

## Web to Phone text message

Description: This feature will enable user to chat with multiple mobile numbers via web chat window.

User will use this feature and can chat from website to multiple phone numbers, this will be a normal text message which will be sent using DID API.

No need to login anywhere. User can start chatting with multiple numbers from multiple chat windows on the page.

Set Webhook Url:

<http://domainname.com/MultiUserChat/ReceiveSms>

## Web.config settings:

```
<add key="DidNumber" value="1XXXXXXXXX"/>
<add key="SmsApiBaseUrl"
value="https://api.didforsale.com/didforsaleapi/index.php/api/SMS/send"/>
<add key="ApiKey" value="Get API Key from your DIDforsale Portal account"/>

<connectionStrings><add name="DIDWebToPhoneTextEntities"
connectionString="metadata=res://*/Models.didwebtophonetextchat.csdl|res://*/Models.didwe
btophonetextchat.ssd1|res://*/Models.didwebtophonetextchat.msl;provider=System.Data.SqlCl
ient;provider connection string=&quot;;data
source=(LocalDB)\MSSQLLocalDB;attachdbfilename=|DataDirectory|\DIDWebToPhoneText.mdf;inte
grated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" /></connectionStrings>
```

## Coding Description:

DIDWebToPhoneText/Controllers/HomeController.cs

```
using DIDWebToPhoneText.Models;
using DIDWebToPhoneText.Services;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;
using System.Web;
using System.Web.Configuration;
using System.Web.Mvc;
```

```

namespace DIDWebToPhoneText.Controllers
{
    public class MultiUserChatController : Controller
    {
        // GET: MultiUserChat

        public string didNumber =
WebConfigurationManager.AppSettings["DidNumber"].ToString();
        public string apiKey = WebConfigurationManager.AppSettings["ApiKey"].ToString();
        public string smsApiBaseUrl =
WebConfigurationManager.AppSettings["SmsApiBaseUrl"].ToString();
        public ActionResult Index()
        {
            using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
            {
                var result = db.GetDistNumber().ToList();
                return View(result);
            }
        }
        [HttpPost]
        public ActionResult Message(string number)
        {
            if (!string.IsNullOrEmpty(number))
            {
                using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
                {
                    var result = db.Inboxes.Where(g => g.PhoneNumber == number).ToList();
                    return Json(result);
                }
            }
            return Json("");
        }
        [HttpPost]
        public ActionResult AllMessages()
        {
            using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
            {
                var result = db.Inboxes.ToList();
                return Json(result);
            }
        }
        public ActionResult SentSms(string Number, string message)
        {
            string status = Utility.SMSType.failed;
            try
            {
                var response = HttpPostRequest(Number, message);
                if (response.Contains("100"))
                {
                    status = Utility.SMSType.sent;
                }
            }
            catch
            {
            }
            Inbox inbox = new Inbox();
            inbox.DateTime = DateTime.Now;
        }
    }
}

```

```

inbox.PhoneNumber = Number;
inbox.DedicatedNumber = didNumber;
inbox.Message = message;
inbox.Type = status;
inbox.Status = false;

using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
{
    var q = db.Inboxes.Add(inbox);
    db.SaveChanges();
    return Json(status);
}
}
public string HttpPostRequest(string mobileNumber, string text)
{
    string result = string.Empty;
    //DIDforsale api only accept json data format

    var httpWebRequest = (HttpWebRequest)WebRequest.Create(smsApiBaseUrl +
apiKey);
    httpWebRequest.ContentType = "application/json";
    httpWebRequest.Method = "POST";
    using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
    {
        var json = "{\"from\":\"" + didNumber + "\",\"to\":\"" + mobileNumber +
"\",\"text\":\"" + text + "\"}";
        streamWriter.Write(json);
        streamWriter.Flush();
        streamWriter.Close();
    }
    ServicePointManager.ServerCertificateValidationCallback = delegate (object s,
X509Certificate certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors) { return
true; };
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Ssl3 |
SecurityProtocolType.Tls12 | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls;
    var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
    {
        result = streamReader.ReadToEnd();
    }
    return result;
}
}
}

```

DIDWebToPhoneText\Views\MultiUserChat\Index.cshtml

```

@{
    Layout = null;
}

@model IEnumerable<DIDWebToPhoneText.Models.GetDistNumber_Result>

<!DOCTYPE html>

<html>

```

```

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link href="@Url.Content("~/css/style-new.css")" rel="stylesheet">
  <!-- Bootstrap CSS -->
  <link href="@Url.Content("~/css/bootstrap.min.css")" rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css" rel="stylesheet">
  <link href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css"
rel="stylesheet">

  <title>Inbox</title>
</head>
<body>
  <div class="tbl_section">
    <div class="container">
      <div class="full_tbl">
        <h2 class="active">Inbox</h2>
        <div class="table-responsive">
          <table class="table">
            <thead class="thead-default">
              <tr>
                <th>Phone Number</th>
                <th>Received Message</th>
                <th>DateTime</th>
                <th></th>
              </tr>
            </thead>
            <tbody>
              @if (Model.Count() > 0)
              {
                foreach (var item in Model)
                {
                  <tr>
                    <th scope="row">@item.PhoneNumber</th>
                    <td>@item.Message</td>
                    <td>@item.DateTime</td>
                    <td><a href="#" data-number="@item.PhoneNumber"
class="viewchat">View Chat</a></td>
                  </tr>
                }
              }
              else {
                <tr>
                  <td></td>
                  <td></td>
                  <td>No Records</td>
                  <td></td>
                </tr>
              }
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

```



```

        message = '<li>\
                    <div class="message-data">\
                        <span class="message-data-
name"><i class="fa fa-circle online"></i>'+ value.PhoneNumber + '</span>\
                        <span class="message-data-time">'
+ eval(value.DateTime.replace(/\Date\((\d+)\)\//gi, "new Date($1)").toUTCString() +
'</span>\
                    </div>\
                    <div class="message my-message">'+
value.Message + '</div>\
                    </li>';
    }
    else {
        message = '<li class="clearfix">\
                    <div class="message-data righttxt">\
                        <span class="message-data-time">'
+ eval(value.DateTime.replace(/\Date\((\d+)\)\//gi, "new Date($1)").toUTCString() +
'</span> &nbsp;';\
                    <span class="message-data-
name">'+ value.DedicatedNumber + '<i class="fa fa-circle me"></i></span>\
                    </div>\
                    <div class="message other-message
float-right">\
                        '+ value.Message + '\
                    </div>\
                    </li>';
    }
    test.find('.chat-history ul').append(message);
    test.find('.chat-history').animate({ scrollTop:
test.find('.chat-history').prop("scrollHeight") }, 100);
    });
    },
    error: function (result) {
        alert(result.message);
    },
    });
    });
    $(document).on('click', '.fa.fa-window-close-o', function () {
        //$('.table-responsive').find('a[data-number="12032661755"]').on('click',
a);
        var test = $(this).closest('.chatpopup').remove();
    });
    $('#_m_chat').on('keypress', '.tbl_section.chatpopup .newmessage', function
(e) {
        if (e.which == 13) {
            e.preventDefault();
            var currentnewmessage = $(this);
            var number =
$(this).closest('.chatpopup').find('.chatnumber').html();
            var text = $(this).val().trim();
            var url = "@Url.Action("SentSms", "MultiUserChat)";
            if (text != null && text != '') {
                $.ajax({
                    url: url,
                    method: "POST",
                    data: { Number: number, message: text },
                    datatype: "application/json",

```

```

        success: function (result) {
            $(currentnewmessage).val('');
        },
        error: function (result) {
            alert(result.message);
        },
    });
    }
});

setInterval(function () {
    var url = "@Url.Action("AllMessages", "MultiUserChat")";
    $.ajax({
        url: url,
        method: "POST",
        success: function (result) {
            $.each(result, function (index, value) {
                var message;
                if (value.Type ==
'@DIDWebToPhoneText.Services.Utility.SMSType.received') {
                    message = '<li>\
                                <div class="message-data">\
                                    <span class="message-data-
name"><i class="fa fa-circle online"></i>'+ value.PhoneNumber + '</span>\
                                    <span class="message-data-time">
+ eval(value.DateTime.replace(/\Date((\d+)\))/gi, "new Date($1)").toUTCString() +
'</span>\
                                </div>\
                                <div class="message my-message">'+
value.Message.trim() + '</div>\
                                </li>';
                }
                else {
                    message = '<li class="clearfix">\
                                <div class="message-data righttxt">\
                                    <span class="message-data-time">
+ eval(value.DateTime.replace(/\Date((\d+)\))/gi, "new Date($1)").toUTCString() +
'</span> &nbsp;\
                                <span class="message-data-
name">'+ value.DedicatedNumber + '<i class="fa fa-circle me"></i></span>\
                                </div>\
                                <div class="message other-message
float-right">'+ value.Message.trim() + '</div>\
                                </li>';
                }
            }
            $(".m_chat").find('.chatnumber').each(function () {
                var number = $(this).html();
                if (value.PhoneNumber == number) {
                    var messageIsExist = false;
                    $(this).closest('.chatpopup').find('.chat-history
.message').each(function () {
                        if ($(this).html().toLowerCase().trim() ==
value.Message.toLowerCase().trim()) {
                            messageIsExist = true;
                        }
                    })
                }
            })
        }
    });
}
});

```

```

        if (!messageIsExist) {
            $(this).closest('.chatpopup').find('.chat-history
ul').append(message);
            $(this).closest('.chatpopup').find('.chat-
history').animate({ scrollTop: $(this).closest('.chatpopup').find('.chat-
history').prop("scrollHeight") }, 1000);
        }
    });

    //$(this).closest('.chatpopup').find('.chat-
history').append('<ul>' + message + '</ul>');
    });
    },
    error: function (result) {
        alert(result.message);
    },
    });
    }, 3000);
    });
</script>
</body>
</html>

```

DIDWebToPhoneText\Services\Utility.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace DIDWebToPhoneText.Services
{
    public class Utility
    {
        public class SMSType
        {
            public const string received = "Received", sent = "Sent", failed =
"Failed";
        }
    }
}

```

In this project user need to add a MS SQL Database file in App\_Data with name DIDWebToPhoneText.mdf and add table (Account & Inbox) and procedure(GetDistNumber).

Add Stored Procedure:

```

Create proc [dbo].[GetDistNumber]
as
select
Inbox.Id,Inbox.PhoneNumber,Inbox.DedicatedNumber,Inbox.Message,Inbox.DateTime,Inbox.Type,
Inbox.Status from Inbox, (SELECT MAX(Id) as Id FROM Inbox where Inbox.Type='Received'
GROUP BY PhoneNumber) as temp where Inbox.Id=temp.Id

```



Then add an ADO.Net Entity Data Model with name didwebtophonetextchat.edmx after that you can access table model in project like as Account model and Inbox model.

DIDWebToPhoneText/Models/Account.cs

```
namespace DIDWebToPhoneText.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Account
    {
        public int Id { get; set; }
        public string UserName { get; set; }
        public string Password { get; set; }
    }
}
```

DIDWebToPhoneText/Models/Inbox.cs

```
namespace DIDWebToPhoneText.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Inbox
    {
        public int Id { get; set; }
        public string PhoneNumber { get; set; }
        public string DedicatedNumber { get; set; }
        public string Message { get; set; }
        public System.DateTime DateTime { get; set; }
        public string Type { get; set; }
        public Nullable<bool> Status { get; set; }
    }
}
```