Watson Assistant Weather Skill Using OpenWeather Free API

By Antonel Neculai

Here is how to build weather capabilities in your Watson Assistant chatbot: you need to create an action in IBM Functions to call the OpenWeather API. We will use Node.js to make the API call and to process the JSON response. The code will save the weather elements in a variable that is passed on to Watson Assistant when the Dialog node uses a webhook to call the action defined in the IBM Functions. Here is a simplified diagram:

WA dialog node<---- webhook call----> IBM Functions Action <----API call to OpenWeather----> OpenWeather

Remember, first you have to create an account on OpenWeather so you get an API_ID that you have to use when calling the OpenWeather API from the IBM Functions. Also, IBM Functions will generate for you a weblink to use as webhook in Watson Assistant.

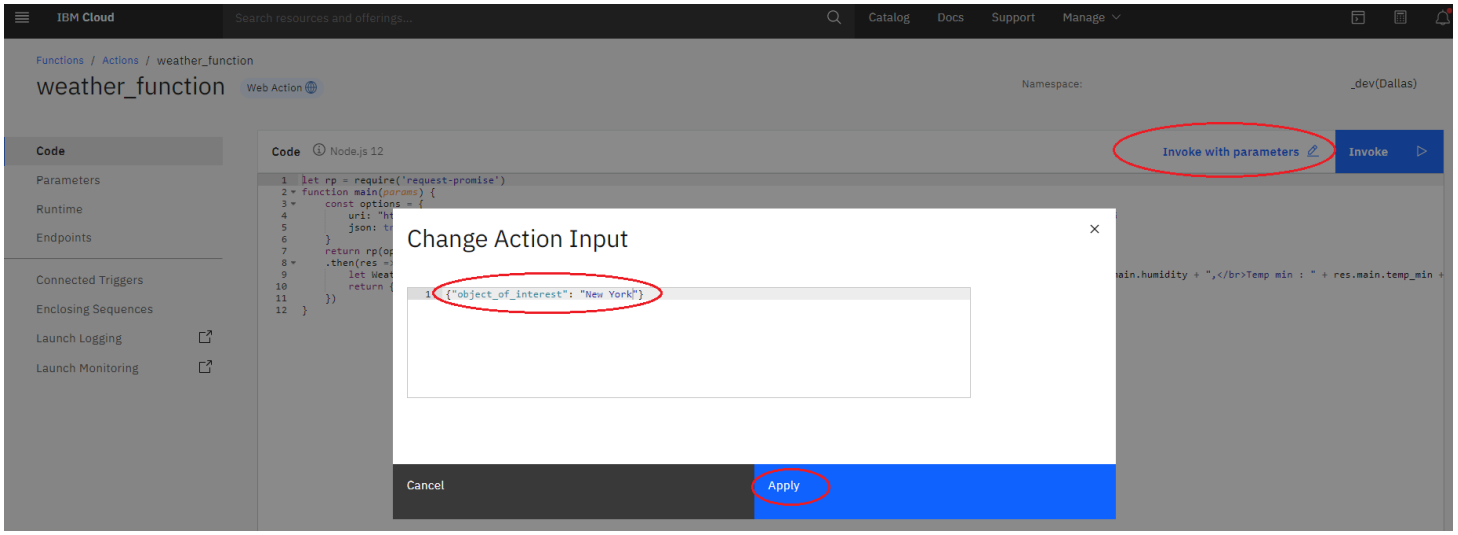
There is one more “trick” you need to be aware of: you will use the city you want the weather conditions for as a parameter. This way the city is not hard-coded in the chatbot and you can define multiple values for a weather entity, each value for a different city. This way the chatbot can tell the weather for different locations instead of just one.

So here is the Node.js code to use when creating the IBM Functions action (remember to replace the here_you_add_your_openweather_api_id with the actual openweather API_id you got when you created the account on that platform). I did not write the code, I found it as a question asked by the user AltShift on StackOverflow: https://stackoverflow.com/questions/68053601/calling-openweather-api-from-watson-assistant-direct-cloudfunctions-call-was-n I only tweaked the code a little on the return portion to make sure the answer is passed on as a message back to Watson Assistant.

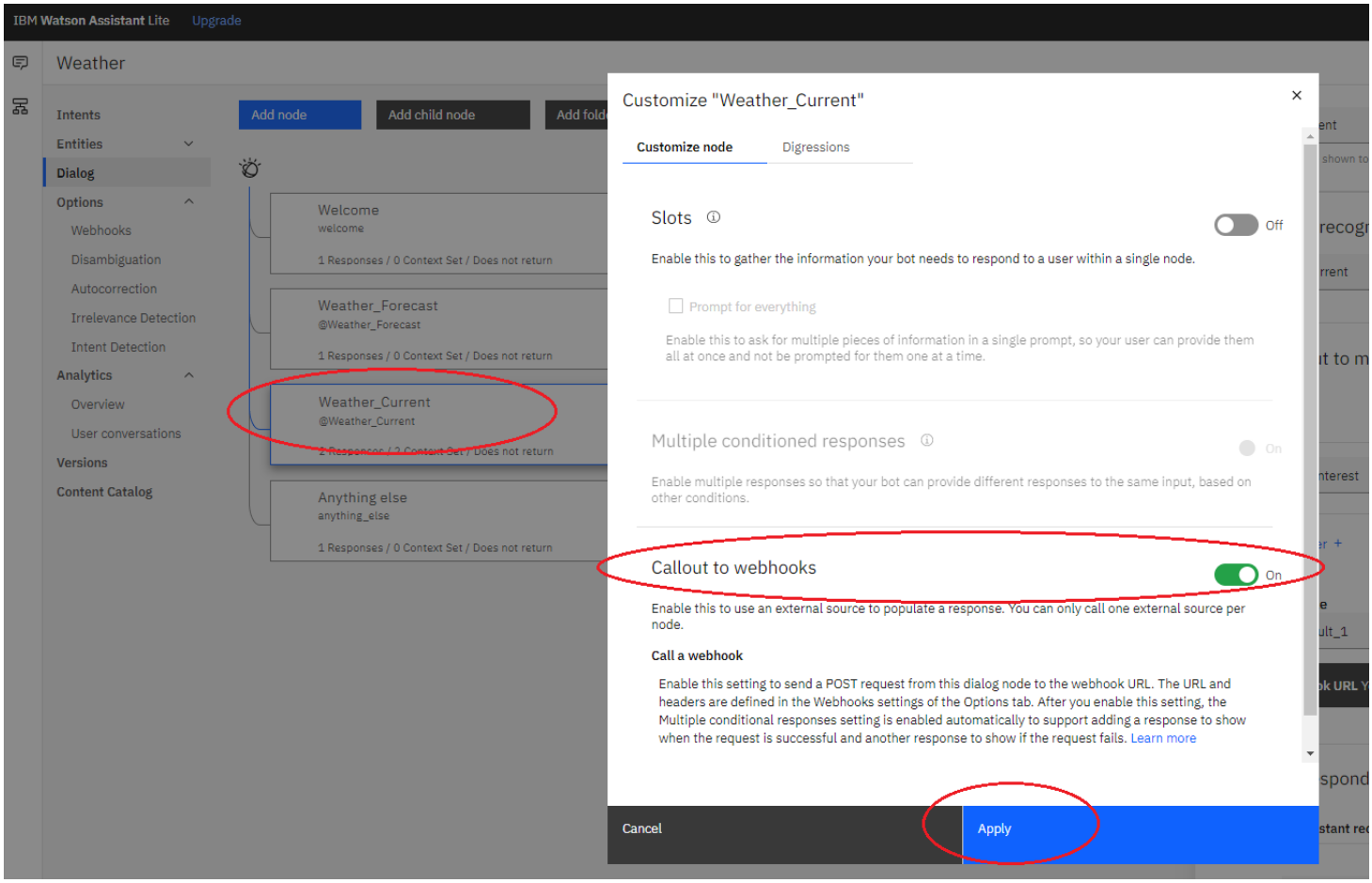
```
let rp = require('request-promise')

function main(params) {
  const options = {
    uri: "http://api.openweathermap.org/data/2.5/weather?q=" + encodeURIComponent(params.object_of_interest)+ "&units=imperial&APPID=here_you_add_your_openweather_api_id" ,
    json: true
  }
  return rp(options)
  .then(res => {
    let WeatherReport = "Current Temperature : " + res.main.temp + ",</br>Pressure : " + res.main.pressure + ",</br>Humidity : " + res.main.humidity + ",</br>Temp min : " + res.main.temp_min + " ,</br>Temp max : " + res.main.temp_max
    return {message: WeatherReport }
  })
}
```

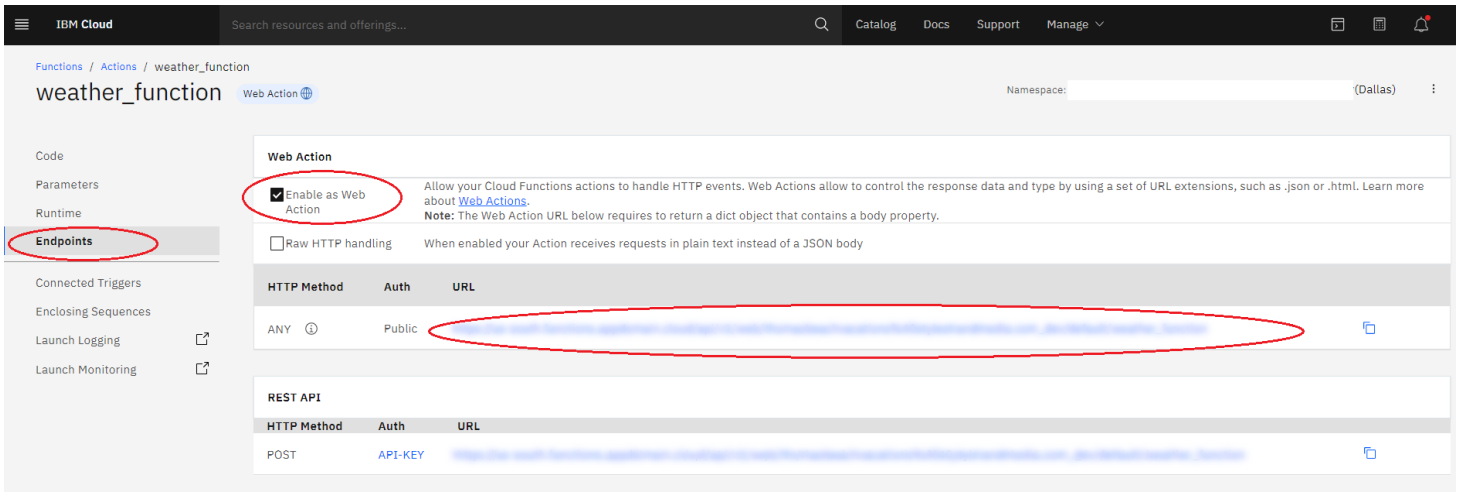
If you want to invoke the action, you need to do it with “Invoke with parameters” where you add the parameter for the city, otherwise you get an error. This is the parameter to use to invoke the function:



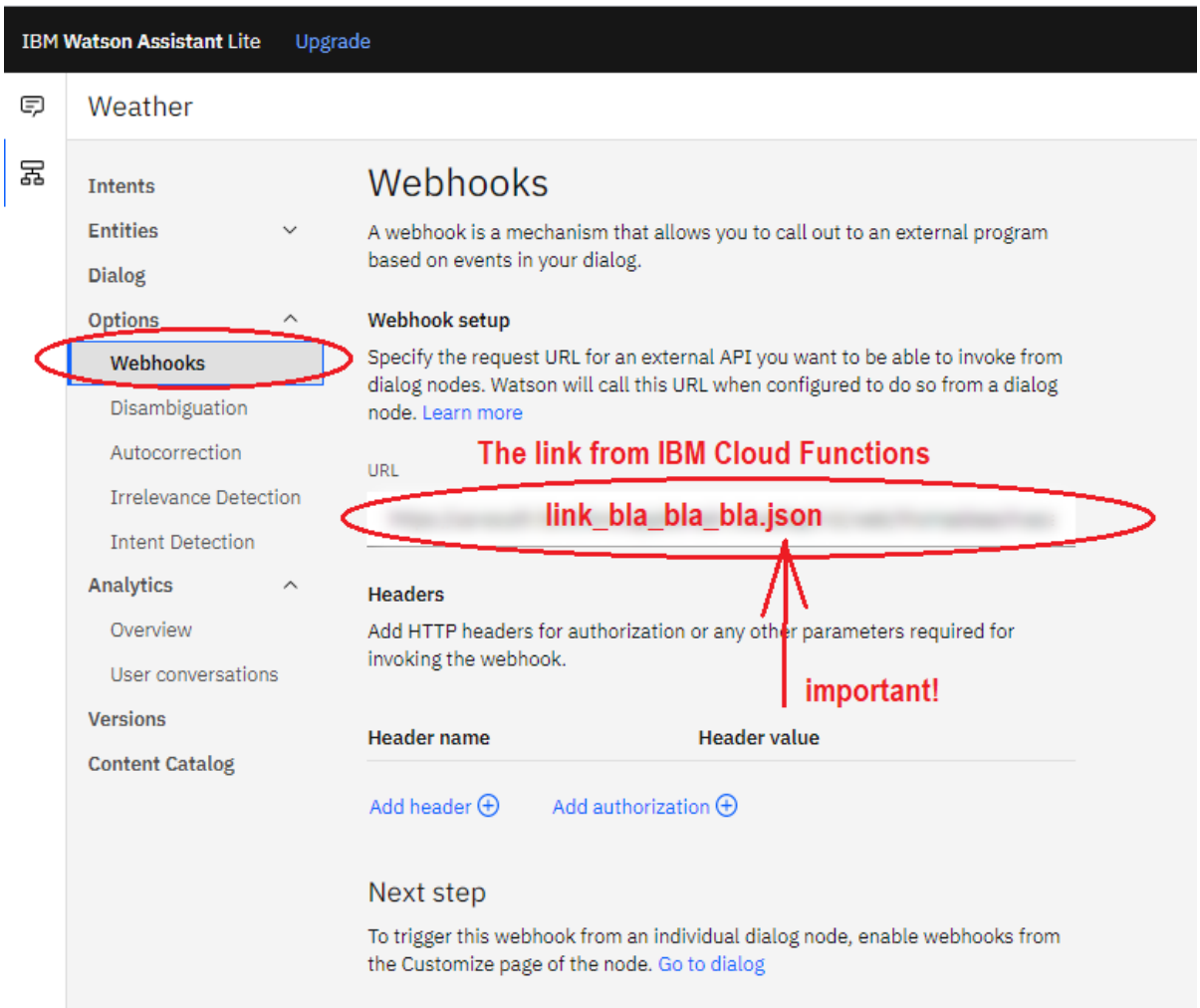
Once the function is created, go to Watson Assistant and create a dialog node for the weather and make sure to select the options gear icon to enable the webhooks (don't forget to click "Apply"):



You will use as webhook the url link you got from the IBM Functions. On IBM Cloud Functions you find the link under “Endpoint” and make sure you check the box for “Enable as web action”. The link you need to copy is the url under “HTTP Method” :



Now that you know where to get the webhook url link from, add it to Watson Assistant as a webhook. Make sure you manually add .json at the end of the link, to tell Watson Assistant to expect a json file as a response, otherwise you will get an error:



The last thing you need to do at this point is to define the elements for the weather node, as following:

- under “If assistant recognizes” area add the weather intent or entity you created for the dialog

- add a parameter when you call out the webhook – define a key called “object_of_interest” which is the city you want the weather for. The value is the name of the city

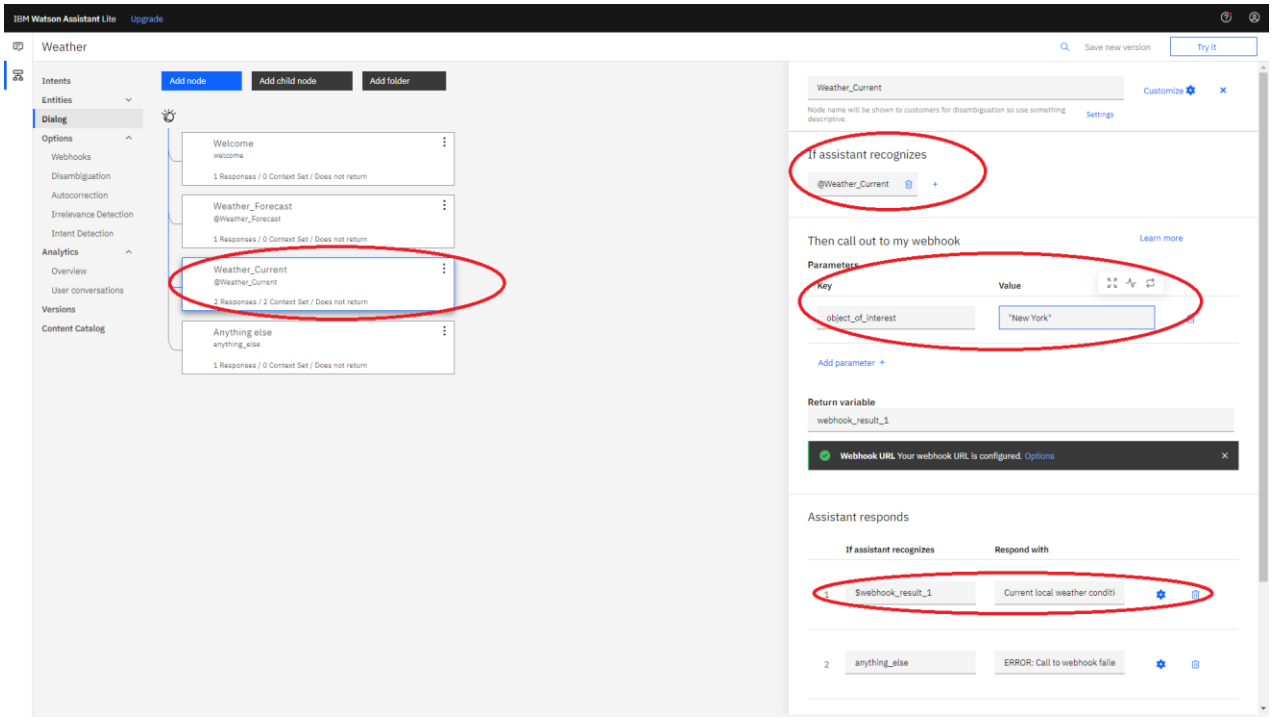
- under “Assistant Responds” area add under “If assistant recognizes” the default return variable presented by Watson Assistant, which usually is $webhook_result_1. This way the weather details will be displayed by the chatbot if the response has been returned by the IBM Cloud Functions (which means that it successfully called the OpenWeather API and got back the answer with the weather details).

- The “Respond with” is:

Current local weather conditions: </br><? $webhook_result_1.message ?>

If you know HTML< you understand why we add the </br> tag: to beautify the answer a little. We also used HTML tags in the IBM Cloud Functions code on the response section to make the weather details returned by OpenWeather API look nicer, listed one under the other.

Here are the settings for the weather dialog node:



And here is the answer offered by the chatbot when we try it: