

DIDforsale Multilevel IVR system.

This is an IVR system with multiple levels. User can create an IVR system with multiple levels like:

User will select 1st option in IVR then system will provide multiple options to proceed further. Then again user will select 2nd option, this too reveal further options to select.

This way multiple levels can be created as per the business need.

Follow this: Set the webhook URL for DID phone number after logging in to user portal of DIDforSale.

Login> Go to DID configs under Products > Select DID number> Choose action manage routing and apply> SELECT YOUR CONFIGURATION AND CHOOSE FORWARD TO WEB URL

Webhook URL:

<http://domainname.com/multilevelivr>

Web.Config Settings

```
<add key="WebsitePath" value="http://domainname.com" />
<add key="SupportEmail" value="user@domainname.com" />
```

You must set website path and email for voicemail forwarding to email address after it's recorded.

MultilevelIVRController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Mail;
using System.Web;
using System.Web.Configuration;
using System.Web.Mvc;

namespace DidIVR.Controllers
{
    public class MultilevelIVRController : Controller
    {
        // GET: MultilevelIVR
        public ActionResult Index()
        {
            string response = "<Gather timeout=\"10\" action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
```

```

"/MultilevelIVR/GetDigits\" numDigits=\"1\" ><Say>If you are current client press 1 or
Not press 2? </Say></Gather>";
    return this.Content("<Response>" + response + "</Response>", "text/xml");
}
public ActionResult GetDigits(string digits)
{
    string response = string.Empty;
    switch (digits)
    {
        case "1":
            response = "<Gather timeout=\"10\" action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/GetSecondDigits\" numDigits=\"1\"><Say>To speak with a Sales
representative, press 1. To reach our Billing department, press 2. If you would like to
speak with an Operator, press 0.</Say></Gather>";
            break;
        case "2":
            response = "<Say>We are connecting you shortly. hold the line stay
with us.</Say><Dial action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/CallEvents\" ><Number>1XXXXXXXXXX</Number></Dial>";
            break;
        case "9":
            response = "<Say>Please Record your message after beep, to finish
reording press *.</Say><Record maxLength=\"30\" finishOnKey=\"*\" action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/RecordingUrl\" />";
            break;
        default:
            response = "<Say>We are not able to recognized your option thanks to
contact us. Good bye</Say><Hangup />";
            break;
    }
    return this.Content("<Response>" + response + "</Response>", "text/xml");
}
public ActionResult GetSecondDigits(string digits)
{
    string response = string.Empty;
    switch (digits)
    {
        case "0":
            response = "<Say>All Customer Support agent are busy on another call
please hold the line and stay with us.</Say><Dial action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/CallEvents\" ><Number>1XXXXXXXXXX</Number></Dial>";
            break;
        case "1":
            response = "<Say>Welcome you in sale department. All sale persons are
busy on another call. Please hold the line and stay with us</Say><Dial action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/CallEvents\" ><Number>1XXXXXXXXXX</Number></Dial>";
            break;
        case "3":
            response = "<Say>Welcome you in billing department.All persons are
busy on another call. Please hold the line and stay with us</Say><Dial action=\"\" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/CallEvents\" ><Number>1XXXXXXXXXX</Number></Dial>";
            break;
    }
}

```

```

        default:
            response = "<Say>We are not able to recognized your option thanks to
contact us. Good bye</Say><Hangup />";
            break;
        }
        return this.Content("<Response>" + response + "</Response>", "text/xml");
    }
    [HttpPost]
    public ActionResult CallEvents(string dialCallStatus)
    {
        string response = string.Empty;
        if (dialCallStatus == "completed")
        {
            return this.Content("<Root></Root>", "text/xml");
        }
        response = "<Gather timeout=\"10\" action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() +
"/MultilevelIVR/GetDigits\" numDigits=\"1\"><Say>Sorry our representative currently not
able to take your call, Please press 9 to start voicemail reording.</Say></Gather>";
        return this.Content("<Response>" + response + "</Response>", "text/xml");
    }

    [HttpPost]
    public ActionResult RecordingUrl(string recordingUrl, string callerid)
    {
        string response = string.Empty;
        if (string.IsNullOrEmpty(recordingUrl))
        {
            response = "<Say>No message recorded, Good bye.</Say><Hangup />";
        }
        else {
            response = "<Say>Thanks, we contact you shortly. Good bye.</Say><Hangup
/>";

            //To forwarded recorded voicemail to email
            SendEmail("New Voicemail Message from " + callerid, "You have a new
voicemail from " + callerid + ".<br/><br/><br/> Click this link to listen to the
message:<br/>" + recordingUrl);
        }
        return this.Content("<Response>" + response + "</Response>", "text/xml");
    }
    public void SendEmail(string subject, string body)
    {
        SmtpClient client = new SmtpClient();
        client.Port = 587;
        client.Host = "smtp.gmail.com";
        client.EnableSsl = true;
        client.Timeout = 10000;
        client.DeliveryMethod = SmtpDeliveryMethod.Network;
        client.UseDefaultCredentials = false;
        client.Credentials = new System.Net.NetworkCredential("fromemailaddress",
"password");
        MailMessage mm = new MailMessage("fromemailaddress",
WebConfigurationManager.AppSettings["SupportEmail"].ToString(), subject, body);
        mm.IsBodyHtml = true;
        mm.DeliveryNotificationOptions = DeliveryNotificationOptions.OnFailure;
        client.Send(mm);
    }
}

```

}

In the above code we have used gather verb to get the keypress input. On the keypress it will redirect to the action with key. We will get the key input using GetDigits action. Then the given logic above will be followed.

Again same process will be followed to redirect user then we will get the 2nd action "GetSecondDigits" by 2nd input and so on.