Akademia Marynarki Wojennej

|  |  |  |  |
| --- | --- | --- | --- |
| Imię i nazwisko | | Grupa | Data ćwiczenia |
| Maciej Maliszewski | | 125NCI\_A | 06.12.2015 r. |
| Tytuł ćwiczenia  **6** | **Web Socket - komunikacja w czasie rzeczywistym.** | | |
| ***Programowanie równoległe i rozproszone***  ***dr inż.A.Zacniewski*** | | | |

Za pomocą wybranego przez siebie języka programowania zademonstruj działanie technologii Web Socket.

Program który wykonałem przyjmuję wiadomość z formularza i wysyła ją do servera napisanego dzięki technologii ASP.Net 4.5.

Utworzono pusty projekt ASP.Net w VS 2012 po czym dodano poprzez Nuget pakiet Microsoft.Websockets.

Ściągnięto bibliotekę Jquery również przez Nuget.

Dodano plik strony html o nazwie index oraz wpisano w nią kod:

<!doctype html>

<head>

<script src="Scripts/jquery-1.7.2.min.js" type="text/javascript"></script>

<script type="text/javascript">

$(document).ready(function () {

var name = prompt('what is your name?:');

var url = 'ws://' + window.location.hostname + window.location.pathname.replace('index.htm', 'ws.ashx') + '?name=' + name;

alert('Connecting to: ' + url);

ws = new WebSocket(url);

ws.onopen = function () {

$('#messages').prepend('Connected <br/>');

$('#cmdSend').click(function () {

ws.send($('#txtMessage').val());

$('#txtMessage').val('');

});

};

ws.onmessage = function (e) {

$('#chatMessages').prepend(e.data + '<br/>');

};

$('#cmdLeave').click(function () {

ws.close();

});

ws.onclose = function () {

$('#chatMessages').prepend('Closed <br/>');

};

ws.onerror = function (e) {

$('#chatMessages').prepend('Oops something went wront <br/>');

};

});

</script>

</head>

<body>

<input id="txtMessage" />

<input id="cmdSend" type="button" value="Send" />

<input id="cmdLeave" type="button" value="Leave" />

<br />

<div id="chatMessages" />

</body>

</html>

Następnie dodano http handler o nazwiews.ashx i wpisano w niego kod

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using Microsoft.Web.WebSockets;

namespace WebSockets

{

public class WSHttpHandler : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

if (context.IsWebSocketRequest

context.AcceptWebSocketRequest(new TestWebSocketHandler());

}

public bool IsReusable

{

get

{

return false;

}

}

}

}

Utworzono nową klasę o nazwie TestWebSocketHandler i uzupełniono ją kodem:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

using System.Web;

using Microsoft.Web.WebSockets;

namespace WebSockets

{

public class TestWebSocketHandler : WebSocketHandler

{

private static WebSocketCollection clients = new WebSocketCollection();

private string name;

public override void OnOpen()

{

this.name = this.WebSocketContext.QueryString["name"];

clients.Add(this);

clients.Broadcast(name + " has connected.");

}

public override void OnMessage(string message)

{

clients.Broadcast(string.Format("{0} said: {1}", name, message));

}

public override void OnClose()

{

clients.Remove(this);

clients.Broadcast(string.Format("{0} has gone away.", name));

}

}

}