**Source Code snippet scrapping from Web page using NodeJS**

**Objective**: The objective of this project is to develop a chrome extension plugin which should scan for code-snippets from any active URL opened in the Chrome browser and should store them in MySQL database.

**Requirement:**

1. Need to develop Chrome Browser extension plugin, which on-click should find the source code snippet from the active URL in the Web browser and store the code snippet to MySQL database.
2. Incase if the URL is not valid, it should display the message saying invalid URL
3. Incase if the URL don’t have any code snippets, it should display the message saying no code snippets found and no record should be inserted to the MySQL database.

**GitHub link:**

<https://github.com/shyamnarayan2001/NodeJS_Projects.git>

**Installation Steps:**

1. Setup all the required pre-requisite softwares (as mentioned under the section **Pre-requisites**)
2. Git clone the project (For example, in my case, under D:\Projects\)
3. Once the clone is complete, execute the DB scripts available in D:\Projects\NodeJS\_Projects\SourceCodeFind\db\_scripts\**db\_scripts.sql** in your local MySQL database.

**CREATE DATABASE nodemysql;**

**USE nodemysql;**

**CREATE TABLE `code\_capture` (**

**`id` int(11) NOT NULL AUTO\_INCREMENT,**

**`url\_scanned` varchar(300) DEFAULT NULL,**

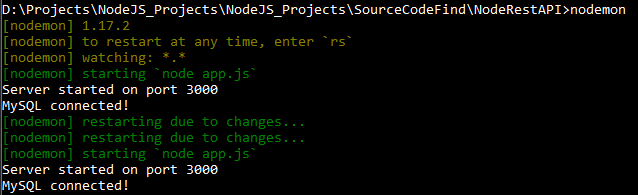
**`code\_snippet` varchar(1000) DEFAULT NULL,**

**`last\_updated\_dt` datetime DEFAULT NULL,**

**PRIMARY KEY (`id`)**

**) ENGINE=InnoDB AUTO\_INCREMENT=68 DEFAULT CHARSET=latin1;**

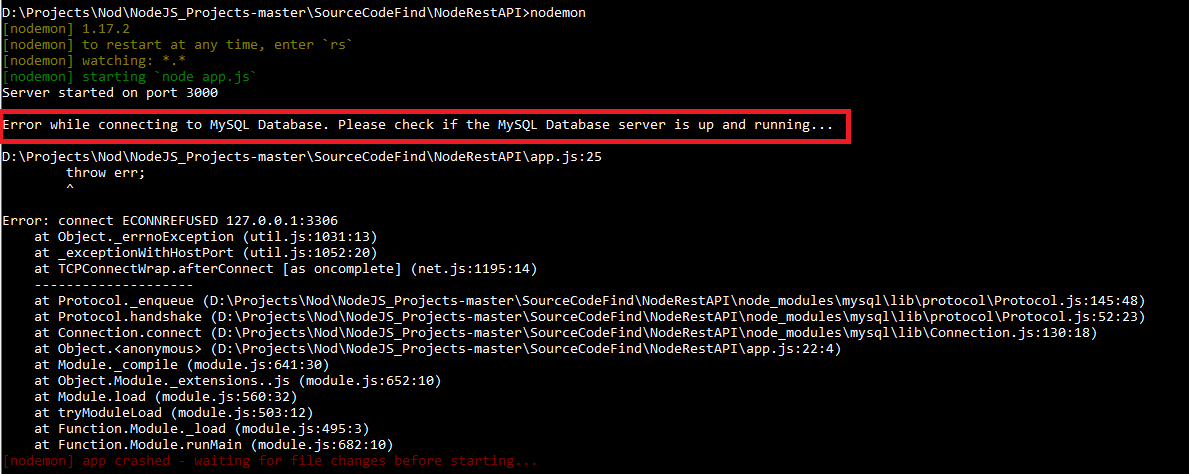
1. Now open the command prompt and cd to path D:\Projects\NodeJS\_Projects\SourceCodeFind\NodeRestAPI and execute the command "npm install"
2. Now execute the command “nodemon”



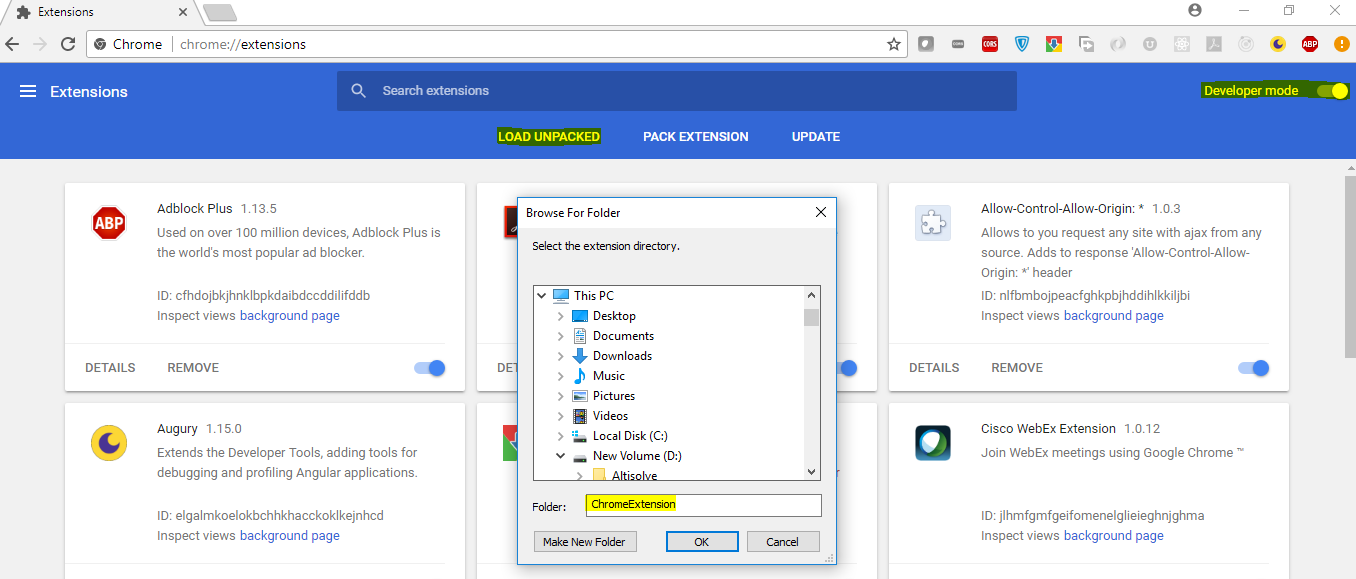
Note: Please don’t close this. This is the Server code which should be always running. Incase if you want to stop this, you can use CTRL + C and close the window.

Error scenario:

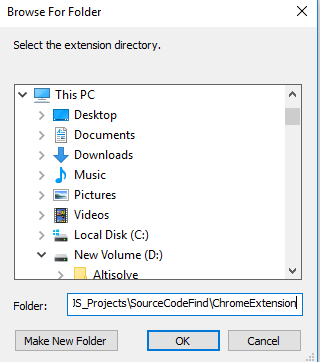
Incase if you get the below error, it means that MySQL server is not up and running. Please start the MySQL and run the command “nodemon”.



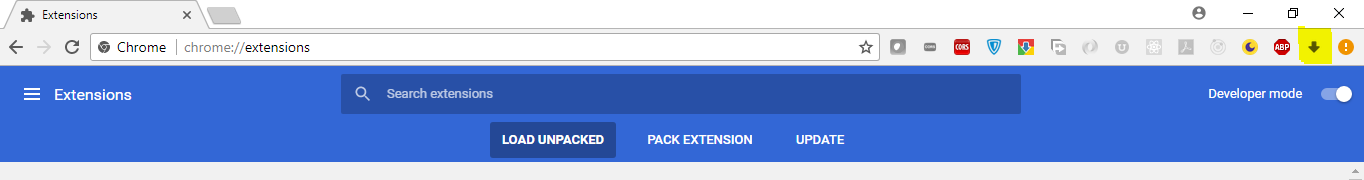
1. Now install the Chrome Extension plugin. For this, open Chrome browser and type “chrome://extensions” in tab, enable the “Developer mode” on the left hand top, click “LOAD UNPACKED”. It will open the “Browse For Folder” option (as shown in the below screen shot) and enter the path D:\Projects\NodeJS\_Projects\SourceCodeFind\ChromeExtension and click Ok button.



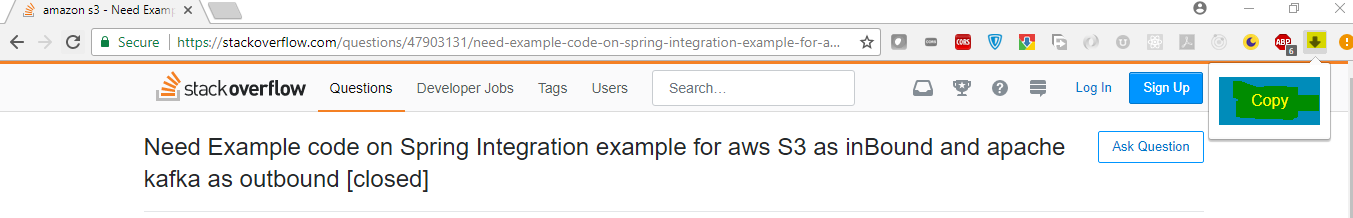
.

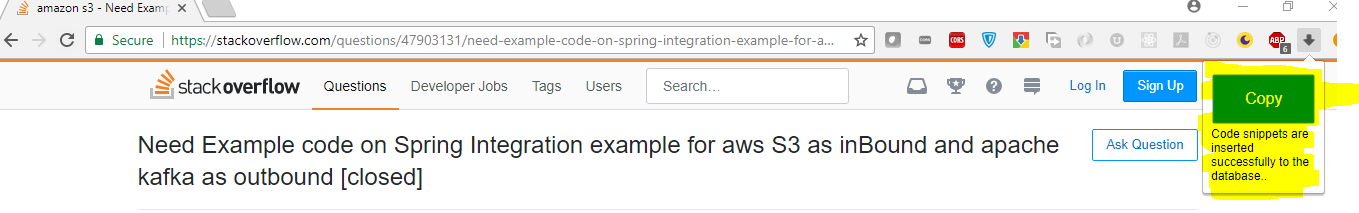


Now you will see the extension added and you will see a blue icon added on the left hand side (as highlighted in yellow).



Now you can open any valid URL in the browser where you can find code snippet (like stackoverflow page) and click this icon (highlighted in yellow), you will see the “Copy” button. Now click this “Copy” button, it will display “Code snippets are inserted successfully to the database”. Note that the Server component in Step 5 is running for this to be working.





You can query the MYSQL database table (**code\_capture)** to see a latest entry, where you can see the following values:

Id: latest auto increment no

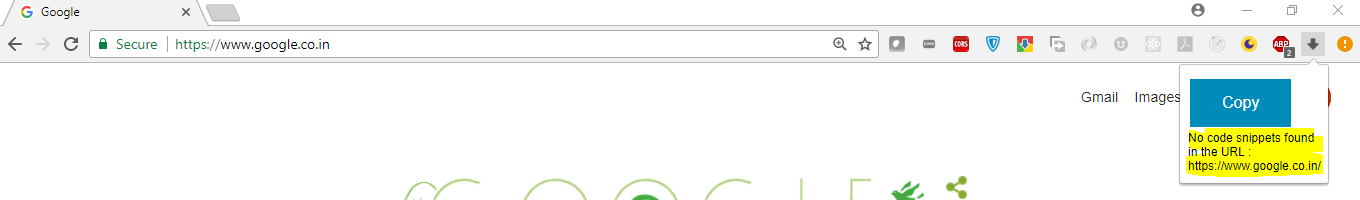
url\_scanned: the URL active in the browser

code\_snippet: The Code snippet from the URL

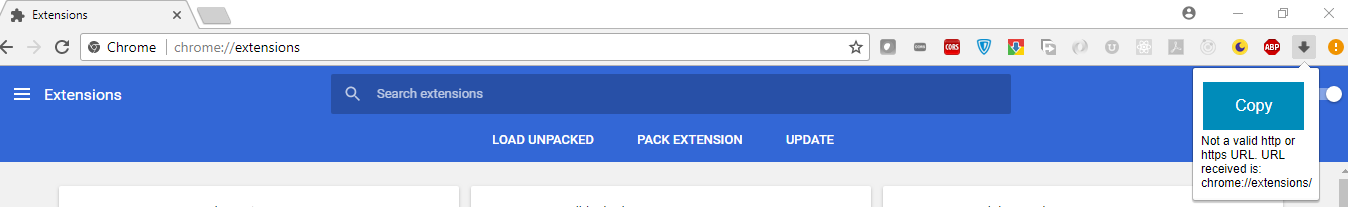
last\_update\_dt: The date time stamp of the insert

Negative scenarios:

1. If you try to hit a URL where you find any code snippet (For ex: [www.google.com](http://www.google.com)), you will receive the message saying – “No code snippets found in the URL: <URL>”.



1. If you try to hit an invalid http (or https) URL (for ex: chrome://extensions/), then you will receive the message saying – “Not a valid http or https URL. URL received is: <URL>”.



**Pre-requisites:**

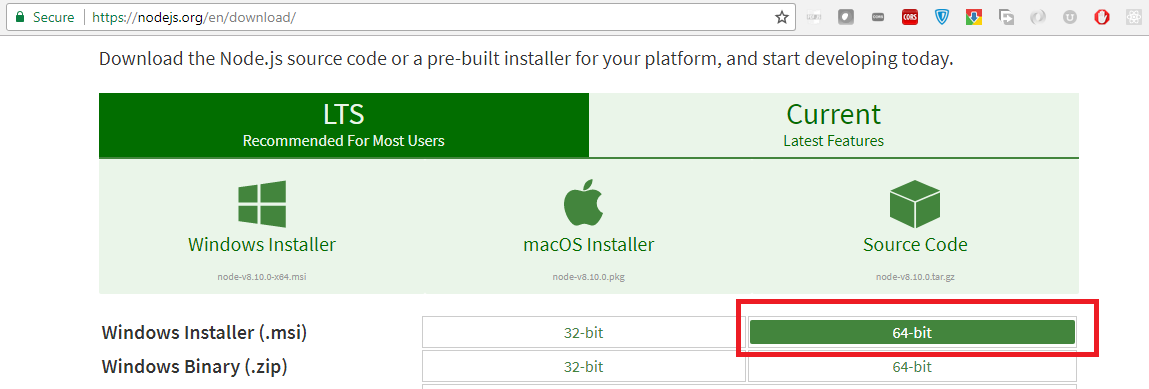
**1. Node JS:**

The pre-requisite for this project is to install NodeJS in your laptop. So let’s now install Node JS.

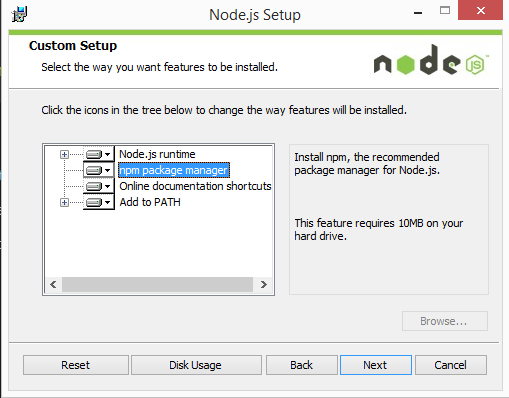
Note: If NodeJS is already installed on your machine, you can skip this step and move to “**Node and NPM upgrade on Windows”** section of this document, to ensure you have the latest npm version installed.

**Installation Steps:**

* Download the Windows installer 64 bit from the Nodes.js® web site - <https://nodejs.org/en/download/>



* Run the installer (the .msi file you downloaded in the previous step.)
* Follow the prompts in the installer (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).

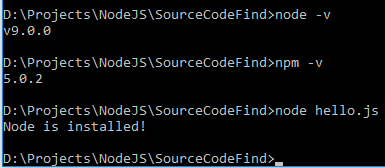


* Restart your computer. You won’t be able to run Node.js® until you restart your computer.

**Test the installation:**

Make sure you have Node and NPM installed by running simple commands to see what version of each is installed and to run a simple test program:

* **Test Node:** To see if Node is installed, open the Windows Command Prompt, Powershell or a similar command line tool, and type node -v. This should print a version number, so you’ll see something like this v6.10.2.
* **Test NPM:** To see if NPM is installed, type npm -v in Terminal. This should print NPM’s version number so you’ll see something like this 5.4.2
* **Create a test file and run it:** A simple way to test that node.js works is to create a JavaScript file: name it hello.js, and just add the code **console.log('Node is installed!');.** To run the code simply open your command line program, navigate to the folder where you save the file and type node hello.js. This will start Node and run the code in the hello.js file. You should see the output Node is installed!.



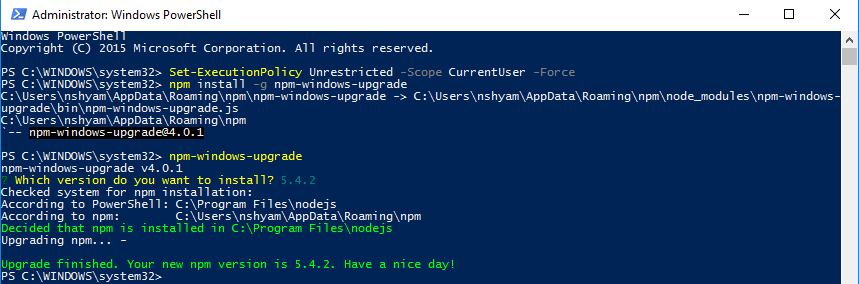
**Node and NPM upgrade on Windows:**

1. Run PowerShell as Administrator
2. Run the following commands in the PowerShell to upgrade

**Set-ExecutionPolicy Unrestricted -Scope CurrentUser -Force**

**npm install -g npm-windows-upgrade**

**npm-windows-upgrade**



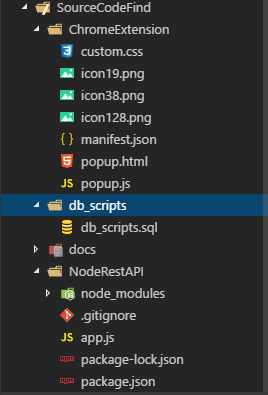
**Explanation of the Source Code:**

**Project Structure:**

1. The below is the project structure for this implementation:

**\ChromeExtension** – holds the Chrome Extension related code

**\db\_scripts** – holds the db\_scripts.sql file which has the database script to create database and table.  
**\NodeRestAPI** – holds the Server side Node JS code, which will have the logic to persist the code-snippet details to the MySQL table via REST apis.



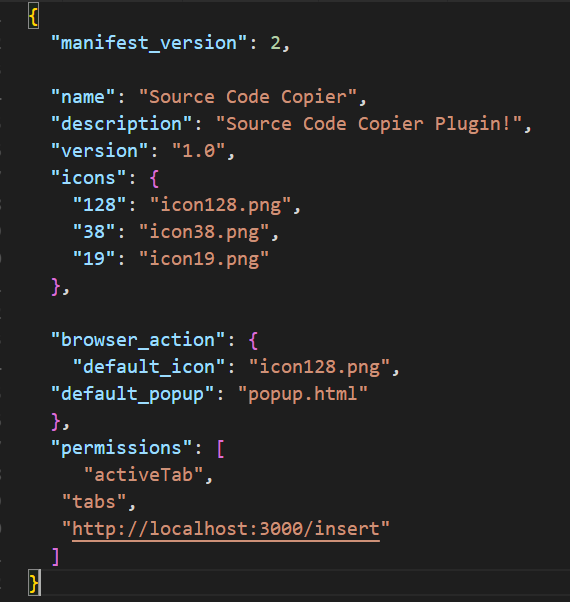
1. **SourceCodeFind\ChromeExtension\manifest.json:**

Chrome Extension needs manifest.json, which holds the metadata details like:

**Icons to be used** – icon128.png, icon38.png, icon19.png

**Browser\_action** – with **default\_popup** value having the html file to be invoked.

**Permission** – holding the Permission details to tell only activeTab to be considered, allow permission to invoke the Node JS API via the URL http://localhost:3000/insert



icon19.png, icon38.png, icon128.png are the icons of various sizes used.

1. **SourceCodeFind\ChromeExtension\popup.html:**

It has logic to display just a button named “Copy” and initialize the javascript file (popup.js) and CSS file (custom.css).



1. **SourceCodeFind\ChromeExtension\popup.js**

On-click of the copy button, the addEventListner() function will be invoked, which will inturn invoke the function triggerCopy(). This triggerCopy() function will fetch the active URL from the ChromeTab, invokes the REST API POST URL <http://localhost:3000/insert> via fetch API. The Fetch API call will accept parameters like:

Method: POST

Headers: for Accept and Content-Type

Body: URL value fetched from Chrome tab

Once the Fetch API response is received, it will replace the div id “output” (defined in popup.html) using innerHTML() method.

document.getElementById('copy').addEventListener('click', triggerCopy);

function triggerCopy(){

    let url = '';

    //Fetch the URL from Chrome Tab

    chrome.tabs.query({currentWindow: true, active: true}, function(tabs){

        url = tabs[0].url;

        console.log(tabs[0].url);

        console.log("Incoming url ="+url);

        //Invoke the POST API call for /insert with valid headers and pass the URL as request body

        fetch("http://localhost:3000/insert", {

        method: "POST",

        headers: {

            'Accept': 'application/json, text/plain, \*/\*',

            'Content-Type': 'application/json'

        },

        //JSON Stringify the URL

        body: JSON.stringify({

            url: url

        })

    })

    .then((res) => res.text())

    .then((responseFromServer) => {

        console.log("Response from Server="+responseFromServer);

        //Replace the div id "output" (in popup.html) with response from server (using innerHTML)

        document.getElementById('output').innerHTML = responseFromServer;

    });

    });

}

1. **SourceCodeFind\ChromeExtension\custom.css**:

.button will apply the css attributes for button

.div.sansserif will apply the font attributes for the text displayed below the “Copy” button.



1. **SourceCodeFind\db\_scripts\db\_scripts.sql:**

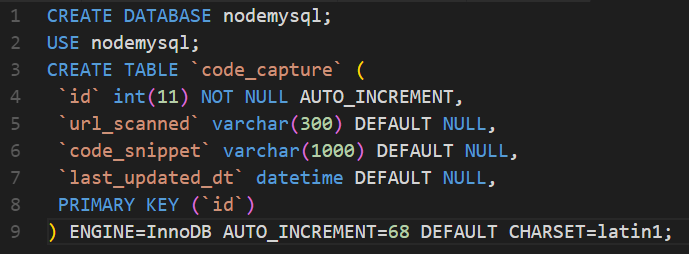
This holds the MySQL database script to create database, create table.

Id - is the Auto\_Increment number field

url\_scanned – holds the URL that is active in the browser and will be scanned for code snippet

code\_snippet – holds the code snippet found in the URL.

Last\_updated\_dt – holds the latest datetime value when the record got inserted.



1. **SourceCodeFind\NodeRestAPI\app.js:**

This is the Node component which will perform the following componets:

1. Import all the required dependencies via require().

express – Node library to start the server

mysql – Node library to connect to MySQL Database

cheerio – Node library to connect to HTML web scrape and look for keyword <pre> in our case in any website

request – Node library to hit the URL and get the request object

body-paser – Node library to parsing the request body to accept only json and URL encoding

moment-timezone – Node library to handle with the timezone and time, required while set today’s date time with Indian timezone (required while inserting the last\_updated\_dt in MYSQL database)

//Import all the required dependencies

const express = require('express');

const mysql = require('mysql');

const cheerio = require('cheerio');

const request = require('request');

const bodyParser = require("body-parser");

const momentTz = require('moment-timezone');

1. Initialize the Node express server, set the bodyparser middleware to accept only json and url encoding and start the express server to listen on port 3000.

const app = express();

//bodyParser returns middleware that only parses json

app.use(bodyParser.json());

//bodyParser returns middleware that only parses urlencoded bodies

app.use(bodyParser.urlencoded({extended : true}));

//Start the Express server on port 3000

app.listen('3000', () => {

console.log('Server started on port 3000');

});

1. Create MySQL database connection and connect to the database

//Create MySQL Database Connection

const db = mysql.createConnection({

host : 'localhost',

user : 'root',

password : '',

database : 'nodemysql'

});

//Connect MySQL Database

db.connect((err) => {

if(err){

console.log('\nError while connecting to MySQL Database. Please check if the MySQL Database server is up and running...\n');

throw err;

}

console.log('MySQL connected!');

});

1. Set the timezone to Indian Timezone using Moment Timezone library and date-time to current date time using Date.now().

//Set the moment Timezone to India

momentTz.tz.add("Asia/Calcutta|HMT BURT IST IST|-5R.k -6u -5u -6u|01232|-18LFR.k 1unn.k HB0 7zX0");

momentTz.tz.link("Asia/Calcutta|Asia/Kolkata");

let currentDateTimestamp = momentTz(Date.now()).format('YYYY-MM-DD HH:mm:ss');

1. Invoke the Post API method /insert, which should perform the following activities:

* Fetch the url (selected in the Chrome browser) from request body.
* Check if the incoming URL is valid URL (starting with http or https) using regex pattern
* If the URL is valid, invoke the URL via request library, load the website body into cheerio library and scan for the HTML keywords <pre> </pre>. Usually all the code snippets in any website will be marked between the <pre> and </pre> html code. Once you get the code snippets, insert them to MySQL Database via Insert script.

//Check if the URL is valid with regex pattern - to check if it starts with http:// or https://

var pattern = /(http|https):\/\/(\w+:{0,1}\w\*@)?(\S+)(:[0-9]+)?(\/|\/([\w#!:.?+=&%@!\-\/]))?/;

if (pattern.test(urlSelected)) {

//Cheerio is library to scrape the website URL provided

request(urlSelected, (err, response, body) => {

if (err) throw err;

if (!err && response.statusCode == 200) {

//Load the entire website content

$ = cheerio.load(body);

//Code Snippets will be tagged under <pre> </pre> tags. Now search for keyword <pre> tag to fetch the code snippet in between <pre> and </pre>.

codeSnippet = $('pre').text();

if(codeSnippet === ''){

console.log("No code snippets found in the URL: "+urlSelected);

res.send('No code snippets found in the URL : '+urlSelected);

} else{

console.log("Returned code snippet:"+codeSnippet);

//Populate all the values to the table columns

let post = {url\_scanned: urlSelected, code\_snippet: codeSnippet, last\_updated\_dt: currentDateTimestamp};

//Now execute the INSERT sql script to insert the records to MySQL database

let sql = 'INSERT INTO code\_capture SET ?';

let query = db.query(sql, post, (err, result) => {

if(err) throw err;

console.log(result);

res.send('Code snippets are inserted successfully to the database..');

});

}

}

Complete Code snippet:

//Import all the required dependencies

const express = require('express');

const mysql = require('mysql');

const cheerio = require('cheerio');

const request = require('request');

const bodyParser = require("body-parser");

const momentTz = require('moment-timezone');

//Intialize the Node express server

const app = express();

//bodyParser returns middleware that only parses json

app.use(bodyParser.json());

//bodyParser returns middleware that only parses urlencoded bodies

app.use(bodyParser.urlencoded({extended : true}));

//Start the Express server on port 3000

app.listen('3000', () => {

console.log('Server started on port 3000');

});

//Create MySQL Database Connection

const db = mysql.createConnection({

host : 'localhost',

user : 'root',

password : '',

database : 'nodemysql'

});

//Connect MySQL Database

db.connect((err) => {

if(err){

console.log('\nError while connecting to MySQL Database. Please check if the MySQL Database server is up and running...\n');

throw err;

}

console.log('MySQL connected!');

});

//Set the moment Timezone to India

momentTz.tz.add("Asia/Calcutta|HMT BURT IST IST|-5R.k -6u -5u -6u|01232|-18LFR.k 1unn.k HB0 7zX0");

momentTz.tz.link("Asia/Calcutta|Asia/Kolkata");

let currentDateTimestamp = momentTz(Date.now()).format('YYYY-MM-DD HH:mm:ss');

//POST API call for /insert

app.post('/insert', (req, res) => {

//Fetch the URL from from request body

//console.log("req.body.url="+req.body.url);

urlSelected = req.body.url;

console.log("urlSelected="+urlSelected);

if(urlSelected === ''){

console.log("urlSelected is empty. There is some problem..");

res.send('Oops..URL is coming as null. There is some problem..');

}else {

//Check if the URL is valid with regex pattern - to check if it starts with http:// or https://

var pattern = /(http|https):\/\/(\w+:{0,1}\w\*@)?(\S+)(:[0-9]+)?(\/|\/([\w#!:.?+=&%@!\-\/]))?/;

if (pattern.test(urlSelected)) {

//Cheerio is library to scrape the website URL provided

request(urlSelected, (err, response, body) => {

if (err) throw err;

if (!err && response.statusCode == 200) {

//Load the entire website content

$ = cheerio.load(body);

//Code Snippets will be tagged under <pre> </pre> tags. Now search for keyword <pre> tag to fetch the code snippet in between <pre> and </pre>.

codeSnippet = $('pre').text();

if(codeSnippet === ''){

console.log("No code snippets found in the URL: "+urlSelected);

res.send('No code snippets found in the URL : '+urlSelected);

} else{

console.log("Returned code snippet:"+codeSnippet);

//Populate all the values to the table columns

let post = {url\_scanned: urlSelected, code\_snippet: codeSnippet, last\_updated\_dt: currentDateTimestamp};

//Now execute the INSERT sql script to insert the records to MySQL database

let sql = 'INSERT INTO code\_capture SET ?';

let query = db.query(sql, post, (err, result) => {

if(err) throw err;

console.log(result);

res.send('Code snippets are inserted successfully to the database..');

});

}

}

});

} else {

res.send('Not a valid http or https URL. URL received is: '+urlSelected);

}

}

});

1. **SourceCodeFind\NodeRestAPI\package.json:**

This file holds the node dependencies required to create node\_modules folder. The “npm install” command will use this package.json file to install all the required dependencies mentioned for this project.

