# Task1-Automation:

Write a script that downloads the “alexa top 1m sites”. This is a csv file available publically. Once it’s downloaded, put the domains that rank 40->49 inclusive into a text file.

The following lines are the alexa\_40to49 js file content:

const fs = require('fs');

const fetch = require('node-fetch');

const unzipper = require('unzipper');

const csv2 = require('csv2');

const AbortController = require('abort-controller');

const URL = 'http://s3.amazonaws.com/alexa-static/top-1m.csv.zip';

async function download(destFileName,f,l) {

const controller = new AbortController();

fetch(URL, { signal: controller.signal })

.then(res => {

let counter = 1;

const dest = fs.createWriteStream(destFileName);

res.body

.pipe(unzipper.ParseOne())

.pipe(csv2())

.on('data', (d) => {

if (counter <= l) {

if (counter>=f){

dest.write(d[1] + '\n'); }

counter++;

if (counter > l) {

dest.close();

return controller.abort()

}

}

})

})

.catch(console.err)

.finally(\_ => {

// Cleanup

})

}

// Download top 10.. and save it in websites.txt

download('alexa\_40to49.txt',40, 49)

# Task1- Crawling:

Write a script that reads 1 domain at a time from the file you created in the previous step and grabs the visible home page text of that site as seen through a browser.

The first file containing the first file named crolingmethod1.js:

const fs = require('fs');

const phantom = require('phantom');

async function scrape() {

//read the file into a single string

// Not optimised for a large file..

const filedata = fs.readFileSync('websites.txt', 'utf-8');

// split the string on newline and store in array

const arrdata = filedata.split(/[\r\n]/);

// console.log(arrdata)

const instance = await phantom.create()

for (var i = 0; i < arrdata.length - 1; i++) {

// show each line

var newurl = 'http://' + arrdata[i] + '/';

newurl = newurl.replace(/\s+/g, '');

arrdata[i] = newurl;

webdomain = arrdata[i];

console.log(webdomain)

const page = await instance.createPage()

await page.on('onResourceRequested', function (requestData) {

// console.info('Requesting', requestData.url);

});

const status = await page.open(webdomain);

if (status === 'success') {

const content = await page.property('content');

fs.writeFileSync(`${i}.html`, content)

console.log(content);

}

}

await instance.exit()

}

scrape()

The first file containing the second file named crolingmethod2.js:

const fs = require('fs');

const readline = require('readline')

const phantom = require('phantom');

async function scrape(url) {

console.log(`Starting: ${url}`)

return new Promise(async (resolve, reject) => {

try {

const instance = await phantom.create()

const page = await instance.createPage()

await page.on('onResourceRequested', function (requestData) {

// console.info('Requesting', requestData.url);

});

const status = await page.open(url);

if (status === 'success') {

const content = await page.property('content');

await instance.exit()

return resolve(content)

}

return resolve(status)

} catch (e) {

reject(e)

}

})

}

// Achieve parallelism..

function websitePageDownload(filename, callbackFn) {

const rI = readline.createInterface({

input: fs.createReadStream(filename),

output: null,

console: false

});

rI.on('line', function (line) {

if (!line || line === '')

return

let url = `http://${line}/`;

url = url.replace(/\s+/g, '');

scrape(url).then(data => {

callbackFn(line, data)

}).catch(console.err)

});

}

// Saving pages to file..

function callbackFn(url, data) {

console.log('Finished: ', url)

fs.writeFileSync(`${url}.html`, data)

}

websitePageDownload('alexa\_40to49.txt', callbackFn)

# Task3- General Questions:

1. You have a 5GB text file which contains a list of domain names. Some are duplicated. Describe the quickest method for removing the duplicates. You may use PHP or Linux bash/cli
2. A very active website (approx 1m visitors a day) works perfectly for 50% of its users, but the other 50% timeout. Describe a few scenarios that might cause this and how you would investigate.
3. Some users visiting (http://ads.explained.com) find css and js files do not load (404). Why might this be?
4. You need to write a php script that acts as a human and logs remotely into a website. Describe what technologies/code libraries might be required to achieve this