### Assignment for Tech Internship via Internshala

### Step 1: Setting up Selenium with Twitter Login and ProxyMesh in Java

#### Java Maven Dependencies

First, ensure you have the necessary dependencies in your ‘pom.xml’:

<dependencies>

<!-- Selenium -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.0.0</version>

</dependency>

<!-- MongoDB Java Driver -->

<dependency>

<groupId>org.mongodb</groupId>

<artifactId>mongodb-driver-sync</artifactId>

<version>4.3.4</version>

</dependency>

<!-- Gson -->

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.6</version>

</dependency>

</dependencies>

**Selenium Script:**

import com.mongodb.MongoClient;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.model.Filters;

import com.mongodb.client.model.Updates;

import org.bson.Document;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.remote.CapabilityType;

import org.openqa.selenium.remote.DesiredCapabilities;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import java.util.Random;

import java.util.UUID;

public class TwitterScraper {

private static final String[] PROXIES = {

"http://proxy1:port",

"http://proxy2:port",

"http://proxy3:port"

// Add as many proxies as you want

};

public static void main(String[] args) {

String proxy = getRandomProxy();

WebDriver driver = createDriver(proxy);

try {

twitterLogin(driver, "your\_username", "your\_password");

List<String> topTrends = scrapeTrends(driver);

String endTime = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date());

String uniqueId = UUID.randomUUID().toString();

// Save to MongoDB

MongoClient mongoClient = new MongoClient("localhost", 27017);

MongoDatabase database = mongoClient.getDatabase("twitter\_trends");

MongoCollection<Document> collection = database.getCollection("trends");

Document record = new Document("\_id", uniqueId)

.append("trend1", topTrends.get(0))

.append("trend2", topTrends.get(1))

.append("trend3", topTrends.get(2))

.append("trend4", topTrends.get(3))

.append("trend5", topTrends.get(4))

.append("end\_time", endTime)

.append("ip\_address", proxy);

collection.insertOne(record);

System.out.println(record.toJson());

} finally {

driver.quit();

}

}

private static String getRandomProxy() {

Random random = new Random();

return PROXIES[random.nextInt(PROXIES.length)];

}

private static WebDriver createDriver(String proxy) {

System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");

ChromeOptions options = new ChromeOptions();

options.addArguments("--disable-blink-features=AutomationControlled");

options.addArguments("--start-maximized");

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability(CapabilityType.PROXY, proxy);

capabilities.merge(options);

return new ChromeDriver(capabilities);

}

private static void twitterLogin(WebDriver driver, String username, String password) {

driver.get("https://twitter.com/login");

try { Thread.sleep(5000); } catch (InterruptedException e) { e.printStackTrace(); }

WebElement usernameField = driver.findElement(By.name("session[username\_or\_email]"));

WebElement passwordField = driver.findElement(By.name("session[password]"));

usernameField.sendKeys(username);

passwordField.sendKeys(password);

passwordField.submit();

try { Thread.sleep(5000); } catch (InterruptedException e) { e.printStackTrace(); }

}

private static List<String> scrapeTrends(WebDriver driver) {

driver.get("https://twitter.com/explore/tabs/trending");

try { Thread.sleep(5000); } catch (InterruptedException e) { e.printStackTrace(); }

List<WebElement> trends = driver.findElements(By.xpath("//div[@data-testid='trend']"));

return trends.stream().limit(5).map(WebElement::getText).toList();

}

}

### Step 2: Creating the HTML Interface

We'll use Flask to create a simple web server that interacts with the Selenium script and displays the results. For this, we need to ensure Flask is installed:

pip install Flask

**Flask App**

from flask import Flask, render\_template, jsonify

import subprocess

app = Flask(\_\_name\_\_)

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/run\_script", methods=["POST"])

def run\_script():

result = subprocess.run(["java", "-cp", "path/to/your/classes", "TwitterScraper"], capture\_output=True, text=True)

record = eval(result.stdout)

return jsonify(record)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

HTML Template (templates/index.html)

<!DOCTYPE html>

<html>

<head>

<title>Twitter Trends</title>

</head>

<body>

<h1>Click here to run the script.</h1>

<button onclick="runScript()">Run Script</button>

<div id="results"></div>

<script>

function runScript() {

fetch("/run\_script", { method: "POST" })

.then(response => response.json())

.then(data => {

const resultsDiv = document.getElementById("results");

const endTime = new Date(data.end\_time).toLocaleString();

resultsDiv.innerHTML = `

<p>These are the most happening topics as on ${endTime}</p>

<ul>

<li>${data.trend1}</li>

<li>${data.trend2}</li>

<li>${data.trend3}</li>

<li>${data.trend4}</li>

<li>${data.trend5}</li>

</ul>

<p>The IP address used for this query was ${data.ip\_address}.</p>

<p>Here's a JSON extract of this record from the MongoDB:</p>

<pre>${JSON.stringify(data, null, 2)}</pre>

<button onclick="runScript()">Click here to run the query again.</button>

`;

});

}

</script>

</body>

</html>