МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп’ютерних наук

**КОМПЛЕКСНО-ПРАКТИЧНЕ**

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**з навчальної дисципліни “Конструювання програмного забезпечення”**

**На тему: “** **Дослідження Java-технології при вирішенні задач по наданню сервісів Internet.”**

**Виконав:** ст.гр. ПЗС-31

Баюрський А.Р.

**Перевірив:** Сирник О.Й.

ТНЕУ– 2015

**План**:

Вступ

1. Реалізація системи мовою програмування
2. Тестування програми
3. Висновок

Список використаних джерел

**Вступ:**

Технології Internet широко використовується ось уже 55 років. За час винайдення інтернету і до тепер, було проведено дуже багато нових досліджень. Мало хто, в 60-ті роки мі би подумати, що ми досягнемо таких висот.

Протягом 90-их років Інтернет об’єднав у собі більшість існуючих на той час мереж (хоча деякі, як, наприклад, Фідонет, залишились відособленими). Завдяки відсутності єдиного керуючого центру, а також завдяки відкритості технічних стандартів Інтернету, що автоматично робило мережі незалежними від бізнесу чи уряду, об'єднання виглядало неймовірно привабливим. До 1997 року в Інтернеті нараховувалось близько 10 мільйонів комп'ютерів і було зареєстровано понад мільйон доменних назв. Інтернет став дуже популярним засобом обміну інформацією. У 2011 році, 5 січня, кількість користувачів інтернету у світі перевищила два мільярди.

Інтернет складається з багатьох тисяч корпоративних, наукових, урядових та домашніх мереж. Об'єднання різнорідних за архітектурою мереж стало можливо завдяки протоколу IP (англ. Internet Protocol) і принципу маршрутизації пакетів даних. Протокол ІР був спеціально створений агностичним у відношенні до фізичних каналів зв'язку. Тобто будь-яка мережа передачі цифрових даних може передавати інтернет-трафік. На стиках мереж спеціальні маршрутизатори займаються сортуванням та перенаправленням пакетів даних, базуючись на ІР-адресах одержувачів цих пакетів. Протокол ІР утворює єдиний адресний простір у масштабах всього світу, але в кожній окремо взятій мережі може існувати свій власний адресний підпростір. Така організація ІР-адрес дозволяє маршрутизаторам однозначно визначати подальший напрямок для кожного, навіть найменшого, пакету даних. У результаті між різними мережами Інтернету не виникає конфліктів і дані точно і без перешкод передаються від мережі до мережі по всій планеті.

1. **Реалізація системи мовою програмування**

Завдяки наявній великій кількості готових бібліотек, зокрема бібліотек «*BufferedReader*», «*DataOutputStream*», «*InputStream*», «*InputStreamReader*», «*HttpURLConnection*», «*URL*», «*URLEncoder*», які були використані при розробці програми, що надсилає POST-запити на сервер.

Що таке POST-запит?

POST передає призначені для користувача дані (наприклад, з HTML-форми) заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані (у прикладі з блогами — текст коментаря) включаються в тіло запиту. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожного відправлення коментаря з'являтиметься одна копія цього коментаря).

Розглянемо дві основні бібліотеки, що використовувались в розроблюваній системі, а саме:

* URL
* HttpURLConnection

Бібліотека URL містить клас, який в подальшому буде використовуватись як посилання на сервер, що буде обробляти наші дані. Взагалі у цього класу доволі широке використання, його можна застосувати не тільки під час POST-запитів.

Бібліотека HttpURLConnection відповідає за усі HTTP запити у середовищі Інтернету, в моєму випадку, я використав цю бібліотеку для того, щоб мати змогу відправити POST запит з необхідними мені даними на сервер.

Для розробки системи було обрано мову програмування Java та середовище IDE IntellijIdea.

Програма містить один клас – Main та функцію excutePost.

Далі подано лістинг коду головної функції програми:

public static void main(String[] args) {

String smth;

System.out.println("We're trying to POST some token into the codeabby servers:");

smth=excutePost("http://codeabbey.sourceforge.net/say-100.php","token=migbKhs8QTgfM1jRPo4fqzM1");

System.out.println(smth);

int result=100-Integer.parseInt(smth.split(" ")[1].trim());

System.out.println("So, for now, we get the answer! Now we need to send the answer:");

System.out.println("answer = "+result+"\nLet's do it!");

String urlParameters =

"token=" + URLEncoder.encode("migbKhs8QTgfM1jRPo4fqzM1") +

"&answer=" + URLEncoder.encode(String.valueOf(result));

smth=excutePost("http://codeabbey.sourceforge.net/say-100.php",urlParameters);

System.out.println("now we get a response, and it's an our victory token:");

System.out.println(smth);

}

Повний код програми подано у додатку А.

1. **Тестування програми**

Для перевірки працездатності програми був обраний сервер одного маловідомого сайту з задачами з програмування – codeabby. Суть завдання зводиться до наступного – необхідно відправити пустий токен на сервер, після чого сервер у відповідь відправить число від 1 до 99, у відповідь необхідно подати число, яке в сумі із числом, яке прийшло від сервера, утворює 100. [8] Цю задачу, а також сервер, який використовувався можна знайти на їх сайті під назвою «Say 100».

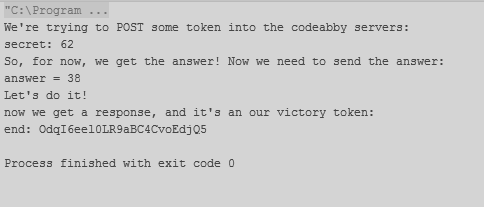


Рисунок 1 – Виконання програми

Як видно з рисунку 1 спочатку програма спробувала відправити POST запит на сервер, після чого отримала відповідь – secret: 62. Це означає що нам потрібно серверу надіслати ще один запит, з відповіддю на це секретне число, а саме – 38. В кінці ми отримали відповідь end – наш токен переможця.

1. **Висновок**

Використовуючи стандартні бібліотеки що надаються такими IDE як NetBeans чи IntelliJIdea ми можемо з легкістю відправляти різного роду запити (POST, GET та інші). Це дозволяє нам навіть імітувати роботу браузера або навіть написати свій інтернет-браузер.

**Список використаних джерел**:

1. [Електронний ресурс]. – Режим доступу: http://www.msdn.com.
2. [Електронний ресурс]. – Режим доступу: http://www.codenet.ru
3. [Електронний ресурс]. – Режим достпу: http://professorweb.ru/
4. [Електронний ресурс]. – Режим достпу: http://uk.wikipedia.org/
5. С#: Пер. С англ./Дейтел Х.,Дейтел П., Листфилд Дж., Нието Т., Йегер Ш., Златкина М. – СПб.: БХВ-Петербург, 2006. – 1056 с.:ил.
6. Джеффри Рихтер - CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке С#. Мастер-класс./Пер. с англ. — М.: Издательство «Русская Редакция»; СПб.: Питер, 2007. — 656 стр. : ил.
7. Кристиан Нейгел, Билл Ивьен, Джей Глинн, Карли Уотсон, Морган Скиннер - С# 2008 и платформа .Net 3.5 для профессионалов.
8. [Електронний ресурус]. – Режим доступу: http://www.codeabbey.com/

**Додаток А**

package com.company;

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLEncoder;

public class Main {

public static void main(String[] args) {

String smth;

System.out.println("We're trying to POST some token into the codeabby servers:");

smth=excutePost("http://codeabbey.sourceforge.net/say-100.php","token=migbKhs8QTgfM1jRPo4fqzM1");

System.out.println(smth);

int result=100-Integer.parseInt(smth.split(" ")[1].trim());

System.out.println("So, for now, we get the answer! Now we need to send the answer:");

System.out.println("answer = "+result+"\nLet's do it!");

String urlParameters =

"token=" + URLEncoder.encode("migbKhs8QTgfM1jRPo4fqzM1") +

"&answer=" + URLEncoder.encode(String.valueOf(result));

smth=excutePost("http://codeabbey.sourceforge.net/say-100.php",urlParameters);

System.out.println("now we get a response, and it's an our victory token:");

System.out.println(smth);

}

public static String excutePost(String targetURL, String urlParameters)

{

URL url;

HttpURLConnection connection = null;

try {

//Create connection

url = new URL(targetURL);

connection = (HttpURLConnection)url.openConnection();

connection.setRequestMethod("POST");

connection.setRequestProperty("Content-Type",

"application/x-www-form-urlencoded");

connection.setRequestProperty("Content-Length", "" +

Integer.toString(urlParameters.getBytes().length));

connection.setRequestProperty("Content-Language", "en-US");

connection.setUseCaches (false);

connection.setDoInput(true);

connection.setDoOutput(true);

//Send request

DataOutputStream wr = new DataOutputStream (

connection.getOutputStream ());

wr.writeBytes (urlParameters);

wr.flush ();

wr.close ();

//Get Response

InputStream is = connection.getInputStream();

BufferedReader rd = new BufferedReader(new InputStreamReader(is));

String line;

StringBuffer response = new StringBuffer();

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

response.append(line);

response.append('\r');

}

rd.close();

return response.toString();

} catch (Exception e) {

e.printStackTrace();

return null;

} finally {

if(connection != null) {

connection.disconnect();

}

}

}

}