Web Scraping using Node.js

R.Gayathri

11/01/2021

SUMMARY:

The purpose of this project is to create a web-based application that helps in web scraping. The tool used to achieve this objective is Node.js which is an open-source server providing an environment for the execution of JavaScript. In this package, the server side of the program aims to extract the filenames present in link format, in the specified URL. The extracted result is stored in an object, which is further used to write a JSON file. The JSON file acts as a database. The client side of the program is displayed in the form of a webpage. This page acts as the user-interface.

INTRODUCTION:

Web scraping saves us the trouble of manually searching and extracting data. It speeds up the process by creating easy access to the data. It has been used in many real-world applications like Finance, marketing and also in research fields such as data science and data analysis. In this project, the module aims at extracting only the links from the given URL. Knowing which component of the web page to extract, and specifying its format narrows down the search process and scraping.

BODY:

The tool used here is Node.js and Editor used is VS Code.

All the necessary dependencies are installed and required libraries are present in ***node_modules***.

There are various modules. Their functionality is listed down as follows:

package.json:

This module contains all the meta-data required for the project. It contains the name of the project, version details, the main program from where the execution should start from, the author and license details.

function.js:

This module contains the web-scraping method fully defined to do the required task. There are some dependencies like cheerio, request and fs to be installed and imported into respected variables. When the execution of this method starts, it makes a request to the given URL and loads the HTML page of that URL into a variable. A looping statement is used to scan the page for "a" tag and "href" attribute. Using the basename() function, the name of that attribute is sliced and stored into the object. This object is converted into a string using the stringify() method, and a JSON file is written.

list.json:

This file contains the list of all the filenames extracted. It is dynamic and can be modified anytime. File names can be added or deleted whenever needed.

index.js:

This is the main program of the project, and the execution begins here. The ScrapeProduct() method from function.js is exported, and index.js imports it into a variable. This module acts like an API and uses express.js to render the contents into a HTML page. The express module is imported into the app variable. The get() function is used to make a request and deliver a response. It calls the ScrapeProduct() function which gives an output, all the contents of the list.json file is loaded into a variable. It also recursively writes each file name into the page as a drop-down list. The port chosen here is 8080.

RESULT:

The execution is done using the command "node index.js" in the VS Code Terminal. The result can be viewed at the localhost webpage - http://localhost:8080/. Thus, the web scraping using Node.js + Express.js has been implemented successfully.

REFERENCES:

1) Download Node.js - https://nodejs.org/en/download/
2) Download VS Code - https://code.visualstudio.com/download
3) https://www.youtube.com/watch?v=7FjhF6Hy9gY&t=391s
4) github - https://github.com/Gayathri99Ravichandran/WebScraping-using-NodeJS

APPENDIX:

Sample code:

```
const express = require('express') ;
const app = express();
const port = 8080;
const scraper = require('./function.js');
const allfiles = require('./function');

app.get("/", async (req,res) =>
{
    var filenames = scraper.scrapeProduct('http://apt.postgresql.org/pub/repos/apt/pool/14/p/postgresql-14/');
    const files = require('./list.json');
    res.setHeader('Content-type','text/html')
    res.write('<html><body><center><h1>National Informatics Centre</h1><h3>FILE LIST</h3><select>');

    for(i=0;i<files.DEBfiles.length; i++)
    {
    res.write('<option>'+files.DEBfiles[i]+'</option>');
    }
    res.write('</select></center></body></html>');
    res.end();
});

app.listen(port, ()=> console.log("Listening on port" + port));
```