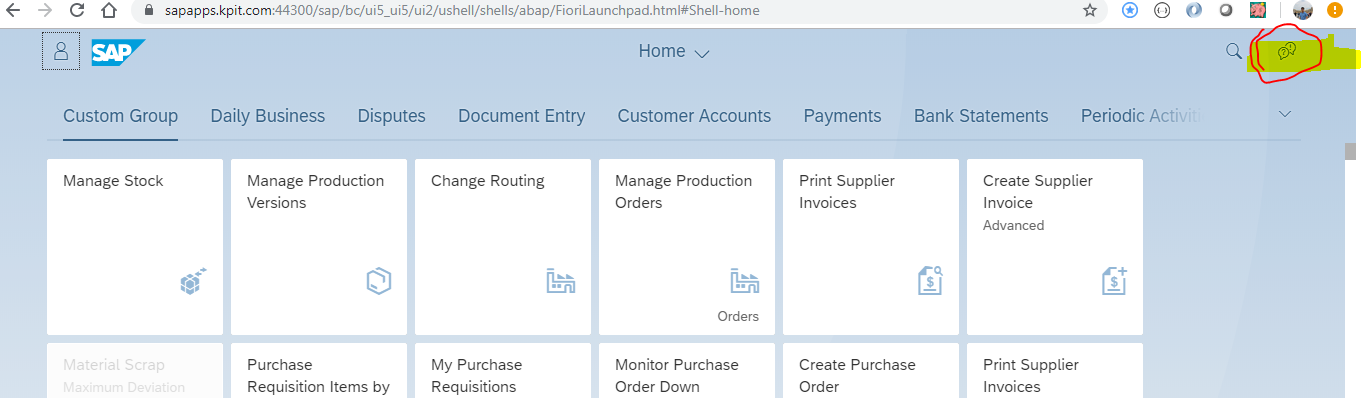


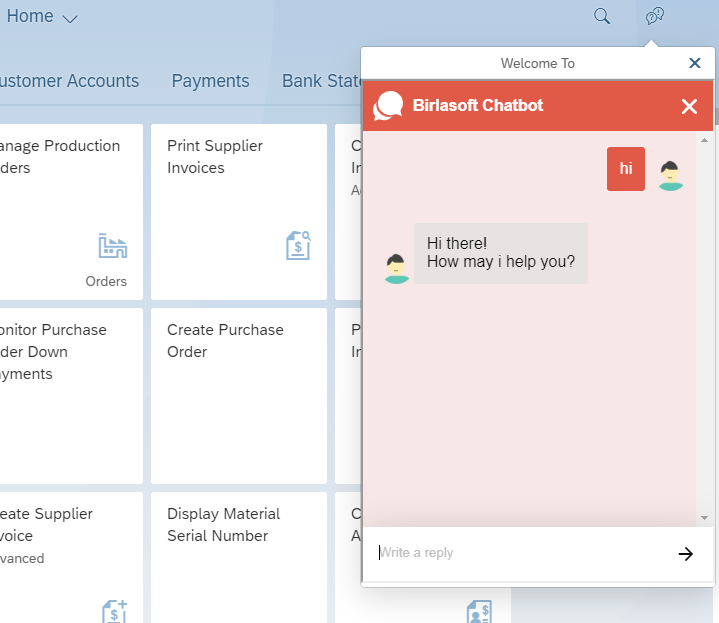
* 1. Now deploy the HTML project onto the ABAP repository.
  2. Create SAP Fiori plugin project and link the above HTML project in the Component.js.



* 1. Deploy the Fiori plugin project into ABAP repository and perform the following steps to configure it in the Fiori Launchpad.
     1. From S4HAN 1809 onwards.
        1. Define (id, UI5 app URL and Id) the plugin by using the TCode **/UI2/FLP\_CONF\_DEF**
        2. Activated the plugin by using the TCode /UI2/FLP\_SYS\_CONF
        3. It will appear for all the users.
     2. S4HANA 1709/previous versions.
        1. Created a catalog in the launch pad designer. ( /UI2/FLPD\_CUST)
        2. Created Target mapping in the catalog (shell, plugin, UI5 App details).
        3. Created a Role in **PFCG** and added the catalog to the role in the **menu** tab.
        4. Added the role to the user.

1. Now the Chatbot is ready to be used, launch the Fiori Launchpad and open the chatbot by clicking on the chatbot icon available in the header toolbar.





1. Reference.

<https://events.sap.com/teched/en/session/48832?source=social-global-sap-developers-linkedin_company-audienceengagement-developers-ai-spr-2870783501&campaigncode=CRM-XB19-MKT-DGEALL>

<https://cai.tools.sap/blog/build-a-pokebot-with-sap-conversational-ai-and-nodejs/>