Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Programare in Retea

Lucrarea de laborator nr.2

HTTP Client cu request-uri paralele

Autor: Negru Igor

lector asistent: Alexandru Gavrisco

lector superior: Dumitru Ciorba

**1. Scopul lucrarii de laborator**

Studierea modelului OSI, protocolului HTTP si implementarea unei aplicatii ce efectueaza multiple HTTP request-uri in paralel.

**2. Implementarea task-ului**

1. Extragerea listei de categorii Categorii se extrag de pe https://evil-legacy-service.herokuapp.com/api/v101/categories/ in format CSV.

// Requesturile GET

private void GetCategoriesOrders()

{

Task.Run(() => MakeRequest(@"https://evil-legacy-service.herokuapp.com/api/v101/orders/?" + $"start={startDate}&end={endDate}", "orders"));

Task.Run(() => MakeRequest(@"https://evil-legacy-service.herokuapp.com/api/v101/categories/", "categories"));

}

//Efectuarea requestului

private void MakeRequest(string url, string type)

{

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

request.Method = "GET";

request.Timeout = 60 \* 1000; // one minute

request.Headers.Add("X-API-Key", "55193451-1409-4729-9cd4-7c65d63b8e76");

request.Accept = "text/csv";

HttpStatusCode responseCode;

string result = "";

try

{

using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())

{

responseCode = response.StatusCode;

result = new StreamReader(response.GetResponseStream()).ReadToEnd();

response.Close();

}

}

catch (WebException webException)

{

throw webException;

}

if (responseCode != HttpStatusCode.OK)

throw new Exception("Invalid respone");

if (type == "categories")

{

foreach (ICsvLine line in CsvReader.ReadFromText(result))

{

categories.AddValidElement(line);

}

}

else if (type == "orders")

{

foreach (ICsvLine line in CsvReader.ReadFromText(result))

{

orders.AddValidElement(line);

}

}

countEventObject.Signal();

}

}

}

**2.Transformarea din format CSV in model Categorii**

//adaugarea elementelor

public void AddValidElement(ICsvLine line)

{

//parsarea categoriilor din CSV

try

{

if (String.IsNullOrWhiteSpace(line["name"])

|| String.IsNullOrWhiteSpace(line["id"]))

{

return;

}

Category cat = new Category(line);

list[Int32.Parse(line["id"])] = cat;

}

catch (Exception e)

{

return;

}

}

}

}

**3.Transformarea din format CSV in model Order**

//parsarea elementului CSV

public void AddValidElement(ICsvLine line)

{

try

{

if (String.IsNullOrWhiteSpace(line["total"])

|| String.IsNullOrWhiteSpace(line["id"])

|| String.IsNullOrWhiteSpace(line["category\_id"]))

{

return;

}

Order ord = new Order(line);

if (list.ContainsKey(Int32.Parse(line["category\_id"])))

{

list[Int32.Parse(line["category\_id"])].total += ord.total;

}

else

{

list[Int32.Parse(line["category\_id"])] = ord;

}

}

catch (Exception e)

{

//do nothing, element is invalid

return;

}

}

}

}

**4. Concluzie**

In cadrul acestei lucrari de laborator am efecuat o aplicatie care aﬁseaza totalurile de comenzi pentru un magazin. Am invatat cum sa fac HTTP request-uri in alt thread si cum de transformat CSV in model.