

Installation and Setup

This guide will be able to help walk you through on how to get started with installing Inncee onto your custom website.

1.1 Create Account

Create an account at <https://beta.inncee.com>. This allows our system to generate

- a unique account ID
- a tracking script code to install on your store
- and a secret key to allow you to authenticate the JSON data that you can send to our API endpoints.

Once generated, all this information will be sent to you via email.

Note: Ignore the pop-up that asks you to connect your store to Shopify.

1.2 Install your tracking script

You will need to install our tracking script library after the opening of the <head> tag found in every page of your website.

```
<script type="text/javascript"
src="https://shopify.inncee.com/tracker/YOUR_ACCOUNT_ID?shop=YOUR_STORE_URL"></script>
```

Once the script has been installed, you'll be able to start recording events on your website.

1.3 Tracking Events

For our platform to be able to analyze your data and look for information, we need to be able to record events that happened on your website. Some events are automatically tracked while others requires you to install code snippets manually or calling our webhook end points in our app to send data to us.

Event Setup

Install these code snippets into your online store to begin tracking the following events.

Event	Description	Code
User logged in	<p>Required. This code block helps us to identify visitors when they are logged in. It should be included on every public page of your website when the user has logged in.</p> <p>It can be omitted if the user is not logged in.</p> <p>innceelD * – Your unique account ID.</p>	<pre>analytics.identify('USER_ID', { 'innceelD': 'YOUR_ACCOUNT_ID' });</pre>
Add to cart	<p>Required. This code block records the event whenever a user clicks on the add to cart button. It should be included on every page that has an "Add To Cart" button.</p> <p>innceelD * – Your unique account ID.</p> <p>categoryID * - The unique ID of your category.</p> <p>productID * – The unique ID of the product being added to cart</p> <p>variantID</p>	<pre>analytics.track("addtocart",{ 'innceelD' : 'YOUR_ACCOUNT_ID', 'categoryID': 'yyyyy', 'productID': 'ppppp', 'variantID': 'ppppp', 'Price': '12.50', 'Quantity': '1', 'Currency': '\$', });</pre>
Checkout	<p>Required. This code block records the event when a user initiates a checkout. It should be included on every page that has a checkout button.</p> <p>innceelD * – Your unique account ID</p>	<pre>analytics.track("Checkout",{ 'innceelD' : 'YOUR_ACCOUNT_ID', 'cartID' : 'qqqqqq' });</pre>

	cartID * - A unique ID of the shopping cart being checked out	
Order Completed	Required. Records the event when an order has been completed innceID * - Your unique account ID orderId * - The unique order ID	<code>analytics.track("OrderComplete",{ 'innceID' : 'YOUR_ACCOUNT_ID', 'orderId' : 'xxxxxx', });</code>

Automatically Tracked Events

There are some events that are automatically tracked by us when you first install our script. No further actions will be required from you.

- Page views
- Add to cart clicks
- UTM Campaign visits

Backend Events

The code snippets above help us to track events that happen on your online store. We will also need to track certain events that happen in the backend of your store as well.

We've setup a series of API end points that ready to receive JSON data from your store when certain backend events happen.

For example, when a new order has been created, you can send a JSON response with information about the new order to our API end point and our platform can begin storing that information for analysis.

Prerequisites

When making API calls to our servers, you are required to first generate a HMAC and attach it to your headers along with the timestamp used to generate it.

The recommended method for generating the HMAC using PHP is as follows

```
$timestamp = time();
$secret = YOUR_SHARED_SECRET;
$domain = "YOUR_DOMAIN.COM";

$message = "store=".$domain."&timestamp=".$timestamp;
$hmac = base64_encode(hash_hmac('sha256', $message, $secret, true));
```

When making calls to our servers, attach the timestamp as a URL parameter

```
$ch = curl_init();  
  
$url = "https://api.inncee.com/hooks/order/create?timestamp".$timestamp  
curl_setopt($ch,CURLOPT_URL, $url);
```

the timestamp should be the same one used to generate the HMAC.

Now attach the HMAC as a header with the key HTTP_X_HMAC_SHA256

```
$hmac_header = "HTTP_X_HMAC_SHA256:".$hmac  
curl_setopt($ch,CURLOPT_HTTPHEADER, array($hmac_header));
```

Once done, you are now good to attach your data and send to our servers.

See the next section for a list of all our API end points and when you can send data to it.

ORDERS API

Orders	POST	/hooks/order/create
Called when a new order has been created.		
Orders	PUT	/hooks/order/update
Called when an order or many orders has been updated		
<pre>{ "orders": [{ "id": 12345667, "created_at": "2017-02-08T14:56:07-05:00", "updated_at": "2017-02-08T14:56:07-05:00", "fulfilled_at": null, "total_price": "199.00", "subtotal_price": "199.00", "total_tax": "0.00", "taxes_included": false, "currency": "SGD", "total_discounts": "0.00", "total_line_items_price": "199.00", "order_reference": "#1002", "order_source": "web", "order_status": "open", "cancelled_at": NULL, "cancel_reason": NULL, "customer_id": 1244345, "discount_codes": [], "line_items": [{ "line_item_id": 1071823175, "product_id": 447654529, "variant_id": 12133424, "product_name": "IPod Touch 8GB", "quantity": 1, "price": "199.00", "sku": "IPOD2009BLACK", "variant_name": "Black", "brand": "Apple", "fulfillable_quantity": 1, "total_discount": "0.00" }], "refunds": [{ "created_at": "", "adjustment": "-3.50", "tax_adjustment": "-1.00", "quantity": "2" }] }] }</pre>		

<pre> }] } </pre>		
Orders	DELETE	/hooks/order/{#orderid}/delete
Called when an order has been marked for deletion		
id	The unique numeric identifier for the order. Order id should be unique across the system. No 2 orders should have the same ids	
created_at	The date and time when the order was created	
updated_at	The date and time when the order was updated.	
fulfilled_at	The date and time when the order was fulfilled. If order is not fulfilled, value should be NULL.	
total_price	The sum of all the prices of all the items in the order, taxes and discounts included. (value must be positive)	
subtotal_price	Price of the order before shipping and taxes.	
total_tax	The sum of all the taxes applied to the order. (value must be positive)	
taxes_included	States whether or not taxes are included in the order subtotal. Valid values include "true" or "false"	
currency	The three letter code for the currency used for payment	
total_discounts	The total amount of discounts to be applied to the price of the order	
total_line_items_price	The sum of all the prices of all the items in the order	
Order_reference	The order reference number that the customer sees.	
order_source	Where the order originated. Valid values include "web", "pos" or "mobile". Mobile refers to orders that originated from a native app	
order_status	Status of the order. Valid values include <ul style="list-style-type: none"> - open – For orders that have not been fulfilled - closed – For orders that have been fulfilled and completed - cancelled – For orders that have been cancelled 	
cancelled_at	The date and time when the order was cancelled. Value is NULL if order is not cancelled	
cancel_reason	The reason why the order was cancelled. If the order was not cancelled, this value is "null." If the order was cancelled, the value will be one of the following: <ul style="list-style-type: none"> - customer: The customer changed or cancelled the order. - fraud: The order was fraudulent. - inventory: Items in the order were not in inventory. - other: The order was cancelled for a reason not in the list above. 	
customer_id	The unique numeric identifier for the customer.	

discount_codes	<p>List of discount codes that can be applied to the order. If no codes exist the value will default to blank.</p> <p>A Discount code will include the following fields:</p> <ul style="list-style-type: none"> - amount: The amount of the discount. The type field determines the unit of this amount. - code: The discount code. - type: The type of discount. Valid values are: <ul style="list-style-type: none"> o fixed_amount: The default value. Applies a discount of amount as a unit of the store's currency. For example, if amount is 30 and the store's currency is USD, then 30 USD is deducted from the order total when the discount is applied. o percentage: Applies a percentage discount of amount. For example, if amount is 30, then 30% of the order total will be deducted when the discount is applied. o shipping: Applies a free shipping discount on orders that have a shipping rate less than or equal to amount. For example, if amount is 30, the discount will give the customer free shipping for any shipping rate that is less than or equal to \$30.
line_items	<p>A list of line item objects, each one containing information about an item in the order. Each line_item object has the following properties:</p> <ul style="list-style-type: none"> - line_item_id: the unique numerical identifier of the line item - product_id: the unique numerical identifier of the product - variant_id: the unique numerical identifier of the product variant - product_name: the name of the product - quantity: the quantity ordered by the customer - price: the price of the product ordered - sku: A unique identifier of the product for fulfillment - variant_name: the name of the product variant - brand: the name of the supplier or brand of the product - fulfillable_quantity: - total_discount: The total discount amount applied to this line item. This value is not subtracted in the line item price.
refunds	<p>List of refunds applied to the order. If no refunds exists, the value will default to blank</p> <p>Each refund object has the following properties:</p> <ul style="list-style-type: none"> - created_at: the date and time that the refund was created - adjustment: - tax_adjustment:

CUSTOMERS API

Customers	POST	/hooks/customers/create
Called when a new customer account has been created.		
Customers	PUT	/hooks/customers/update
Called when a customer account has been updated		
<pre>{ "customers": [{ "id": 543213, "created_at": "2017-02-08T14:56:07-05:00", "updated_at": "2017-02-08T14:56:07-05:00", "accepts_marketing": true, "city": "Ottawa", "province": "Ontario", "country": "Canada", "status": "enabled" }] }</pre>		
Customers	DELETE	/hooks/customers/{#customerid}/delete
Called when a customer account has been deleted		
id	The unique numerical identifier for the customer	
created_at	The date and time that the customer was created	
updated_at	Date and time that the customer's information was updated	
accepts_marketing	Indicates whether the customer has consented to be sent marketing materials via emails. Valid values are "true" and "false"	
city	The customer's city	
province	The customer's province	
country	The customer's country	
status	The status of the customer's account. Valid states are <ul style="list-style-type: none"> - enabled : The customer has verified their email address and created an active account - disabled: The customer has not verified their email addresses and logged into their account yet. Account is considered not active 	

PRODUCTS API

Products	POST	/hooks/products/create
Called when a new product has been created.		
Products	PUT	/hooks/products/update
Called when a product has been updated		
<pre>{ "products": [{ "id": 3245465, "created_at": "2017-02-08T14:56:07-05:00", "updated_at": "2017-02-08T14:56:07-05:00", "product_name": "IPod Touch 8GB", "brand": "Apple", "tags": "music, device, white", "path": "product/android-phone/lenovo-a1000-hitam-/0889800170294", "image_url": "https://api.oktagon.co.id/core/images/product/ZTE%20Kis%203%20V8118w%20Biru.jpg", "variants": [{ "id": 123454534, "product_id": 3245465, "name": "8G", "price": "128.00", "sku": "IPOD2008PINK", "inventory_quantity": null }, { "id": 123454534, "product_id": 3245465, "name": "16G", "price": "128.00", "sku": "IPOD2008PINK", "inventory_quantity": null }] }] }</pre>		
Products	DELETE	/hooks/products/{#productid}/delete
Called when a product has been deleted		
id	The unique numerical identifier for the product	
created_at	The date and time that the product was created	
updated_at	The date and time that the product was updated	

product_name	Name of the product
brand	Supplier or brand name of the product
tags	A categorization that a product can be tagged with, commonly used for filtering and searching. Each comma-separated tag has a character limit of 255.
path	The unique url path visitors use to view the product
image_url	The url to the main image of the product
variants	<p>A list of variant objects, each representing a slightly different version of the product. For example, if a product comes in different sizes and colors, each size and color permutation (such as "small black", "medium black", "large blue"), would be a variant.</p> <p>If your product does not have a variant, then simply create 1 variant record with the name "Default" to record the product's SKU, price and other details.</p> <ul style="list-style-type: none"> - id: The unique numerical identifier to identify the variant. - product_id: The unique product id that the variant belongs to. - name: Name of the variant - price: Price of the variant - sku: SKU number of the variant - inventory_quantity: Total available units of inventory available for sale.

CATEGORY API

Category	POST	/hooks/category/create
Called when a new category has been created.		
Category	PUT	/hooks/category/update
Called when a category's information has been updated		
<pre>{ "categories": [{ "id": 3245465, "created_at": "2017-02-08T14:56:07-05:00", "updated_at": "2017-02-08T14:56:07-05:00", "category_name": "Android Phones", "path": "/products/android-phone/AN03", "parent_id": "2434435" }] }</pre>		
Category	DELETE	/hooks/category/{#categoryid}/delete
Called when a category has been deleted		
id	The unique numerical identifier for the category	
created_at	The date and time that the category was created	
Updated_at	The date and time that the category information was updated	
Category_name	Name of the category displayed to the visitors	
Path	The unique url path that visitors use to view the category page	
Parent_id	The unique identifier of the parent category. If there is no parent category, then this value is set to NULL.	

CATEGORYMAP API

CategoryMap	POST	/hooks/categorymap/create
Called when you add a product to a category		
<pre>{ "categorymap": [{ "id": 3245465, "created_at": "2017-02-08T14:56:07-05:00", "updated_at": "2017-02-08T14:56:07-05:00", "category_id": 3424223, "product_id": 4344542 }] }</pre>		
CategoryMap	DELETE	/hooks/categorymap/delete
Called when a product has been removed from a category		
<pre>{ "categorymap": [{ "id": 3245465, "category_id": 3424223, "product_id": 4344542 }] }</pre>		
id	The unique numeric identifier for the mapping	
created_at	The date and time stamp that the product was added to the category	
updated_at	The date and timestamp that the product mapping was updated.	
category_id	The unique numeric identifier of the category that the product is mapped to.	
product_id	The unique numeric identifier of the product that is being mapped.	