Product Documentation:

Tracking Scripts

Table of Contents

Common Terminology

Terminology	Understanding
HyperText Markup Language or HTML	HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
Script	A script is a programming language that is used to manipulate, customize, and automate the facilities of an existing system.
Header	A website header is a visual typographic strip or menu that typically runs across the top of a website.
Footer	A website's footer is an area located at the bottom of every page on a website, below the main body content.
<head>	The <head> element is a container for metadata.
<body>	The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
Substitutions	A Substitution replaces the element in place with dynamic information.
jQuery	jQuery is a JavaScript Library that greatly simplifies JavaScript programming.
Globally Unique Identifier or GUID	GUID is a 128-bit label used for information in computer systems.

Understanding Tracking Scripts

Tracking Scripts allow you to leverage custom JavaScript and jQuery to customize your Thought Industries platform. Tracking scripts allow you to see what's holding your page visitors' attention, how long they're staying on your page, and how they interact with your content. This kind of information allows you to make improvements to your page design, content, call to action, or even your product.

Tracking scripts are pieces of code that are added by injecting these code pieces into various parts of the system.

The following are common tracking script providers, along with use cases for using tracking scripts:

Google Analytics
Google Analytics is one of most popular and powerful tracking tools that can display in-depth analytics and stats about your site's traffic.

Google Tag Manager
Google Tag Manager allows you to manage all other scripts in one area. Google Tag Manager serves a different purpose from Google Analytics, but works well with it as allows you to customize the data that gets sent to Google Analytics.

Facebook Tracking Pixel
Facebook's Pixel tracks visitors to your site and ties their data to any known Facebook accounts. This makes it possible to retarget those who visit your website, reducing your ad costs and increasing your conversions.
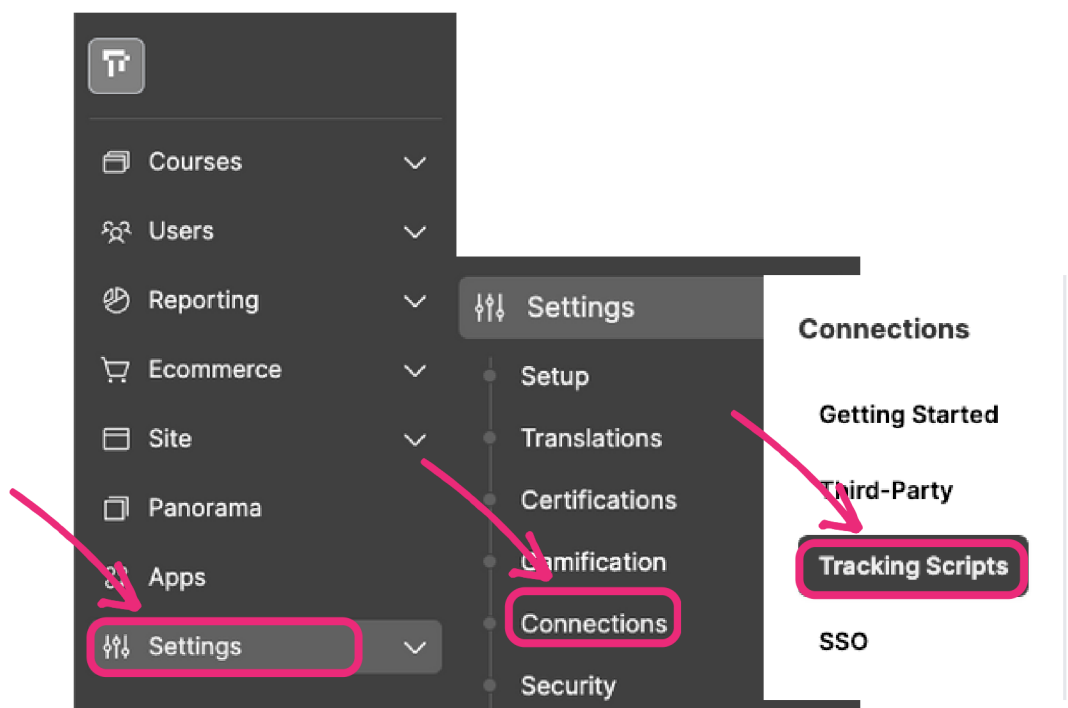
Note: There are dozens of different types of tracking scripts available, each from a range of different providers and each allowing you to understand a different type of interaction.

Prerequisites

- Thought Industries data will be redirected into external data systems after the script action. Please refer to your legal team's guidance before script creation.

- Some knowledge of HTML and JavaScript is helpful to use tracking codes effectively.

- When setting up tracking scripts, you must have access to an administrative resource supporting the script creation and/or provider to troubleshoot.

Script Configuration

1. Navigate to site **Settings** from the admin menu.
2. Select the **Connections** options.
3. Select the **Tracking Scripts** option.
4. Manually enter or copy and paste the appropriate scripts to any of the available fields.

Available Scripting Fields

Scripting Fields	
Header	Inserted before the </head> on every page
Footer	Inserted before the </body> on every page
Order Confirmation	Inserted on order confirmation pages
Email Capture	Executed whenever an email is captured
Page View	Executed whenever a page is viewed
Product View	Executed whenever a product is viewed
Add Cart Item	Executed whenever a cart item is added
Remove Cart Item	Executed whenever a cart item is removed
Checkout - Name/Address	Executed when a learner visits the checkout
Checkout - Information	Executed when a learner proceeds to the credit card page
Section Completed	Executed when a learner completes a section
Course Completed	Executed when a learner completes a course
Onboarding Survey Completed	Executed upon completion of onboarding survey

Header Scripts

Header scripts are among the most popular area to insert tracking scripts, alongside Footer scripts. When you insert a Header script it is inserted, in the source code, before the </head> on every page in the Thought Industries Platform.

Examples you my place in a Header Script:

```
<script src="javascript.js"></script>
```

```
<link href="https://fonts.example.com/css2?family=Roboto&display=swap"
rel="stylesheet">
```

Footer Scripts

Footer scripts are the most popular area to insert tracking scripts. If you are not sure where to put a tracking script, we recommend putting all tracking scripts here. When you insert a footer script it is inserted, in the source code, before the </body> on every page in the Thought Industries Platform.

Example you my place in a Footer Script:

```
<script> alert("Hello World!"); </script>
```

Note: jQuery is already included on every page, so there's no need to include your own version.

Adding your own jQuery could cause conflicts and is not recommend.

7

Order Confirmation Scripts

When you insert an order confirmation script, it is run after a successful action. The action only happens on the order confirmation page, after a successful order, following a checkout transaction or registration for free content.

The following substitutions are available for order confirmation scripts:

Substitutions	
{{totalInDollars}}	Total value of the order in dollars
{{totalInCents}}	Total value of the order in cents
{{orderId}}	The unique ID of the order

Within order confirmation script, you also have access to window.orderItems, which contains a variety of details on everything that was ordered, and access to window.currentUser, which contains information about the user who purchased.

An example order confirmation script that checks if the course with the slug "course-slug-here" was purchased may look like the following:

```
<script type="text/javascript">
window.orderItems.forEach(function(orderItem) { if (orderItem.slug === 'course-slug-here') { console.log('I am purchasing the specific course', orderItem); // insert any code you want to execute here } });
</script>
```

Email Capture Scripts

When you insert an email capture script, it is run after a successful action. The action only happens following an email capture from a course waitlist or paywall.

The following substitutions are available for email capture scripts:

Substitutions	
{{email}}	The email captured during action
{{source}}	The source of the email capture, either "coursePaywall" or "courseWaitlist"
{{courseTitle}}	The title of the course the email was captured on Because course titles can change, we recommend using the slug, SKU, or ID
{{courseSlug}}	The slug of the course the email was captured on
{{courseSku}}	The SKU of the course the email was captured on
{{courseId}}	The ID of the course the email was captured on

An example email capture script look like the following:

```
<script type="text/javascript">
alert('{{email}} captured from {{source}} on course {{courseSku}}');
</script>
```

Page View Scripts

When you insert a page view script, it is run after a successful action. The action only happens for page views on the public site and learner view. This action will not be present on the following pages: sign in, sign out, forgot password, order confirmation, manager interface, and account management.

The following substitutions are available for page view scripts:

Substitutions	
{{url}}	The URL of the page viewed
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields

An example page view script may look like the following:

```
<script type="text/javascript">
if ('{{url}}' === 'URL_TO_THE_COURSE_PAGE_HERE')
{ alert('I am viewing the course'); // insert any code you want to execute here }
</script>
```

Product View Scripts

When you insert a product view script, it is run after a successful action. The action only happens whenever a purchasable item (course, collection, etc.) is viewed.

The following substitutions are available for product view scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields
{{purchasableType}}	The type of the item viewed
{{purchasableTitle}}	The title of the item viewed
{{purchasableSlug}}	The slug of the item viewed
{{purchasableId}}	The UUID of the item viewed

Substitutions	
{{purchasableSku}}	The SKU of the item viewed
{{priceInCents}}	The price (in cents) of the item viewed
{{priceInDollars}}	The price (in dollars) of the item viewed

An example product view script may look like the following:

```
<script type="text/javascript">
alert('{{purchasableTitle}} viewed by {{email}})');
</script>
```

Add Cart Item & Remove Cart Item Scripts

When you insert an add cart item and remove cart script, it is run after a successful action. The action only happens whenever when a item is added or removed.

The following substitutions are available for add cart Item and remove cart scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields
{{purchasableType}}	The type of the item added or removed from the cart
{{purchasableTitle}}	The title of the item added or removed from the cart
{{purchasableSlug}}	The slug of the item added or removed from the cart
{{purchasableId}}	The UUID of the item added or removed from the cart

Substitutions	
{{purchasableSku}}	The SKU of the item added or removed from the cart
{{priceInCents}}	The price (in cents) of the item added or removed from the cart
{{priceInDollars}}	The price (in dollars) of the item added or removed from the cart

An example add cart Item and remove cart script may look like the following:

```
<script type="text/javascript">
alert('{{purchasableTitle}} added to the cart for {{priceInDollars}}');
</script>
```

Checkout Scripts – Step 1 & Step 2

When you insert a checkout script, it is run after a successful action. The action only happens during the checkout flow, before a user completes a purchase. Step 1 is run when the checkout flow is first entered, and step 2 is run after collecting name/address information from the customer.

The following substitutions are available for checkout scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields
{{address1}}	The shipping address line 1
{{address2}}	The shipping address line 2
{{city}}	The shipping city

15

Substitutions	
{{state}}	The shipping state
{{zipCode}}	The shipping zip code
{{country}}	The shipping country

The order is available on the window object, and can be manipulated with javascript window.order.

An example checkout script may look like the following:

```
<script type="text/javascript">
window.order.orderItems.forEach(function(orderItem){ alert(orderItem.title + '
was ordered by {{firstName}} {{lastName}}');}) </script>
```

Section Completed Scripts

When you insert a section completed script, it is run after a successful action. The action only happens whenever a section is completed in the learner view.

The following substitutions are available for section completed scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields
{{sectionTitle}}	The title of the section that was completed. Because section titles can change, we recommend using the slug.
{{sectionSlug}}	The slug of the section that was completed
{{courseTitle}}	The title of the course the section was completed in. Because course titles can change, we recommend using the slug, SKU, or ID.

Substitutions
{{courseSlug}}	The slug of the course the section was completed in
{{courseSku}}	The SKU of the course the section was completed in
{{courseId}}	The ID of the course the section was completed in

Course Completed Scripts

When you insert a course completed script, it is run after a successful action. The action only happens whenever a course is completed in the learner view.

The following substitutions are available for course completed scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user
{{userId}}	The GUID of the currently logged in user
{{ref1}} - {{ref10}}	The user's references fields
{{courseTitle}}	The title of the course completed. Because course titles can change, we recommend using the slug, SKU, or ID.
{{courseSlug}}	The slug of the course completed
{{courseSku}}	The SKU of the course completed
{{courseId}}	The ID of the course completed

19

Onboarding Survey Completed Scripts

When you insert a onboarding survey completed script, it is run after a successful action. The action only happens whenever a Onboarding Survey is completed in the learner view.

The following substitutions are available for onboarding survey completed scripts:

Substitutions	
{{email}}	The email of the currently logged in user
{{role}}	The role of the currently logged in user
{{firstName}}	The first name of the currently logged in user
{{lastName}}	The last name of the currently logged in user

Additionally, `window.onboardingSurveyResponses` will contain the responses to each question in the window object.

The Window Object

Certain context-specific data can be pulled into tracking scripts directly as variable substitutions on a script-by-script basis. However, some data not listed here can be pulled into your tracking scripts via the browser's window object.

To see the entirety of the window object, follow these steps:
1. Open the browser inspector
2. Navigate to the Console tab, then type window and hit enter.

Feel free to explore this object. Most of what is there has to do with how the browser compiles the Thought Industries app and is not useful for customizations or integrations. However, the following may be useful:

- window.CONF.preload contains information on the currentUser (such as their user detail, client, and license) and the instance

- window.location.pathName contains the current URL path and can be useful, for example, for isolating the current courseSlug in the browser (typing window.location.pathname.split("/")[3] will yield the courseSlug itself)

The following variables are available for window.orderItems in the Order Confirmation scripts:

Variables	
couponCode	The coupon code used (if any)
priceInCents	The price in cents of order item
purchasableType	The type of order item
purchasableId	The UUID of order item
quantity	The quantity of item ordered
total	The order item total
SKU	The SKU of the item ordered
title	The name of the product ordered
variationLabel	The particular variation of the ordered item

The following variables are available for window.order in the Checkout scripts:

Variables	
gift	Whether or not an order is a gift. If gift is "True" you can also receive information on giftMessage, giftDeliveryDate, giftRecipientEmail, giftRecipientFirstName and giftRecipientLastName
newsletterSignupChecked	Whether or not the user clicked to sign up for a newsletter
totalInCents	The price of order total in cents as a number
totalInDollars	The price of order total in dollars as a string
orderItems	An array of order items
couponCode	The coupon code used (if any)
interval	The interval at which a subscription is paid (if any)
needShippingAddress	Whether or not an order item needs a shipping address
priceInCents	The price in cents of order item
quantity	The number of items ordered
taxable	Whether or not an order item is taxable
variationLabel	The particular variation of the ordered item

Powering the Business
of Learning

Thought
INDUSTRIES