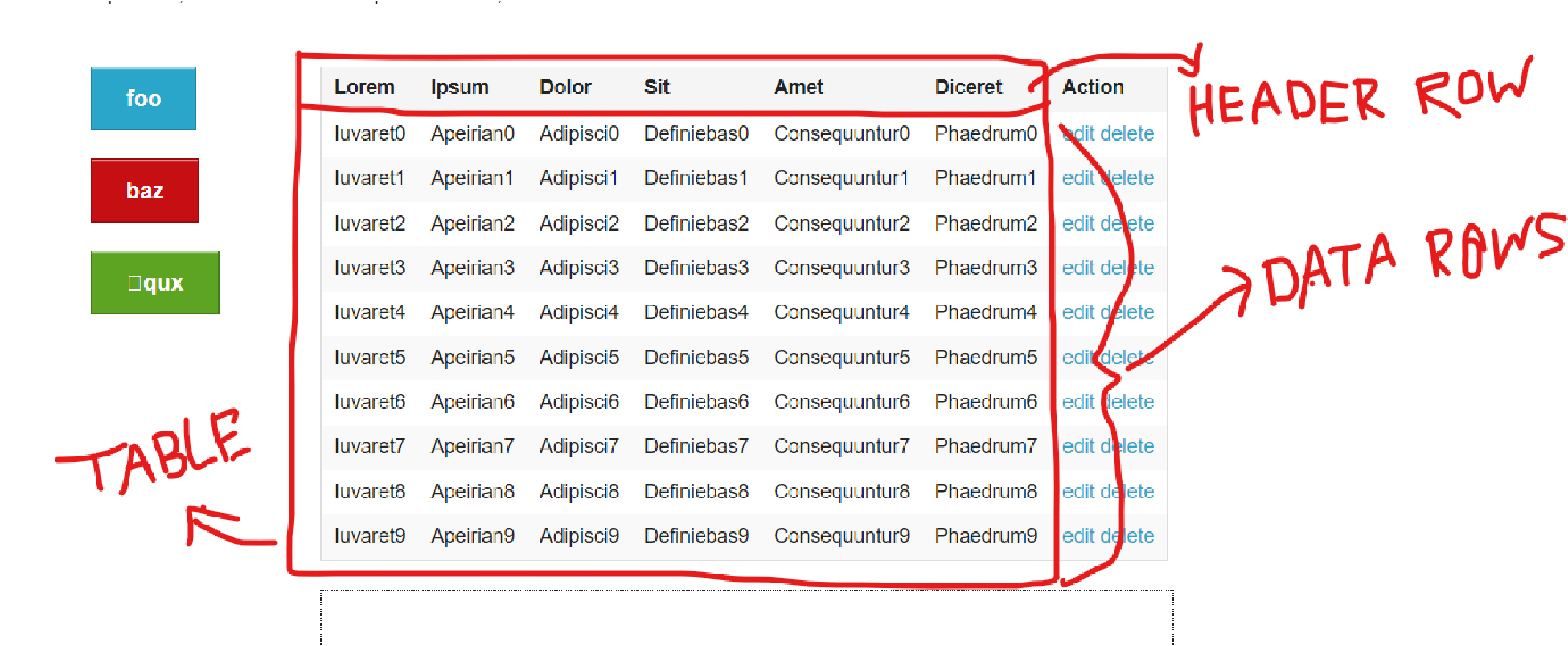
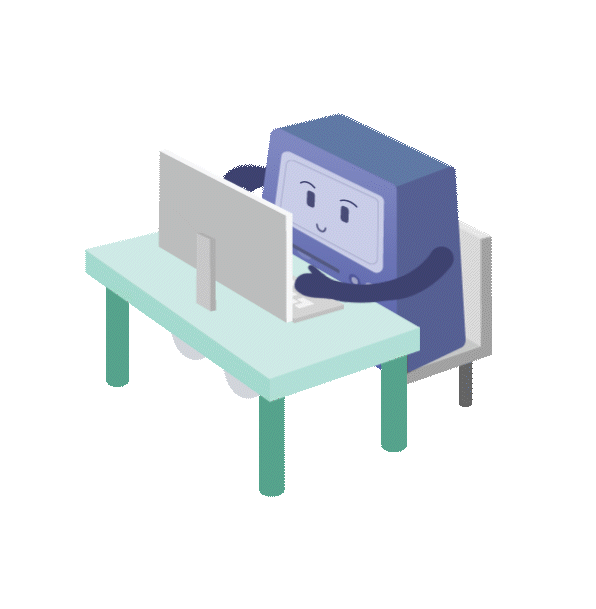
1. Selenium - Web Table to CSV Generate
   1. Go to the website <http://the-internet.herokuapp.com/challenging_dom>



* 1. Read the table as highlighted in the above screenshot
  2. Store the table data into any JAVA collection & print table to the console

Code:

import java.util.ArrayList;

import java.util.List;

public class Main {

static class TableRow {

private String iuvaret;

private String apeirian;

private String adipisci;

private String definiebas;

private String consequuntur;

private String phaedrum;

private String action;

public TableRow(String iuvaret, String apeirian, String adipisci, String definiebas, String consequuntur, String phaedrum, String action)

{

this.iuvaret = iuvaret;

this.apeirian = apeirian;

this.adipisci = adipisci;

this.definiebas = definiebas;

this.consequuntur = consequuntur;

this.phaedrum = phaedrum;

this.action = action;

}

@Override

public String toString() {

return iuvaret + "\t" + apeirian + "\t" + adipisci + "\t" + definiebas + "\t" + consequuntur + "\t" + phaedrum + "\t" + action;

}

}

public static void main(String[] args) {

// Create the table

List<TableRow> table = new ArrayList<>();

for (int i = 0; i <= 9; i++) {

table.add(new TableRow("Iuvaret" + i, "Apeirian" + i, "Adipisci" + i, "Definiebas" + i, "Consequuntur" + i, "Phaedrum" + i, "edit delete"));

}

// Print header

System.out.println("Lorem\t Ipsum\t Dolor\t Sit\t Amet\t Diceret\t Action");

// Print table

for (TableRow row : table)

{

System.out.println(row);

}

}

}

2.

1. Fetch data from this url using any API library and export as csv <https://data.sfgov.org/resource/p4e4-a5a7.json>
   * 1. Print current timestamp in MM-DD-YY-HH-MI-SS format to console
     2. Convert json to a table & print output to console
     3. Using regex find if a particular line is related to “roof”

Code:

import java.io.\*;

import java.net.HttpURLConnection;

import java.net.URL;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.regex.\*;

import org.json.JSONArray;

import org.json.JSONObject;

public class DataFetcher {

public static void main(String[] args) {

try {

// Step 1: Fetch data from the URL

String jsonData = fetchDataFromURL("https://data.sfgov.org/resource/p4e4-a5a7.json");

// Step 2: Print current timestamp in MM-DD-YY-HH-MI-SS format

String timestamp = getCurrentTimestamp();

System.out.println("Current Timestamp: " + timestamp);

// Step 3: Convert JSON to a table & print output to console

convertJsonToTableAndPrint(jsonData);

// Step 4: Using regex, find if a particular line is related to "roof"

findLinesRelatedToRoof(jsonData);

}

catch (Exception e)

{

e.printStackTrace();

}

}

// Step 1: Fetch data from the URL

public static String fetchDataFromURL(String url) throws Exception

{

HttpURLConnection connection = (HttpURLConnection) new URL(url).openConnection();

connection.setRequestMethod("GET");

int responseCode = connection.getResponseCode();

if (responseCode == HttpURLConnection.HTTP\_OK)

{

BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

String inputLine;

StringBuilder response = new StringBuilder();

while ((inputLine = in.readLine()) != null)

{

response.append(inputLine);

}

in.close();

return response.toString();

}

else

{

throw new Exception("Failed to fetch data from URL. Response code: " + responseCode);

}

}

// Step 2: Print current timestamp in MM-DD-YY-HH-MI-SS format

public static String getCurrentTimestamp()

{

SimpleDateFormat dateFormat = new SimpleDateFormat("MM-dd-yy-HH-mm-ss");

Date date = new Date();

return dateFormat.format(date);

}

// Step 3: Convert JSON to a table & print output to console

public static void convertJsonToTableAndPrint(String jsonData)

{

JSONArray jsonArray = new JSONArray(jsonData);

if (jsonArray.length() > 0)

{

JSONObject firstObject = jsonArray.getJSONObject(0);

System.out.println("Table Columns: " + firstObject.keySet());

System.out.println("Table Data:");

for (int i = 0; i < jsonArray.length(); i++)

{

JSONObject jsonObject = jsonArray.getJSONObject(i);

System.out.println(jsonObject);

}

}

else

{

System.out.println("No data found.");

}

}

// Step 4: Using regex, find if a particular line is related to "roof"

public static void findLinesRelatedToRoof(String jsonData)

{

String regex = ".\*roof.\*";

Pattern pattern = Pattern.compile(regex, Pattern.CASE\_INSENSITIVE);

Matcher matcher = pattern.matcher(jsonData);

System.out.println("\nLines related to 'roof':");

while (matcher.find())

{

System.out.println(matcher.group());

}

}

}