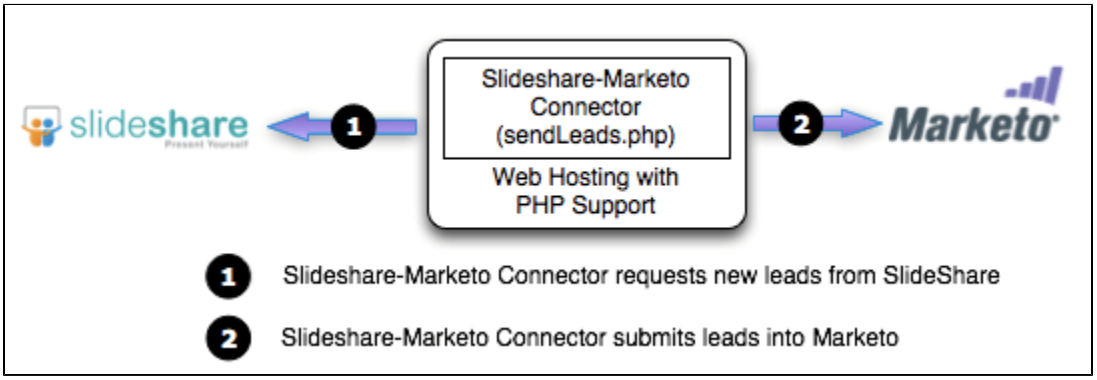
SlideShare-Marketo Connector

Capture Leads Data from SlideShare to Marketo

This page will show details on how to capture leads data from Slideshare and pass it directly inside Marketo, but before we begin **the following is required**:

1. Client must have a **"Slideshare PRO" account**
2. Client must have **"PHP hosting enviroment"** as well as the capability and permissions to put files onto a server via FTP
3. Client will need these files to get things started SlideShare-Marketo-Connector.zip (by Erik Rehn)



How to setup Slideshare Leads Capture Scripts:

Inside SlideShare-Marketo-Connector.zip there are **5 php files**, with **3 key files**:

1. **api_config.php**
2. class.rss.php
3. **marketo.php**
4. **sendLeads.php**
5. SSUtil.php

These 5 files will need to live on the clients hosting server, because as of right now we do not do hosting of custom php scripts.

1 - api_config.php

In this file you will need to get the client to request for their Slideshare API Keys from http://www.slideshare.net/developers/applyforapi

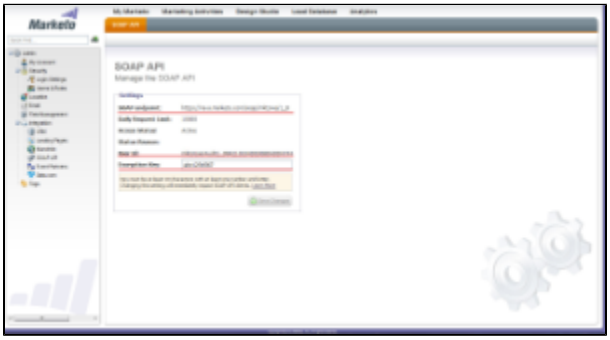
1. They will need to replace "**enter_api_key**"
2. They will need to replace "**enter_secret_key**"

```
<?php
// This key can be obtained from http://www.slideshare.net/developers/applyforapi
// API  public key and secret key
$key='enter_api_key';   // Slideshare API Key
$secret='enter_secret_key';  // Slideshare Shared Secret

// Do not change this
$apiurl='http://www.slideshare.net/api/2/';

?>
```

3 - marketo.php



This file is Marketo SOAP API and the client will need to do the following:

1. In this file the client will need to **find "class mktSampleMktowsClient"**
2. They will need to replace "**enter_marketo_userID**"
3. They will need to replace "**enter_marketo_encrypt_key**"

Below is the code area that they should be finding:

```
class mktSampleMktowsClient
{
  // Change this vale to true to enable debug output.
  const DEBUG = false;

  const CLIENT_TZ = 'America/Los_Angeles';

  const MKTOWS_USER_ID = 'enter_marketo_userID';  // Marketo USER ID
  const MKTOWS_SECRET_KEY = 'enter_marketo_encrypt_key'; // Marketo Encryption Key

  const MKTOWS_NAMESPACE = 'http://www.marketo.com/mktows/'; // DO NOT MODIFY

...more code that they should just ignore
```

4 - sendLeads.php

Last but not least, the brains of the operation. In this file the user will need to modify various areas to complete the connection:

```
$soapEndPoint = 'enter_soap_endpoint';  // Marketo Soap Endpoint
```

Enter their Slideshare username and password:

- Make a note that **the username is NOT the email address used to login but the USERNAME HANDLE**

```
// Slideshare Connection Setup
$user = 'slideshareUser';      // slideshareUser
$pass = 'slidesharePass';      // slidesharePass
```

This piece of code will determine the time frame of what data to pull:

```
$begin = date( "YmdHi", time()-60*20);  // Sync all leads data from 20 minutes ago
```

If all the setup was done correctly you would have a message displayed "**# leads synced from SlideShare.**" when you run "**sendLeads.php**"

The type of data pulled from Slideshare would be within the syncArray:

```
$syncArray = array(
  "FirstName" => $lead->FirstName,
  "LastName" => $lead->LastName,
  "Company" => $lead->Company,
  "Phone" => $lead->PhoneNo
        );
```

The **left side** of the array is the field names of **Marketo**, the **right side** is the sync with fields from **Slideshare**.

If client requests other type of data to pull the following is available:http://www.slideshare.net/developers/documentation#get_user_leads just make sure that the client has prepared the "custom field name" if it isn't already available in Marketo or sendLeads.php will spit out an error that a sync could not happen.

```
<Leads>
    <Lead>
        <SlideshowId>{int}</SlideshowId>
        <Email>{txt}</Email>
        <PhoneNo>{txt}</PhoneNo>
        <Address>{txt}</</Address>
        <PaidAt>{datetime}</</PaidAt>
        <Rating>{int}</Rating>
        <CreatedAt>{datetime}</CreatedAt>
        <UpdatedAt>{datetime}</</UpdatedAt>
        <Message>{txt}</Message>
        <UserComment>{txt}</UserComment>
        <City>{txt}</City>
        <Country>{txt}</Country>
        <Zipcode>{txt}</Zipcode>
        <State>{txt}</State>
        <Mechanism>{txt: download, sidebar, or player}</Mechanism>
        <ReadAt>{datetime}</ReadAt>
        <DeletedAt>{datetime}</DeletedAt>
        <LeadCampaignId>{int}</LeadCampaignId>
        <GeoData>
          <Region></Region>
          <Country></Country>
          <City></City>
        </GeoData>
        <Cost>{$}</Cost>
        <FirstName>{txt}</FirstName>
        <LastName>{txt}</LastName>
    </Lead>
    …
</Leads>
```

Advanced Setup with CRONS

If the user has the ability to setup crons on their server, they can most definitely do it. Using a CRON will automate the run of sendLeads.php at timed intervals so that you would not need to load up the script via URL each time to get leads to sync. For this example we will use GoDaddy.com for setup:

1. Get into the **Hosting Control Center**
2. Under "**Content**" select "**Cron Job Manager**"
3. The settings should be straight forward
    a. Name your cron job
    b. Select a frequency
    c. Select "sendLeads.php" as the file to run