package D2L;

import java.util.Objects;

import java.io.\*;

import java.io.IOException;

import java.io.OutputStreamWriter;

import java.io.OutputStream;

import java.net.URL;

import java.io.InputStreamReader;

import java.net.MalformedURLException;

import java.net.HttpURLConnection;

import java.net.ProtocolException;

import java.io.BufferedReader;

import java.util.Base64;

public class AccessToken {

public String bearerToken;

public String clientId;

public String clientSecret;

public String callBackUrl;

public String authorizeUrl;

public String authTokenUrl;

public String AppKey;

public String AppId;

public String qualifiedAuth;

String redirect = getAuthGrantType(callBackUrl);

public String getAuthGrantType(String callBackUrl) {

return qualifiedAuth;

}

// public AccessToken(String clientId, String clientSecret, String callBackUrl, String authorizeUrl) {

// this.clientId = clientId;

// this.clientSecret = clientSecret;

// this.callBackUrl = callBackUrl;

// this.authorizeUrl = authorizeUrl;

// qualifiedAuth = authorizeUrl + "?response\_type=code&client\_id=" + clientId + "&redirect\_uri=" + callBackUrl + "&scope=core%3A\*%3A\*";

// }

public void GetAuthCode() throws IOException {

String clientId = "e9c18f0d-b7d0-4cd4-92a7-0d716c2d2a6b";

String clientSecret = "0rMzHMLu-4iAuPqGXRYmUq\_mS8\_oVebOFDCMBguTW7w";

String authorizeUrl = "https://auth.brightspace.com/oauth2/auth";

String callBackUrl = "https://oauth.pstmn.io/v1/callback";

qualifiedAuth = authorizeUrl + "?response\_type=code&client\_id=" + clientId + "&redirect\_uri=" + callBackUrl + "&scope=core%3A\*%3A\*";

URL AuthUrl = new URL(qualifiedAuth);

String read = null;

HttpURLConnection connection = (HttpURLConnection) AuthUrl.openConnection();

connection.setRequestMethod("GET");

int codeResponse = connection.getResponseCode();

if (codeResponse == HttpURLConnection.***HTTP\_OK***)

{

// reading the response from the server

InputStreamReader isrObj = new InputStreamReader(connection.getInputStream());

System.***out***.println(isrObj);

BufferedReader bf = new BufferedReader(isrObj);

System.***out***.println(bf);

// to store the response from the servers

StringBuffer responseStr = new StringBuffer();

while ((read = bf .readLine()) != null)

responseStr.append(read);

}

// closing the BufferedReader

bf.close();

// disconnecting the connection

connection.disconnect();

// print the response

System.***out***.println(responseStr.toString());

}

else

{

System.***out***.println("GET Request did not work");

}

}

public void getGradesList() throws IOException

{

String clientId = "e9c18f0d-b7d0-4cd4-92a7-0d716c2d2a6b";

String clientSecret = "0rMzHMLu-4iAuPqGXRYmUq\_mS8\_oVebOFDCMBguTW7w";

String authorizeUrl = "https://auth.brightspace.com/oauth2/auth";

String callBackUrl = "https://oauth.pstmn.io/v1/callback";

qualifiedAuth = authorizeUrl + "?response\_type=code&client\_id=" + clientId + "&redirect\_uri=" + callBackUrl + "&scope=core%3A\*%3A\*";

AccessToken Obj = new AccessToken();

Obj.getAuthGrantType(callBackUrl);

URL AuthUrl = new URL("https://lms.cofc.edu/d2l/api/lp/1.31/dataExport/list");

BufferedReader reader = null;

HttpURLConnection connection = (HttpURLConnection) AuthUrl.openConnection();

connection.setRequestMethod("GET");

int codeResponse = connection.getResponseCode();

if (codeResponse == HttpURLConnection.***HTTP\_OK***)

{

connection.setRequestProperty("Authorization", "Bearer " + bearerToken);

connection.setDoOutput(true);

connection.setRequestMethod("GET");

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

String line = null;

StringWriter out = new StringWriter(connection.getContentLength() > 0 ? connection.getContentLength() : 2048);

while ((line = reader.readLine()) != null) {

out.append(line);

}

String response = out.toString();

System.***out***.println(response);

}

else

{

System.***out***.println("GET Request did not work");

}

}

public static void main (String[] args) throws IOException {

// creating an object of the JavaGETExample class

AccessToken getObj = new AccessToken();

// invoking the method GETRequest()

getObj.GetAuthCode();

}

}