

## **Click-To-Call with Did Number:**

1. Visitor submits a web form with a phone number.
2. Your web application receives the submission and initiates an HTTP request to DID asking to initiate an outbound call.
3. DID receives the request and initiates a call to the user's phone number.
4. The user picks up the call.
5. After the call connects, we provide XML instructions to connect the user to our sales or support teams.

Code description:

### **1) Add some key in web.config file**

```
<add key="DidNumber" value="1XXXXXXXXX"/>
<add key="ApiKey" value="find your apikey from DID portal"/>
<add key="VoiceApiBaseUrl"
value="https://api.didforsale.com/didforsaleapi/index.php/api/V2/CallTermination/" />
```

### **2) Call the controller ClickToCall.Controllers/HomeController.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Collections.Specialized;
using System.IO;
using System.Net.Http;
using System.Threading.Tasks;
using System.Web.Mvc;
using ClickToCall.Services;
using System.Web.Configuration;

namespace ClickToCall.Controllers
{
    public class HomeController : Controller
    {
        public string didNumber =
WebConfigurationManager.AppSettings["DidNumber"].ToString();
        public string apiKey = WebConfigurationManager.AppSettings["ApiKey"].ToString();
        public string voiceApiBaseUrl =
WebConfigurationManager.AppSettings["VoiceApiBaseUrl"].ToString();
        ClickToCallService _clickToCall = new ClickToCallService();

        public ActionResult Index()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Index(string phonenumber)
        {
            Dictionary<string, string> values = new Dictionary<string, string>();
            values.Add("from", didNumber);
            values.Add("to", phonenumber);
        }
    }
}
```

```

        values.Add("apikey", apiKey);
        // if your project is live use url domain/home/calltoadmin
        values.Add("url", "http://www.domain/home/calltoadmin");
        var result = _clickToCall.HttpPostRequest(voiceApiBaseUrl, values);
        ViewBag.Result = result;
        return View();
    }
    public ActionResult CallToAdmin()
    {
        string response = "<Response><Say>Please hold we are connecting
you.</Say><Number>1XXXXXXXXX</Number></Response>";
        return this.Content(response, "text/xml");
    }
}
}
}

```

### 3) Create user input HTML page: ClickToCall/Views/Home/Index.cshtml

```

@{
    Layout = null;
}
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <form name="authform" method="post" action="/">
        <table id="phone">
            <tr><td>Phonenumber:</td><td><input type="text" name="phonenumber"
maxlength="15" id="phonenumber" /></td></tr>
        </table>
        <input type="Submit" name="callme" value="Call Me" />
    </form>
    <div>
        <span>
            @ViewBag.Result
        </span>
    </div>
</body>
</html>

```

### 4) Submit form with phone number.

```

[HttpPost]
public ActionResult Index(string phonenumber)
{
    Dictionary<string, string> values = new Dictionary<string, string>();
    values.Add("from", didNumber);
    values.Add("to", phonenumber);
    values.Add("apikey", apiKey);
}

```

```

        // if your project is live use url domain/home/calltoadmin
        values.Add("url", "http://www.trignobit.com/hello-world.php");
        var result = _clickToCall.HttpPostRequest(voiceApiBaseUrl, values);
        ViewBag.Result = result;
        return View();
    }

```

### 5) Initiate call to user phone number.

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using System.Web.Configuration;

namespace ClickToCall.Services
{
    interface IClickToCallService
    {
        string HttpPostRequest(string url, Dictionary<string, string> postParameters);
        string MyDictionaryToJson(Dictionary<string, string> dict);
    }
    class ClickToCallService : IClickToCallService
    {
        public string HttpPostRequest(string url, Dictionary<string, string>
postParameters)
        {
            string result = string.Empty;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Ssl3 |
SecurityProtocolType.Tls12 | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls;
            var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
            //DIDforsale api only accept json data format
            httpWebRequest.ContentType = "application/json";
            httpWebRequest.Method = "POST";
            using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
            {
                var json = MyDictionaryToJson(postParameters);
                streamWriter.Write(json);
                streamWriter.Flush();
                streamWriter.Close();
            }
            var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
            using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
            {
                result = streamReader.ReadToEnd();
            }
            return result;
        }
        public string MyDictionaryToJson(Dictionary<string, string> dict)
        {
            var entries = dict.Select(d => string.Format("\"{0}\": \"{1}\"", d.Key,
string.Join(",", d.Value)));
            return "{" + string.Join(",", entries) + "}";
        }
    }
}

```

```
    }  
  }  
}
```

## 6. Create XML instructions to connect the user to our sales or support teams.

```
public ActionResult CallToAdmin()  
{  
    string response = "<Response><Say>Please hold we are connecting  
you.</Say><Number>1XXXXXXXXXX</Number></Response>";  
    return this.Content(response, "text/xml");  
}
```