IBM Watson

Where is the ISS?

A Node-RED Approach

Technical Challenge

Version 2021.03.05

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Table of Contents

Preface

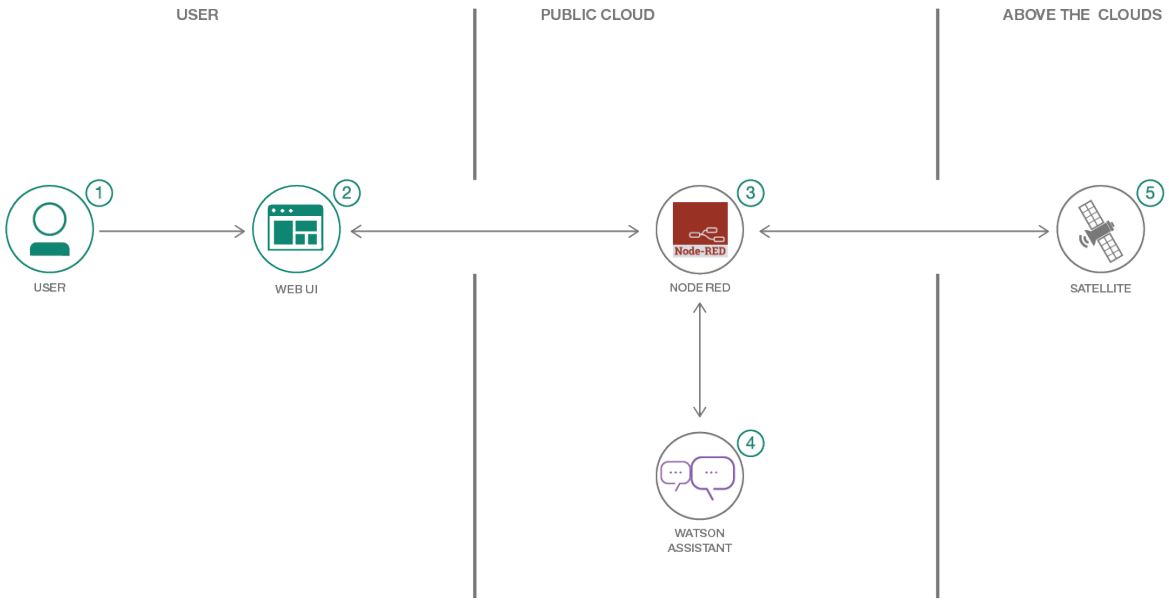
Overview

In this Code Pattern, you will build a satellite tracker using Node-RED. You will connect the **Watson Assistant** Chatbot with a **node-red-contrib-satellites** node, as well as a web UI and **worldmap** node.

Objectives

- Build a complex flow and web UI using simple Node-RED tools.
- Implement a chatbot with Watson Assistant and embed it on a web page with Node-RED.
- Get satellite information for the International Space Station (ISS) and use it in a web app.

Flow



1. User interacts with Web UI to query the chat bot "Where is the ISS?"
2. Web UI communicates with Node-RED.
3. Node-RED processes info and performs HTTP requests.
4. The Node-RED app communicates with Watson Assistant to extract intents and entities.
5. Satellites orbiting the earth send position info which is streamed to Node-RED module.

Tools

- Install Node-RED manually: Node-RED: Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
- Create a chatbot Watson Assistant with a program that conducts a conversation via auditory or textual methods.
- Artificial Intelligence: Artificial intelligence can be applied to disparate solution spaces to deliver disruptive technologies.
- Node.js: An open-source JavaScript run-time environment for executing server-side JavaScript code.

Prerequisites

You have created an account and logged into IBM Cloud platform.

About Node-RED

Node-RED is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the JS Foundation.

Follow this link for more information: https://nodered.org/about/#flow-based-programming

Milestone 1: Install Node-RED Locally

At the core of Node-RED is Node.js, which is a JavaScript runtime that boasts the largest ecosystem of open source components

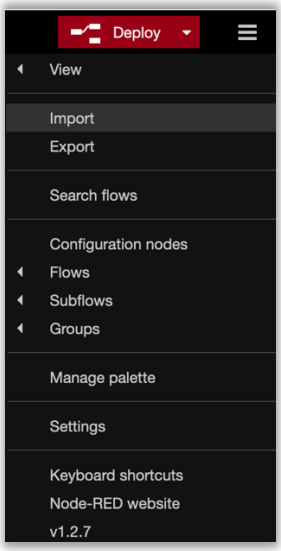
Installing on Mac	Installing on PC

1. Download and Install npm: https://nodejs.org/en/#home-downloadhead
2. Accept the defaults when installing.
3. Open a terminal on your Mac.
4. Enter the following command to install node-RED:
`sudo npm install -g --unsafe-perm node-red`
This command will install Node-RED as a global module along with its dependencies.
You can confirm it has succeeded if the end of the command output looks similar to:
`+ node-red@1.0.0`
`added 332 packages from 341 contributors in 18.494s`
`found 0 vulnerabilities`
5. You are now ready to invoke Node-RED. At the terminal prompt, type:
`$ node-red`
6. Enter the following URL (or per returned command) http://127.0.0.1:1880/
	1. Download the latest 10.x LTS version of Node.js from the official Node.js home page.
	2. Run the downloaded MSI file. Installing Node.js requires local administrator rights; if you are not a local administrator, you will be prompted for an administrator password on install.
	3. Accept the defaults when installing.
	4. After installation completes, close any open command prompts and re-open to ensure new environment variables are picked up.
	5. Once installed, open a command prompt and run the following command to ensure Node.js and npm are installed correctly.
	Using Powershell: `node --version; npm --version`
	Using cmd: `node --version && npm --version`
	You should receive back output that looks similar to:
	`v10.16.3`
	`6.11.3`
	6. Install Node-RED:
	`npm install -g --unsafe-perm node-red`
	7. At the Command prompt Type:
	`node-red`
	8. Enter the following URL (or per returned command) http://127.0.0.1:1880/

Milestone 2: Import Node-RED Flow

You are now ready to import an already built flow onto your canvas. Complete the following steps:

1. Navigate to this repository and copy the JSON code.
2. Download the ISS Node-RED flow in JSON format on your local drive, from this link: https://github.com/apischdo/skillsacademy/blob/master/ISS-Node-RED-Flow.json



3. From the Node-RED hamburger icon in the top right corner select Import.

4. From the **Import** panel, you can either select the ISS Node-RED file for upload or copy/paste the JSON flow in the window.

   The flow may open in an adjacent canvas.

5. Click somewhere in the white space to release the flow and click **Deploy** in the top right corner.
6. Click **Deploy** from the top right corner above the canvas. Notice that some of the nodes appear as unknown.
7. From the hamburger icon, click **Manage Palate** and select the Install tab.

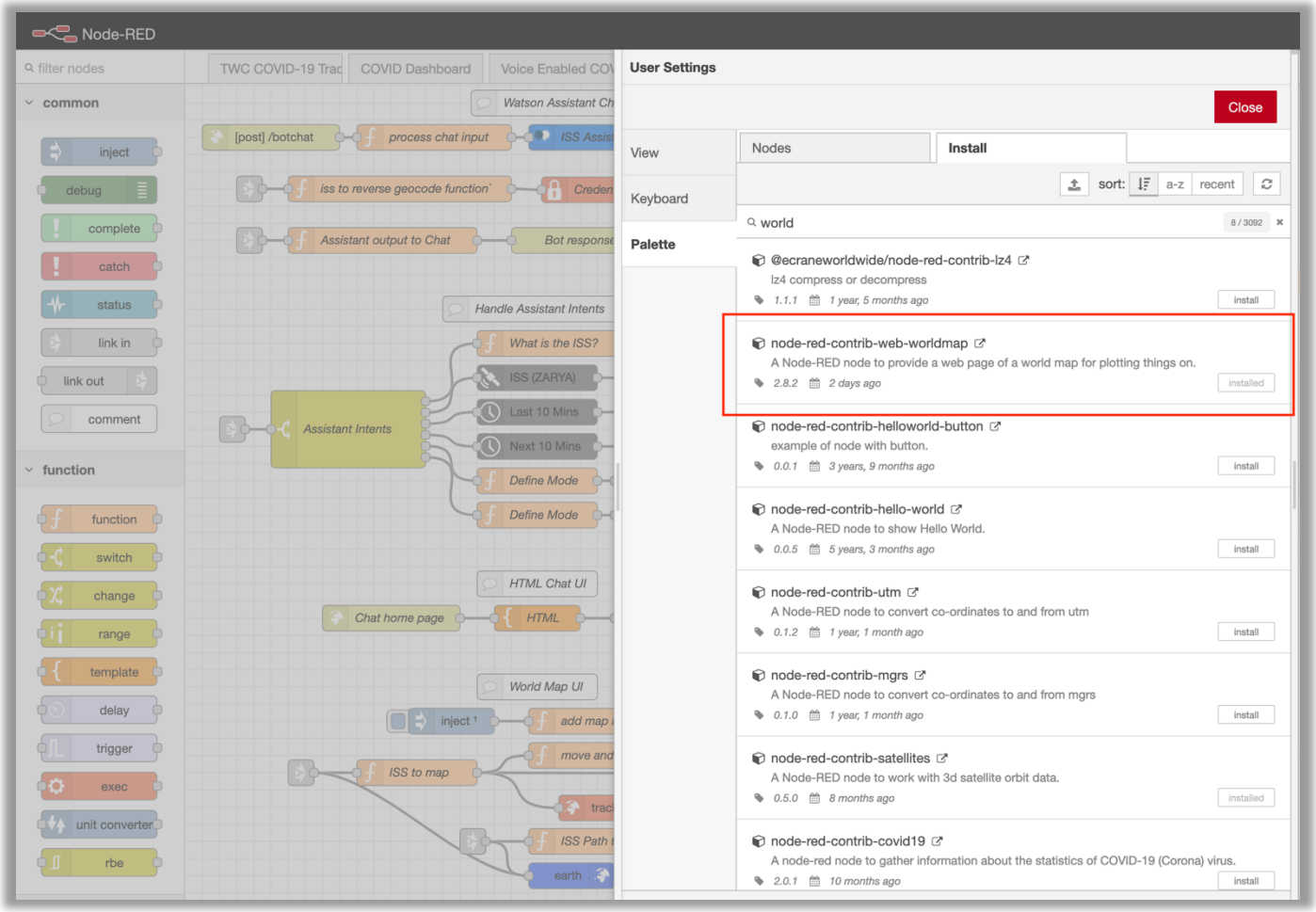
8. Type the name of each of the missing nodes per below (unknown) just as it appears in bold and install them:

   Watson (node-red-node-watson)

   Credentials (node-red-contrib-credentials)

   Worldmap (node-red-contrib-web-worlddmap)

   Satellites (node-red-contrib-satellites) This node may already be available via the flow, if not, install it.

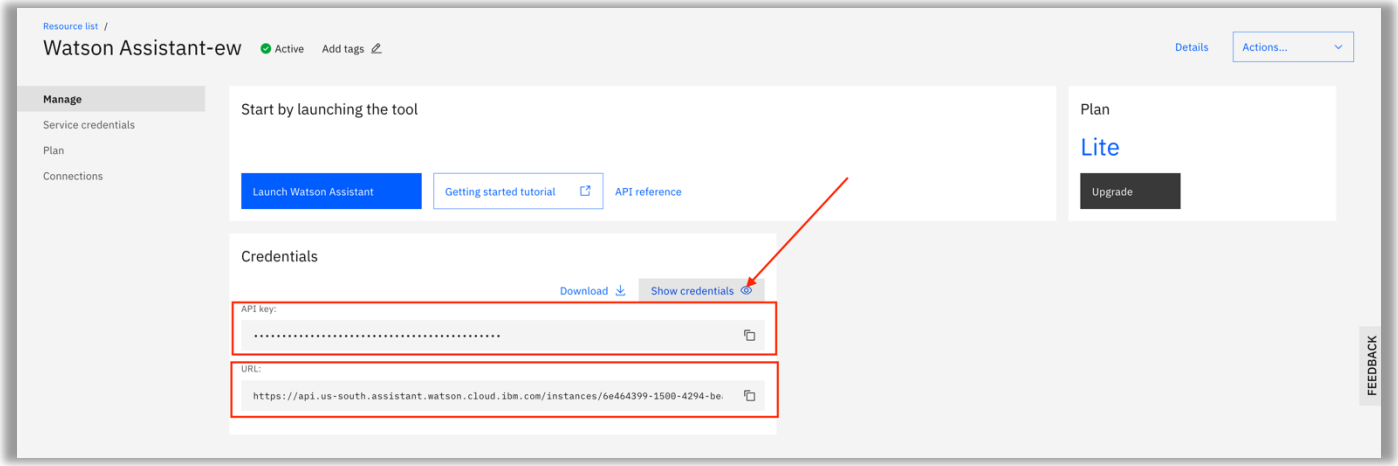


9. Click Deploy after every and any activity that you do on the canvas.

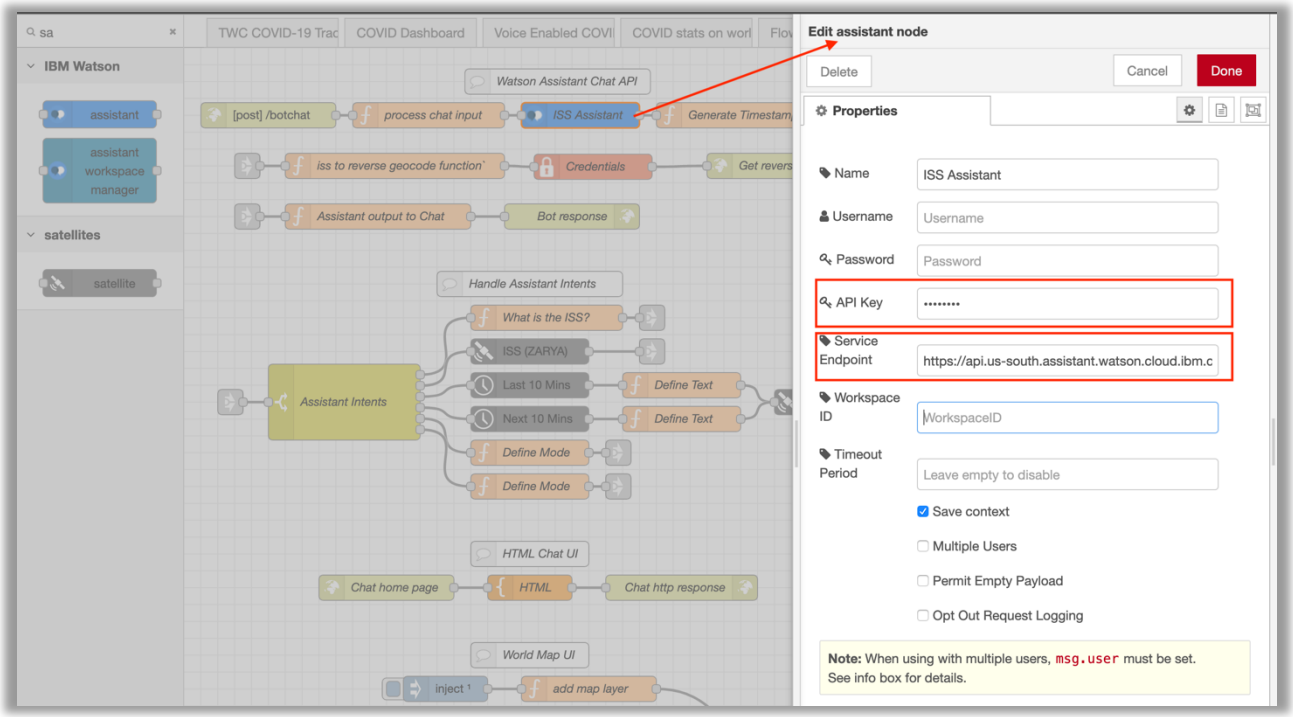
Milestone 3: Populate the Assistant Node with API Credentials

This lab assumes that you have already created the Watson Assistant service from earlier labs, therefore, you do not need to create a new Watson Assistant service.

1. From the IBM Cloud **Resource** list, click and open the **Watson Assistant** service.
2. Copy the **API key** and then the **URL** (in any order, one at a time).



3. Refer to the Node-RED tab in your browser and *paste* the **API key** and then the **URL** (in any order, one at a time) inside the **ISS Assistant** node.

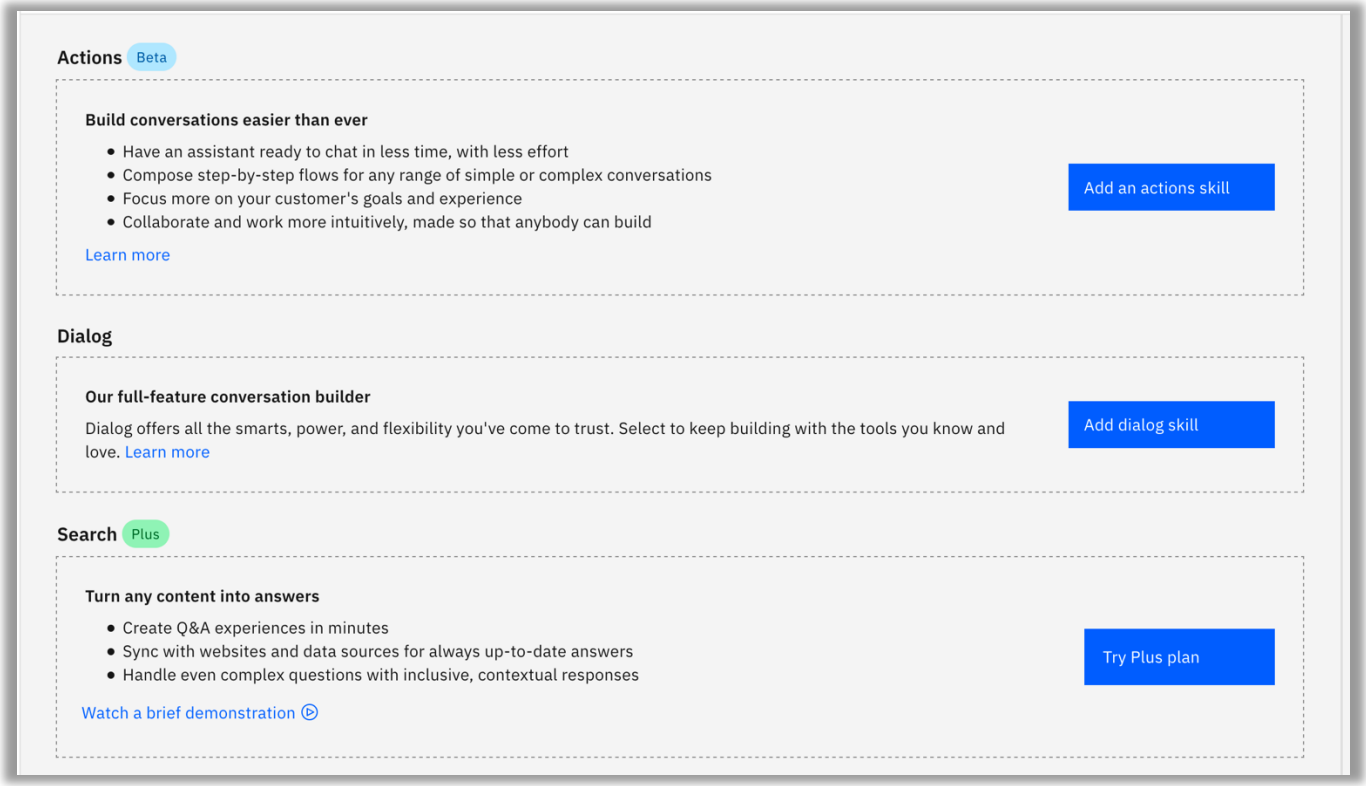


Now you need the Workspace ID.

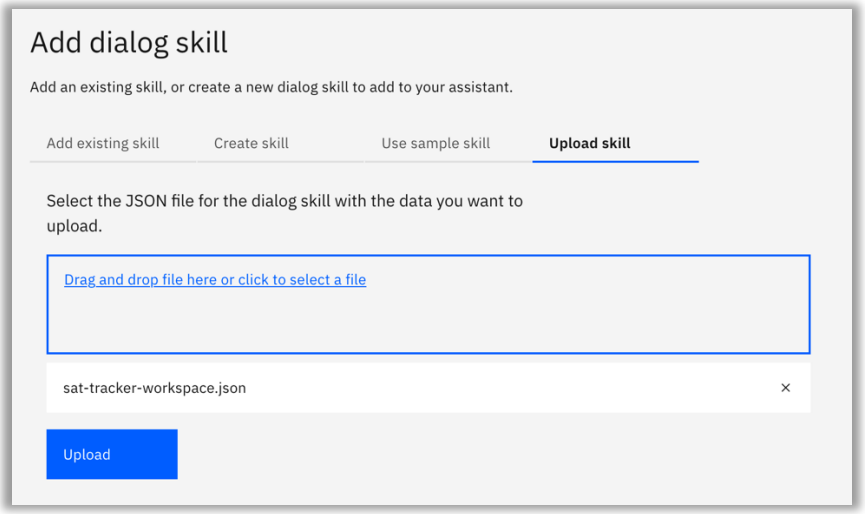
Upload Dialog Skill

You are now ready to upload an already build skill and use the ID from that Skill to direct the Node to the particular conversation therein.

1. Download the sat-tracker-workspace.json to your local drive.
2. Back to the Assistance service Manage page, Click **Launch Watson Assistant**.
3. Click **Create Assistant** and give it a meaningful name such as ISS tracker and create the Assistant.
4. Click **Add dialog skill** (the middle box).

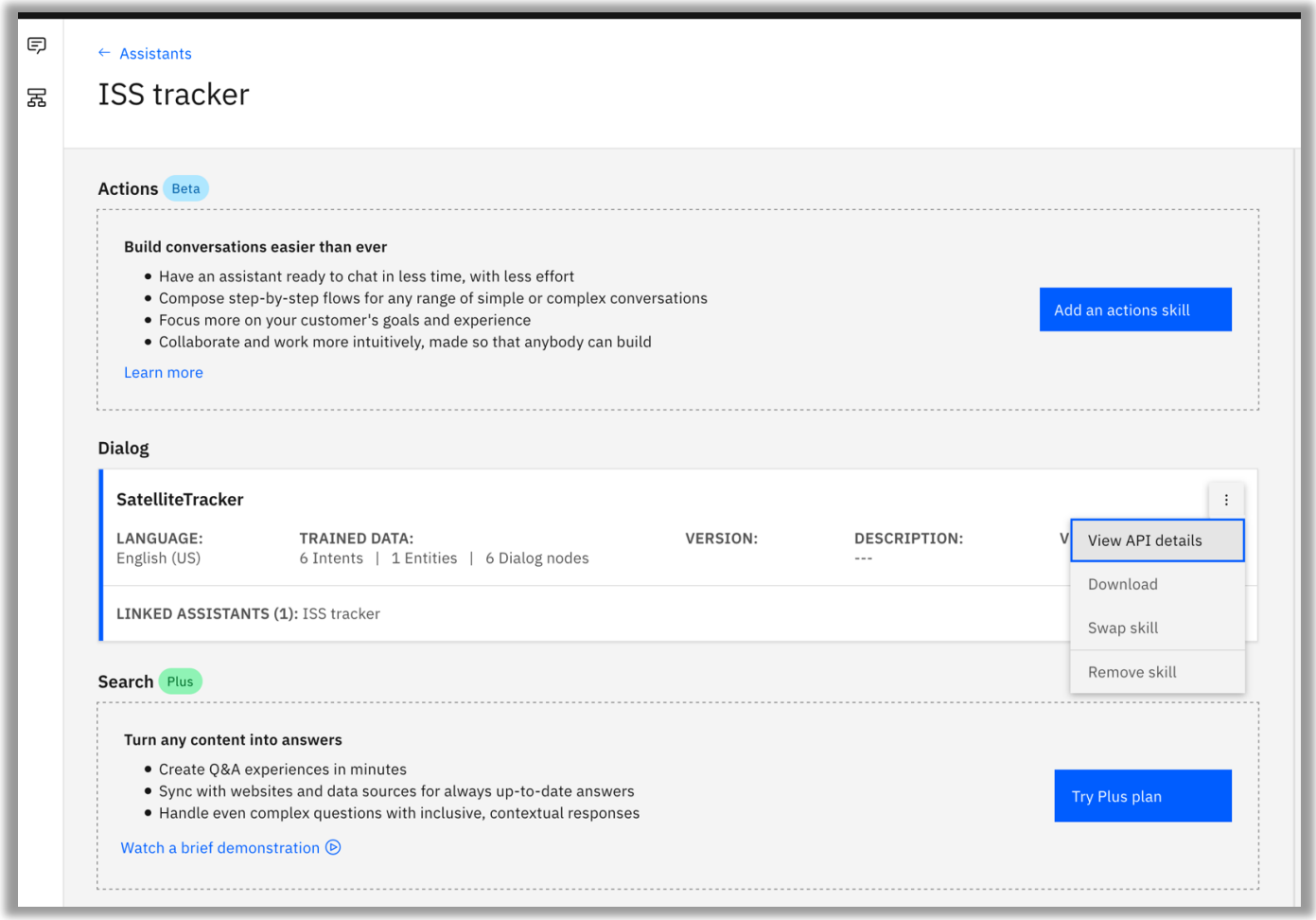


5. Select the **Upload skill** option and then navigate to the sat-tracker-workspace.json on your local drive and upload it.

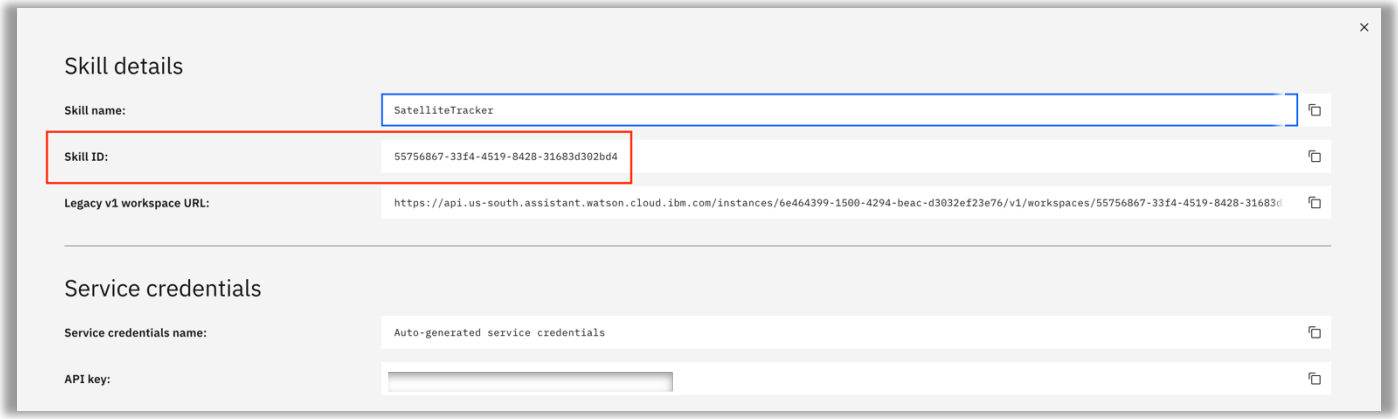


You are now ready to open the Dialog Skill and obtain the Workspace ID (same as Skill ID) for the Assistant node that is on your Node-RED canvas. Complete the following steps:

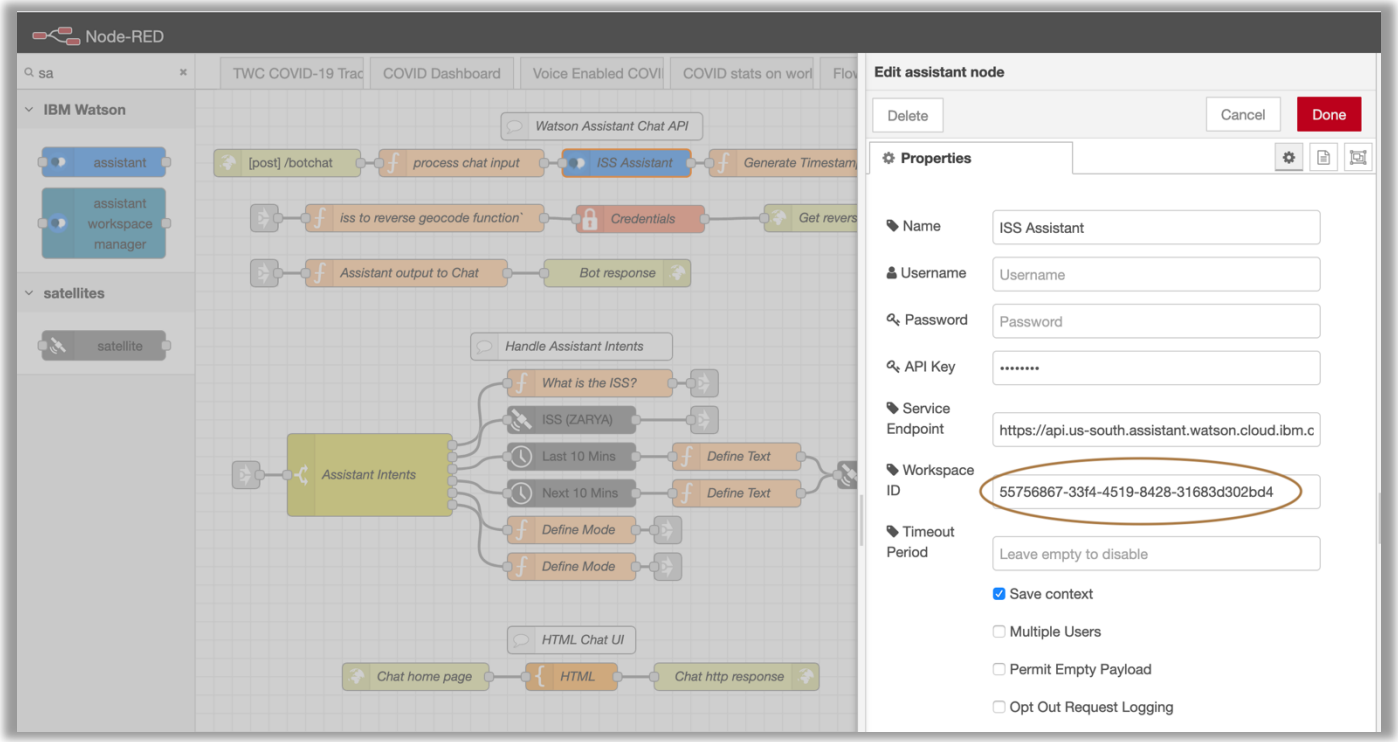
6. Click the *three dots* inside the **SatelliteTracker** Dialog to the right of the page and click **View API details**.



7. Copy the Skill ID (same as Workspace ID inside the node).



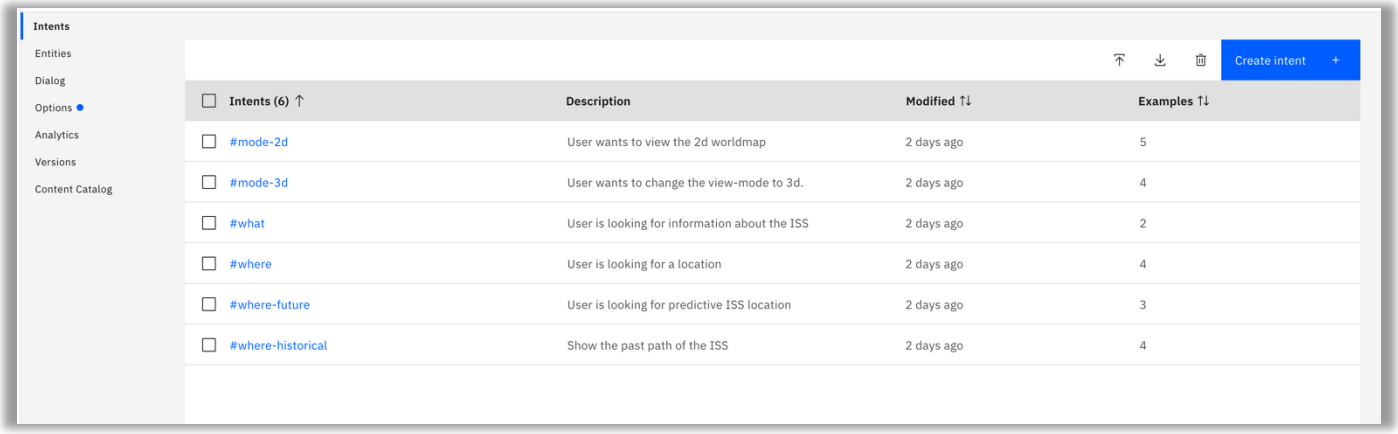
8. Paste it in the **ISS Assistant** node on your Node-RED canvas



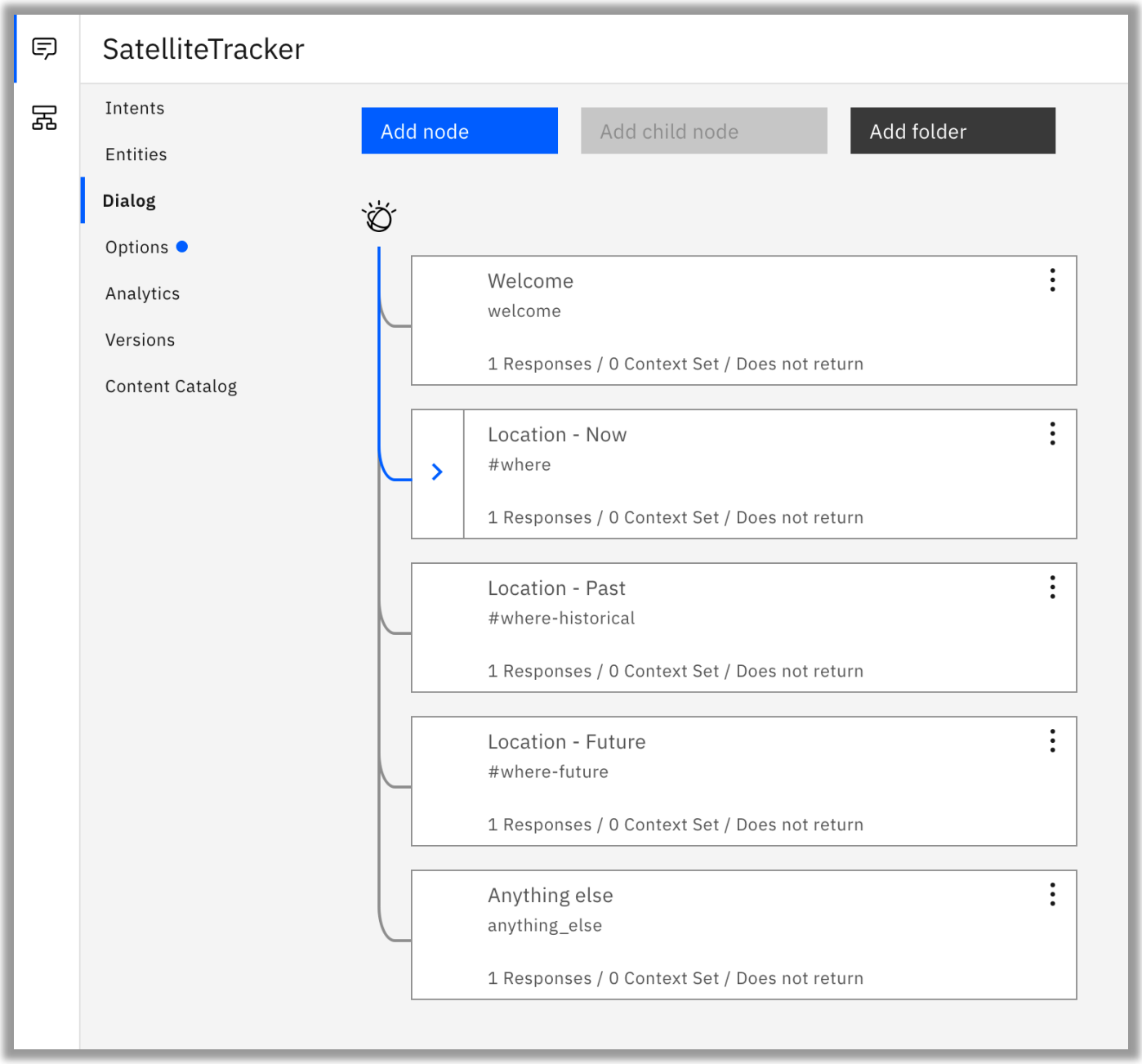
9. Click **Deploy**. Each time you make any changes to the canvas, click **Deploy**. It is akin to saving your work.

Getting to know the ISS Tracker chatbot

10. Before you move on to the next section as a team, explore the Intents, entities and the dialog constructs of the ISS Tracker chatbot.
11. Open the Intents tab and notice how these would relate to questions you may ask the bot.



12. Next, open the Dialog section and observe the entries, care to change any of them. Remember Intents are verbs and Entities are nouns, and both refer to what the human thinks and says. The Dialog is what the chatbot replies and we specify that verbiage.

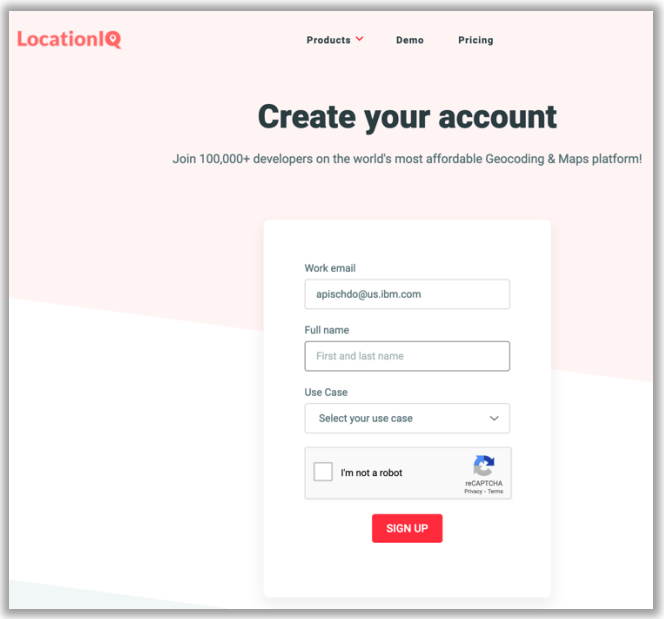


Although not part of the exercises, feel free to make any changes you'd like to the chatbot.

Milestone 4: Configure LocationIQ Credentials

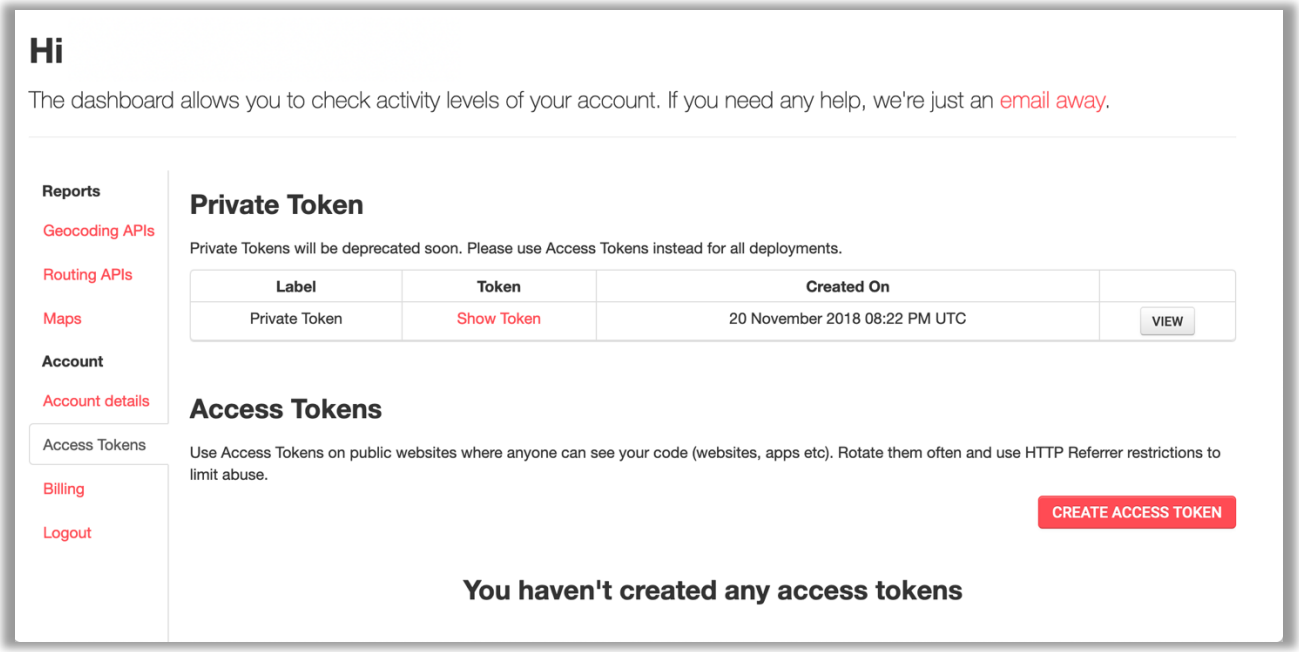
You are now ready to populate the last API, and it goes inside the **Credentials** node. But first you need to go to an external and trusted site and obtain an API key for the services that LocationIQ provides.

1. Navigate to this web site https://locationiq.com/ to obtain your key.
2. Enter your email address and click **Get Started**.
3. Enter all necessary information.

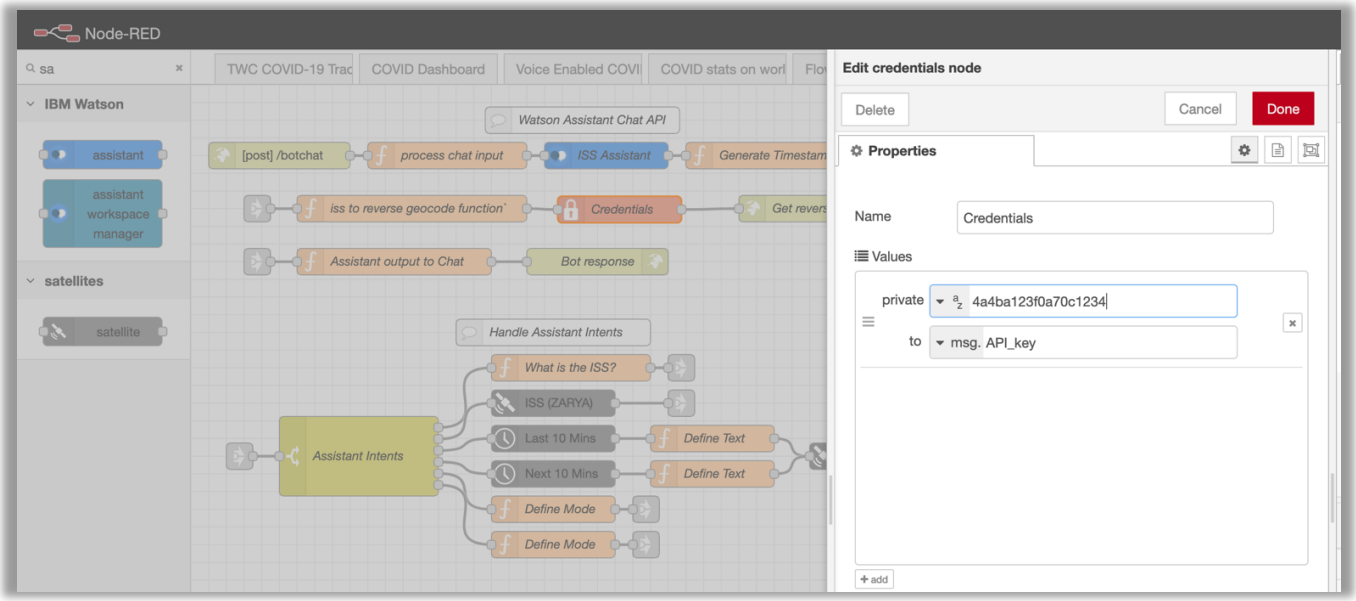


4. You must now login either using password or email. Select your preferred option.

5. Click the **Access Tokens** link and then click **View** to see your API key (token).



6. Back to the Node-RED canvas. Click to open the **Credentials** node.
7. Copy and paste the token from LocationIQ inside the **private** box and click **Done**.



8. **Deploy** the flow

Milestone 5: Reveal Satellite Location

You are now ready to bring together the results of your efforts. Complete the following steps:

1. Open a new tab on your browser and copy or type the following URL:

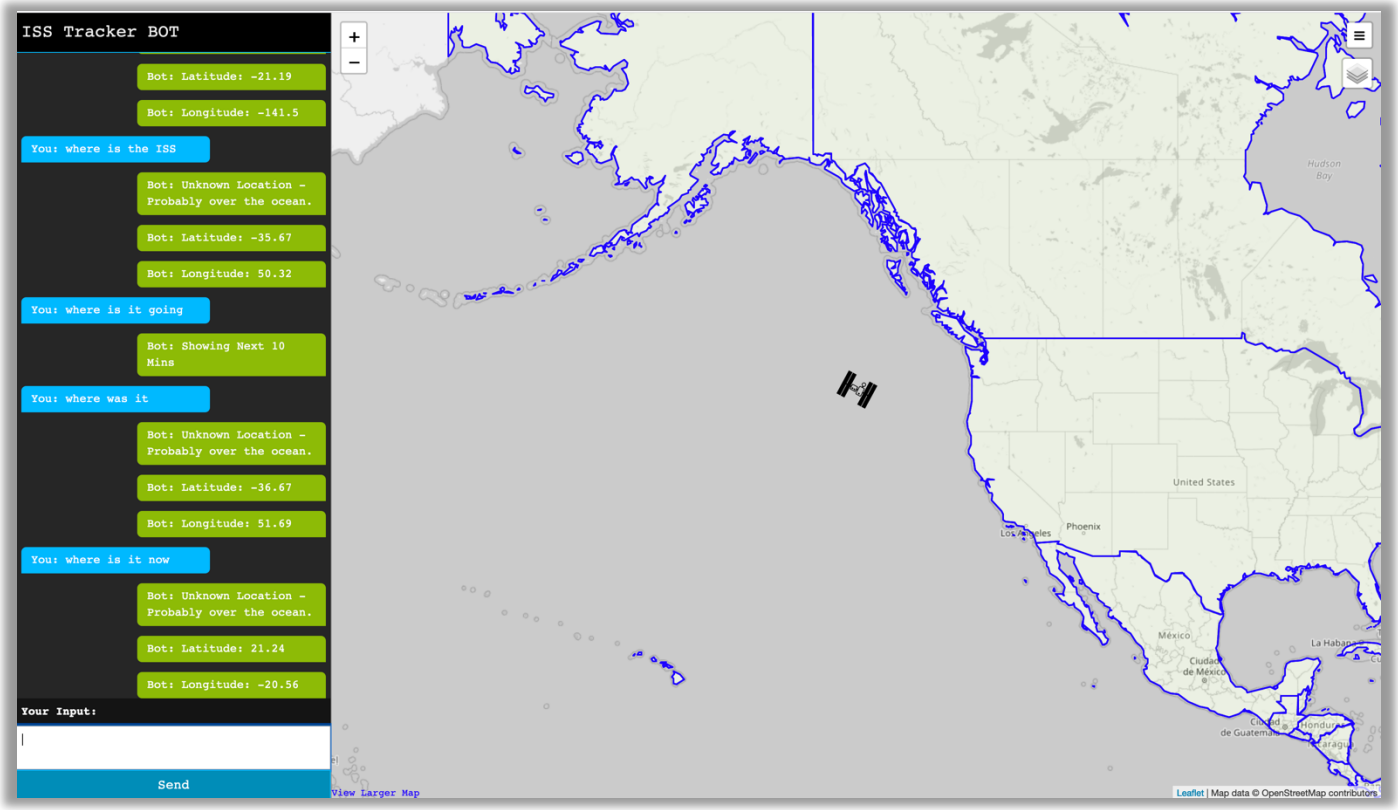
   http://127.0.0.1:1880/bot

2. Begin to ask questions, such:

   Where is ISS?

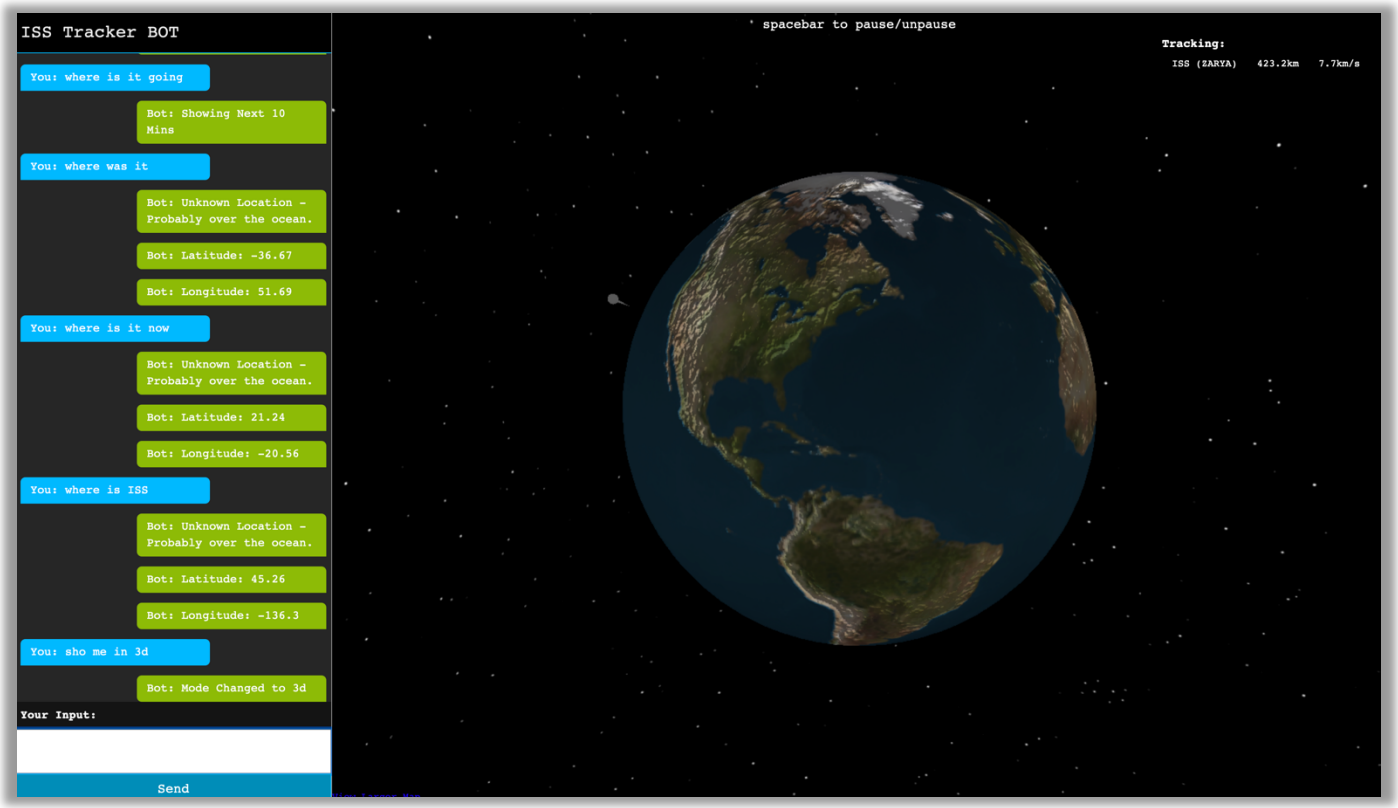
   Where is it going next?

   Where has it been?



Try a new question: show me in 3D

Once you see the globe, now ask: where is ISS?



Please Recycle