# Conversion Tracking

*Conversion Tracking* is used to track orders that originate from an email sent from iPost. Conversion Tracking provides detailed information about these orders, such as items ordered, the amount of money spent, and any other custom data.

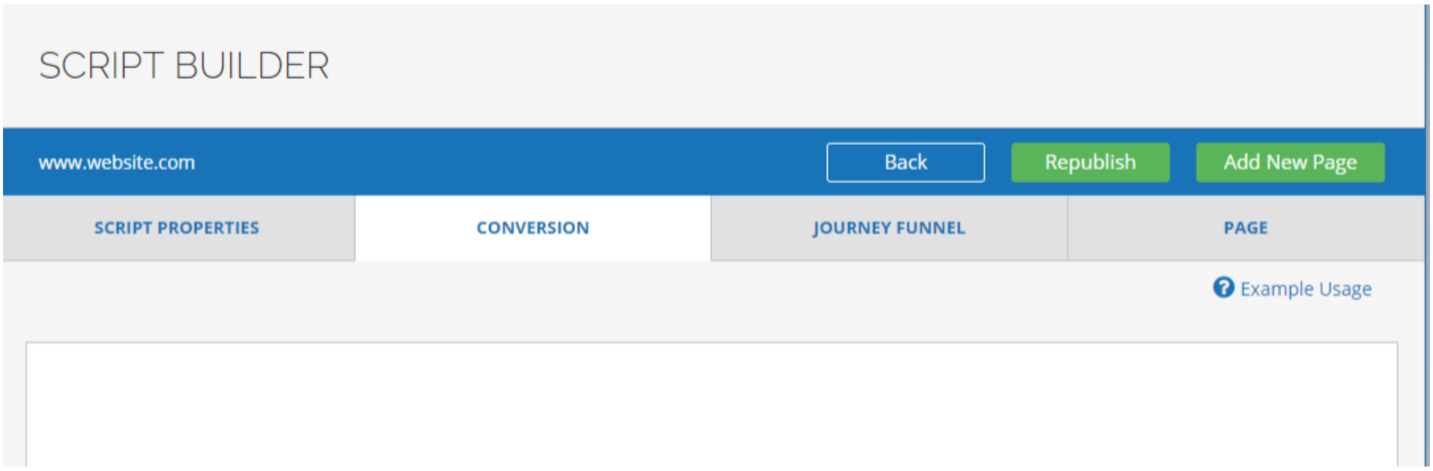
As the Conversion Tracking configuration relies on JavaScript tagging, it can also be used to track other information besides orders.  Custom conversion tracking can also be applied to the site to track information such as form submission or donations.

## How It Works

When the contact reaches the order confirmation page, the iPost JavaScript code checks for the existence and validity of the iqs cookie: if present, the code retrieves any conversion data (a purchase or custom conversion data) and sends it back to iPost, where it will get displayed in the Email Ticket for that send.

## Setting Up Conversion Tracking

From the *Web Tracking Manager* setting for your website, click on the Conversion tab, and then Click the Add New Page button.



A pop up module will appear with the various fields that will need to be configured in order to track the conversions.

## Page Name

The first step to configure Conversion Tracking  is to let the system recognize the page from where conversions will be tracked.  The page name is the URL of the page you want to track data from.

The options available are:

- Equals to
- Contains
- Begins with
- Ends with

The script will track data from the pages with a URL that is equal to the inserted string, contains it, begins with it or ends with it.

For example, if the page you want to track conversions from has URL “www.website.com/conversion_start.html”, you can select the option “Contains” and in the input box write “conversion_start.html”.

# Conversion Type

Currently, two types of conversions can be tracked: order or custom conversions. Order is conversion data associated with an ecommerce order. Custom conversions are used to track events such as if a form is completed, a video is watched, or a donation is made.

## Order

For each tracked element, some JavaScript code needs to be inserted that will extract the desired value from the web page and “return” it.

Special attention should be paid to using a set of functions that will find a unique location on the web page and return its value – unless where an array is specifically requested, in which case a loop will need to be present to collect data for each element of the array.

This type of Tracking data can accept the following elements:

| **Name** | **Description** | **Data Type** | **Required** |
|---|---|---|---|
| **OrderId** | The unique identifier for the order | String | Yes |
| OrderGrandTotal | The total amount for the order | Decimal | Yes |
| OrderDate | The date of the purchase | Datetime | Yes |
| OrderSubTotal | The total amount before taxes | Decimal | |
| OrderTax | The total tax amount | Decimal | |
| OrderShipping | The shipping costs | Decimal | |
| CouponUsed | Indicates if a coupon was used | Yes/No | |

| Name | Description | Data Type | Required |
|---|---|---|---|
| OnSale | Indicates if the item was on sale | Yes/No | |
| OrderDomain | The domain of the order | String | |
| OrderSource | The source for the order | String | |
| StoreId | The ID of the Store | String | |
| Items.Title | Name of the Item | String | Yes |
| Items.SKU | The SKU for the Item | String | Yes |
| Items.Price | The item price | Decimal | Yes |
| Items.Quantity | The number of items | Integer | Yes |
| Items.Description | The product description | String | |
| Items.Categories | | String | |
| Items.Brand | The  name of the brand of the item | String | |
| Items.Gender | | String | |
| Items.Color | The item color | String | |
| Items.Size | The size of the item | String | |
| Items.Additional1 | An additional field | String | |

Conversions will be tracked even if the Items object is empty.  If present, each order item will need to have a valid value for the 4 required fields within.

## Custom

This option lets the user track any other type of events happening on the selected page. The two items for each custom variable are: “eventTitle” and “details”.

## EventTitle

This is the name of the event you want to track.

```
return document.getElementById('event_title').innerHTML;
```

Click to copy

### Details

For each event, you can track an array of elements such as their title, quantity and description.

```
var detailSelector = document.querySelectorAll('.detail');
var detailCount = detailSelector.length;

for(var i=0;i<detailCount;i++) {
 var detailObject = {};
 detailObject.Title = detailSelector[i].querySelector('.Title').innerText;
 detailObject.Quantity = detailSelector[i].querySelector('.Quantity').innerText;
 detailObject.Description = detailSelector[i].querySelector('.Description').innerText;

 details.push(detailObject);
}

return details;
```

Click to copy

## Useful Functions

### querySelector()

The querySelector() method returns the first element that matches a specified CSS selector(s) in the document.

```
querySelector(".cartItemPrice")
```

Click to copy

This will return the first item with class “cartItemPrice”.

### querySelectorAll

The querySelectorAll() method returns all the elements in the document that matches a specified CSS selector.

It is useful to get to a specific location in the web page. Then, you can loop through all elements and extract the value you need to return.

```
document.querySelectorAll(".section-left div")
```

Click to copy

The above code will return all the <div> tags within the elements of class “section-left”.

```
cartArray = document.querySelectorAll(".cartItem")
cartArray.forEach(function (ea, i) {
        price = ea.querySelector(".cartItemPrice");
}
```

Click to copy

This will get the elements within the section of class “cartItem” and then loop through it to get the element with class “cartItemPrice”.

## getElementById

The getElementById() method returns the element that has the ID attribute with the specified value.

The below code will return the element that has id=”cartTable”.

```
document.getElementById("cartTable")
```

Click to copy

The code returning each value does not need to be written on just one line. It can be written on multiple lines. The only necessary step is “returning” the value at the end of the script.

If you need to customize the process, for example in case of page URLs  don’t change on the different conversion pages, you can refer to the documentation on “Conversion Tracking - Manually triggered”.

# Conversion Tracking - Manually Triggered

Our Manually Triggered option allows you to prepare conversion data and have a JS function send it to iPost, to be displayed in the Conversion Tracking data.

This option provides the flexibility to support almost any website, including single-page applications. As conversion data will be manually injected into the iPost Tracking Script, a page does not need to be configured under the conversion tab.

**NOTE:** Conversion pages are only required if you want to track data when a contact loads a particular page. If you are tracking both manually injected data and discreet pages, Conversion page(s) will need to be added to the Conversion section.

## Configuring Your Website To Track Data

Please follow these steps:

1. Copy the Script Loader content (screenshot above) and add it to your website code (before the end of the </body> tag).
2. Make sure to include the script in all the pages of your website in which you want to track conversion data from.
3. Our web tracking system allows two types of Conversion data: **order** and **custom**.
4. For each tracking data type, we have defined specific properties (parameters) that you can pass in your manual data object to track.
5. If you pass some other properties which are not listed in our properties list, then our web tracking system will ignore it and not show them in the reports.
6. Below is a list of properties that we allow for order and custom type.

### Order

```
var orderObj = {
 conversion: {
    ConversionPage: '', // key and value both are required
    ConversionType: 'order', // key and value both are required
    OrderID: '', // key and value both are required
    OrderGrandTotal: '', // key and value both are required
    OrderDate: ‘’, // key and value both are required
    OrderSource: ‘’,
    OrderStoreId: ‘’,
    OrderSubTotal: '',
    OrderTax: '',
    orderShipping: '',
    CouponUsed: '',
    OnSale: '',
    OrderDomain: '',
    Items: [ // the Items object is optional, but if present, the required fields below
will need to be passed
    {
        Title: '', // key and value both are required
        Sku: '', // key and value both are required
        Price: '', // key and value both are required - unit price for product
        Quantity: '', // key and value both are required
        Description: '',
        Categories: '',
        Brand: '',
        Color: '',
        Tags: ['tag1', 'tag2', 'tag3'],
        Collection: ['Collection1', 'Collection3', 'Collection3'],
        Variant: ''
    },
    {
        Title: '', // key and value both are required
        Sku: '', // key and value both are required
        Price: '', // key and value both are required - unit price for product
        Quantity: '', // key and value both are required
        Description: '',
        Categories: '',
        Brand: '',
        Color: '',
        Tags: ['tag1', 'tag2', 'tag3'],
        Collection: ['Collection1', 'Collection3', 'Collection3'],
        Variant: ''
    }]
 }
}
```

Click to copy

## Custom

```
var customObj = {
   conversion: {
            ConversionPage: '', // key and value both are required
            ConversionType: 'custom', // key and value both are required
            EventTitle: '', // key and value both are required
            ConversionDate: ‘’, // key and value both are required
            Details: [
                        {
                                    Title: '', // key and value both are required
                                    Quantity: '',
                                    Description: ''                },
                        {
                                    Title: '', // key and value both are required
                                    Quantity: '',
                                    Description: ''
                        }
            ]
```

```
  }
}
```

Click to copy

**NOTE**

- Key name should always be the same.
- URL encoding should be used when passing URLs.

## Tracking Data

After creating the objects, they will need to be sent to iPost, using the "send" function, as in the examples below:

- Order data:
  - window.iPostAnalytic.send(orderObj);
- Custom Data:
  - window.iPostAnalytic.send(customObj);

You can add them inside any event (button click, form submit, ajax call etc.).

You can also, at any point, run this command in the browser console log, to check if the data is being sent to iPost and the response received from it:

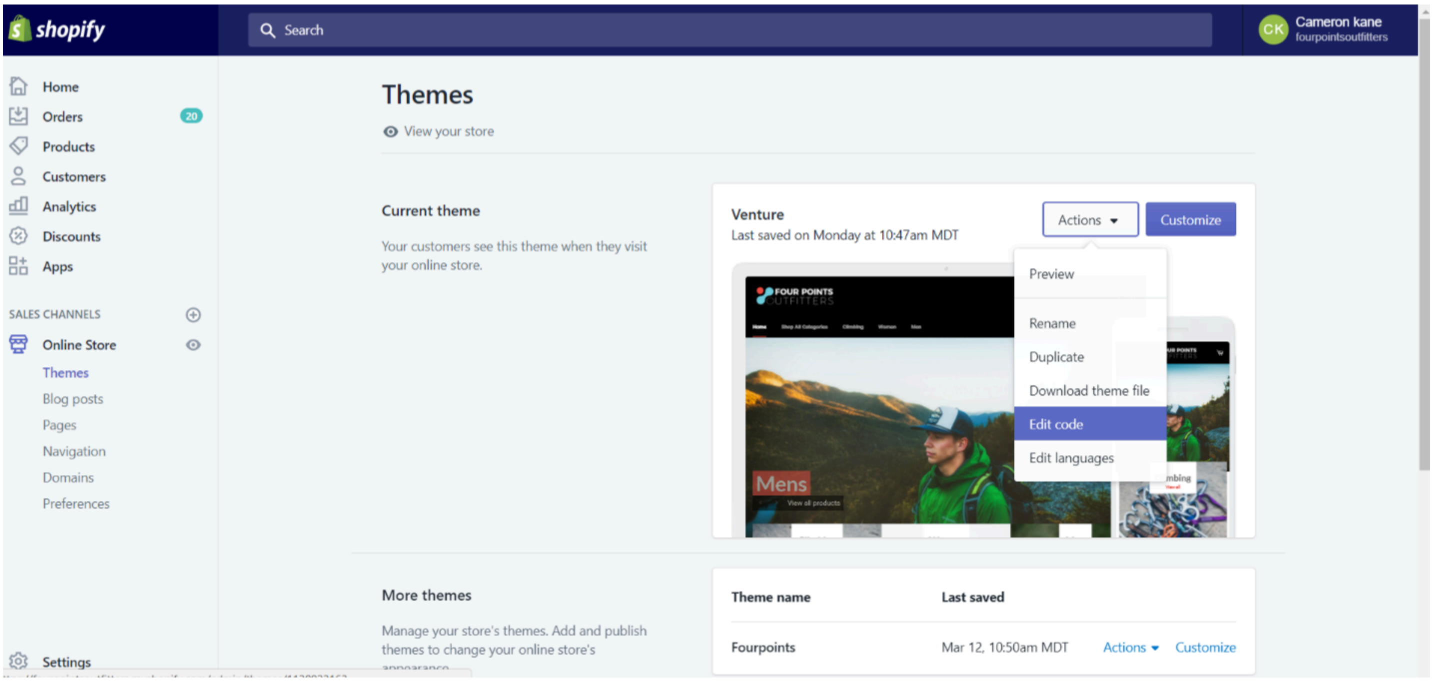
- window.iPostAnalytic;

# Conversion Tracking on Shopify

Follow the steps below to configure conversion tracking for Shopify.

1. Log into your Shopify admin account

2. Go to the Themes page and select Edit *code* to edit the theme currently being used:



Themes

Current theme

Your customers see this theme when they visit your online store.

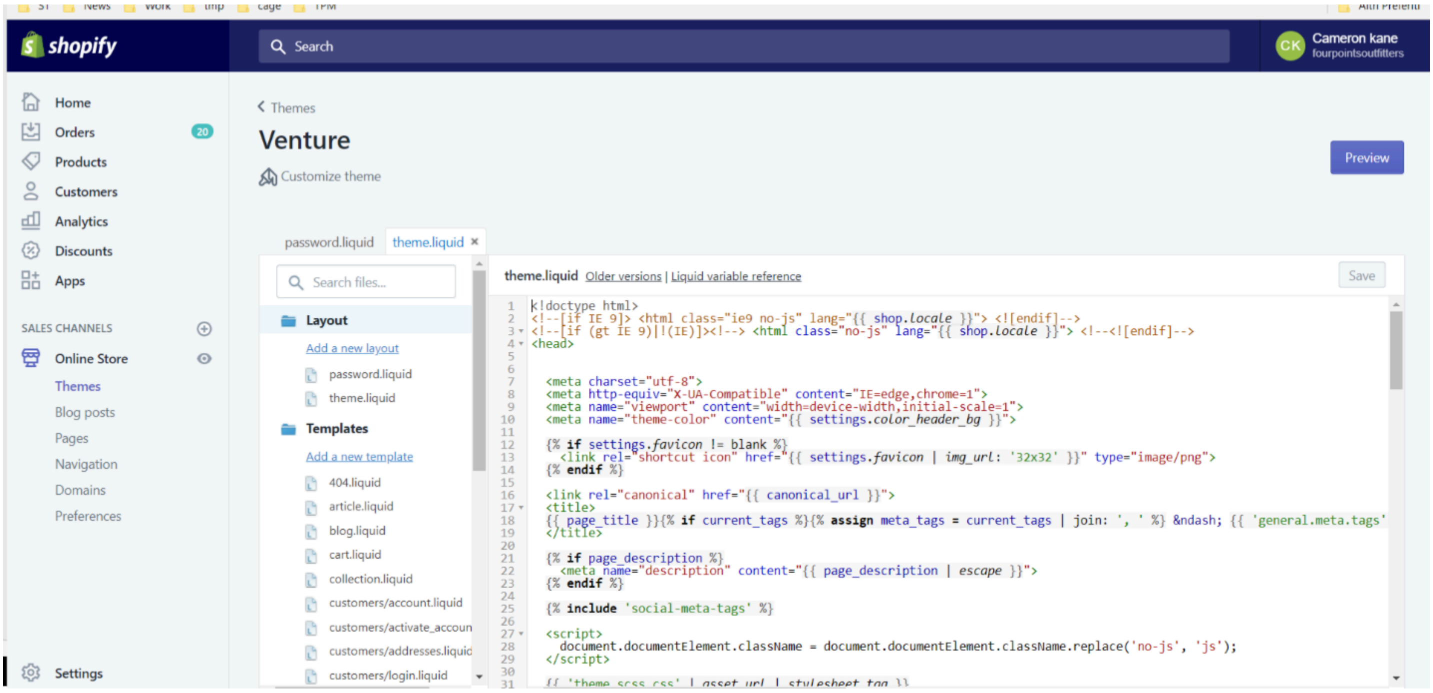
Venture
Last saved on Monday at 10:47am MDT

More themes

Manage your store's themes. Add and publish themes to change your online store's appearance.

| Theme name | Last saved |
|---|---|
| Fourpoints | Mar 12, 10:50am MDT |

3. Click on each layout element to open and edit it:



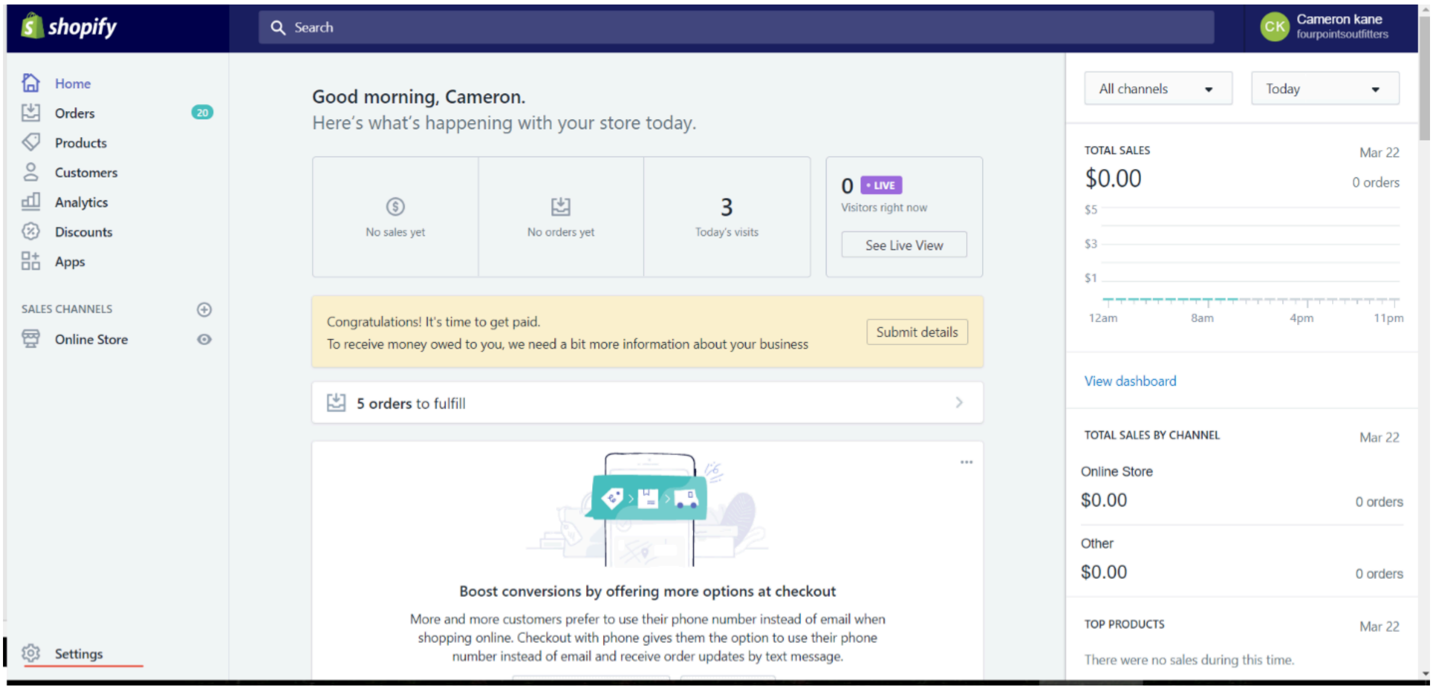
Venture

theme.liquid

```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}"> <![endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html class="no-js" lang="{{ shop.locale }}"> <!--<![endif]-->
<head>


  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <meta name="theme-color" content="{{ settings.color_header_bg }}">

  {% if settings.favicon != blank %}
    <link rel="shortcut icon" href="{{ settings.favicon | img_url: '32x32' }}" type="image/png">
  {% endif %}

  <link rel="canonical" href="{{ canonical_url }}">
  <title>
  {{ page_title }}{% if current_tags %}{% assign meta_tags = current_tags | join: ', ' %} &ndash; {{ 'general.meta.tags'
  </title>

  {% if page_description %}
    <meta name="description" content="{{ page_description | escape }}">
  {% endif %}

  {% include 'social-meta-tags' %}

  <script>
    document.documentElement.className = document.documentElement.className.replace('no-js', 'js');
  </script>

  {{ 'theme.scss.css' | asset_url | stylesheet_tag }}
```

4. In each of these pages, insert the “Script Loader” JavaScript code right before the end of </body> closing HTML tag.

5. After saving the pages, go to Settings:



Good morning, Cameron.
Here's what's happening with your store today.

No sales yet | No orders yet | 3 Today's visits | 0 LIVE Visitors right now

Congratulations! It's time to get paid.
To receive money owed to you, we need a bit more information about your business

5 orders to fulfill

Boost conversions by offering more options at checkout

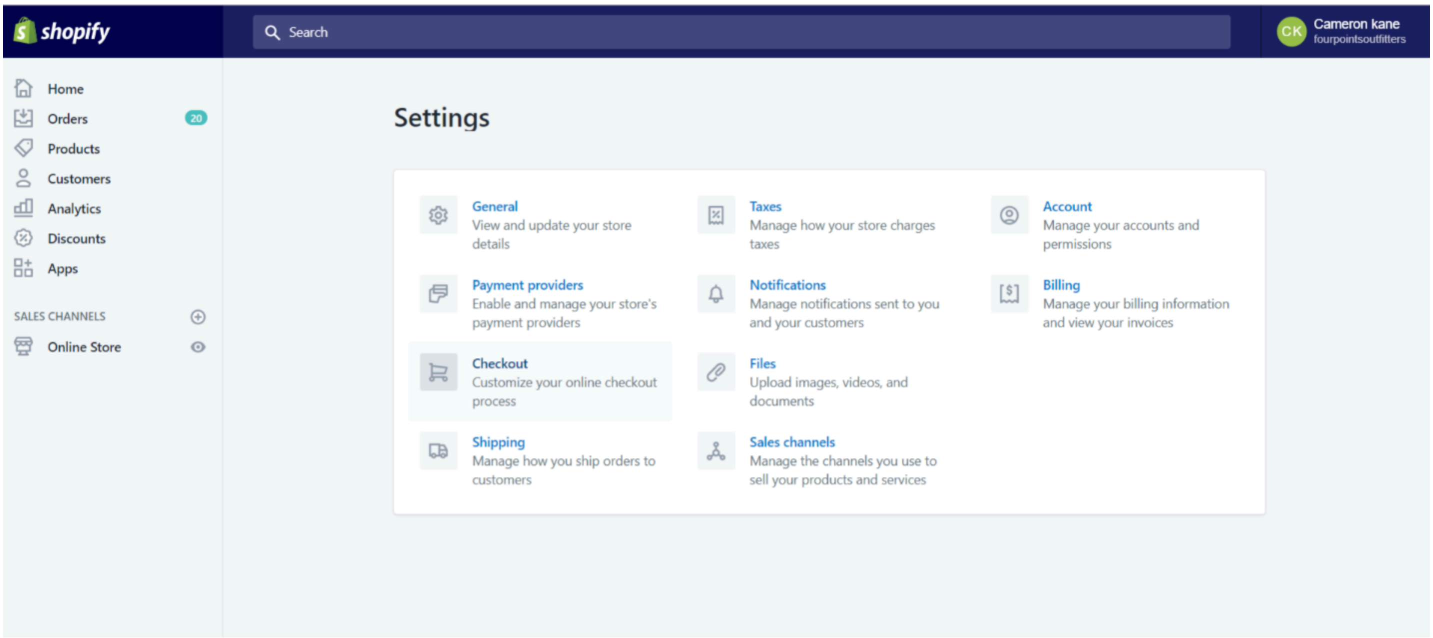
More and more customers prefer to use their phone number instead of email when shopping online. Checkout with phone gives them the option to use their phone number instead of email and receive order updates by text message.

TOTAL SALES Mar 22
$0.00 0 orders

TOTAL SALES BY CHANNEL Mar 22
Online Store
$0.00 0 orders
Other
$0.00 0 orders

TOP PRODUCTS Mar 22
There were no sales during this time.

6. Click on Checkout



Settings

General
View and update your store details

Taxes
Manage how your store charges taxes

Account
Manage your accounts and permissions

Payment providers
Enable and manage your store's payment providers

Notifications
Manage notifications sent to you and your customers

Billing
Manage your billing information and view your invoices

Checkout
Customize your online checkout process

Files
Upload images, videos, and documents

Shipping
Manage how you ship orders to customers

Sales channels
Manage the channels you use to sell your products and services

7. In the Order processing section and in the text box under Additional scripts, insert the "Script Loader" JavaScript code.

8. After inserting the code above, insert the code displayed below in the same box.

```
<script>
var today = new Date();
var orderObj = {
               ConversionPage: 'thankyou',
               ConversionType: 'order',
               OrderID: '{{ order.order_number }}',
               OrderGrandTotal: '{{ order.total_price | money_without_currency  }}',
               OrderDate: today,
               OrderSource: '',
               OrderStoreId: '{{ shop.name }}',
               OrderSubTotal: '{{ order.subtotal_price | money_without_currency  }}',
               OrderTax: '{{ order.tax_price | money_without_currency  }}',
               orderShipping: '{{ order.shipping_price | money_without_currency  }}',
               CouponUsed: '{% if order.discounts %}true{% endif %}',
               OnSale: '',
               OrderDomain: '',
               Items: [
                              {% for item in order.line_items %}
                              {
                                             Title: '{{ item.title }}',
                                             Sku: '{{ item.sku }}',
                                             Price: '{{ item.price |
money_without_currency  }}',
                                             Quantity: '{{ item.quantity }}'
                              },
                              {% endfor %}
               ]
}
console.log(orderObj);
window.onload = function(e){
   window.iPostAnalytic.send(orderObj);
 };
</script>
```

Click to copy

9. Save the edits on this page

You are now set and can start sending emails. Be sure to remember to enable *Web Tracking* in the send configuration.

Please contact iPost Support If you want additional data sent to iPost.

## List of Errors

If you had a successful conversion, but you still don't see it appear in the UI in the next few minutes, here is a spreadsheet with the list of possible errors:

https://docs.google.com/spreadsheets/d/13VEwMuytszw-qCA5viHDLdBwrEv_-GqdLx_XuMQHx8s/edit?usp=sharing