

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.

© Copyright International Business Machines Corporation 2021.

This document may not be reproduced in whole or in part without the prior written permission of IBM. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

Preface	5
Overview	5
Objectives	5
Flow	5
Tools	6
Prerequisites	6
About Node-RED	6
Milestone 1: Install Node-RED Locally	7
Milestone 2: Import Node-RED Flow	8
Milestone 4: Populate the Assistant Node with API Credentials	10
Upload Dialog Skill	11
Getting to know the ISS Tracker chatbot	15
Milestone 5: Configure LocationIQ Credentials	17
Milestone 6: Reveal Satellite Location	19

Preface

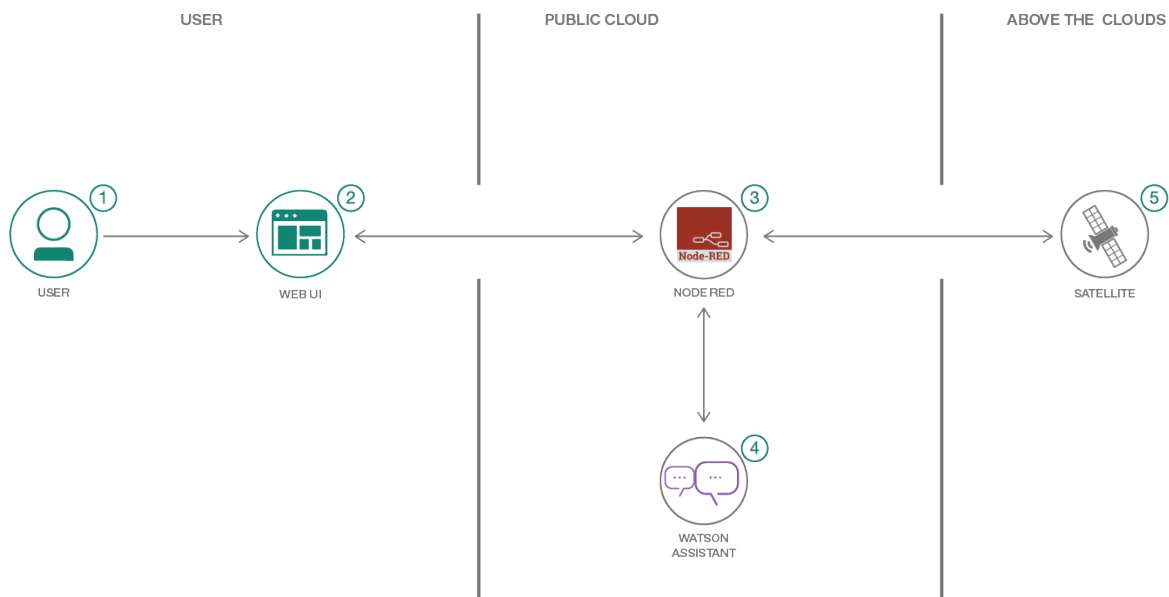
Overview

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

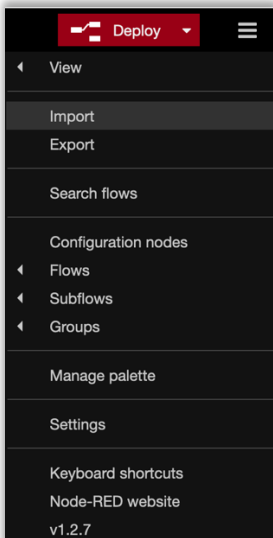
At the core of Node-RED is Node.js, which is a JavaScript runtime that [boasts the largest ecosystem of open source components](#)

Installing on Mac	Installing on PC
<ol style="list-style-type: none"> Download and Install npm: https://nodejs.org/en/#home-downloadhead Accept the defaults when installing. Open a terminal on your Mac. Enter the following command to install node-RED: <code>sudo npm install -g --unsafe-perm node-red</code> 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: <code>+ node-red@1.0.0 added 332 packages from 341 contributors in 18.494s found 0 vulnerabilities</code> You are now ready to invoke Node-RED. At the terminal prompt, type: <code>\$ node-red</code> Enter the following URL (or per returned command) http://127.0.0.1:1880/ 	<ol style="list-style-type: none"> Download the latest 10.x LTS version of Node.js from the official Node.js home page. 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. Accept the defaults when installing. After installation completes, close any open command prompts and re-open to ensure new environment variables are picked up. Once installed, open a command prompt and run the following command to ensure Node.js and npm are installed correctly. Using Powershell: <code>node --version; npm --version</code> Using cmd: <code>node --version && npm --version</code> You should receive back output that looks similar to: <code>v10.16.3 6.11.3</code> Install Node-RED: <code>npm install -g --unsafe-perm node-red</code> At the Command prompt Type: <code>node-red</code> 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 Palette** 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-worldmap)

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

The screenshot shows the Node-RED interface with a flow titled "TWC COVID-19 Trac" and "COVID Dashboard". The flow includes nodes for "Assistant Intents", "ISS (ZARYA)", "Last 10 Mins", "Next 10 Mins", "Define Mode", "HTML Chat UI", "Chat home page", "HTML", "World Map UI", "inject 1", "add map", "move and", "ISS Path", and "earth".

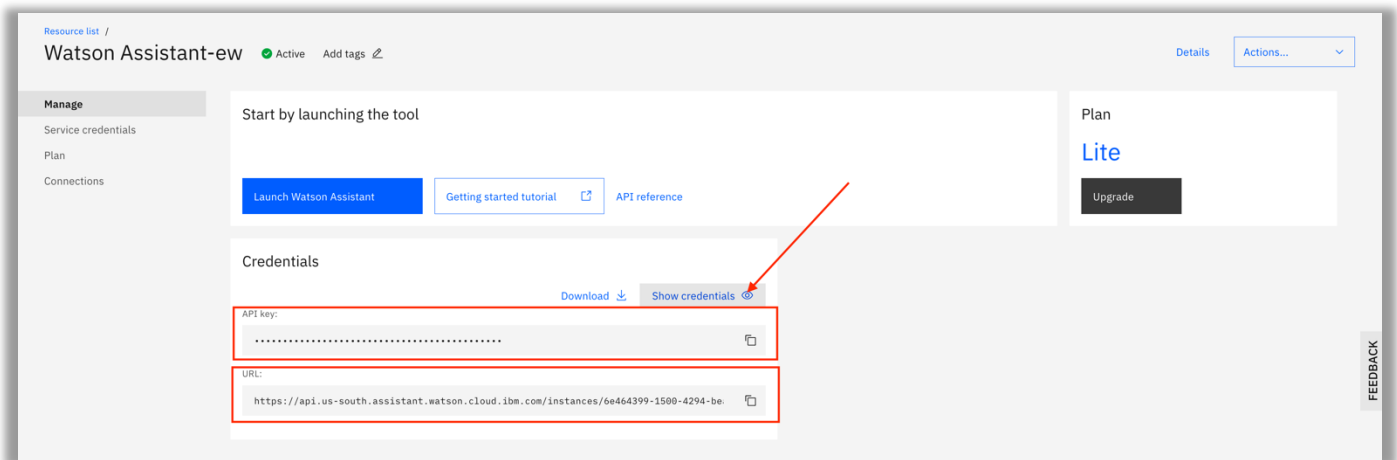
The "User Settings" panel is open, showing the "Nodes" tab. The search bar contains "world". The "Palette" shows a list of nodes, with "node-red-contrib-web-worldmap" highlighted in a red box. The node description for "node-red-contrib-web-worldmap" is: "A Node-RED node to provide a web page of a world map for plotting things on." The version is 2.8.2, released 2 days ago, and it is marked as "installed".

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

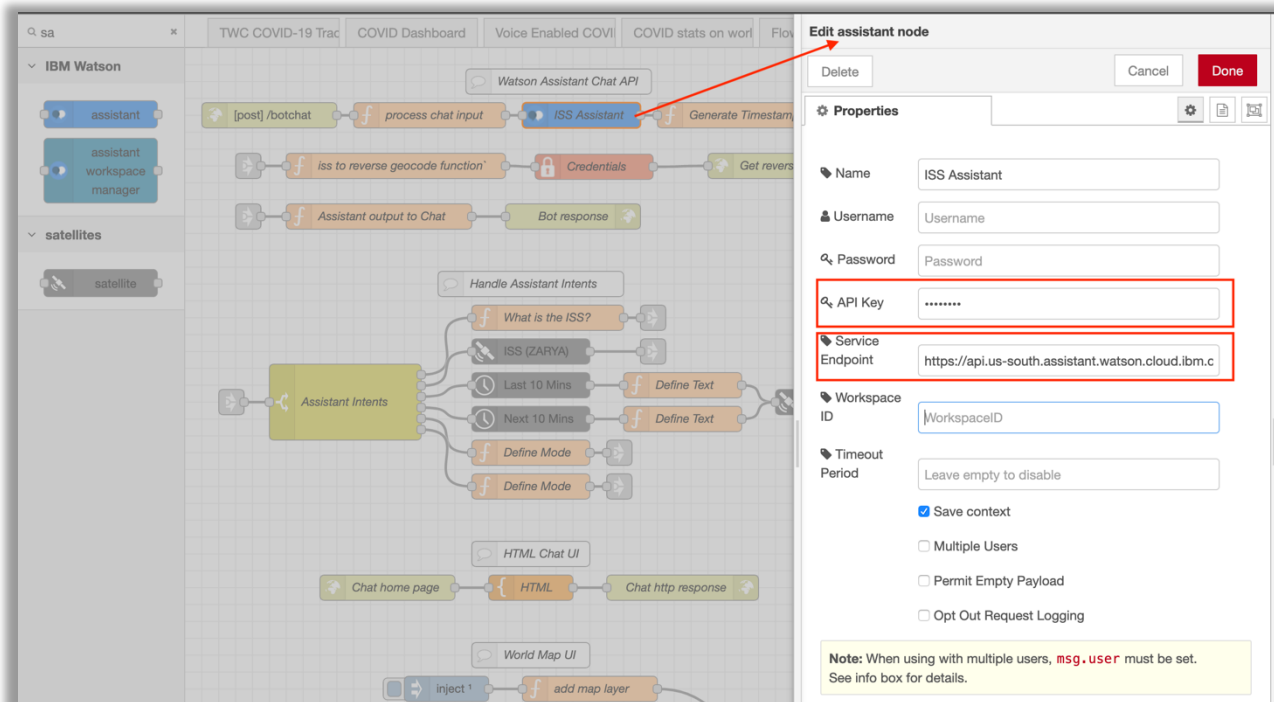
Milestone 3: Populate the Assistant Node with API Credentials

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).

Actions Beta

Build conversations easier than ever

- Have an assistant ready to chat in less time, with less effort
- Compose step-by-step flows for any range of simple or complex conversations
- Focus more on your customer's goals and experience
- Collaborate and work more intuitively, made so that anybody can build

[Learn more](#)

[Add an actions skill](#)

Dialog

Our full-feature conversation builder

Dialog offers all the smarts, power, and flexibility you've come to trust. Select to keep building with the tools you know and love. [Learn more](#)

[Add dialog skill](#)

Search Plus

Turn any content into answers

- Create Q&A experiences in minutes
- Sync with websites and data sources for always up-to-date answers
- Handle even complex questions with inclusive, contextual responses

[Watch a brief demonstration](#)

[Try Plus plan](#)

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

Add dialog skill

Add an existing skill, or create a new dialog skill to add to your assistant.

[Add existing skill](#) [Create skill](#) [Use sample skill](#) **[Upload skill](#)**

Select the JSON file for the dialog skill with the data you want to upload.

[Drag and drop file here or click to select a file](#)

sat-tracker-workspace.json ×

Upload

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**.

The screenshot shows the IBM Skills Academy interface for configuring a dialog skill. The main heading is "ISS tracker". Below it, there are three main sections: "Actions", "Dialog", and "Search".

Actions (Beta)

Build conversations easier than ever

- Have an assistant ready to chat in less time, with less effort
- Compose step-by-step flows for any range of simple or complex conversations
- Focus more on your customer's goals and experience
- Collaborate and work more intuitively, made so that anybody can build

[Learn more](#)

Dialog

LANGUAGE:	TRAINED DATA:	VERSION:	DESCRIPTION:
English (US)	6 Intents 1 Entities 6 Dialog nodes		---

LINKED ASSISTANTS (1): ISS tracker

Search (Plus)

Turn any content into answers

- Create Q&A experiences in minutes
- Sync with websites and data sources for always up-to-date answers
- Handle even complex questions with inclusive, contextual responses

[Watch a brief demonstration](#)

The "SatelliteTracker" skill is highlighted in the "Dialog" section. A dropdown menu is open, showing the following options: "View API details", "Download", "Swap skill", and "Remove skill". The "View API details" option is highlighted.

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

Skill details

Skill name: SatelliteTracker

Skill ID: 55756867-33f4-4519-8428-31683d302bd4

Legacy v1 workspace URL: https://api.us-south.assistant.watson.cloud.ibm.com/instances/6e464399-1506-4294-beac-d3032ef23e76/v1/workspaces/55756867-33f4-4519-8428-31683d302bd4

Service credentials

Service credentials name: Auto-generated service credentials

API key: [Redacted]

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

Node-RED

IBM Watson

- assistant
- assistant workspace manager

satellites

- satellite

Canvas: Watson Assistant Chat API, [post]/botchat, process chat input, ISS Assistant, Generate Timestamp, iss to reverse geocode function, Credentials, Get reverse geocode, Assistant output to Chat, Bot response, Handle Assistant Intents, What is the ISS?, ISS (ZARYA), Last 10 Mins, Next 10 Mins, Define Text, Define Mode, Define Mode, HTML Chat UI, Chat home page, HTML, Chat http response

Edit assistant node

Delete Cancel Done

Properties

Name: ISS Assistant

Username: Username

Password: Password

API Key:

Service Endpoint: https://api.us-south.assistant.watson.cloud.ibm.c

Workspace ID: 55756867-33f4-4519-8428-31683d302bd4

Timeout Period: Leave empty to disable

Save context

Multiple Users

Permit Empty Payload

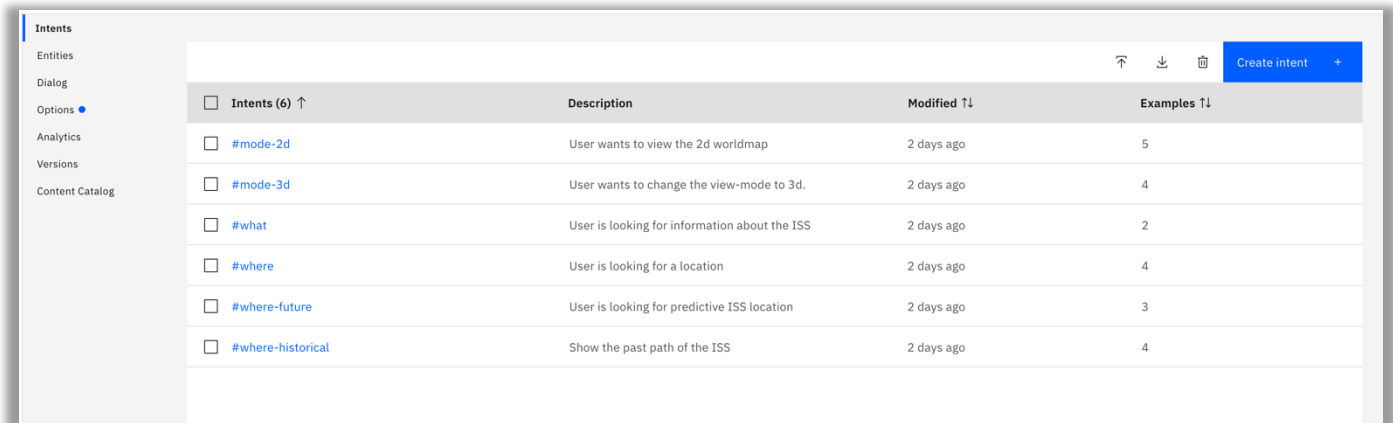
Opt Out Request Logging

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.



<input type="checkbox"/> Intents (6) ↑	Description	Modified ↑↓	Examples ↑↓
<input type="checkbox"/> #mode-2d	User wants to view the 2d worldmap	2 days ago	5
<input type="checkbox"/> #mode-3d	User wants to change the view-mode to 3d.	2 days ago	4
<input type="checkbox"/> #what	User is looking for information about the ISS	2 days ago	2
<input type="checkbox"/> #where	User is looking for a location	2 days ago	4
<input type="checkbox"/> #where-future	User is looking for predictive ISS location	2 days ago	3
<input type="checkbox"/> #where-historical	Show the past path of the ISS	2 days ago	4

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.

The screenshot shows the IBM Watson Assistant console for a chatbot named "SatelliteTracker". The left sidebar contains navigation options: Intents, Entities, Dialog (selected), Options, Analytics, Versions, and Content Catalog. The main area displays the Dialog section with a lightbulb icon and three buttons: "Add node" (blue), "Add child node" (grey), and "Add folder" (dark grey). Below these buttons, a list of dialog nodes is shown, each with a title, a list of utterances, and a status bar. The nodes are:

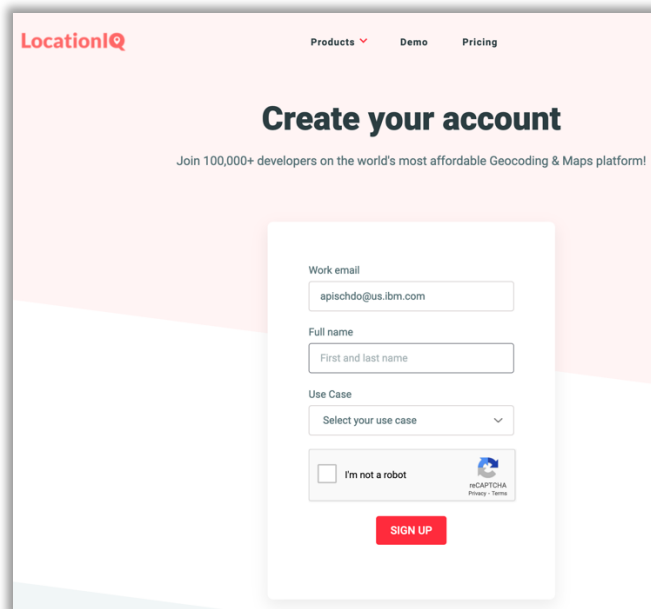
- Welcome**: utterances: "welcome"; status: "1 Responses / 0 Context Set / Does not return"
- Location - Now**: utterances: "#where"; status: "1 Responses / 0 Context Set / Does not return"
- Location - Past**: utterances: "#where-historical"; status: "1 Responses / 0 Context Set / Does not return"
- Location - Future**: utterances: "#where-future"; status: "1 Responses / 0 Context Set / Does not return"
- Anything else**: utterances: "anything_else"; status: "1 Responses / 0 Context Set / Does not return"

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.

A screenshot of the LocationIQ website's account creation page. The page has a light pink and white background. At the top left is the 'LocationIQ' logo. To the right are links for 'Products', 'Demo', and 'Pricing'. The main heading is 'Create your account' with a subtext 'Join 100,000+ developers on the world's most affordable Geocoding & Maps platform!'. The form is centered and contains the following fields: 'Work email' with the value 'apischdo@us.ibm.com', 'Full name' with the placeholder 'First and last name', and 'Use Case' with a dropdown menu showing 'Select your use case'. Below these is a checkbox for 'I'm not a robot' next to a reCAPTCHA logo. At the bottom of the form is a red 'SIGN UP' button.

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

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

Hi

The dashboard allows you to check activity levels of your account. If you need any help, we're just an [email away](#).

Reports

- Geocoding APIs
- Routing APIs
- Maps

Account

- Account details
- Access Tokens
- Billing
- Logout

Private Token

Private Tokens will be deprecated soon. Please use Access Tokens instead for all deployments.

Label	Token	Created On	
Private Token	Show Token	20 November 2018 08:22 PM UTC	VIEW

Access Tokens

Use Access Tokens on public websites where anyone can see your code (websites, apps etc). Rotate them often and use HTTP Referrer restrictions to limit abuse.

[CREATE ACCESS TOKEN](#)

You haven't created any access tokens

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

Node-RED interface showing a flow with a **Credentials** node. The **Edit credentials node** panel is open, showing the **private** field with the value `4a4ba123f0a70c1234` and the **to** field set to `msg.API_key`. The **Done** button is highlighted in red.

- 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:

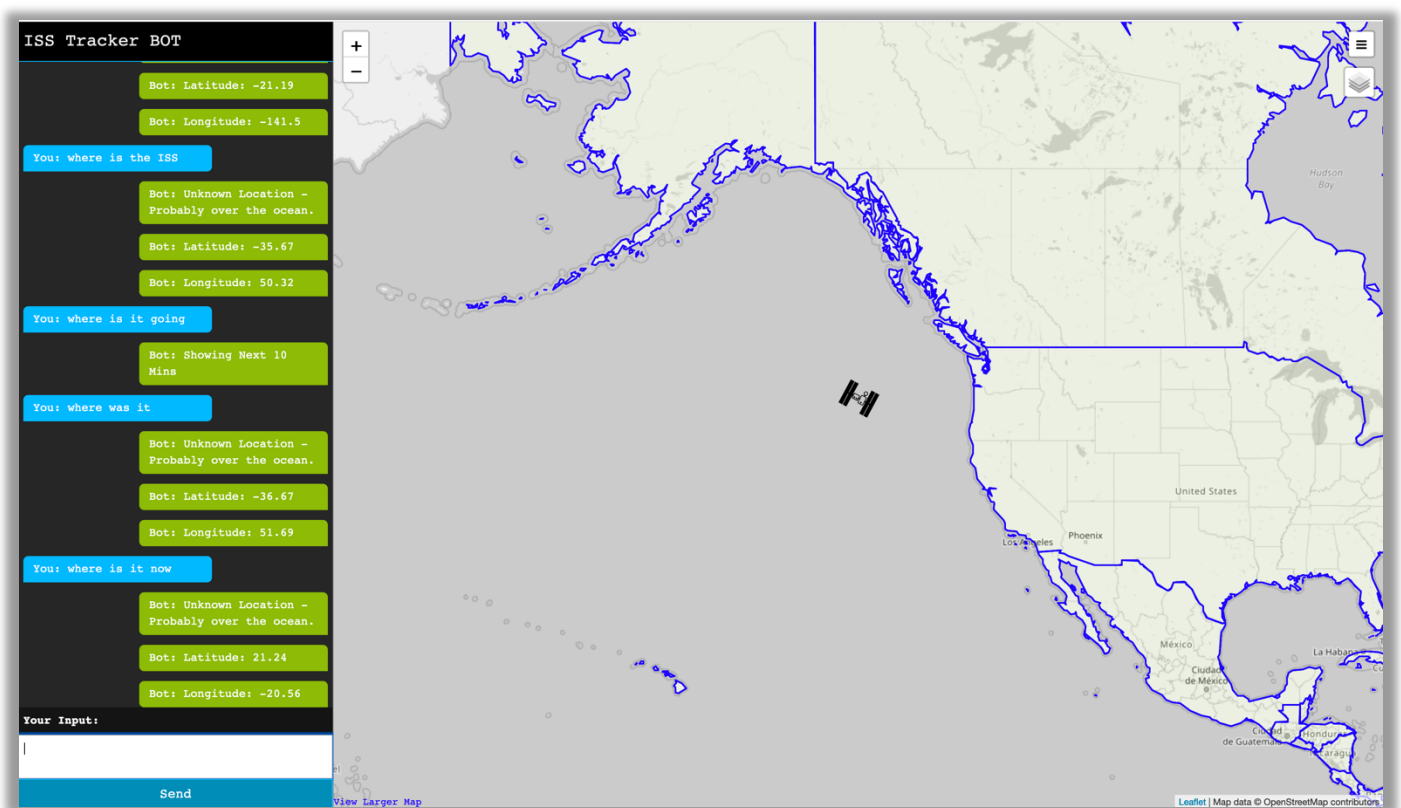
`http://127.0.0.1:1880/bot`

2. Begin to ask questions, such:

Where is ISS?

Where is it going next?

Where has it been?



The screenshot displays the 'ISS Tracker BOT' interface. On the left is a chat window with a black background and green text bubbles for bot responses and blue text bubbles for user questions. The chat history shows the following interactions:

- Bot: Latitude: -21.19
- Bot: Longitude: -141.5
- You: where is the ISS
- Bot: Unknown Location - Probably over the ocean.
- Bot: Latitude: -35.67
- Bot: Longitude: 50.32
- You: where is it going
- Bot: Showing Next 10 Mins
- You: where was it
- Bot: Unknown Location - Probably over the ocean.
- Bot: Latitude: -36.67
- Bot: Longitude: 51.69
- You: where is it now
- Bot: Unknown Location - Probably over the ocean.
- Bot: Latitude: 21.24
- Bot: Longitude: -20.56

At the bottom of the chat window is a 'Your Input:' field and a 'Send' button. To the right of the chat is a map of North America and the surrounding oceans. A small satellite icon representing the ISS is positioned in the Pacific Ocean, west of the United States coast. The map includes labels for 'United States', 'Mexico', 'Los Angeles', 'Phoenix', 'La Habana', 'Ciudad de México', 'Ciudad de Guatemala', and 'Paraguay'. A 'View Larger Map' link is visible at the bottom left of the map area, and a 'Leaflet | Map data © OpenStreetMap contributors' footer is at the bottom right.

Try a new question: show me in 3D

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

The screenshot displays the 'ISS Tracker BOT' interface. On the left is a chat window with a dark background and light text. The chat history shows several user questions and bot responses. The bot's responses include location information and coordinates for the ISS. At the bottom of the chat is an input field labeled 'Your Input:' and a 'Send' button. On the right is a 3D rendering of the Earth from space, with the International Space Station (ISS) visible as a small white object in the upper left quadrant. The background is a starry space. In the top right corner of the 3D view, there is a 'Tracking:' section with the following data: ISS (ZARYA) 423.2km 7.7km/s. At the top center of the 3D view, the text 'spacebar to pause/unpause' is visible.

ISS Tracker BOT

spacebar to pause/unpause

Tracking:
ISS (ZARYA) 423.2km 7.7km/s

You: where is it going

Bot: Showing Next 10 Mins

You: where was it

Bot: Unknown Location - Probably over the ocean.

Bot: Latitude: -36.67

Bot: Longitude: 51.69

You: where is it now

Bot: Unknown Location - Probably over the ocean.

Bot: Latitude: 21.24

Bot: Longitude: -20.56

You: where is ISS

Bot: Unknown Location - Probably over the ocean.

Bot: Latitude: 45.26

Bot: Longitude: -136.3

You: sho me in 3d

Bot: Mode Changed to 3d

Your Input:

Send



© Copyright IBM Corporation 2021.

The information contained in these materials is provided for informational purposes only and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.



Please Recycle
