1.2 NodeJS: HTTP



This section will guide you to:

* Create a JavaScript project in your IDE
* Write a program in NodeJS to show geocoding

This lab has five subsections, namely:

1.2.1 Writing a program in NodeJS to take the address and convert it into a latitude-longitude pair

1.2.2 Setting up an error handler for the above program

1.2.3 Modifying the geocoding program using callbacks

1.2.4 Modifying the geocoding program to implement callback chaining and accept location via command line argument

1.2.5 Pushing the code to your GitHub repositories

*NodeJS and Visual Studio Code are already installed in your lab. (Refer MEAN: Lab Guide - Phase 3)*

**Step 1.2.1:** Writing a program in NodeJS to take an address and convert it into a latitude-longitude pair

* Go to [https://www.mapbox.com](https://www.mapbox.com/) and create an account
* Click on *Map web services*
* You will find the URL for forward geocoding under **example request for forward geocoding**
* You need to modify the URL according to the desired city
* Go to <https://darksky.net/dev> and create an account
* Once you log in, you will find *Sample API Call*. You need to use that URL in your program to find the weather of a place
* Open Visual Studio Code. Open the **weather** folder
* Create a file named geocoding.js in your directory named weather
* Using Visual Studio Code, add the given code by resolving the warnings and errors due to compatibility-related issues

const request = require('request')

const url = 'https://api.darksky.net/forecast/0f117a9db5a9407fd73d76e7555804b1/37.8267,-122.4233'

request({ url: url, json: true }, (error, response) => {

console.log(response.body.daily.data[0].summary + ' It is currently ' + response.body.currently.temperature + ' degrees out. There is a ' + response.body.currently.precipProbability + '% chance of rain.')

})

const geocodeURL = 'https://api.mapbox.com/geocoding/v5/mapbox.places/Bengaluru.json?access\_token=pk.eyJ1Ijoic2hhbGluaTk4IiwiYSI6ImNqejRsZDVlZDAybzEzZW56NXpxYTRwdG0ifQ.V\_Oz9BJ972Q9Qvh9wMnpQQ&limit=1'

request({ url: geocodeURL, json: true }, (error, response) => {

const latitude = response.body.features[0].center[0]

const longitude = response.body.features[0].center[1]

console.log(latitude, longitude)

})

* *Run the following command:*

*node geocoding.js*

**Output:**

*77.59796 12.96991*

* *Partly cloudy throughout the day. It is currently 66.94 degrees out. There is a 0% chance of rain.*

**Step 1.2.2:** Setting up an error handler for the above program

* Add the code shown below by resolving the warnings and errors due compatibility-related issues

const request = require('request')

const url = 'https://api.darksky.net/forecast/0f117a9db5a9407fd73d76e7555804b1/37.8267,-122.4233'

request({ url: url, json: true }, (error, response) => {

if(error){

console.log('Network not available');

}

else{

console.log(response.body.daily.data[0].summary + ' It is currently ' + response.body.currently.temperature + ' degrees out. There is a ' + response.body.currently.precipProbability + '% chance of rain.')

}

})

const geocodeURL = 'https://api.mapbox.com/geocoding/v5/mapbox.places/Bengaluru.json?access\_token=pk.eyJ1Ijoic2hhbGluaTk4IiwiYSI6ImNqejRsZDVlZDAybzEzZW56NXpxYTRwdG0ifQ.V\_Oz9BJ972Q9Qvh9wMnpQQ&limit=1'

request({ url: geocodeURL, json: true }, (error, response) => {

if(error){

console.log('Network not available');

}

else{

const latitude = response.body.features[0].center[0]

const longitude = response.body.features[0].center[1]

console.log(latitude, longitude)

}

})

* Turn off your network, and run the following command:

*node geocoding.js*

* **Output:**

*Network not available*

**Step 1.2.3:** Modifying the geocoding program using callbacks

* Create a folder named utils
* Add a file named geocode.js in it
* Add the code shown below in geocode.js by resolving the warnings and errors due to compatibility-related issues

const request = require('request')

const geocode = (address, callback) => {

const url = 'https://api.mapbox.com/geocoding/v5/mapbox.places/' + address + '.json?access\_token=pk.eyJ1Ijoic2hhbGluaTk4IiwiYSI6ImNqejRsZDVlZDAybzEzZW56NXpxYTRwdG0ifQ.V\_Oz9BJ972Q9Qvh9wMnpQQ&limit=1'

request({ url: url, json: true }, (error, response) => {

if (error) {

callback('Unable to connect to location services!', undefined)

} else if (response.body.features.length === 0) {

callback('Unable to find location. Try another search.', undefined)

} else {

callback(undefined, {

latitude: response.body.features[0].center[0],

longitude: response.body.features[0].center[1],

location: response.body.features[0].place\_name

})

}

})

}

module.exports = geocode

* Add the code shown below in app.js

const request = require('request')

const geocode = require('./utils/geocode')

geocode('Kolkata', (error, data) => {

console.log('Error', error)

console.log('Data', data)

})

* Run the following command:

*node app.js*

* **Output:**

*Error undefined*

*Data { latitude: 88.33778,*

*longitude: 22.54111,*

*location: 'Kolkata, West Bengal, India' }*

**Step 1.2.4:** Modifying the geocoding program to implement callback chaining and accept location via command line argument

* Add the following code in geocode.js

const request = require('request')

const geocode = (address, callback) => {

const url = 'https://api.mapbox.com/geocoding/v5/mapbox.places/' + address + '.json?access\_token=pk.eyJ1Ijoic2hhbGluaTk4IiwiYSI6ImNqejRsZDVlZDAybzEzZW56NXpxYTRwdG0ifQ.V\_Oz9BJ972Q9Qvh9wMnpQQ&limit=1'

request({ url: url, json: true }, (error, response) => {

if (error) {

callback('Unable to connect to location services!', undefined)

} else if (response.body.features.length === 0) {

callback('Unable to find location. Try another search.', undefined)

} else {

callback(undefined, {

latitude: response.body.features[0].center[1],

longitude: response.body.features[0].center[0],

location: response.body.features[0].place\_name

})

}

})

}

module.exports = geocode

* Add a file in utils named forecast.js and add the following code in it:

const request = require('request')

const forecast = (latitude, longitude, callback) => {

const url = 'https://api.darksky.net/forecast/0f117a9db5a9407fd73d76e7555804b1/' + latitude + ',' + longitude

request({ url: url, json: true }, (error, response) => {

if (error) {

callback('Unable to connect to weather service!', undefined)

} else if (response.body.error) {

callback('Unable to find location', undefined)

} else {

callback(undefined, response.body.daily.data[0].summary + ' It is currently ' + response.body.currently.temperature + ' degrees out. There is a ' + response.body.currently.precipProbability + '% chance of rain.')

}

})

}

module.exports = forecast

* Add the following code in app.js:

const geocode = require('./utils/geocode')

const forecast = require('./utils/forecast')

const address = process.argv[2]

if (!address) {

console.log('Please provide an address')

} else {

geocode(address, (error, data) => {

if (error) {

return console.log(error)

}

forecast(data.latitude, data.longitude, (error, forecastData) => {

if (error) {

return console.log(error)

}

console.log(data.location)

console.log(forecastData)

})

})

}

* Run the following command:

*node app.js Kolkata*

* **Output:**

*Kolkata, West Bengal, India*

*Rain and humid throughout the day. It is currently 86.1 degrees out. There is a 0.25% chance of rain.*

**Step 1.2.5:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master