**Klasa LogEntry**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vezbe8novembar

{

//4.189.158.117 - - [21/Sep/2015:07:59:14 -0400] "GET /favicon.ico HTTP/1.0" 404 621

internal class LogEntry

{

private string ipAddress;

private DateTime accessTime;

private string request;

private int statusCode;

private int bytesReturned;

public LogEntry(string ipAddress, DateTime accessTime, string request, int statusCode, int bytesReturned)

{

this.ipAddress = ipAddress;

this.accessTime = accessTime;

this.request = request;

this.statusCode = statusCode;

this.bytesReturned = bytesReturned;

}

public string IpAddress

{

get { return ipAddress; }

set { ipAddress = value; }

}

public DateTime AccessTime

{

get { return accessTime; }

set { accessTime = value; }

}

public string Request

{

get { return request; }

set { request = value; }

}

public int StatusCode

{

get { return statusCode; }

set { statusCode = value; }

}

public int BytesReturned

{

get { return bytesReturned; }

set { bytesReturned = value; }

}

//4.189.158.117 - - [21/Sep/2015:07:59:14 -0400] "GET /favicon.ico HTTP/1.0"

public override string ToString()

{

return ipAddress+" "+accessTime+" "+request+" "+statusCode+" "+bytesReturned;

}

}

}

**Klasa LogParser**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vezbe8novembar

{

internal class LogParser

{

//4.189.158.117 - - [21/Sep/2015:07:59:14 -0400] "GET /favicon.ico HTTP/1.0" 404 621

public static string cutter(StringBuilder line, string delim)

{

string oneLine = line.ToString();

int x = oneLine.IndexOf(delim);

int length = delim.Length;

if (x == -1)

{

x = oneLine.Length;

length = 0;

}

string result = oneLine.Substring(0, x);

line.Remove(0, x + length);

Console.WriteLine("Ostatak: " + line.ToString());

return result;

}

// 621

public static LogEntry parseLine(string line)

{

//StringBuilder logLine = new StringBuilder();

// ovo je linija koda zbog koje je bila greska na casu i zbog cega je izbacivalo gresku za metod parseDate

// treba da bude konstruktor sa prametrom, a parameter je ulazni podatak tipa string (line) metode LogEntry

//sve ostalo sto smo radili na casu je dobro

**StringBuilder logLine = new StringBuilder(line);** //🡨 ovako je ispravno

Console.WriteLine(line);

string ip = cutter(logLine, " ");

Console.WriteLine("IP adresa: " + ip);

cutter(logLine, " "); //brisemo deo podataka koji nam ne treba

cutter(logLine, "[");

string dateString = cutter(logLine, " ");

Console.WriteLine("Datum: "+dateString);

DateTime date=parseDate(dateString);

Console.WriteLine("Datum (kao DateTime tip):"+date.ToString());

cutter(logLine, " \"");

string request = cutter(logLine, "\"");

Console.WriteLine("Request: "+request);

cutter(logLine, " ");

string statusStr= cutter(logLine, " ");

int status=int.Parse(statusStr);

Console.WriteLine("Status: "+status);

string returnedBytesStr=cutter(logLine, " ");

int returnedBytesInt=int.Parse(returnedBytesStr);

Console.WriteLine("Bytes: "+returnedBytesInt);

return new LogEntry(ip,date,request,status,returnedBytesInt);

}

private static DateTime parseDate(string dateString)

{

//21/Sep/2015:07:59:14

StringBuilder date = new StringBuilder(dateString);

int x = dateString.IndexOf(":");

date.Replace(":", " ", x, 1);

return DateTime.Parse(date.ToString());

}

}

}

**Glavni program**

namespace Vezbe8novembar

{

internal class Program

{

static void Main(string[] args)

{

string path = @"c:\tmp\Log.txt";

string line= "4.189.158.117 - - [21/Sep/2015:07:59:14 -0400] \"GET /favicon.ico HTTP/1.0\" 404 621";

LogParser.parseLine(line);

}

}

}

Rezultat izvrsavanja  
A screenshot of a computer

Description automatically generated