Integrate VoipNow calls into your CRM

This tutorial explains how to integrate VoipNow with your CRM. With the help of this integration, you can log into your CRM all the calls that are placed and received through VoipNow.

Why use it

For various reasons, most companies find it useful to keep CRM records of phone conversations they had with a particular customer.

With the help of this tutorial, you will discover that automatically storing information in your CRM for each customer call is not that difficult.

The CRM notes in question can store information such as the time the call was placed or received, the identity of the staff taking or placing that call, the duration of the call, etc.

Setup

In order to integrate VoipNow with your CRM, you must use Call Events API and the API provided by your CRM.

Let's consider an example of logging all incoming calls from our customers to their CRM profile. Let's assume that we have an extension that is tied to an external phone number, i.e. our Customer Support line.

For this, we must create a simple script in any programming language. The script should identify the customer based on caller ID and should place a note on their CRM account.

If we name the script LogIncomingCalls.php and host it on http://example.com, we must configure the Phone Call Events from the VoipNow interface so that it runs our script for the Dial-In event. More details here.

For multiple extensions, we must configure each of them as described below.

How it works

When VoipNow receives an incoming call on our Customer Support line, our script will be triggered with a series of call parameters described here.

The script (earlier named LogIncomingCalls.php) should have the following structure:

```
if (isset($_GET['CallerIDNum']) && isset($_GET['CalledExtension'])) {
        // use the CRM API to identify the customer ID based on the caller ID number
        $CRMcustomerID = identifyCustomer($_GET['CallerIDNum']);

        // use some internal logic, or SystemAPI to determine employee name based on the relation phoneNumber -
staff member
        $employeeName = identifyEmployee($_GET['CalledExtension']);

        // use the CRM API to leave a note on the customer
        leaveNote($CRMcustomerID, "Incoming call at ".date('Y-m-d H:i:s').' answered by '.$employeeName.'
('.$_GET['CalledExtension'].')');
}
```

After executing the script above, a CRM note reading "Incoming call at 2013-04-13 14:51:23 answered by Mary Sue (003*021)" will be posted on the profile of the customer that placed the call.

This code sums up the logical steps recommended for a successful integration.

You can use the example from Track My Calls as a starting point of your integration and replace the database requests in this example with API calls to your CRM application.

Tips and tricks

You can use System API to fetch more log information about the extension (staff member) that answered the call.

To find out the duration of the call, you can combine Hangup and Dial-In events. On a Hangup event, you can update the note that was previously added to the customer.

If you want a more comprehensive example on a specific CRM, please add a comment to this page and we will consider building a demo for your CRM.

Need help? Ask a question in our GetSatisfaction community.