

WEATHER SERVICE

Prepared By: Shantanu Ranjan

Summary:

This service allows user to display past weeks service based on location. User Interface has been developed in Angularjs while making third party RestFul Call through Express/NodeJs. Third Party Service call has been hosted on Google Cloud engine and is located at:

To Get last weeks weather based on default location:

<https://weatherapp-174405.appspot.com/weather> - Default City: New York

To Get last weeks weather based on location:

<https://weatherapp-174405.appspot.com/weather?city=San Diego>

Third Party Service Call

There are many third party apis that expose weather related service to public. This module uses

<http://api.openweathermap.org>

Please use the following link to setup your account and generate appid key in order to make third party api call.

<https://openweathermap.org/appid>

Once signed up, you can generate appid key in your dashboard:

The screenshot shows the OpenWeatherMap API keys dashboard. At the top, there's a navigation bar with the OpenWeatherMap logo and links for Weather, Maps, API, Price, Partners, Stations, Widgets, News, and About. Below this, the 'API keys' section is active, showing a list of existing keys and a 'Create key' form. The list of keys includes a 'Default' key and a key for 'api.openweathermap.org'. The 'Create key' form has a 'Name' field and a 'Generate' button. A blue notification box states: 'Activation of an API key for Free and Startup accounts takes 10 minutes. For other accounts it takes from 10 to 60 minutes. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.'

Key	Name	Create key
acfcf37ae83b7a590e1e95de88d9f678	Default	<input type="checkbox"/> <input type="checkbox"/>
529f851338c27706f843e2cca4ff3d30	api.openweathermap.org	<input type="checkbox"/> <input type="checkbox"/>

Google Cloud Engine:

Instructions:

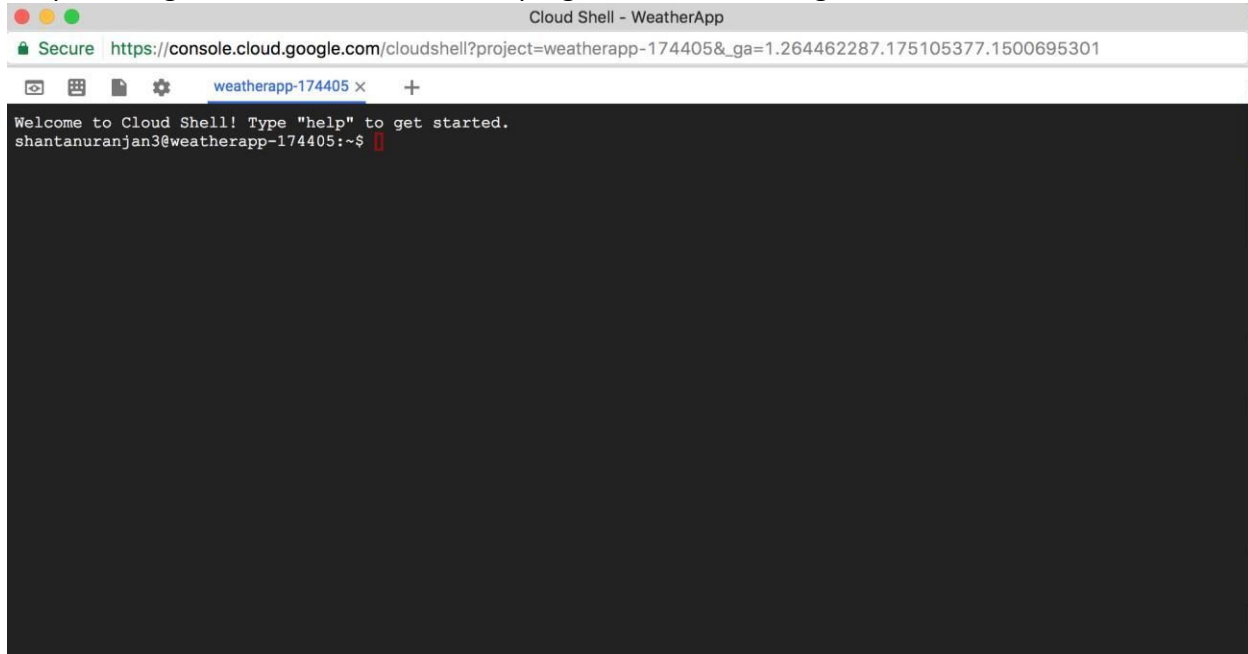
1. Before deploying code at Google Cloud Engine, first setup the environment for App Engine using the following link:

<https://cloud.google.com/nodejs/getting-started/hello-world>

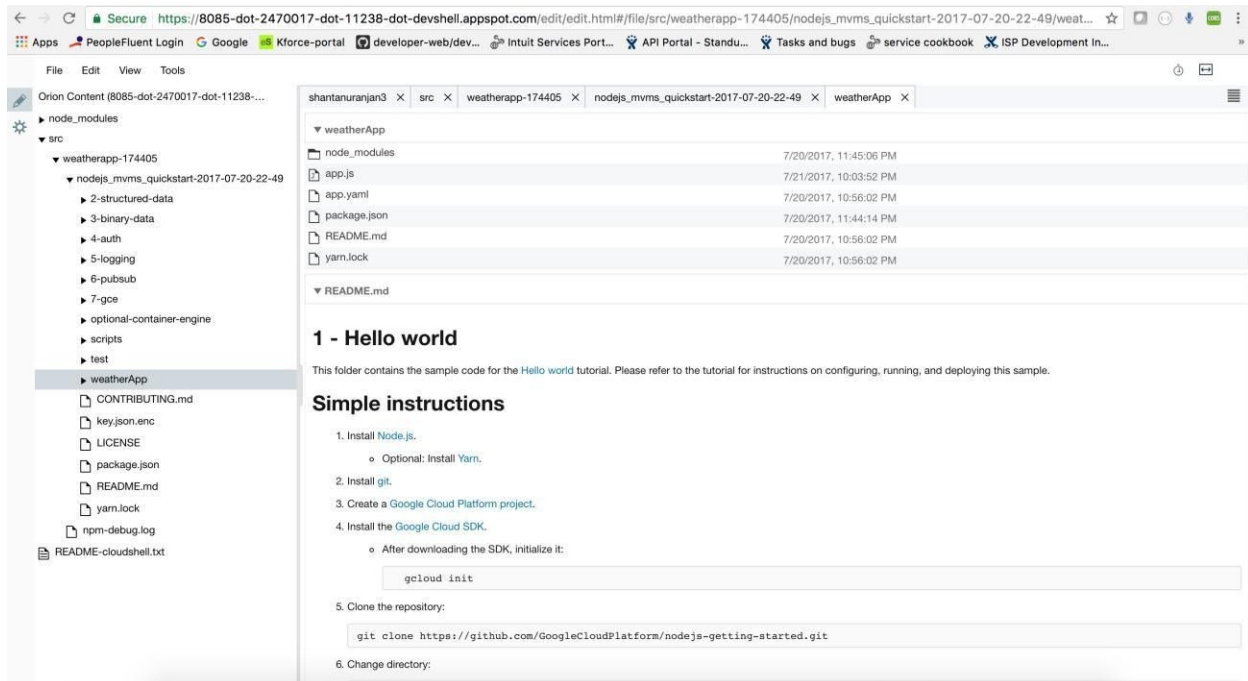
2. Open your App Engine Dashboard



3. Open Google cloud shell located at top right corner with >-sign



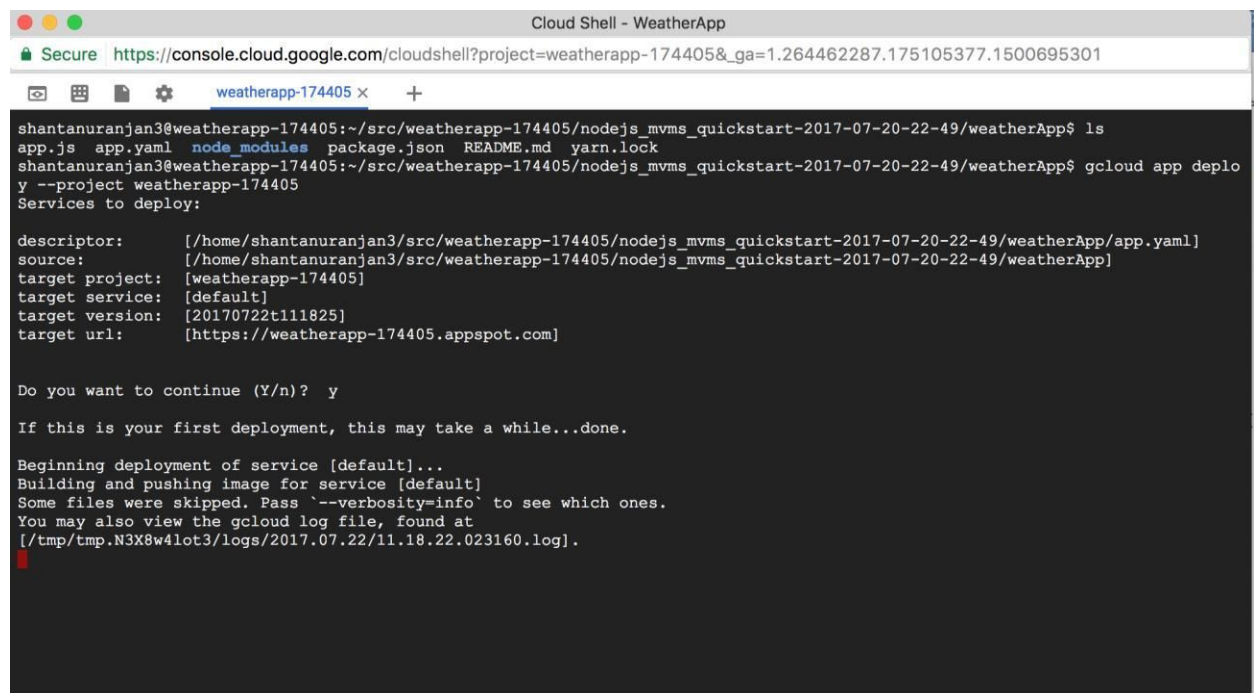
4. Click on Files -> Launch Code Editor



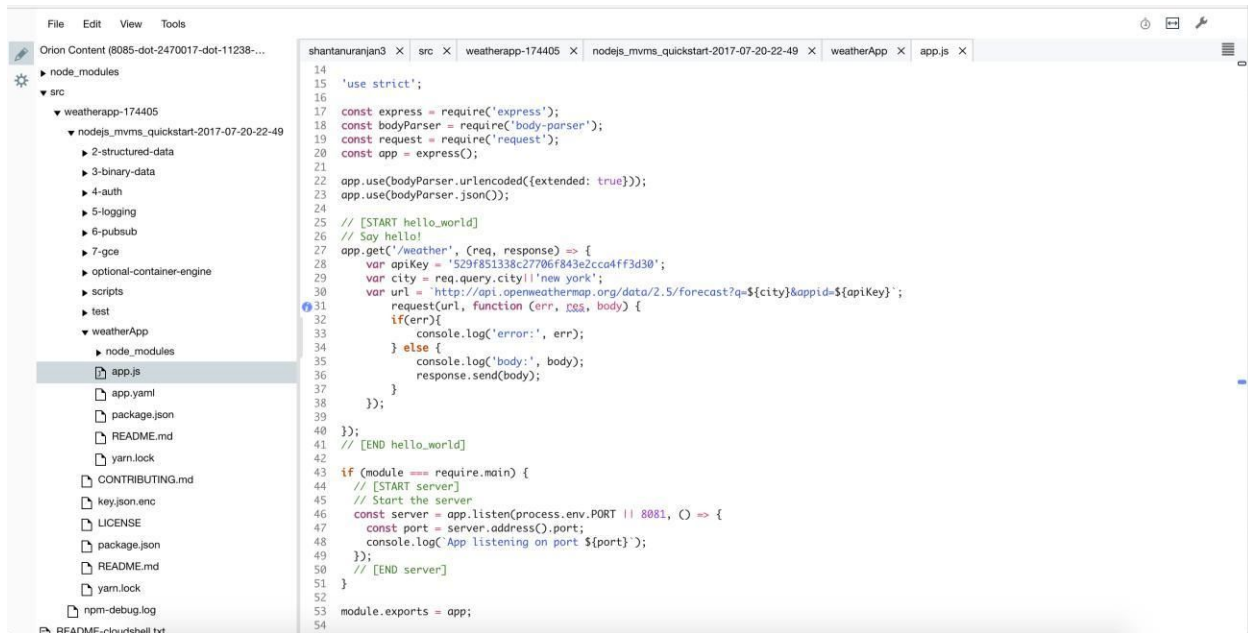
5. You can deploy your code in code editor. To deploy use the following command through cloud shell:

`gcloud app deploy --project <<project-name>>` eg: `weatherapp-174405`

`gcloud app browse` eg: <https://weatherapp-174405.appspot.com/weather>



6. Server Making Third party api call to fetch weather related information



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'weatherapp-174405' containing a 'src' directory. The 'src' directory contains a file named 'app.js'. The code editor shows the contents of 'app.js', which is a Node.js application that makes a third-party API call to fetch weather information.

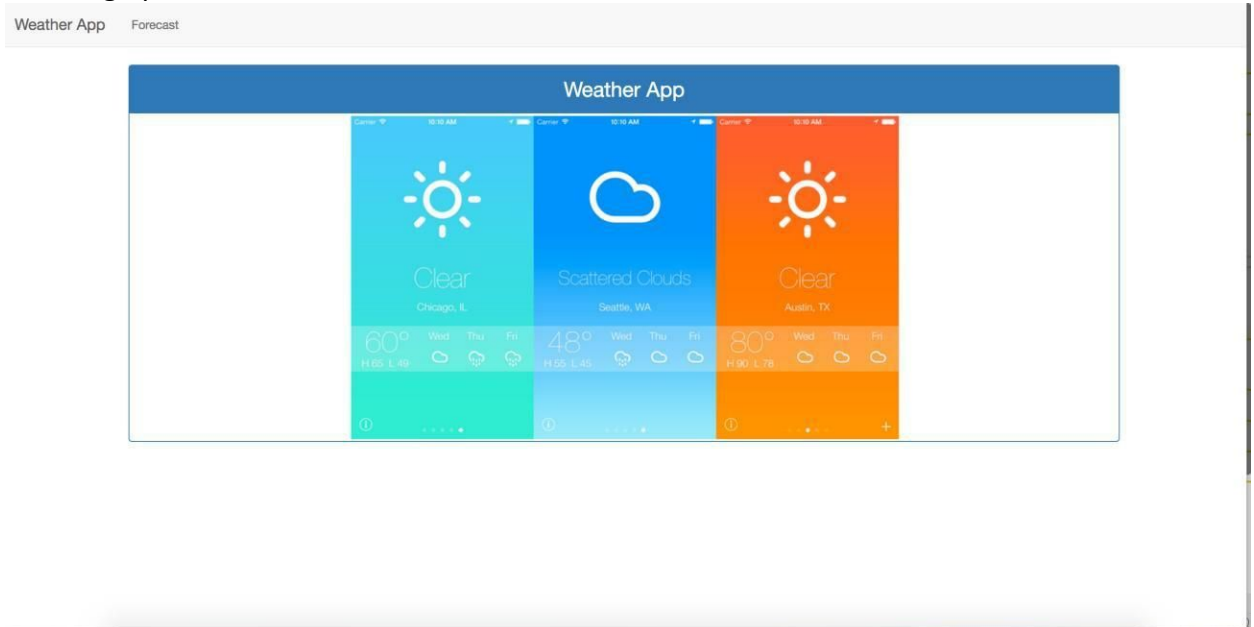
```
14 'use strict';
15
16
17 const express = require('express');
18 const bodyParser = require('body-parser');
19 const request = require('request');
20 const app = express();
21
22 app.use(bodyParser.urlencoded({extended: true}));
23 app.use(bodyParser.json());
24
25 // [START hello_world]
26 // Say hello!
27 app.get('/weather', (req, response) => {
28   var apiKey = '529f851338c27706f843e2cca4ff3d30';
29   var city = req.query.city || 'new york';
30   var url = 'http://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${apiKey}';
31   request(url, function (err, res, body) {
32     if(err){
33       console.log('error:', err);
34     } else {
35       console.log('body:', body);
36       response.send(body);
37     }
38   });
39 });
40 // [END hello_world]
41
42 if (module === require.main) {
43   // [START server]
44   // Start the server
45   const server = app.listen(process.env.PORT || 8081, () => {
46     const port = server.address().port;
47     console.log('App listening on port ${port}');
48   });
49   // [END server]
50 }
51
52 module.exports = app;
```

Here use the value of appKey generated from third party api **openweathermap.org**

Client CodeBase:

1.Copy the folder and extract it on your local machine:

2. Go to application path
and run npm install
3. use “gulp connect” to launch it on browser



Hamburg San Francisco Berlin Athens Tokyo New York Moscow

New York, US Lon: -74.01 Lat: 40.71

Friday, July 21, 2017

 C Clouds

broken clouds High: 296.29 C Low: 293.37 C

Day: C Night: C Pressure: 1020.38 hPa Humidity: 87%

Saturday, July 22, 2017

 C Clear

clear sky High: 294.264 C Low: 294.264 C

Day: C Night: C Pressure: 1024.94 hPa Humidity: 85%

Sunday, July 23, 2017

 C Rain

moderate rain High: 295.146 C Low: 295.146 C

Day: C Night: C Pressure: 1014.1 hPa Humidity: 100%

Monday, July 24, 2017

 C Rain

light rain High: 293.292 C Low: 293.292 C

Day: C Night: C Pressure: 1016.45 hPa Humidity: 100%

Tuesday, July 25, 2017

 C Clear