Федеральное государственное образовательное бюджетное учреждение высшего образования «Финансовый университет при Правительстве РФ»

КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

Лабораторная работа № 8

по дисциплине инфокоммуникационные системы и сети

Тема: HTTPLisener, HTTPClient

Выполнил студент

Ефимов К.С

Группа 3ПКС-420

Вариант: 6

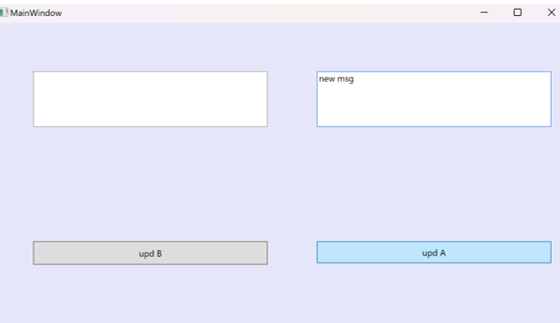
Проверил

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

оценка подпись

Москва, 2023

Цель: разобрать тему HTTPLisener, HTTPClient и работу с HTTP запросами на языке С# с помощью разработки WPF -приложения

Выполнение программы:  



Код программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Net;

using System.Net.Http;

using System.IO;

namespace HTTP\_lab8

{

public static class Ext\_HttpListener

{

public static System.Net.HttpListener Set\_Prefixes\_Add(this System.Net.HttpListener \_this, string \_str = "http://127.0.0.1:8888/connection/", System.Boolean \_IsOpen = false)

{

\_this.Prefixes.Add(\_str);

if (\_IsOpen) System.Diagnostics.Process.Start(\_str);

return \_this;

}

/// <summary>

/// Начинаем фоном прослушивать входящие подключения

/// </summary>

/// <param name="\_this"></param>

/// <returns></returns>

public static System.Net.HttpListener Set\_Start(this System.Net.HttpListener \_this)

{ \_this.Start(); return \_this; }

public static System.Net.HttpListener Set\_Stop(this System.Net.HttpListener \_this)

{ \_this.Stop(); return \_this; }

public static System.Net.HttpListener Get\_ContextAsync(this System.Net.HttpListener \_this, System.Action<HttpListenerContext> A)

{ A(\_this.GetContextAsync().GetAwaiter().GetResult()); return \_this; }

public static System.Net.HttpListener Get\_ContextAsync\_WhileTrue(this System.Net.HttpListener \_this, System.Action<HttpListenerContext> A)

{ while (true) \_this.Get\_ContextAsync(A); return \_this; }

}

public static class Ext\_String

{

public static byte[] Get\_Encoding\_UTF8\_Bytes(this System.String \_this) { return Encoding.UTF8.GetBytes(\_this); }

}

public static class Ext\_HttpListenerResponse

{

public static System.Net.HttpListenerResponse Set\_Bytes(this System.Net.HttpListenerResponse \_this, byte[] \_Bytes)

{

\_this.ContentLength64 = \_Bytes.Length;

Stream output = \_this.OutputStream;

output.WriteAsync(\_Bytes, 0, \_Bytes.Length);

output.FlushAsync();

return \_this;

}

}

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

///

public partial class MainWindow : Window

{

public MainWindow()

{

System.Threading.Tasks.Task \_TaskA = new Task(() =>

new HttpListener()

.Set\_Prefixes\_Add("http://127.0.0.1:8881/connection/", \_IsOpen: false)

.Set\_Start()

.Get\_ContextAsync\_WhileTrue(a => {

System.String \_strResponse = this.TextBoxA.Text + " from adr " + a.Request.RemoteEndPoint + ":" + a.Request.Url.ToString().Split('?')[1]);

Dispatcher.InvokeAsync(() => this.TextBoxA.Text += \_strResponse);

a.Response.Set\_Bytes(\_strResponse.Get\_Encoding\_UTF8\_Bytes());

})

);

\_TaskA.Start();

System.Threading.Tasks.Task \_TaskB = new Task(() =>

new HttpListener()

.Set\_Prefixes\_Add("http://127.0.0.2:8882/connection/", \_IsOpen: false)

.Set\_Start()

.Get\_ContextAsync\_WhileTrue(a => {

System.String \_strResponse = this.TextBoxA.Text + " from adr " + a.Request.RemoteEndPoint + ":" + a.Request.Url.ToString().Split('?')[1]);

Dispatcher.InvokeAsync(() => this.TextBoxB.Text += \_strResponse);

a.Response.Set\_Bytes(\_strResponse.Get\_Encoding\_UTF8\_Bytes());

})

);

\_TaskB.Start();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

new HttpClient().GetStringAsync("http://127.0.0.2:8882/connection/").GetAwaiter().GetResult();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

new HttpClient().GetStringAsync("http://127.0.0.1:8881/connection/").GetAwaiter().GetResult();

}

}

}

Вывод: разобрана тема HTTPLisener, HTTPClient и работа с HTTP запросами на языке С# с помощью разработки WPF-приложения.