

# Sailthru



## Integrating with Sailthru

Gigya's integration with [Sailthru](#) enables the use of Sailthru **Smart Lists** to segment social data. Sailthru can access Gigya data which can be leveraged in reporting and email campaigns. Note that this data is not currently used in on site recommendations because only tagged content on-site is used for those recommendations. Please contact your Implementation Consultant if you are interested in on-site personalization using Gigya data through Sailthru.

Currently this is a one-way integration with user profile data passing from Gigya to Sailthru.

## Getting Started

Ask Gigya's Implementation Consultant to enable [Social Login](#) or [Registration-as-a-Service](#) on your account and select the social providers you want to support. Once you have the social login or registration service feature enabled, contact your Sailthru Customer Engagement Executive to help you create a **callback script**, and provide some sample code to capture Gigya's login event.

Your Sailthru implementation team will need to know your technology stack to guide you with the callback script. For example PHP, Rails, Java etc.

 Unknown macro: 'includeplus'

An example of the server code that Sailthru provides is shown below. Your implementation team will help you set the security signature which ensures that the data received was posted from Gigya.

For example the following script might be called `SailthruGigya.php` (see [PHP SDK](#)):

```
<?php
include '../lib/sailthru/Sailthru_Client.php';
include '../lib/sailthru/Sailthru_Social.php';

include 'GSSDK.php'; //This is Gigya's PHP SDK.
$sailthru_api_key = "your_api_key";
$sailthru_api_secret = 'api_secret';
$client = new Sailthru_Social($sailthru_api_key, $sailthru_api_secret);
$data = json_decode($_POST['json']);

$gigya_secret_key = "<gigya_secret_key>";
// The security signature check, verifying the data's origin, will follow:
if (SigUtils::validateUserSignature("<UID_from_$data>",
"<signatureTimestamp_from_$data>", $gigya_secret_key,
"<UIDSignature_from_$data>")) {
    // Post data to Sailthru
    $result = $client->social_login($data);
}
?>
```

## Adding JavaScript

Once the **callback script** and your Gigya account is set up, you will need to add the following JavaScript snippet to your client site code. The name of the callback script used in the previous example is passed to the JavaScript as the **callback URL** and must match the **callback\_url** parameter. This should be passed over an SSL connection to ensure that user profile data is encrypted and to maintain security [best practices](#) as described by the Gigya documentation.

```
<script src="//"></script>
  <script type="text/javascript">
    SailthruGigya.callback_url = 'https://yourdomain.com/<ac:link
ac:anchor="SailthruGigya.php"><ri:page ri:content-title="Sailthru"
ri:space-key="GD"/><ac:plain-text-link-body><![CDATA[SailthruGigya.php
'; //the Gigya onLogin event calls SailthruSync
gigya.socialize.addEventHandlers({ onLogin: }); ]]>
```

You may view the details of the Sailthru sync.js file .

## What Information is Shared?

Gigya has a standard schema of user information that we share via the [onLogin event](#). If you are using Registration-as-a-Service, then it is [this schema](#), and if Social Login [it is here](#). Different information is available from different services. If the service does not provide the information in the schema, or the user has not provided it, then it will not be passed to Sailthru. Only information which the user explicitly allowed a service to share will be provided.

After the data is passed to Sailthru it will become available in the user profile. The format for reference is **servicename\_variable**.

As these are user variables you will be able to use them in the Sailthru query builder for reports and Smart Lists, opening up opportunities to further refine your lists to provide quality, relevant content for your users.