TECHNICAL TRAINING

Iframe integration for Forms.

INTEGRA

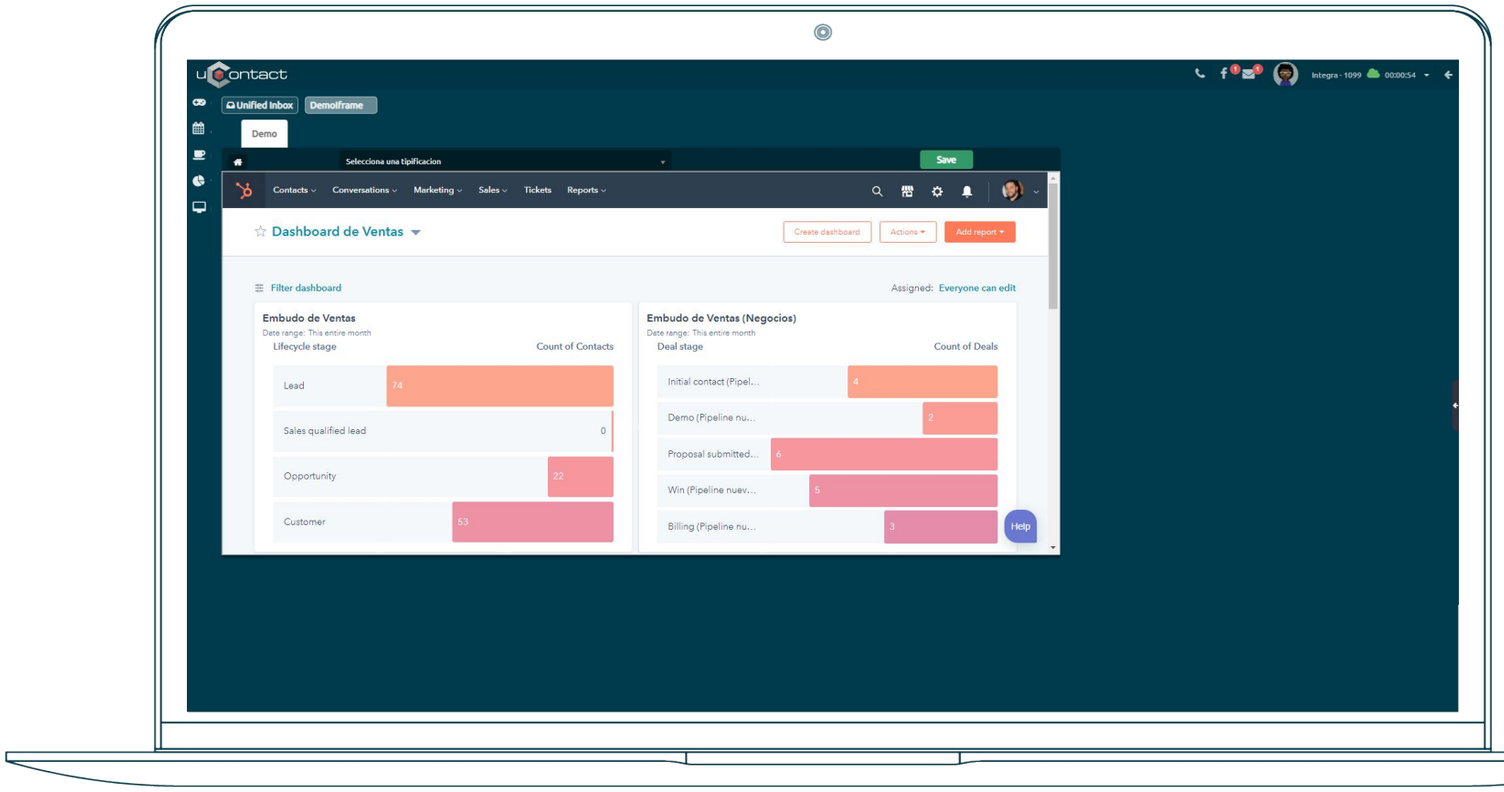
Table of contents

1. Introduction

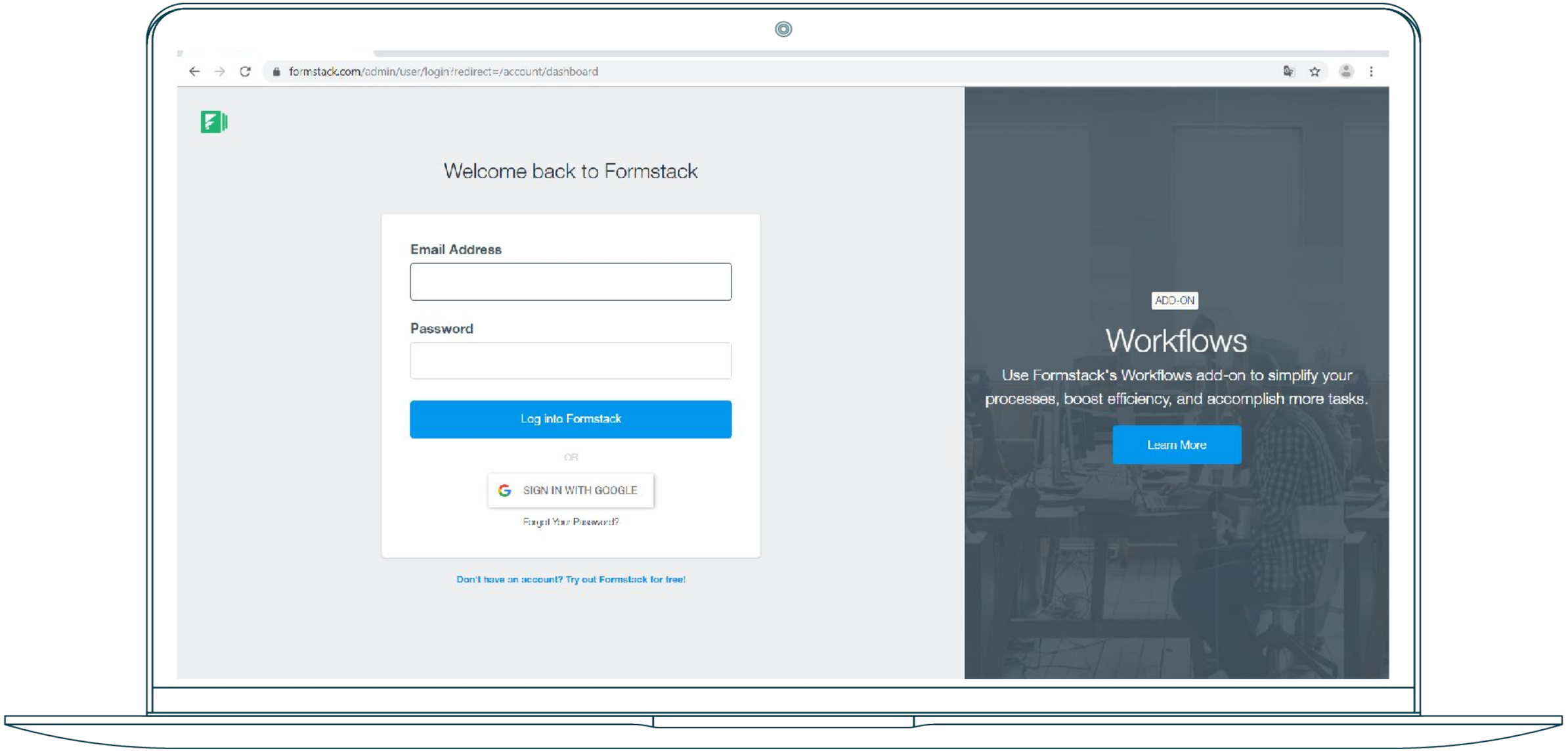
What is an IFrame?

An IFrame is a "container" or document within the HTML code of uContact's forms, which allows you to display another external page that is not part of the platform into it.

The system uses this methodology to deploy third-party systems used by the client into the system, allowing them to access the site in question without the need to change application or tab (through the form).



1.2 Look & feel (example)



2. Implementation

Below, you'll find a detailed example of the HTML code that must be included in order to create a brand new, tailor-made form using uContact's low-code programming Form Designer:

```
<section style="width: 100%;">
        <iframe id="page" src="" style="height: 100%;width: 100%;"></iframe>
</section>
```

In the 'Source' (src) variable, you must include the URL of the page you wish to be displayed by the iFrame. If it does not include sharing parameters (fixed), it can be included directly in the HTML code. Otherwise, it must be added using JavaScript, as shown in the following slide.

Javascript

```
function cargarIFrame() {
        var page =
'https://10.44.1.206/promociontelefonica_ucontact/index.php?cliente=${pyv.cliente}&username=99999999&answernumber=${ctiParseado.Callerid}campaign=${ctiParse
ado.Campaign}_';
        $('#EmbeddedPage').attr('src', page);
}
```

In the **'PAGE'** variable, you define the URL that will be invoked when the solution processes a client interaction (either inbound, outbound or through any interface) and the form loads the iFrame.

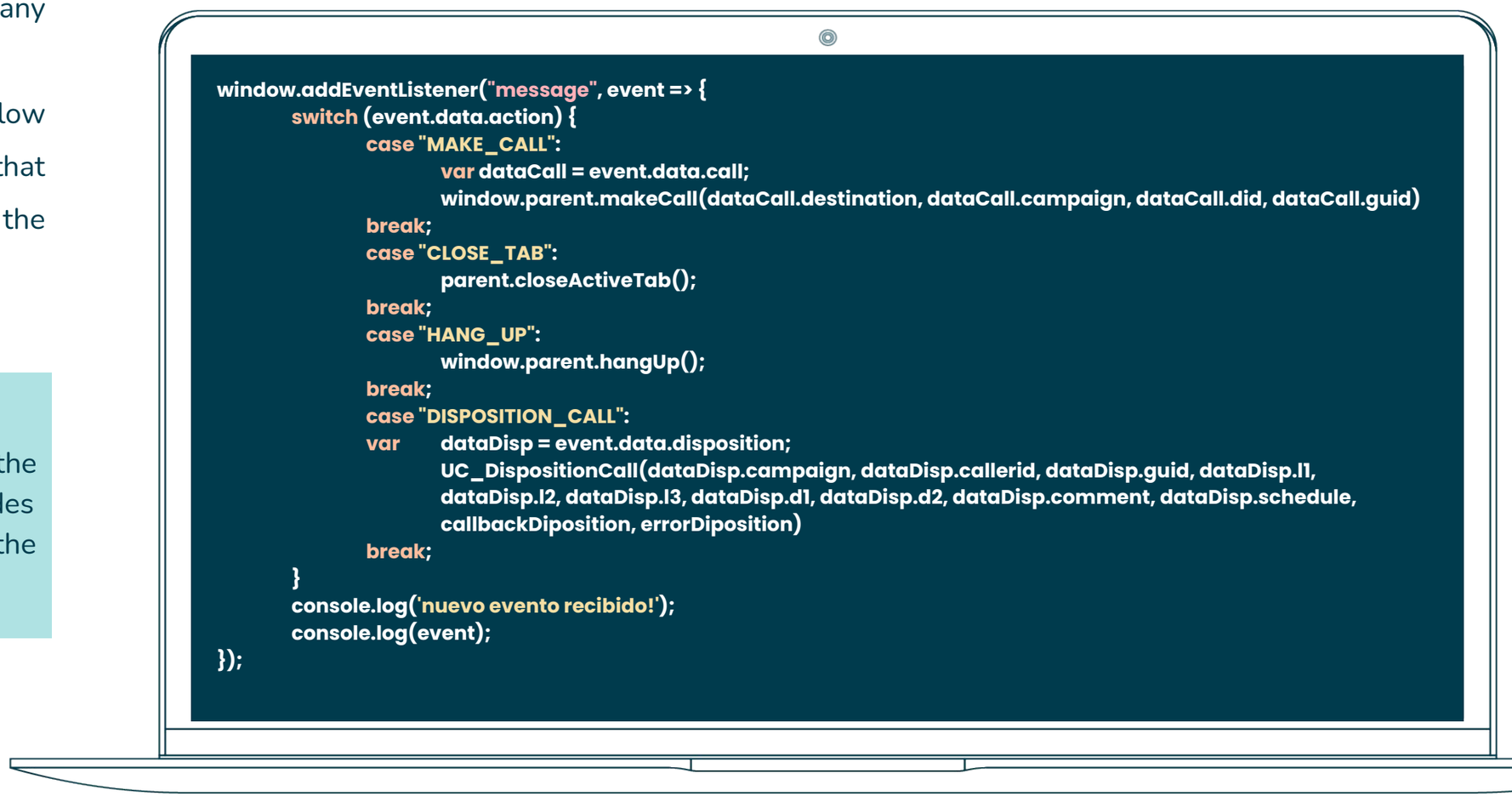
The URL of the third-party system must implement the variable capture as shown in the code in the image above, so that these variables can be worked by the external system. This is useful since there is certain information that is only handled by uContact (e.g.: the client's phone number) that will be necessary if you want to invoke any available method of the uContact API.

JavaScript Implementation

This implementation (part of the JavaScript code to be included in the uContact IFrame form) enables handling events in a form IFrame, in order to execute an specific behaviour and there is no need for any plugins to make this work.

The JavaScript code detailed below includes the instruction “SWITCH” that allows to identify the type of event of the variable event.data.action:

IMPORTANT:
This piece of code must be included in the uContact IFrame from only. Further slides shows the code to be included in the third-party system development.



```
window.addEventListener("message", event => {
        switch (event.data.action) {
                case "MAKE_CALL":
                        var dataCall = event.data.call;
                        window.parent.makeCall(dataCall.destination, dataCall.campaign, dataCall.did, dataCall.guid)
                break;
                case "CLOSE_TAB":
                        parent.closeActiveTab();
                break;
                case "HANG_UP":
                        window.parent.hangUp();
                break;
                case "DISPOSITION_CALL":
                var     dataDisp = event.data.disposition;
                        UC_DispositionCall(dataDisp.campaign, dataDisp.callerid, dataDisp.guid, dataDisp.l1,
                        dataDisp.l2, dataDisp.l3, dataDisp.d1, dataDisp.d2, dataDisp.comment, dataDisp.schedule,
                        callbackDiposition, errorDiposition)
                break;
        }
        console.log('nuevo evento recibido!');
        console.log(event);
});
```

Handled Events

The events that can be handled with the event.data.action variable are:

- **MAKE_CALL:** This event can be triggered from a button (or similar) within the third-party system implementation. uContact will call the number specified in the requested parameter.
- **HANG_UP:** This event is used to terminate the call. It is useful to terminate the call properly since sometimes the client forgets to hang up.
- **CLOSE_TAB:** This event allows to close the IFrame Form shown in the agent screen. This can be implemented as part of a button behaviour (or similar) after the call is terminated to clean the screen for the next interaction.
- **DISPOSITION_CALL:** This event will allow third-party to send the interaction (call) disposition once it is finished. The format in which must be sent is explained in next slides.

JavaScript Implementation (Third-party system)

- **MAKE_CALL:**

A dataSet needs to be defined in order to send the information related to the call that will be made.

The dataSet includes 4 variables as shown below. Destination and Campaign are mandatory since it is requested by the service in uContact. The rest of the variables are optional.
After the dataSet definition, you can see the line of code that invokes the MAKE_CALL event.

```
let dataCall = {
                destination: $CustomerNumber, (Number that will be called)
                campaign: $Campaign, (Name of the campaign, must be a manual outbound campaign created in the system)
                did: $SourceNumber, (Number displayed when the call is made) (Not Mandatory)
                guid: $CallID, (Call ID) (Not Mandatory)
};

window.parent.postMessage({ action: 'MAKE_CALL', call:dataCall}, '*');
```

JavaScript Implementation (Third-party system)

- **HANG_UP:**

Just include the following line in the JavaScript code implemented in the third-party system
In general, this is included in the behaviour of a button or similar as part of the actions to be performed after the on-click event.

```
window.parent.postMessage({ action: 'HANG_UP'}, '*');
```

JavaScript Implementation (Third-party system)

- **CLOSE_TAB:**

Just include the following line in the JavaScript code implemented in the third-party system. In general, this is included in the behaviour of a button or similar as part of the actions to be performed after the one-click event.

```
window.parent.postMessage({ action: 'CLOSE_TAB'}, '*');
```

JavaScript Implementation (Third-party system)

- **DISPOSITION_CALL:**

A dataSet needs to be defined in order to send the information related to the call that will be made. Campaign, CallerID & guid are obtained from the variables that are shared in the URL.
After the dataSet definition, you can see the line of code that invokes the DISPOSITION_CALL event.

```
let dataDisp = {
        campaign: $Campaign, (Name of the campaign. Obtained from parameter shared through URL invocation)
        callerid: $CustomerNumber, (Customer Number to be called. Obtained from parameter shared through URL invocation)
        guid: $CallID, (Name of the campaign. Obtained from parameter shared through URL invocation)
        l1: $DispositionL1, (Level 1 disposition)
        l2: $DispositionL2, (Level 2 disposition) (Not Mandatory)
        l3: $DispisitionL3, (Level 3 disposition) (Not Mandatory)
        d1: $AdditionalData1, (Additional information 1, If needed) (Not Mandatory)
        d2: $AddionalData2, (Additional information 2, If needed) (Not Mandatory)
        comment: $Comment, (Manual comment, If needed)
        schedule: $ScheduleDate (example date format 2019-06-18 15:55:55), (In case the call is rescheduled) (Not Mandatory)
};

window.parent.postMessage({ action: 'DISPOSITION_CALL', disposition:dataDisp}, '*');
```

3. Parameters

Parameters

When a third-party URL is integrated into uContact, it is possible to pass variables so that they can be worked on on the external server side. For this, it is necessary to include the parameters that are required to be shared in the invocation of the iFrame.

The inclusion of these parameters must be implemented by the client according to the logical needs of the business.

Invocation format:

```
URL.index.php?param1=valor&param2=otrovalor
```

Parameter transfer

The **‘PAGE’** variable, that will then contain the URL meant to be invoked by the iFrame, uses other variables with values that will then be passed to the URL. These parameters are placed after the question mark (?), which indicates that everything that happens after corresponds to parameters, which are concatenated with the symbol ‘&’.

```
function loadIFrame() {

        var page = ‘https://10.44.1.206/telephonypromotion_ucontact/index.php?client=$ {pvy.client}&username=99999999’
                +’answernumber=${ctiParseado.Callerid}&clientid=${pyv.clientid}&tipo_operation?${pyv.typeoperation}’
                +’&promotype=${pyv.promotype}&call_id=${ctiParseado.Guid}&agent=${parent.userid}&campaign=${ctiParseado.Campaign}’;
                $(’EmbeddedPage’).attr(’src’,page);
}
```

Conditions

The URL that will be invoked from uContact must have the parameter pass implemented. Otherwise, it will **not** be possible to share variable values to the third-party system.

The parameters that will be available in the passage through the URL are up to the implementer's consideration, according to the information that needs to be processed in the external system.

The format of the invocation of the parameters will depend on the implementation made by the client on the server side.

The configuration that allows one URL to be embedded within another must be enabled on the server side. It cannot be: *x-frame-options: SAMEORIGIN* as this does not allow the page to be displayed within an iframe.

Passing parameters is very useful when you want to use uContact API methods that require variables that are only handled in our system.

Conditions

The information that can be shared from uContact is:

Data obtained from the database, as well as the information registered manually by the agents throughout the call/interaction.

Call data:

- Caller ID
- GUID (call identification)
- Call duration.
- Campaign.
- Agent associated with the interaction.
- All relevant information provided by the client during the interaction or captured by a database consult.

4. Practical example

Parameters

Below is an example of the variables that can be worked on in the parameter passage of the URL. As mentioned above, the variables expected in the URL is at the discretion of the client's implementation, that is, it must be defined beforehand which variables to share in the invocation must be.

https://ejemploURLCLiente/index.php?llamada_id=5f9a8d30-d423-41fa-b440-f03a5c0d8d05&agent=Integra&campaign=entrante<-&callerid=2626363

In this case, the URL expects the parameters "**call_id**", "**agent**", "**campaign**" and "**callerid**" which can then be processed by the client when invoking our API.

Below is an invocation example and where to use the variables obtained in the parameter pass shown above. For this example, the typing method is used, which is useful to avoid double typing, since it is only necessary to type in the client's system, and this triggers the invocation of the method with the parameters obtained from the invocation of its URL.

Invoking uContact's API

Previously, the implementation in the external system must capture the variables that were passed as parameters in the invocation of the URL. Assuming that the capture of the variables has been as follows:

```
$CampaignName = campaign, $CustomerNumber = callerid, $Call_ID = llamada_id
```

The JS code that should be included in the development of the external system is the following:

```
let dataDisp = {
        campaign: $CampaignName,
        callerid: $CustomerNumber,
        guid: $Call_ID,
        l1: $Varlevel1,
        l2:  $Varlevel2,
        l3:  $Varlevel3,
        d1:  $data1 (Additional data),
        d2:  $Vardata2 (Additional data),
        comment: $AdditionalComment,
        schedule: '2019-05-04 15:55:55' (Example of a date format, can be handled in another Var)
};

window.parent.postMessage({ action: 'DISPOSITION_CALL' , disposition:dataDisp}, '*');
```

Invoking uContact's API

Relative URL: Integra/resources/api/DispositionCall
Method: POST
content-type: application/x-www-form-urlencoded; charset=UTF-8
Params:

campaign	(variable "campaign" obtained in the parameter pass, since it is only known by uContact)
agent	(variable "agent" obtained in the parameter pass, since it is only known by uContact)
callerid	(variable "callerid" obtained in the parameter pass, since it is only known by uContact)
guid	(variable "call_id" obtained in the parameter pass, since it is only known by uContact)
l1	(nivel 1 typification, defined in the client's system when typing)
l2	(nivel 2 typification, defined in the client's system when typing)
l3	(nivel 3 typification, defined in the client's system when typing)
d1	(extra information to be saved, defined in the client's system when typing)
d2	(extra information to be saved, defined in the client's system when typing)
comment	(comment, defined in the client's system when typing)
schedule	(if there is a date with the format 2019-02-26 13:45:00 and the dialer action is *'Reschedule'*, schedule a call to that number) - Result: 1

Thank you!

INTEGRA