

Integrating Phone Validation into Marketo

Step by Step Using “Webhooks”



Regardless of the industry or the department, communicating via telephone is essential for successfully connecting with your customers and prospects. Whether you are in marketing, sales, or customer service, you need to have accurate telephone numbers for your customers. Poor phone data quality leads to negative customer experiences and failed engagements.

Utilizing our Phone Validation service within Marketo ensures your customer and prospect phone numbers are accurate from the second that you acquire them.

This integrated solution verifies phone numbers in real-time within Marketo, recording the results in custom fields. By using Marketo's Webhooks and Informatica's Phone Validation REST API, you can verify phone numbers anywhere within your smart campaigns and programs in Marketo.

For example, verify phone numbers as soon as they enter your system via a web form. Or you may choose to add the Webhook to a Smart Campaign at other points – like conducting a batch cleanse on a regular basis (quarterly, semi-annually, etc.) to keep your contacts up-to-date.

This document outlines the use case of leads being verified within a campaign.

There are 3 steps in this process:

1. The Webhook must be configured to communicate with Informatica's Global Phone Validation API, including mapping the proper output data.
2. A campaign must be created to trigger the webhook.
3. A second Smart Campaign must be created to call the trigger campaign.

In addition, you must have credentials (a license key) from StrikeIron – an Informatica company - to successfully validate phone numbers. These can be obtained via subscription from us at sales@strikeiron.com. Free trials are available.

Step 1: Configure the Webhook

The Marketo Web application communicates with us using a “Webhook,” which uses the REST protocol to communicate with Informatica’s Phone Validation API. REST, or Representational State Transfer, is a Web-based architecture for enabling different hardware and software platforms to communicate via a common text-based XML format.

Here is an example of a REST call that verifies global phone numbers using Strikelron’s API:

http://ws.strikeiron.com/Strikelron/GlobalPhoneValidation1/GlobalPhoneValidation/ValidateNumber?LicenseInfo.RegisteredUser.UserID=<Strikelron_License_Key>&ValidateNumber.PhoneNumber=9194674545

You can test the above REST call by copying and pasting the information above into the address URL of any web browser such as Firefox, Google Chrome, or Internet Explorer. Be sure to edit the REST call information by supplying your Strikelron license key obtained from Strikelron and any phone number you wish. Do this by replacing the Strikelron_License_Key field with your 18-22 digit license key, and replace the phone number listed after “ValidateNumber.PhoneNumber=” with the phone number that you will be verifying.

It is the various components of the above REST call that we will insert into the Marketo Webhook.

Note the question mark (?) in the middle of the actual REST call. This question mark separates the endpoint of the API (the URL where the API lives on the Internet, i.e. Strikelron servers) from the parameters that are supplied along with the REST call.

Once logged into your Marketo account, go to “Admin.” Within Admin, you will see the ability to “Manage Webhooks.” Click on “New Webhook.”

Webhooks

Admin... [New Webhook](#) [Webhooks Actions](#)

Business Data Append

Details [EDIT](#)

URL: `http://ws.strikeiron.com/Strikelron/CorteraBusinessVitals2/CorteraBusinessVitals/BusinessDemographicsByCompany`

Payload Template: `LicenseInfo.RegisteredUser.UserID=xxxxxxxxxxxxxxxxxxxx&BusinessDemographicsByCompany.Company.CompanyName={{company.CompanyName:default=edit me}}&BusinessDemographicsByCompany.Company.AddressLine={{company.BillingAddress:default=edit me}}&BusinessDemographicsByCompany.Company.City={{(company.BillingCity:default=edit me)}}&BusinessDemographicsByCompany.StateOrProvince={{(company.BillingState:default=edit me)}}&BusinessDemographicsByCompany.Company.ZIPCodeOrPostalCode={{(company.BillingPostalCode:default=edit me)}}&BusinessDemographicsByCompany.Company.Country={{(company.BillingCountry:default=edit me)}}string`

Request Token Encoding: WebhookAdmin.NONE

Request Type: POST

Response Format: XML

Used By: 1

Response Mappings [EDIT](#)

Response Attribute	Field
	Marketo Field

Search... Webhooks

- Business Data Append
- Email Verification + Hygiene v6
- Email Verification and Hygiene
- Email Verification v5
- Email Verification v6
- EpsilonTest
- Global Address Verification
- Global Phone Validation
- MobileID
- North American Address Verification
- Phone Number Validation
- Reverse Phone Append
- SMS Alerts
- SMS Marketing Demo
- SMS Notification
- SMS Tracking
- SMS Tracking - Test

The following form will appear and allow you to create your Email Verification and Hygiene Webhook:

The screenshot shows a form titled "Edit Webhook" with a close button (X) in the top right corner. The form contains the following fields:

- Webhook Name:** * Global Phone Validation
- Description:** (empty text box)
- URL:** * INSERT TOKEN
http://ws.strikeiron.com/Strikelron/GlobalPhoneValidation1/GlobalPhoneValidation/ValidateNumber?
- Request Type:** * POST
- Template:** INSERT TOKEN
`LicenseInfo.RegisteredUser.UserID=xxxxxxxxxxxxxxxxxxxx&ValidateNumber.PhoneNumber={{lead.Phone Number.default=edit me}}`
- Request Token Encoding:** None
- Response type:** XML

At the bottom right of the form are two buttons: "CANCEL" and "SAVE".

Use the following configuration for the Global Phone Validation Webhook:

- **Webhook Name:** Use a name that will make sense in your campaigns. Here we've used "Global Phone Validation."
- **Description:** Enter a description for your Webhook. This will help others who come in later to view your Webhook to know exactly its purpose.
- **URL:** This is the base URL for the Strikelron REST call. Copy the following URL: <http://ws.strikeiron.com/Strikelron/GlobalPhoneValidation1/GlobalPhoneValidation/ValidateNumber?>
- **Request Type:** POST

- **Template:** Two parameters must be included here – License Key and Phone Number

Here is a sample of the template text:

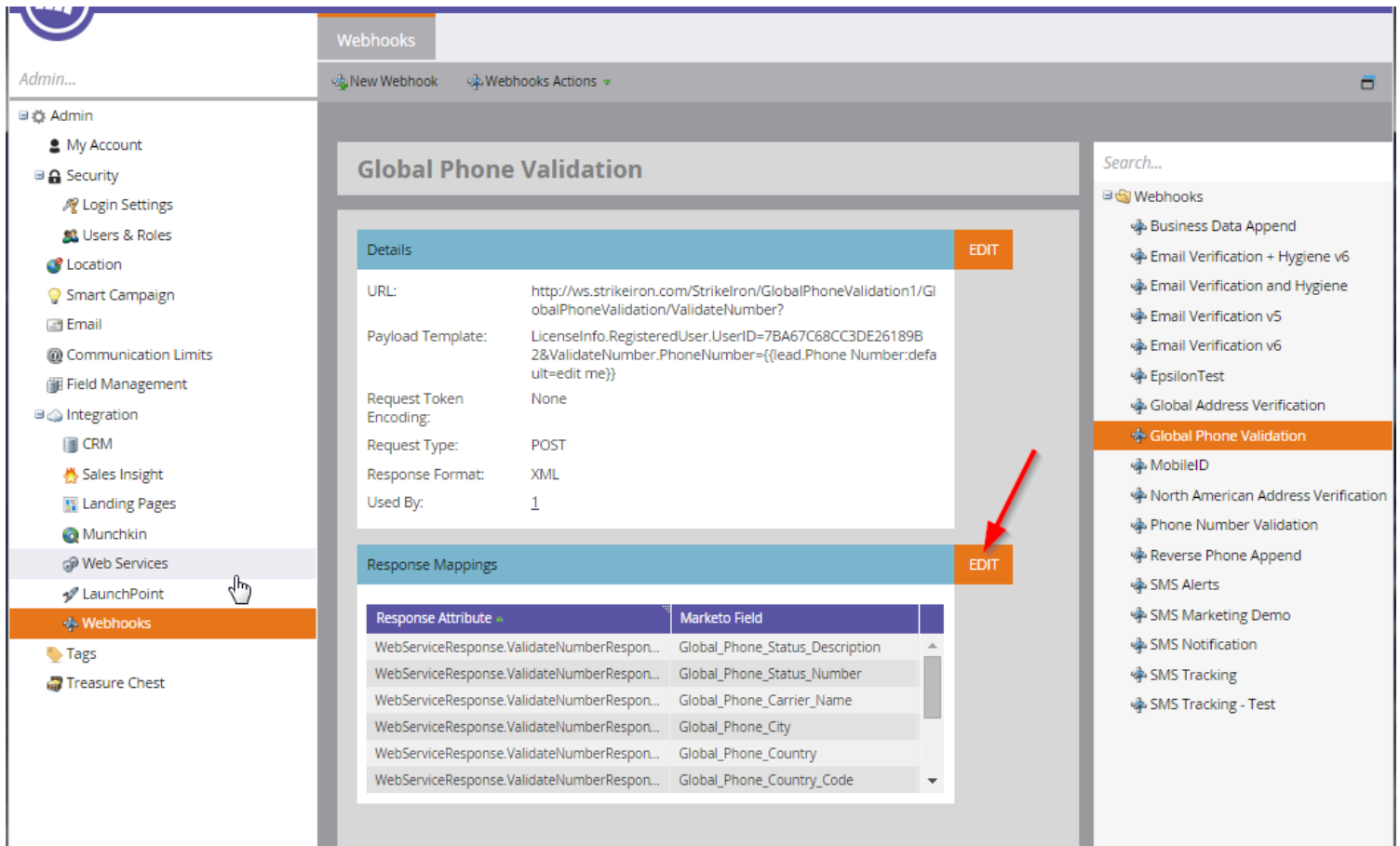
- LicenseInfo.RegisteredUser.UserID=xxxxxxxxxxxxxxxxxxxx&ValidateNumber.PhoneNumber=9194674545
- You can also use Marketo Tokens, which would look something like this:
- LicenseInfo.RegisteredUser.UserID=xxxxxxxxxxxxxxxxxxxx&ValidateNumber.PhoneNumber={{lead.Phone Number:default=edit me}}
- LicenseInfo.RegisteredUser.UserID – Strikelron License Key.
Replace the XXXXXXXXXXXXXXXXXXXX in this example with your Strikelron license key.
- ValidateNumber.PhoneNumber – The phone number that you want to verify. You can add a token here, so that a lead’s phone number populates this field within REST URL parameters. To include a token in any part of the template section, move your cursor to the correct insertion place and click on “Insert Token” button. Choose the field that you wish to insert – most likely: {{lead.PhoneNumber}}. Tokens indicate to Marketo where to integrate actual lead data into the Webhook, as the field will be populated by the lead data for the record that triggers the invocation of the Webhook.

- **Request Token Encoding:** None
- **Response Type:** XML

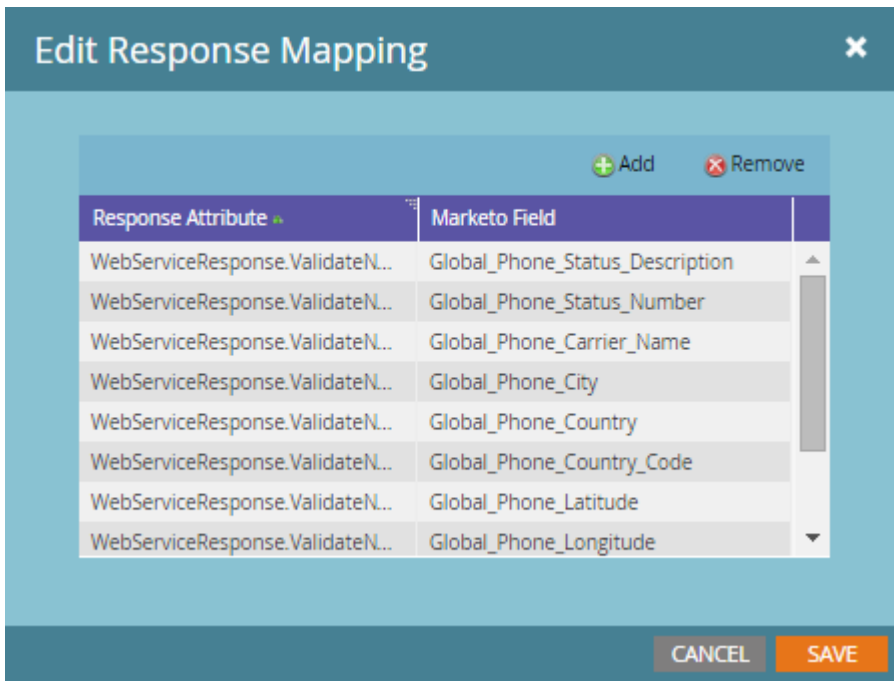
Once you save these options, it is time to add Response Mappings to your Webhook. These map the return values that the Informatica Phone Validation service returns to the Marketo fields. Before you map the Response Mappings, it is recommended that you create the new fields. The names we use are below, but you can name that whatever you wish. You can map as many or as few fields as you like, your options are below.

- **Global Phone Status Number**
- **Global Phone Status Description**
- **Global Phone Number**
- **Global Phone Country Code**
- **Global Phone Intenational Prefix**
- **Global Phone National Prefix**
- **Global Phone Cell Prefix**
- **Global Phone NDC**
- **Global Phone SN_From**
- **Global Phone SN_To**
- **Global Phone Uses_NDC**
- **Global Phone Number Type**
- **Global Phone Carrier Name**
- **Global Phone Language**
- **Global Phone Country**
- **Global Phone ISO**

- Global Phone Region Code
- Global Phone Region Name
- Global Phone City
- Global Phone Time Zone
- Global Phone UTC
- Global Phone DST
- Global Phone Latitude
- Global Phone Longitude
- Global Phone Error Code



Once the fields are created, you will click “Edit” to edit the Response Mappings and then “Add” to include additional mappings.



Map the following Response Attributes to your newly created fields. Make sure that you copy the Response Attribute exactly as you see it here and make sure there are no spaces before or after the text. Only copy the sting in black to the Response Mapping section, **DO NOT copy anything in orange.**

Global Phone Status Number =

`WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.ItemStatus.StatusNbr`

Global Phone Status Description =

`WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.ItemStatus.StatusDescription`

Global Phone Number =

`WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Number`

Global Phone Country Code =

`WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.CountryCode`

Global Phone International Prefix =

`WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.InternationalPrefix`

Global Phone National Prefix =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.NationalPrefix

Global Phone Cell Prefix =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.CellPrefix

Global Phone NDC =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.NDC

Global Phone SN From =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.SN_From

Global Phone SN To =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.SN_To

Global Phone Uses NDC =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Uses_NDC

Global Phone Number Type =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Number_Type

Global Phone Carrier Name =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.CarrierName

Global Phone Language =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Language

Global Phone Country =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Country

Global Phone ISO =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.ISO

Global Phone Region Code =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.RegionCode

Global Phone Region Name =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.RegionName

Global Phone City =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.City

Global Phone Time Zone =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.TimeZone

Global Phone Uses UTC =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.UTC

Global Phone DST =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.DST

Global Phone Latitude =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Latitude

Global Phone Longitude =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.Longitude

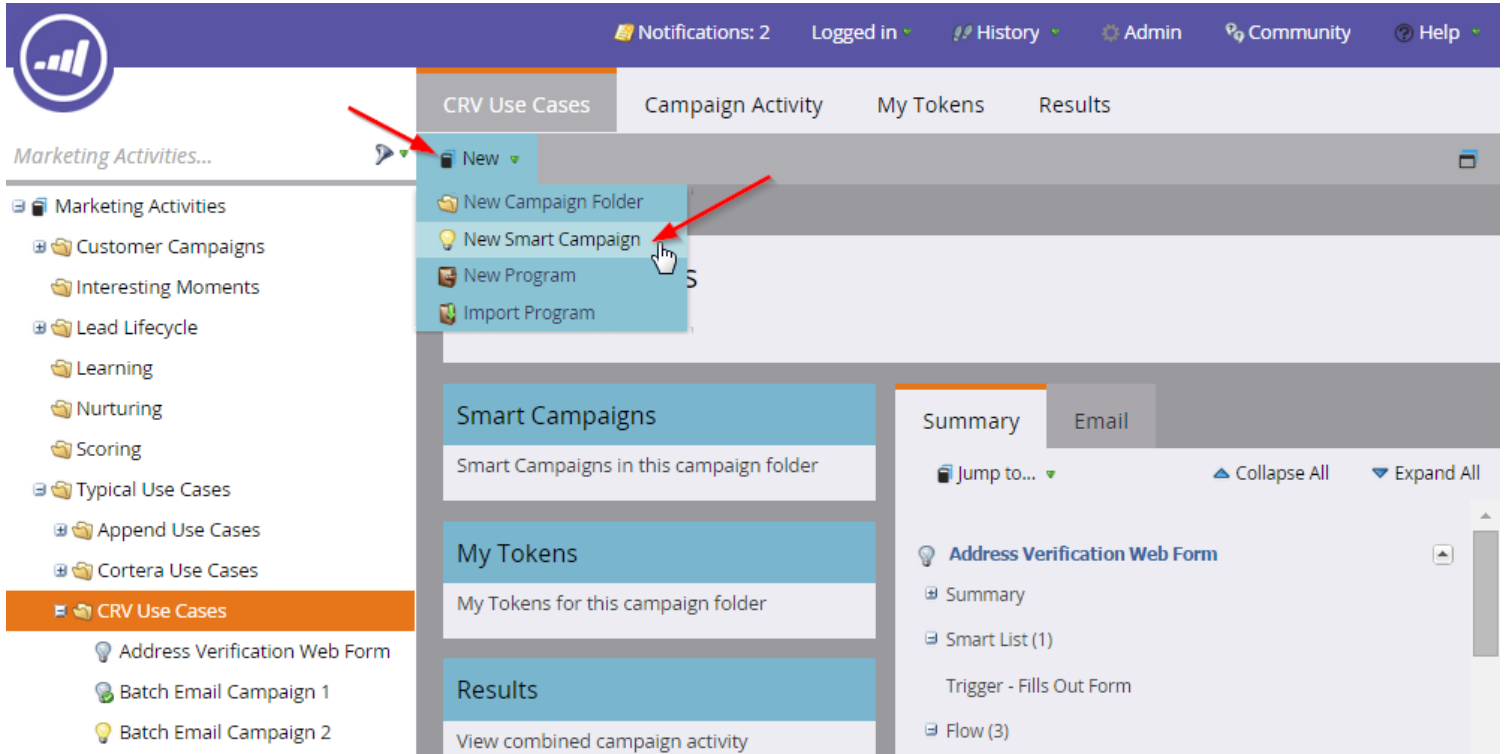
Global Phone Error Code =

WebServiceResponse.ValidateNumberResponse.ValidateNumberResult.ServiceResult.Results.GlobalPhoneEntryWithStatus.Listing.ErrorCode

Save your Response Mappings by clicking “Save” at the bottom of the screen. The Webhook is now ready to be used by the Marketo system.

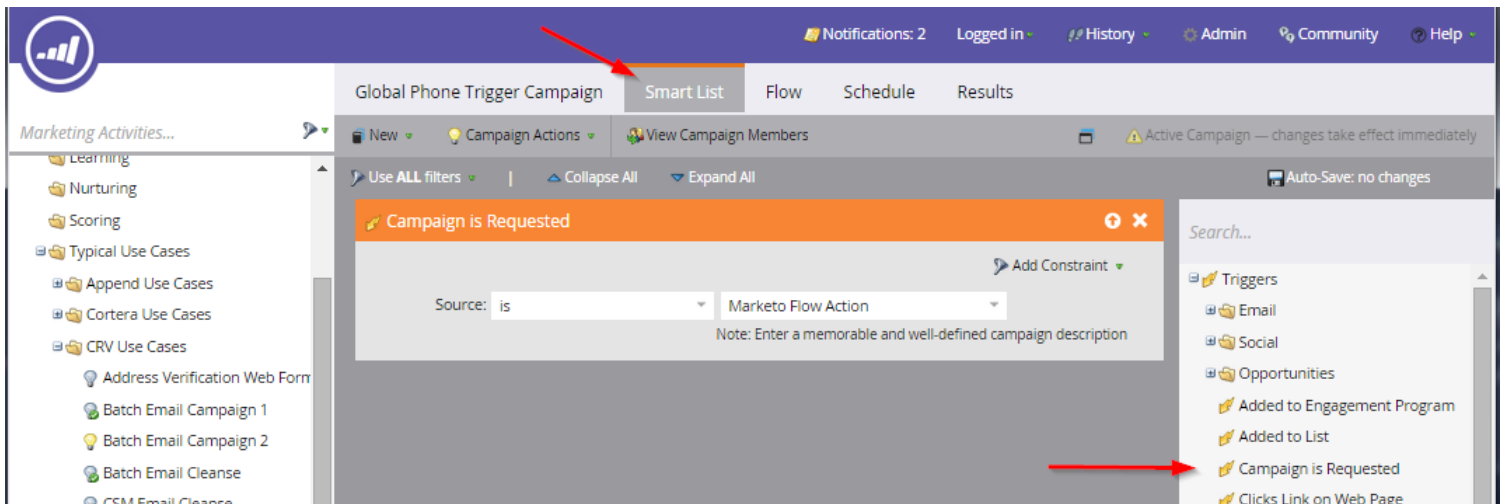
Step 2: Create the first Smart Campaign with Trigger

Creating a smart campaign is a standard activity within Marketo. More information can be found within the Marketo help documentation. Within a Campaign Folder you will click 'New' and then 'New Smart Campaign'



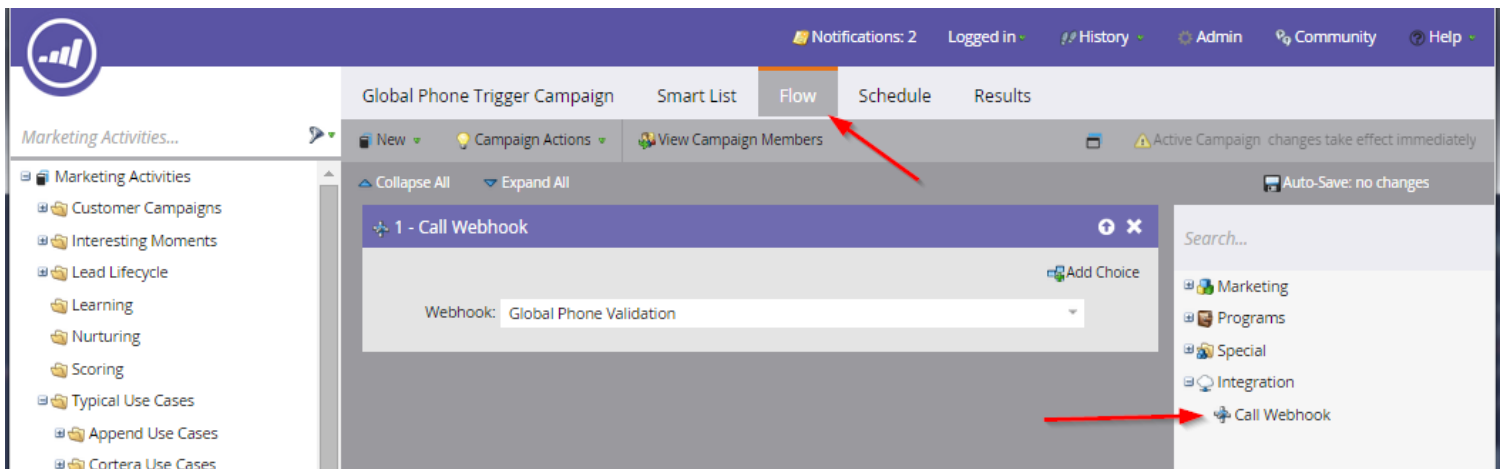
When the Smart Campaign button is clicked, you will see the New Smart Campaign window pop up, fill in the information accordingly.

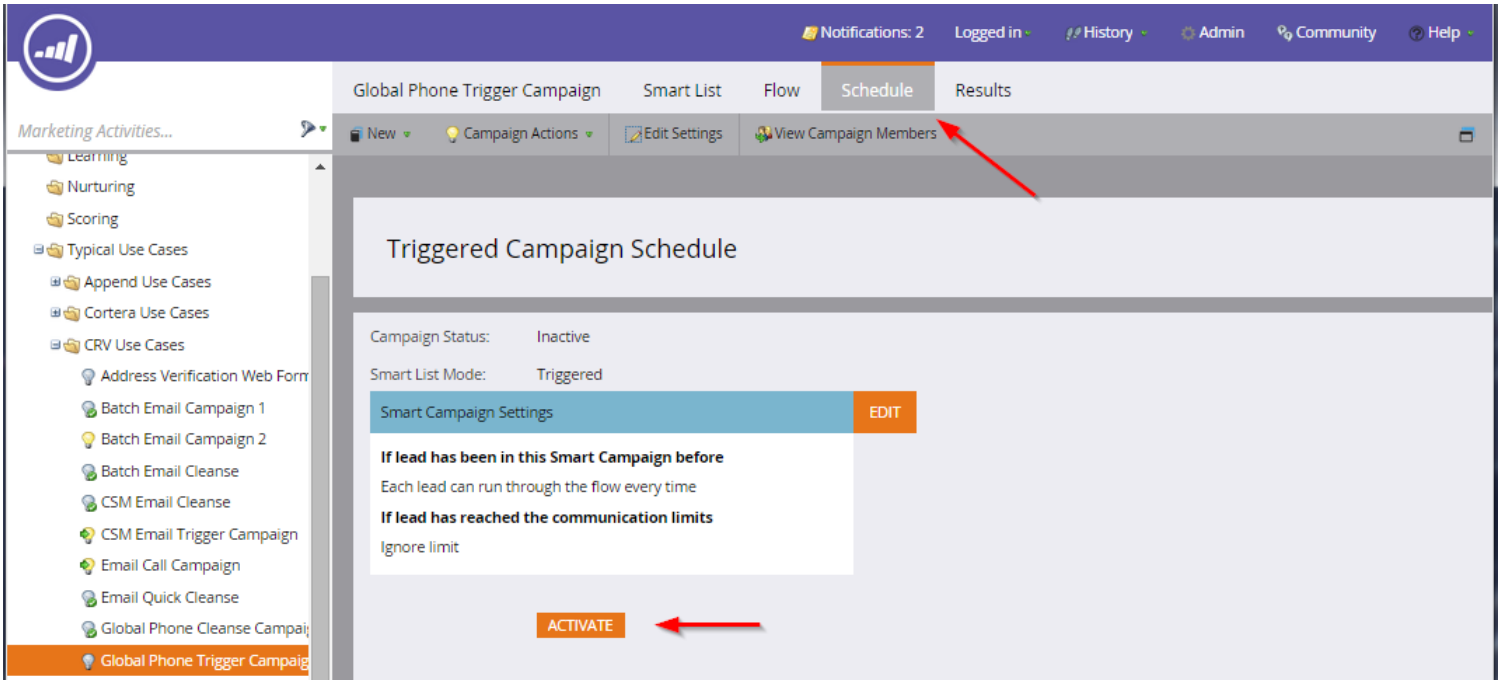
A screenshot of the 'New Smart Campaign' dialog box. The title bar says 'New Smart Campaign' with a close button (X). The form has three fields: 'Folder:' with a dropdown menu showing 'CRV Use Cases', 'Name:' with a text input field containing 'Global Phone Trigger Campaign', and 'Description:' with a text area containing 'Campaign to trigger global phone validation webhook'. At the bottom right, there are two buttons: 'CANCEL' and 'CREATE'.



Once the 'Smart Campaign' is created, you will need to set it up accordingly. The first step is to set up the Smart List, here is where you will add the trigger that sets the campaign off. You will want to drag the 'Campaign is Requested' trigger from the right hand side into the Smart List. Leave it set up just the way it is with the source being the Marketo Flow Action.

The next step is to setup the Flow. All you really need to do here is drag the 'Call Webhook' function to the Flow, then choose the webhook to run. In this case it is the Global Phone Validation Webhook.





The last part of this step is to activate the campaign underneath the 'Schedule' tab. Just simply click the 'Active' button to activate the campaign.

Keep in mind, in this step you also have some control over how many times a lead can run through the Flow. You can choose either Only Once; Once every X days; or Every Time. Just know that if you choose 'Every Time' that each time you run the Flow the lead can run through and be verified again and consume another transaction.

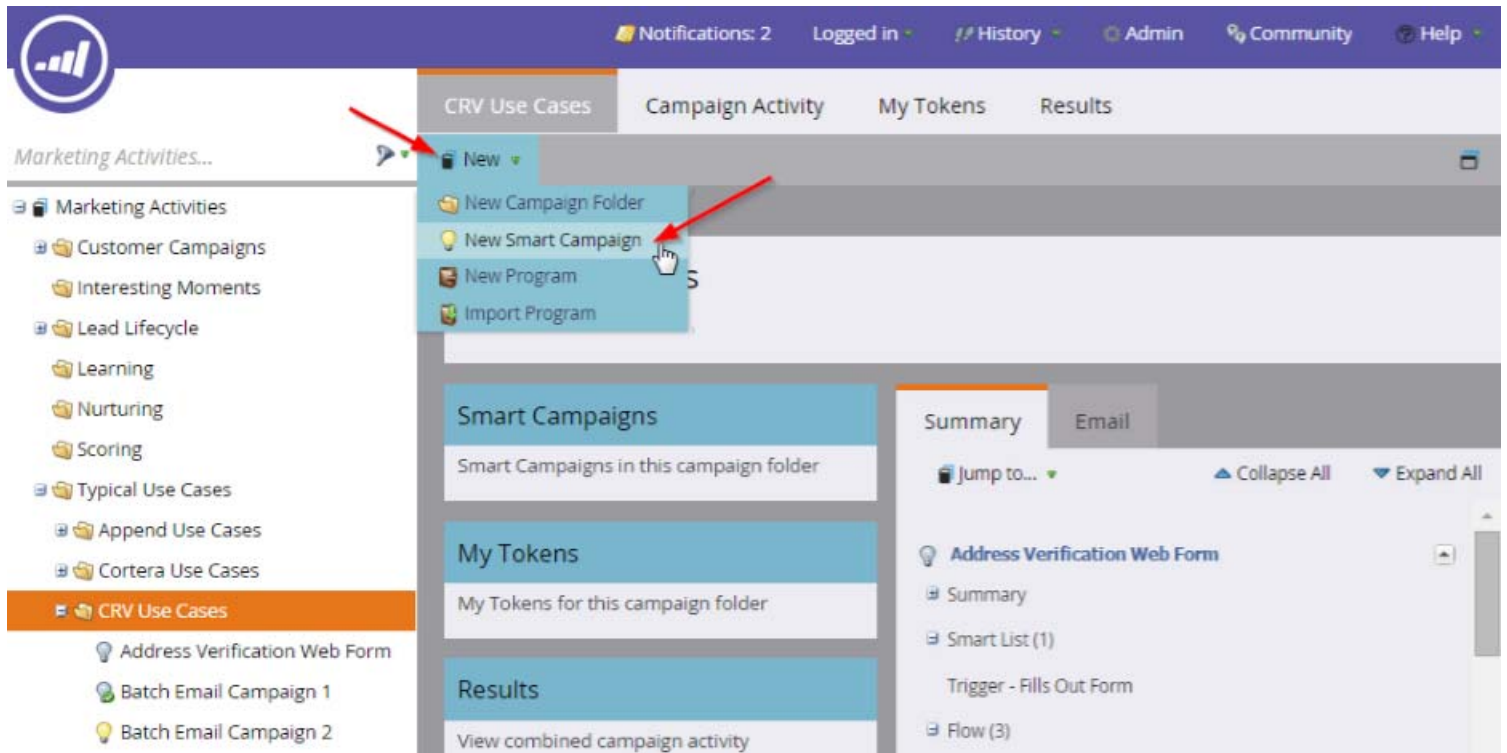


Step 3: Create the second Smart Campaign to call Trigger

This also is a standard activity within Marketo. More information can be found within the Marketo

help documentation if necessary about creating Smart Campaigns.

Click on your Smart Campaign Folder and then “New” and “New Smart Campaign.” Give the campaign a name and decide which folder to store it in.



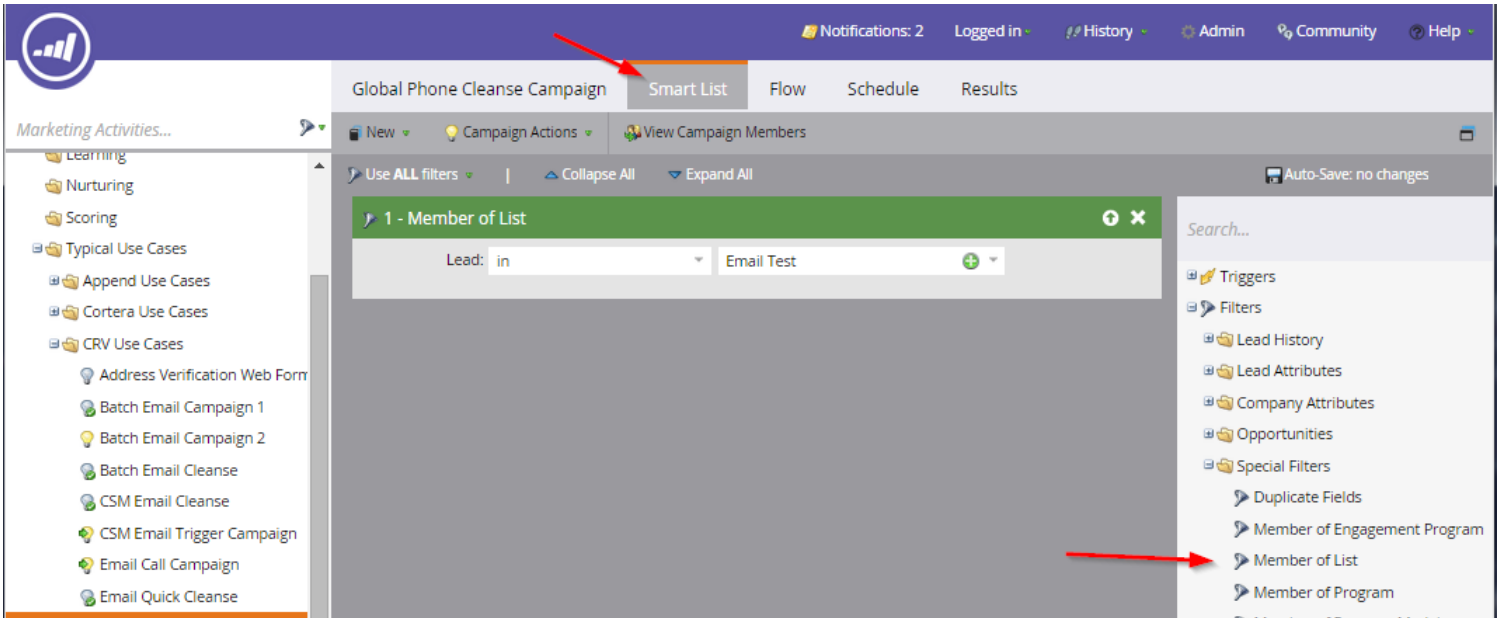
New Smart Campaign ✕

Folder: * CRV Use Cases

Name: * Global Phone Cleanse Campaign

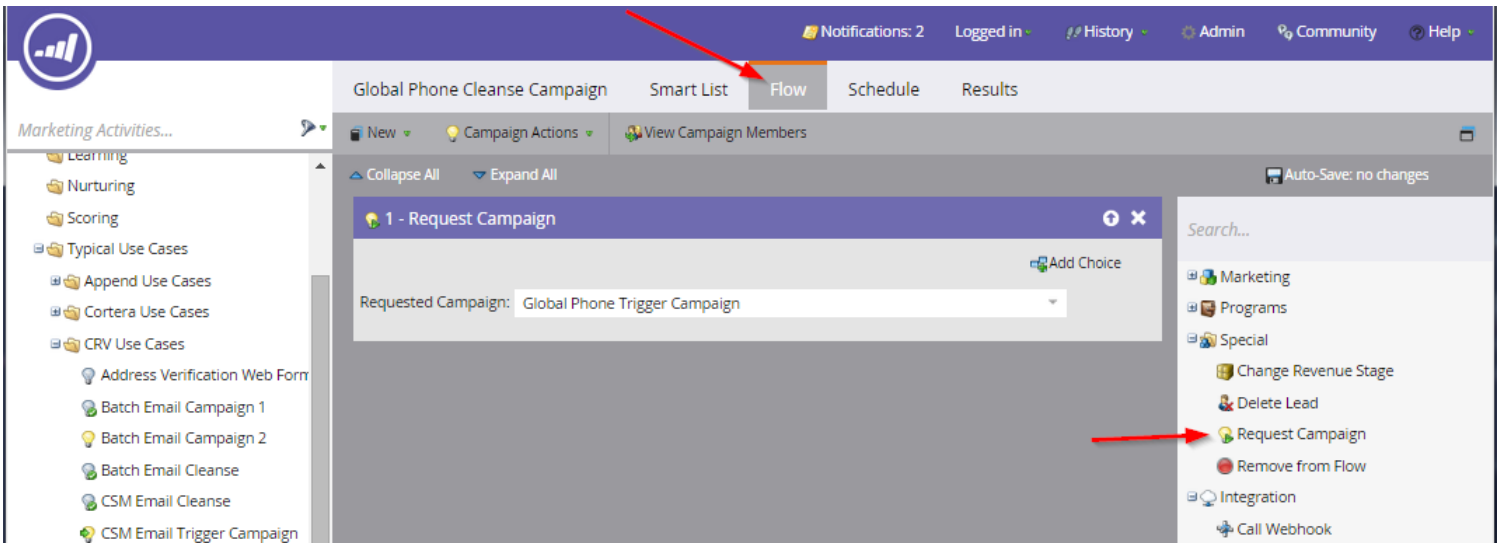
Description: Campaign to validate global phone numbers

CANCEL CREATE



Next, click the “Smart List” tab. Drag the “Member of List” Trigger from the right to the blank canvas, and then choose the list that you want to have the webhook run against.

Move onto the next tab, “Flow.” Drag “Request Campaign” onto the blank canvas and choose the ‘Smart Campaign’ that was created in step 2.



Lastly, you will just need to either schedule the campaign to run at a predetermined interval, or you can leave it as is and run it on demand with the 'Run Once' button. To schedule the campaign just click on the 'Schedule' tab, then 'Schedule Recurrence'. This will pop up the scheduling wizard. Set it up accordingly. The campaign is now ready.

The screenshot shows the Marketo interface for configuring a campaign. The top navigation bar includes 'Notifications: 2', 'Logged in', 'History', 'Admin', 'Community', and 'Help'. The main header shows the campaign name 'Global Phone Cleanse Campaign' and tabs for 'Smart List', 'Flow', 'Schedule', and 'Results'. A red arrow points to the 'Schedule' tab. Below the header, there are buttons for 'New', 'Campaign Actions', 'Run Once', 'Schedule Recurrence', 'Edit Settings', and 'View Campaign Members'. The main content area is titled 'Batch Campaign Schedule' and displays the following information:

- Campaign Status: Last run: Mar 20, 2015 11:27 AM EDT
- Smart List Mode: Batch
- Smart List Status: Set to affect around 4 existing leads (may change over time). Around 0 leads are blocked from mailings.
- Smart Campaign Settings: EDIT
- If lead has been in this Smart Campaign before: Each lead can run through the flow every time.
- If lead has reached the communication limits: Ignore limit.
- Recurrence: No repeating schedule has been set.

The 'Schedule Recurrence' dialog box is shown with the following configuration:

- Schedule: Weekly
- First run: 01/20/2015 at 4:12 PM EST
- Repeat every: 1 weeks on: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
- End on: No end date

Buttons for 'CANCEL' and 'SAVE' are visible at the bottom right.

Congratulations! You have successfully integrated Informatica's Phone Validation API into Marketo. You now have the ability to create marketing campaigns with maximum email deliverability within the Marketo platform.