[#chat\_box](https://www.youtube.com/results?q=%23chat_box){ position:fixed; right:5px; bottom:0px; width:200px; font-size: 12px; line-height: 16px; border: [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); } [#chat\_body](https://www.youtube.com/results?q=%23chat_body){ background:#edeef1; height:400px; padding:10px 10px; border: solid 1px [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); } .chat\_head{ background:#f39c12; color:white; padding: 10px; height: 36px; vertical-align:middle; font-size: 14px; font-weight: bold; cursor:pointer; border-radius:5px 5px 0px 0px; border: [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); } .msg\_box{ position:fixed; bottom:-5px; width:250px; background:white; border-radius:5px 5px 0px 0px; border: [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); } [#msg\_head](https://www.youtube.com/results?q=%23msg_head){ background:#47639e; color:white; padding: 5px; height: 36px; vertical-align:middle; font-size: 14px; font-weight: bold; cursor:pointer; border-radius:5px 5px 0px 0px; border: [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); position: relative; } [#msg\_body](https://www.youtube.com/results?q=%23msg_body){ background:#edeef1; height:200px; font-size:12px; padding:10px; overflow: inherit; overflow-x: hidden; border: solid 1px [#abb2c1](https://www.youtube.com/results?q=%23abb2c1); } .msg\_input{ width:100%; padding: 10px 8px 6px; border: 1px solid white; border-top:1px solid [#DDDDDD](https://www.youtube.com/results?q=%23DDDDDD); -webkit-box-sizing: border-box; /\* Safari/Chrome, other WebKit \*/ -moz-box-sizing: border-box; /\* Firefox, other Gecko \*/ box-sizing: border-box; word-wrap: break-word; } .closechat{ float:right; font-size: 16px; cursor:pointer; } .minimize{ float:right; cursor:pointer; padding-right:5px; } [#userchat](https://www.youtube.com/results?q=%23userchat){ position:relative; padding:5px 25px; } [#userchat](https://www.youtube.com/results?q=%23userchat):before{ content:''; position:absolute; background:#009e1d; height:10px; width:10px; left:5px; top:10px; vertical-align:middle; border-radius:5px; } [#userlist](https://www.youtube.com/results?q=%23userlist){ position:relative; padding: 5px 5px 5px 30px; vertical-align:middle; } [#userlist](https://www.youtube.com/results?q=%23userlist):hover{ background:#dcdee3; cursor:pointer; margin-left: 25px; } [#userlist](https://www.youtube.com/results?q=%23userlist):after{ content:''; position:absolute; background:#009e1d; height:8px; width:8px; right: 8px; top:8px; border-radius:5px; } [#msg\_a](https://www.youtube.com/results?q=%23msg_a){ word-wrap: break-word; position:relative; background:#e0edff; padding:10px; min-height:10px; margin-bottom:5px; margin-right:10px; border-radius:5px; border: 1px solid [#bcc7d6](https://www.youtube.com/results?q=%23bcc7d6); } /\*#msg\_a:before{ content:""; word-wrap: break-word; position:absolute; width:0px; height:0px; border: 10px solid; border-color: transparent [#e0edff](https://www.youtube.com/results?q=%23e0edff) transparent transparent; left:-20px; top:7px; }\*/ [#msg\_b](https://www.youtube.com/results?q=%23msg_b){ word-wrap: break-word; background:#fefefe; padding:5px; min-height:15px; margin-bottom:5px; position:relative; margin-left:30px; border-radius:5px; border: 1px solid [#d5d5d5](https://www.youtube.com/results?q=%23d5d5d5); vertical-align:middle; } .avatar { float:left; width: 25px; height: 25px; vertical-align:middle; }﻿

using System; using System.Collections.Generic; using System.Web.Security; using System.Web.UI; using BIX2.BAL; using BIX2.DAL; namespace BIX2.Admins { public partial class Admin : MasterPage { protected static string userName { get; set; } protected static string userImage { get; set; } protected static string userFullName { get; set; } protected void Page\_Load(object sender, EventArgs e) { if (!Page.User.Identity.IsAuthenticated) { FormsAuthentication.RedirectToLoginPage(); } if (!Page.IsPostBack) { getUserInfo(); } } private void getUserInfo() { if (!string.IsNullOrEmpty((string) Session["UserId"])) { userName = (string) Session["UserName"]; var userId = (string) Session["UserId"]; var listU = UsersService.Users\_GetById(userId); userImage = listU[0].UserImage; var EmpId = listU[0].EmpID; var listE = new List<Employee>(); listE = EmployeeService.Employee\_GetById(EmpId); userFullName = listE[0].LastName + " " + listE[0].FirstName; } } } }﻿

HTML <!-- jQuery 2.1.4 --> <script src="../Scripts/jquery-2.1.4.min.js"></script> <!-- SignalR chat js--> <script> $(document).ready(function () { $("#chat\_body").slideToggle("slow"); $(".msg\_wrap").slideToggle("slow"); }); $(function () { startChatHub(); }); var nickname = ""; var avatarurl = ""; function startChatHub() { var chat = $.connection.chatHub; // Get the user name. nickname = "<%= userName %>"; avatarurl = "<%= userImage %>"; chat.client.online = function (name) { // Update list of users if (name == nickname) { $("#chat\_body").append("<img src=" + avatarurl + " class=\"avatar\" /><div id=\"userlist\">" + name + "</div>"); $("#msg\_head").append("<div id=userchat>" + name + "</div>"); } else { $("#chat\_body").append("<img src=\"../Images/user.jpg\" class=\"avatar\" /><div id=\"userlist\">" + name + "</div>"); $("#users").append("<option value=\"" + name + "\">" + name + "</option>"); } }; chat.client.enters = function (name) { $("#msg\_body").append("<div ><i>" + name + " signs in</i></div>"); $("#users").append("<option value=\"" + name + "\">" + name + "</option>"); //$("#chat\_body").append("<div id=userlist>" + name + "</div>"); $("#chat\_body").append("<img src=\"../Images/user.jpg\" class=\"avatar\" /><div id=\"userlist\">" + name + "</div>"); }; // Create a function that the hub can call to broadcast chat messages. chat.client.broadcastMessage = function (name, message) { //Interpret smileys message = message.replace(":)", "<img src=\"../Images/Emoticons/smile-2.png\" class=\"smileys\" />"); message = message.replace(":D", "<img src=\"../Images/Emoticons/smile-1.png\" />"); message = message.replace(":o", "<img src=\"../Images/Emoticons/smile-6.png\" />"); //display the message $("#msg\_body").append("<div class=\"border\"><span style=\"color:orange\">" + name + "</span>: " + message + "</div>"); //$("#msg\_body").append("<img src=" + avatarurl + " class=\"avatar\" />", message); }; chat.client.disconnected = function (name) { //Calls when someone leaves the page $("#msg\_body").append("<div ><i>" + name + " signs out</i></div>"); $("#chat\_body div").remove(":contains('" + name + "')"); $("#users option").remove(":contains('" + name + "')"); }; //scroll bar $(".chat\_head").click(function () { $("#chat\_body").slideToggle("slow"); }); $("#msg\_head").click(function () { $(".msg\_wrap").slideToggle("slow"); }); $(".close").click(function () { $(".msg\_box").hide(); }); $("#userlist").click(function () { $(".msg\_wrap").show(); $(".msg\_box").show(); }); // Start the connection. $.connection.hub.start().done(function () { //Calls the notify method of the server chat.server.notify(nickname, $.[connection.hub.id](http://connection.hub.id/)); $("#messagebox").keypress(function (e) { if (e.keyCode == 13) { var msg = "<div id=msg\_b>" + $("#messagebox").val() + "</div>"; $("#messagebox").val(""); if ($("#users").val() == "All") { //Call the Send method on the hub. chat.server.send(nickname, msg); $("#msg\_body").scrollTop($("#msg\_body")[0].scrollHeight); } else { chat.server.sendToSpecific(nickname, msg, $("#users").val()); //Clear text box and reset focus for next comment. $("#messagebox").val("").focus(); $("#msg\_body").scrollTop($("#msg\_body")[0].scrollHeight); } } }); }); } </script> <script src="../Scripts/jquery.signalR-2.2.0.min.js"></script> <script src="/signalr/hubs"></script>

ChatHub.cs using System; using System.Collections.Concurrent; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Web; using Microsoft.AspNet.SignalR; namespace BIX2 { public class ChatHub : Hub { static ConcurrentDictionary<string, string> dic = new ConcurrentDictionary<string, string>(); public void Send(string name, string message) { Clients.All.broadcastMessage(name, message); } public void SendToSpecific(string name, string message, string to) { Clients.Caller.broadcastMessage(name, message); Clients.Client(dic[to]).broadcastMessage(name, message); } public void Notify(string name, string id) { if (dic.ContainsKey(name)) { Clients.Caller.differentName(); } else { dic.TryAdd(name, id); foreach (KeyValuePair<String, String> entry in dic) { Clients.Caller.online(entry.Key); } Clients.Others.enters(name); } } public override Task OnDisconnected(bool stopCalled) { var name = dic.FirstOrDefault(x => x.Value == Context.ConnectionId.ToString()); string s; dic.TryRemove(name.Key, out s); return Clients.All.disconnected(name.Key); } } }﻿