Personalization Engine Setup

Configuring Personalization Engine and adding Sailthru's Personalize JavaScript on your site will enable you to import your site's content into Sailthru and track your users' behavior and interests. W this data, Sailthru will enable you provide optimal personalized content to your users.

The simplest approach to get up and running requires just two steps:

1. Embed a single line of Sailthru's Personalize JavaScript on your pages.
   ```
   <script src="//ak.sail-horizon.com/onsite/personalize.v0.0.4.min.js"
   type="text/javascript"></script>
   ```
2. In your Settings, enter your site domain(s) and tell Sailthru which kinds of URLs contain content that should or should not be pulled into your Content Library.

However, there are many additional features to consider that require additional configuration steps. For example:

- Set custom metadata on your content items
- Make direct updates to the Content Library via API
- Manually track user actions or user interests via JavaScript

Review the configuration steps below and speak with your Customer Success Manager to determine which methods will work best for you.

The configuration steps covered on this page:

1. Customize and Embed the **Personalize JavaScript**
2. Implement **Meta Tags** and/or the **Content API**
3. Set **Content Spidering** Preferences
4. Understand and Use **Sailthru Cookies**

1 Customize and Embed the Personalize JavaScript

Sailthru offers two modes for the Personalize JavaScript that you must include on your site.

- Most clients can use the default **Single-Line Mode** by simply including the following script:
  ```
  <script src="//ak.sail-horizon.com/onsite/personalize.v0.0.4.min.js"
  type="text/javascript"></script>
  ```
- However, if you need to use any additional JavaScript functions described in this document, you can enable **Instrumented Mode** by using the following code and including those functions after it.
  ```
  <script src="//ak.sail-horizon.com/onsite/personalize.v0.0.4.min.js" type="text/javascri
  data-sailthru-setup="true"></script>
  ```

Use the following feature-comparison table to decide.

Note that this is a per-page setting, so you can opt to enable both modes on different types of pages on you site. If you already have a legacy Sailthru JavaScript code embedded on your site, you can replace it with th latest version of the Personalize JS.

Sailthru Personalize JS Mode Features	
Single-Line Mode (Default)	**Instrumented Mode**
Auto-tracking of pageviews.	Manual tracking option available for pageviews. (Ideal for single-page views, such as infinite scroll, and dynamic single-page sites.) Auto-tracking is the default.
Tags to be recorded as user interests when an item is viewed are those spidered by Sailthru or sent via the Content API and stored for each item in your Content Library.	Tags for viewed items to be recorded as user interests can be manually specified in the JS code, regardless of whether others are stored for the content item in the Content Library.
URL of current page is identified automatically, with query strings included or excluded based on sitewide Spider Preferences.	URL of current page can optionally be explicitly specified in the JS.
Whether to spider page is determined by sitewide Spider Preferences.	Whether to spider page can be determined in the JS.

Single-Line Mode uses the default behavior/values for all functions below, and does not require them to be called.

If you are using single-line mode, you can skip the rest of this section and go directly to section 2, Implemer Meta Tags and/or the Content API.

Under **Instrumented Mode**, the *setup* function below is required and an additional *trackPageview* function i optional.

Functions for Instrumented Mode with Advanced Options

setup

```
Sailthru.SPM.setup(customerId, {optionalConfigObject});
```

This function requires your Sailthru Customer ID which can be found in the My Sailthru interface on the API and Postbacks page under Settings > Setup.

Required Parameter	Type	Default	Description
customerId	String	N/A	Customer ID, for example b682bb3746796686c27164ba015c3da7. Replace customerId with the one displayed for your account in My Sailthru on the Settings page API & Postbacks.

Click to view keys for optionalConfigObject (a JSON object containing key-value pairs): autoTrackPageVi useStoredTags, url, tags, and spider

Optional Key	Type	Default	Description
autoTrackPageView	Bool	true	Specify whether pageviews should be tracked automatically on page-load. You can track pageviews manually using the trackPageView function, which is useful when a single URL may display an array of content items.
useStoredTags	Bool	true	Specify which interest tags to assign to the user based on the item; if true, use existing tags in your Content Library (whether added by the Personalize JS spider or Content API calls); if false, use tags sent in the

			trackPageView function.
url	Bool	[URL of the current page as visited]	If your page has multiple valid URLs, specify the one primary URL (a.k.a. the canonical URL) that Sailthru should use when spidering content. For pages with multiple URLs, this is required in order to prevent your content library from acquiring duplicate content under multiple URLs.
tags	String of comma-separated values or an array of strings	true	By default, for the user who views the page, Sailthru will attribute the stored interest tags for the current item that exist in your Content Library. If you wish to pass other dynamic interest tags dynamically upon a page view, use the tags parameter to send those tags. Note: If you are manually tracking page views (i.e. you have set autoTrackPageView to false), you should instead pass tags via

			the trackPageView function.
spider	Bool	true	Determines whether Sailthru should spider the page on which the Javascript appears. Set this to false to prevent spidering. (Note that when set to false, the page will not be spidered or count pageviews.)

Include any keys as key-value pairs using object-literal syntax. For example:

```
Sailthru.SPM.setup(123456789, {
 autoTrackPageviews: false,
 useStoredTags: false
});
```

trackPageview

Optional. Used to manually register pageviews. This is Ideal for single-page views, such as infinite scroll, ar other dynamic single-page sites.

You can log a pageview using the spm.trackPageView function at any time. This is typically used when autoTrackPageView is set to false. Note that if you leave autoTrackPageView set to true and use spm.trackPageView, multiple pageviews will be logged.

```
spm.trackPageView(url, {
tags: tagsVar,
onSuccess : function() {console.log('trackPageView success note');},
onError : function() {console.log('trackPageView failure note');}}
});
```

Parameters for trackPageView:

- **Required**: Replace **url** with a variable representing the URL for which you want to track a pageview.
- **Required**: JSON **object** parameter containing key-value pairs. (May be empty if no optional keys are included.)
  - **Optional**: If you wish to attribute a custom array of interest tags to the user who views this page rather than those which may be stored for the associated URL in the Content Library, replace **tagsVar** with your tags: either a string of comma-separated tags or an array of strings, c for each tag.
  - **Optional**: Include **onSuccess** or **onError** callbacks.

According to the instructions above, you can customize and include the following sample JavaScrip code for Instrumented Mode.

```
<script language="javascript" type="text/javascript" src="//ak.sail-horizon.com/onsite/persor
ize.v0.0.4.min.js" data-sailthru-setup="true"></script>

<script type="text/javascript">

var customerId = 'your customerId';
 // via http://my.sailthru.com/settings/api_postbacks

Sailthru.SPM.setup(customerId, {
 autoTrackPageviews: false,
 useStoredTags: false
});

Sailthru.SPM.trackPageView(url, {
 tags: "tag1,tag2,tag3",
 onSuccess : function() {},
 onError : function() {}
});

</script>
```

2 Implement Meta Tags and/or the Content API

Personalization Engine crawls pages that are found by our Spider or submitted using the Content API. Simi to search bots, the Spider records meta tags to determine the key attributes of each item-specific page, including interest tags, though you can also submit content metadata using the Content API.

Personalization Engine also measures content popularity by pageviews, which can further weight content in Personalization Engine-powered products. For a full list of algorithms that you can use to determine which items are displayed to users, see the Zephyr function personalize().

You can add Sailthru-specific meta tags to the content pages of your site. The text in “sailthru.tags” should correspond broadly to the interests of users. For example:

```
<meta name="sailthru.date" content="Tue, 05 Apr 2012 23:26:57 -0400" />
<meta name="sailthru.title" content="Belichick named top coach in the NFL" />
<meta name="sailthru.tags" content="sports, football, nfl, new-england-patriots" />
```

- If you already have a meta “date” tag or suitable tags as meta “keywords”, you do not need to provide those as custom Sailthru tags. However, we recommend that interest tags are Sailthru-specific (i.e. th “sailthru.tags” is used) for optimum performance.
- Additional supported Sailthru meta tags exist, beyond these examples. For details and best practices, see Personalization Engine Meta Tags.
- To start using Sailthru without suitable meta keywords or sailthru.tags, consider Sailthru’s Auto-Tagging feature.

For more information on using the Content API to submit page metadata or request spidering of pages, see the Content API page.

3 Set Content Spidering Preferences

You can leverage Sailthru's spider to automatically add your content pages and their metadata to your Sailtl Content Library.

Clients with advanced content needs can alternatively add content/attributes exclusively using Sailthru's Content API, or use a hybrid approach with both features: for example, spider content, then use the API to a additional metadata, update content, delete outdated content, or request a respider.

If you intend to spider any content:

- Specify which URLs and URL structures to allow Sailthru to crawl in Spider Preferences. For example you can choose to "include" your entire domain, but "exclude" category and landing pages.
    - Note: An alternative way to exclude pages is through the JavaScript on each page using the "spider" parameter (requires Instrumented Mode and personalize JS v0.0.3+ described on this page). To exclusively use the JS on each page to determine whether the page is spidered, leave "Include Rules" blank. If you simply want to exclude pages using JS, you can add "Include Rules and use the JS to exclude pages from that set of matching URLs.
- To learn more about setting up these rules, see Personalization Engine Setup: Include/Exclude Pages
- Note that query strings appended to URLs (for example, example.com/?queryparameter=queryvalue) ignored by the spider. If you would like certain query parameters and values to be included as part of t stored URL in the Content Library, add those parameter names on the Spider Preferences page. Ente them in the "Spiderable Query Params" section, one per line.
- The first time a spiderable page is loaded, its metadata is added to the Content Library.
- If your site is login-only
    - Click for instructions on how to let the spider in automatically

      The Spider passes an HTTP User-Agent header that is unique to your account. Follow the ste below to obtain your unique Spider User-Agent name, and then allow the agent access to you site.

      1. Login to my.sailthru, and go to your Settings page.
      2. Click the lock icon under the heading API Key and Secrets at the top left of the page.
      3. Copy the text from the Spider User-Agent section. For example: "Sailthru Content Spide [Account Name/abcefgh12345]"
      4. On your login website, you will typically check if a user is logged in and if not, redirect th to the login page. To this existing code, you just need to add a check of the SERVER/US AGENT. This check uses the Content Spider key. The website recognizes the key, and allows the Horizon spider to login to gather user interests/clicks.

      For example (in PHP):

```
<?php
    . . .
    //ensure user is logged in
    if (!isset($_SESSION['user']) && $_SERVER["HTTP_USER_AGENT"]=="Sailthru
Content Spider [Sailthru Client Name/1233456783248l348asfasdf]") {
        redirect("login_page");
    }
    . . . web page desired to be spidered . . .
```

If you are having difficulty editing your site to allow our spider access or are unsure of the bes way to go about doing this, please contact Sailthru's Support Team.

If you intend to manage content exclusively using the Content API, with no spidering:

- Do not enter any "Include Rules" on the Spider Preferences page.

Note: If the web server your site is hosted on is forcing GZIP compression, the pages cannot be spidered. I this case, please use the Content API instead.

4 Understand and Use Sailthru Cookies

As users interact with your site and email, browser cookies are dropped to track their activity and identify known users.

The following cookies are dropped automatically by the Personalize JavaScript, as needed. The only configuration steps you may require would pertain to the described API integrations.

- `sailthru_content` – Tracks recent pageviews for all visitors, and can be used to populate a new use profile.
  Note: It is highly recommended that you configure your site to submit this cookie value through the Us API on each user signup. See Anonymous User Conversion.
- `sailthru_hid` – Uniquely identifies known users in order to track their onsite activity. It is dropped ur the following scenarios:
  - Signup – When a new user signs up on your site.
  - Site Login – When an existing user logs in, if the cookie was not already present.
  - Email Click – When an uncookied user clicks through an email from Sailthru and visits your site.
  - Purchase – When a user makes a purchase on your site and the email address does not yet be to a known user. (Commerce sites only. Requires Purchase API integration.)
- `sailthru_bid` – Set when a subscriber opens and clicks an email link, to attribute onsite actions to a particular email campaign. (Note that this requires that you have set up a CNAME for link rewriting an have added this to the Link Domain setting in your account.) Commerce clients should pass this cooki as message_id when making any purchase API call.
- `sailthru_pc` – Tracks SPM clicks. This only applies if you have implemented Sailthru's Site Personalization Manager.