

Web to Phone text message

Description: This feature will enable user to send messages from website to phone number.

User will use this feature and can send messages from website to phone number, this will be a normal text message which will be sent using DID API.

User will login to index page, he will see inbox on the page where all the messages will be available from various numbers to DID number. By clicking over right side button "view chat" chat thread will open for that particular number and user can send message from below typing box.

Set Webhook Url:

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

Web.config settings:

```
<add key="DidNumber" value="16505340994"/>
<add key="SmsApiBaseUrl"
value="https://api.didforsale.com/didforsaleapi/index.php/api/SMS/send/" />
<add key="ApiKey" value="SMjKQ1v67K320e6af48fa9d99b4778132a11ae1d31"/>

<connectionStrings><add name="DIDWebToPhoneTextEntities"
connectionString="metadata=res://*/Models.didwebtophonetextchat.csdl|res://*/Models.didwebtophonetextchat.ssd1|res://*/Models.didwebtophonetextchat.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=(LocalDB)\MSSQLLocalDB;attachdbfilename=|DataDirectory|\DIDWebToPhoneText.mdf;integrated 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 HomeController : Controller
    {
        // GET: Home
        // Get values from web.config file
        public string didNumber =
WebConfigurationManager.AppSettings["DidNumber"].ToString();
        public string apiKey = WebConfigurationManager.AppSettings["ApiKey"].ToString();
        public string smsApiBaseUrl =
WebConfigurationManager.AppSettings["SmsApiBaseUrl"].ToString();
        public ActionResult Index()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Index(Account user)
        {
            if (ModelState.IsValid)
            {
                using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
                {
                    var result = db.Accounts.Where(u => u.UserName.ToLower() ==
user.UserName.ToLower().Trim() && u.Password == user.Password).FirstOrDefault();
                    if (result != null)
                    {
                        Session["UserId"] = result.Id;
                        return RedirectToAction("Inbox");
                    }
                    else {
                        ViewData["error"] = "Invalid User Name Or Password.";
                    }
                }
            }
            else {
                ViewData["error"] = "Invalid Input";
            }
            return View();
        }

        public ActionResult Inbox()
        {
            if (Session["UserId"] == null)
            {
                return RedirectToAction("index");
            }
            using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
            {
                var result = db.GetDistNumber().ToList();
            }
        }
    }
}

```

```

        return View(result);
    }
}
public ActionResult Message(string number)
{
    if (Session["UserId"] == null)
    {
        return RedirectToAction("index");
    }
    if (!string.IsNullOrEmpty(number))
    {
        ViewBag.Number = number;
        using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
        {
            var result = db.Inboxes.Where(g => g.PhoneNumber == number).ToList();
            return View(result);
        }
    }
    return View();
}

public ActionResult ReceiveSms()
{
    String from_number = Request.Form["From"]; // Sender's phone number
    String to_number = Request.Form["To"]; // Receiver's phone number
    String text = Request.Form["Text"]; // The text which was received

    if (!string.IsNullOrEmpty(from_number))
    {
        from_number = from_number.Replace("+", "");
        from_number = from_number.Replace(" ", "");
    }
    Inbox inbox = new Inbox();
    inbox.DateTime = DateTime.Now;
    inbox.PhoneNumber = from_number;
    inbox.DedicatedNumber = to_number;
    inbox.Message = text;
    inbox.Type = Utility.SMSType.received;
    inbox.Status = false;

    using (DIDWebToPhoneTextEntities db = new DIDWebToPhoneTextEntities())
    {
        var q = db.Inboxes.Add(inbox);
        db.SaveChanges();
    }
    return RedirectToAction("index");
}

public ActionResult SentSms(string Number, string message)
{
    if (Session["UserId"] == null)
    {
        return RedirectToAction("index");
    }
    try
    {
        var status = HttpPostRequest(Number, message);
        if (status.Contains("100"))

```

```

        {
            Inbox inbox = new Inbox();
            inbox.DateTime = DateTime.Now;
            inbox.PhoneNumber = Number;
            inbox.DedicatedNumber = didNumber;
            inbox.Message = message;
            inbox.Type = Utility.SMSType.sent;
            inbox.Status = false;

            using (DIDWebToPhoneTextEntities db = new
DIDWebToPhoneTextEntities())
            {
                var q = db.Inboxes.Add(inbox);
                db.SaveChanges();
            }
        }
    }
    catch
    {
    }
    return RedirectToAction("Message", new { number = Number });
}
public ActionResult Logout()
{
    Session["UserId"] = null;
    Session.Abandon();
    return RedirectToAction("index");
}
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\Home\Index.cshtml

```
@{  
    Layout = null;  
}  
  
<!DOCTYPE html>  
  
<html>  
<head>  
    <!-- Required meta tags -->  
    <meta charset="utf-8">  
    <title>Login</title>  
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
    <link href="@Url.Content("~/css/style.css")" rel="stylesheet">  
    <!-- Bootstrap CSS -->  
    <link href="@Url.Content("~/css/bootstrap.min.css")" rel="stylesheet">  
    <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>  
    <script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  
  
</head>  
<body>  
    <div class="wrapper fadeInDown">  
        <div class="loginform">  
            @if (ViewData["error"] != null)  
            {  
                <div class="alert alert-danger in">  
                    <a href="#" class="close" data-dismiss="alert">&times;</a>  
                    <strong>Error!</strong> @ViewData["error"]  
                </div>  
            }  
            <!-- Tabs Titles -->  
            <h2 class="active text-center">Login</h2>  
            <!-- Login Form -->  
            @using (Html.BeginForm("index", "home", FormMethod.Post))  
            {  
                <input type="text" id="login" class="second" name="UserName"  
placeholder="User Name">  
                <input type="password" id="password" class="third" name="Password"  
placeholder="Password">  
                <input type="submit" class="fourth" value="Log In">  
            }  
            @*@using (Html.BeginForm("ReceiveSms", "home", FormMethod.Post))  
            {  
                <input type="text" id="From" class="first" name="From"  
placeholder="From">  
                <input type="text" id="To" class="second" name="To" placeholder="To">
```



```

                <td>@item.DateTime</td>
                <td><a href="@Url.Action("Message", new { number
= item.PhoneNumber }) ">View Chat</a></td>
            </tr>
        }
    }
    else {
        <tr>
            <td></td>
            <td></td>
            <td>No Records</td>
            <td></td>
        </tr>
    }
</tbody>
</table>
</div>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
<script>
    $(document).ready(function () {
        $('table').DataTable();
    });
</script>
</body>
</html>

```

DIDWebToPhoneText\Views\Home\Message.cshtml

```

@{
    Layout = null;
}
@model IEnumerable<DIDWebToPhoneText.Models.Inbox>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link href="@Url.Content("~/css/style.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">
    <title>Message</title>
</head>
<body>
    <div class="tbl_section">
        <div class="container">
            <div class="full_tble">
                <div class="row">
                    <div class="col-md-12">

```



```
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; }
    }
}
```