

# Technical Integration and Testing

First, Last Name | Title

[email@listrak.com](mailto:email@listrak.com)

DIRECT: 717.625.5734

529 E MAIN STREET, LITITZ PA 17543

Technical Document

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Document

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Document

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Document

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

Technical Doc

# Everything that’s highlighted in yellow is for your PM use. Remember to un-highlight it or delete it when you’re done.

Contents

[Step 1: Apply Listrak’s Javascript Framework 3](#_Toc417898595)

[Step 2: Collect Order Information 4](#_Toc417898596)

[Step 3: Apply Conversion Tracking 6](#_Toc417898597)

[Step 4: Collect Cart Information 7](#_Toc417898598)

[Step 5: Capture Email Addresses 9](#_Toc417898599)

[Step 6: Collect Browsing Activity 10](#_Toc417898600)

[Step 7: Setup Subscription Points 12](#_Toc417898601)

[Step 8: Facebook Integration 16](#_Toc417898602)

[Step 8: Install Preference Center 19](#_Toc417898603)

[Step 9: Install Refer a Friend 20](#_Toc417898604)

[Step 10: Install Sweepstakes 20](#_Toc417898605)

[Step 11: File Import Specifications 21](#_Toc417898606)

[Step 12: Website Recommendations 27](#_Toc417898607)

[Appendix A: Function Reference 34](#_Toc417898608)

[Appendix B: Developer Tools 36](#_Toc417898609)

# Step 1: Apply Listrak’s Javascript Framework

The script block referred to as Listrak’s Javascript Framework is a segment that builds the core foundation used by every Listrak service.

Insert the Javascript Framework

This specific block needs to load at the end of your document **directly above the </body> tag**, and **after all other Listrak segment blocks**.

<!-- Listrak Analytics – Javascript Framework -->

<script type="text/javascript">

var biJsHost = (("https:" == document.location.protocol) ? "https://" : "http://");

(function (d, s, id, tid, vid) {

var js, ljs = d.getElementsByTagName(s)[0];

if (d.getElementById(id)) return; js = d.createElement(s); js.id = id;

js.src = biJsHost + "cdn.listrakbi.com/scripts/script.js?m=" + tid + "&v=" + vid;

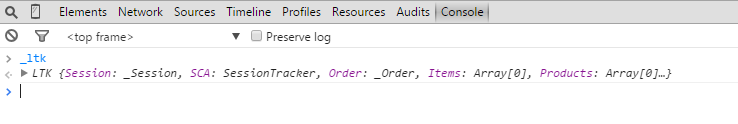
ljs.parentNode.insertBefore(js, ljs);

})(document, 'script', 'ltkSDK', '<<MERCHANT\_ID>>', '<<VERSION>>');

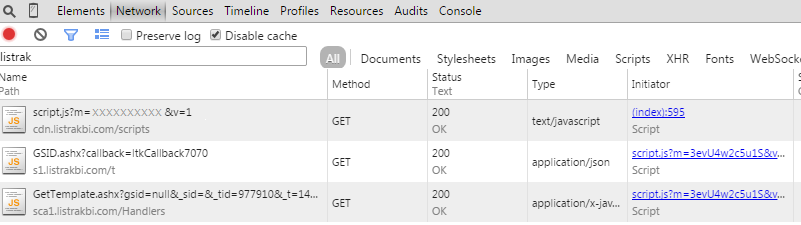
</script>

</body> *<- the closing body tag already on your website*

Test the Javascript Framework

Now that you have the Listrak Javascript Framework on your page, open your browser’s console tab, reload the page, and type “\_ltk” + enter to check for the existence of the \_ltk object.

Troubleshooting

If the \_ltk object does not exist, double-check your script block and verify that the tracking ID is correct. Also open the Network tab and set the filter to ‘listrak’ to check for a successful call and response to the Listrak servers.

Congratulations!

*You are now ready to send email if conversion tracking is not a requirement for you. Please see* [*Step 7*](#_Step_7:_Setup) *for instructions on linking your website’s email newsletter subscription points to Listrak.*

# Step 2: Collect Order Information

When a shopper’s transaction is complete, we need to ensure that we grab the order information and transmit it to Listrak to properly track conversions and other metrics.

Pay attention to the base object: the ORDER handles passing the order information while the Shopping Cart Abandonment (SCA Removal) object handles removing the customer from the automated messaging. Both are critical components to a successful integration.

*Insert the Script*  
Place this block on the **order confirmation page only**. Update all \_ltk.Order and \_ltk.SCA values with code to inject values from the specific order.

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

/\*\* Handle ORDER \*\*/

\_ltk.Order.SetCustomer('customer@domain.com', 'John', 'Smith');

    \_ltk.Order.OrderNumber = '10025736';

    \_ltk.Order.ItemTotal = '25.00';

    \_ltk.Order.ShippingTotal = '6.99';

    \_ltk.Order.TaxTotal = '1.50';

    \_ltk.Order.HandlingTotal = '0.50';

    \_ltk.Order.OrderTotal = '33.99';

    \_ltk.Order.AddItem('SKU-123', 3, '5.00'); // one line per item ordered

    \_ltk.Order.Submit();

/\*\* Handle SCA Removal \*\*/

\_ltk.SCA.SetCustomer('customer@domain.com', 'John', 'Smith');

    \_ltk.SCA.OrderNumber = '10025736';

\_ltk.SCA.Submit();

/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

As a failsafe, we also have an image-based component which ensures that the customer is removed from the Cart Abandonment conversation in the event that the JavaScript function call fails.

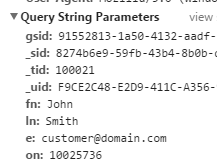
Place this image element directly after the opening <body> tag on the **order confirmation page only**.

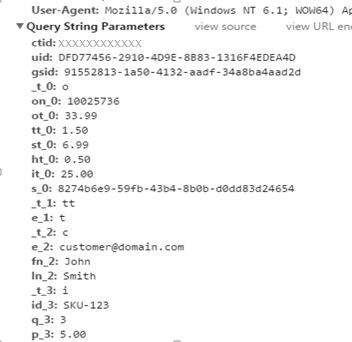
<body> *<- the opening body tag already on your website*

<img src="https://fp.listrakbi.com/fp/<<MERCHANT\_ID>>.jpg" height="1" width="1" />

Test the Script

Open the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. If successful, you will see two calls (T.ashx and Set.ashx)

Click on the Set.ashx entry, click on the Headers tab, and observe the order number. Make sure this value matches the order number from your test.

Click on the T.ashx entry, click on the Headers tab, and observe the order information for integrity.

# Step 3: Apply Conversion Tracking

Now that we are capturing order information, we need a way to make sure that we can track those who make a subsequent purchase as a result of a marketing email. To accomplish this, we have a function that listens for tracking data passed in the URL.

Insert the Script

Place this block in a global context so we can ensure that it is loaded on every page of the site.

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

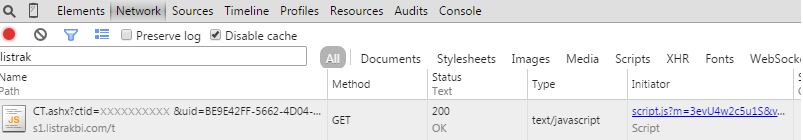
\_ltk.Click.Submit();

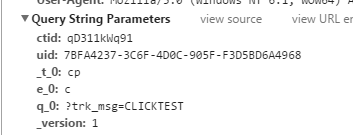
});

</script>

Test the Script

Open the network tab in your browser’s toolbar and, if possible, set the filter to ‘**listrak**’.

In your browser’s address bar, append the following parameter: **?trk\_msg=CLICKTEST**. Hit the Enter key to refresh the page. You should see a call to CT.ashx

Click on this entry, click on the Headers tab, and verify that you are passing the tracking code.

Congratulations!

*You are now ready to send email and track conversions. Your conversion metrics will be available for you within the Listrak application.**Please see* [*Step 7*](#_Step_7:_Setup) *for instruction on linking your website’s email newsletter subscription points to Listrak.*

# Step 4: Collect Cart Information

To make sure we have the latest cart information, it is very important to receive this call every time the shopper has the ability to modify the cart contents. Every time Listrak receives this call, the cart will be recreated from scratch, so it is also important to ensure that every update contains the full cart feed: **THIS IS NOT AN INCREMENTAL PROCESS**. If the user removes all of their items from the cart, call the ClearCart function with no other calls to Listrak after that. This way, we have a means of determining implicit removal versus just another abandon.

Insert the Script

\_ltk.SCA.AddItemWithLinks('<productID>', <quantity>, '<price>', '<productName>', '<productImageUrl>', '<productUrl>')

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.SCA.AddItemWithLinks('sku1', 1, '19.95', 'prod 1', '/images/product1.jpg', '/product1.html');

\_ltk.SCA.AddItemWithLinks('sku2', 1, '29.95', 'prod 2', '/images/product2.jpg', '/product2.html');

\_ltk.SCA.AddItemWithLinks('sku3', 1, '39.95', 'prod 3', '/images/product3.jpg', '/product3.html');

\_ltk.SCA.Submit();

/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

*Note: If you have an AJAX style add to cart feature or “minicart”, you will need to make sure you are calling the \_ltk.SCA.AddItemWithLinks calls on the OnSuccess or OnComplete function of the AJAX call. One \_ltk.SCA.AddItemWithLinks call should be made for every item in the cart.*

**\_ltk.SCA.ClearCart() - ONLY WHEN ALL ITEMS ARE REMOVED FROM THE CART**

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.SCA.ClearCart();

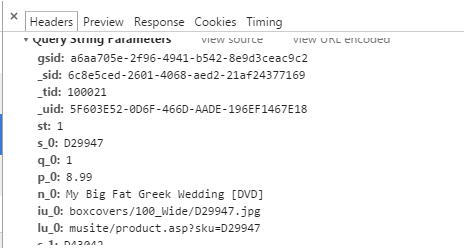
/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

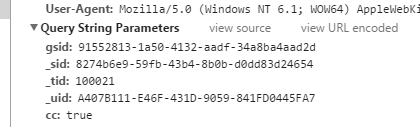
Test Cart Item Entry

Screen ClippingOpen the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. If successful, you will see a call to Set.ashx

Click on the Set.ashx entry and glance at the header values to ensure integrity.

Test Clear Cart

Screen ClippingOpen the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. Remove all items from your cart. If successful, you will see a call to Set.ashx. It is critical for this to be the very last call to Listrak in the stack.

Click on the Set.ashx entry, click on the Headers tab, and look for cc = true.

# Step 5: Capture Email Addresses

One of the most critical components of a successful Shopping Cart Abandonment program is the ability to reach the user who abandons his or her cart. To accomplish this, we have a function which binds to the blur events of input elements (by ID or name attribute). For maximum efficiency, we recommend capturing every possible input element on your site that is requesting a user’s email address.

Insert the Script

Place this block in a global context so we can ensure that it is loaded on every page of the site.

**\_ltk.SCA.CaptureEmail('**<id or name attribute of input element>**');**

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.SCA.CaptureEmail('email');

\_ltk.SCA.CaptureEmail('login\_email');

\_ltk.SCA.CaptureEmail('billing\_email');

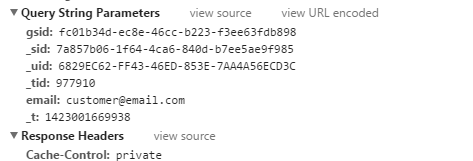
/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

Test the Script

Screen ClippingOpen the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. Enter a valid email address in the input box and hit ‘TAB’ on your keyboard. If successful, you will see a call to Update.ashx.

Click on this entry and view the headers to verify that you are passing the email address.

Troubleshooting

If you do not see Update.ashx fire, first verify that the name or ID of the element is correct. Second, verify that the input element exists when the capture email code fires.

# Step 6: Collect Browsing Activity

To be able to collect and utilize customer browsing activity, one of the two following script blocks should be placed on each page of your website. The type of page will determine which script block should be used.

*Insert the Script*  
On **product detail pages only**, place this script block. The sku of the current product should be passed into this function.

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.Activity.AddProductBrowse('sku1');

    \_ltk.Activity.Submit();

/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

On **every other page**, place this script block:

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.Activity.AddPageBrowse();

    \_ltk.Activity.Submit();

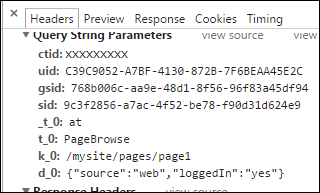
/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

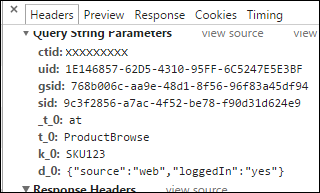
Test Page Browse Activity Entry

Open the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. If successful, you will see a call to Set.ashx  


Click on the Set.ashx entry and glance at the header values to ensure integrity, specifically checking the ‘type’ of activity, t\_0, for ‘PageBrowse’, ‘key’ value, k\_0, which should match the page entered, and the data object, d\_0, for expected values.  
  


Test Product Browse Activity Entry

Open the network tab in your browser’s toolbar and, if possible, set the filter to ‘listrak’. If successful, you will see a call to Set.ashx.  
  


Click on the Set.ashx entry, click on the Headers tab, and look for the ‘type’ value, t\_0, to be ‘ProductBrowse’, the ‘key’ value, k\_0, to be the SKU of the product you browsed, and that the data, d\_0, has any expected values.  
  


# Step 7: Setup Subscription Points

JavaScript Input Values

*<<Internal PM Notes – remember to remove!>>*

*Replace the input tags below with the custom tags you’ve generated in the Subscription Point CD. Note that these tags are not necessary to supply to Magento clients, as we can use the JavaScript injector without them.*

*<</Internal PM Notes>>*

Please insert the following input tags on the appropriate pages as noted below. Each tag should appear only once per page and should be inserted **immediately inside the opening <body> tag**. Once you have completed this step, please advise your project manager so we can begin testing.

*Note: When we set this up for you now, it will be based upon how your site is currently built (or how it’s built in a staging environment, depending upon your situation). If you ever decide to modify the forms or the structure of your site, you may inadvertently break the JavaScript we have in place. So, please notify your project manager or account manager in the future if you change anything that could affect our JavaScript code.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Acquisition Point** | | **URL** | **Input Value** |
| **Account Creation** | http://www.site.com/account/create | <input type="hidden" id=”ListrakDataHook” value="AccountCreate" /> |
| **Newsletter Signup** | | http://www.site.com/newsletter | <input type="hidden" id=”ListrakDataHook” value="Newsletter" /> |

HTML Form Code

*<<Internal PM Notes – remember to remove!>>*

*When generating HTML subscribe code in the Listrak application, remember to apply the correct external event and redirect URL and eliminate all segmentation fields except those that are required for that particular subscription point. Make sure that the source value is hidden and set to “on”. Replace the code below with the custom code you’ve generated for the client.*

*<</Internal PM Notes>>*

Please insert the following code on the appropriate pages as noted below.

**URL:** single-field form in global footer

**HTML Code:**

<form action="http://t.lt02.net/q/wawjy8zdyJcsGdiXxQCxXjb4k6YvKhBSSd" method="post">

<input type="hidden" name="crvs" value="LsUJiZQI8ZHwvzZenSaq\_VLyMDpqNGxuWfrKMeyBWEJOOAg4ef6mXY5t0M8rh9VKu4yVU-ofbzk92466EKmJVQ"/>

<table cellpadding=”0” cellspacing=”0”>

<tr>

<td align="left">Email Address</td>

<td><input type="text" name="email" size="40" maxlength="100" value=""/></td>

</tr>

<tr>

<td align="left" colspan="2><input type="hidden" name="CheckBox.Source.Footer " value="on"/></td>

</tr>

<tr>

<td>&nbsp;</td>

<td align="left"><input type="submit" id="submit" value="Sign Up"/></td>

</tr>

</table>

</form>

API Information

*<<Internal PM Notes – remember to remove!>>*

*Create separate API user. Do not use FAUser\_######. An IP address will need to be defined within Listrak -- this is the IP address of the server making the API calls. Ask for this information and define it under Manage > Accounts > API IP Authorization menu*

*<</Internal PM Notes>>*

Listrak’s APIs use the standards-based protocol SOAP. SOAP is an acronym for Simple Object Access Protocol. Simply stated, SOAP is a communications tool that allows applications to transfer data over a computer network. SOAP provides an effective way to communicate information between applications using different operating systems, different technologies, and different programming languages. This gives organizations an easy way to share XML-based information between their programs and databases with Listrak.

You can find our complete API Integration Guide here: <http://webservices.listrak.com/>

1. Click on the Soap & WSDL link in the navigation
2. Scroll down to the SetContact call – this is the call you will need to use to subscribe through our API
3. SetContact will show you a sample request and response. The necessary API ID information has been provided below, and events and source attributes are provided for the following subscriptions points:
   1. Footer Miniform
   2. Account Creation
   3. Guest Checkout
4. When defining the <ProfileUpdateType> tag, you’ll want to use the Update attribute. Also, there is the tag <OverrideUnsubscribe> - if set to True, it will force subscription to someone who has previously unsubscribed. The value False will honor the unsubscribe and does not complete the subscribe method. It is recommended that you set this to False.

For more information about API, visit <http://webservices.listrak.com>. If you require assistance, please contact your project manager and he or she will set up an appointment with our development team.

**User Name:** APIUSER\_Clientname  
**Password:** ####

### **Email Acquisition Point:** Footer Mini-Form

**URL:** single-field form in global footer

|  |  |  |
| --- | --- | --- |
| **Attribute** | | **API ID** |
| ListID for Master List | #### | |
| AttributeID for Source  (Value must be set to “on”) | | #### |
| ExternalEventID for MiniForm | | #### |

### **Email Acquisition Point:** Account Creation

**URL:** http://www.site.com/account/create

|  |  |  |
| --- | --- | --- |
| **Attribute** | | **API ID** |
| ListID for Master List | #### | |
| AttributeID for First Name (Value must be passed/ imported via API into this text segmentation field) | | #### |
| AttributeID for Last Name (Value must be passed/ imported via API into this text segmentation field) | | #### |
| AttributeID for Source  (Value must be set to “on”) | | #### |
| ExternalEventID for AccountCreate | | #### |

### **Email Acquisition Point:** Guest Checkout

**URL:** https://www.site.com/cart/checkout

|  |  |  |
| --- | --- | --- |
| **Attribute** | | **API ID** |
| ListID for Master List | #### | |
| AttributeID for First Name (Value must be passed/ imported via API into this text segmentation field) | | #### |
| AttributeID for Last Name (Value must be passed/ imported via API into this text segmentation field) | | #### |
| AttributeID for Source  (Value must be set to “on”) | | #### |
| ExternalEventID for GuestCheckout | | #### |

**Dynamic Coupon Codes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Coupon Offer** | | **Segmentation Field Value** | **API ID** |
| **15% Off** | AttributeID for Message Coupon | #### |
| *Dynamic Coupon Codes need to be generated by you and imported via API into a text segmentation field for use within Listrak messages. If you choose to change the value or character length of these coupons, please advise your project manager, as message creative may have to be changed accordingly* | | | |

# Step 8: Facebook Integration

*<<Internal PM Notes – remember to remove!>>*

*Edit the following parts of the code to relate to the client’s merchant ID and subscription settings. You may suggest verbiage for the parts in yellow or leave them to the client for editing. Once you’re done, delete this entire section and paste (in black text) into the client-facing section below.*

**We need to send them the HTML and the JavaScript in separate sections. The new iframe needs to have them separate.**

Green needs to be replaced with the client’s merchant ID

Navy parts can be edited by the client to suit their needs

Aqua part needs to match your subscription setting name

Gray part needs to match your source value and whatever other segmentation you’re using

*<</Internal PM Notes>>*

Below is the subscription code you need to insert into your Facebook email signup tab.

Note: This method utilizes a third party Facebook application, making your use of the app bound by their terms and conditions, privacy policy and availability. If you would like assistance, please contact your project manager.

Step 1: Save this code

Copy the following code to a local text file for later use. You may change the language in the navy highlighted portions to suit your needs, but please do not change the rest.

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js" type="text/javascript"></script>

<script src=”https://cdn.listrakbi.com/scripts/script.js?m=######&v=1” type="text/javascript"></script>

<div id="form-container">

<form id="signup-form" method="get" class="form-container">

<div class="form-header">

<h2>Subscribe to our emails</h2>

<p>Thanks for your interest in us! To receive exciting news about upcoming sales, new products, and other emails tailored to your interests, please enter your email address below.</p>

</div>

<div class="form-title">Email</div>

<input class="form-field" id="email" type="text" name="email" /><br />

<div class="submit-container"><input class="submit-button" id="btnSubmit" type="submit" value="Submit" /></div>

</form>

</div>

<div id="thank-you" style="display:none;">

<h2>Thank You!</h2>

<p>Check your inbox for an email from us!</p>

</div>

<script type="text/javascript">

// <![CDATA[

jQuery(document).ready(function () {

jQuery('#signup-form').submit(function (e) {

e.preventDefault();

\_ltk.Subscriber.List = 'Facebook';

\_ltk.Subscriber.Email = jQuery('#email').val();

\_ltk.Subscriber.Profile.Add('Checkbox.Source.Facebook', "on");

\_ltk.Subscriber.Submit();

setTimeout(function () {

jQuery('#form-container').css('display', 'none');

jQuery('#thank-you').removeAttr('style');

}, 1000);

});

});

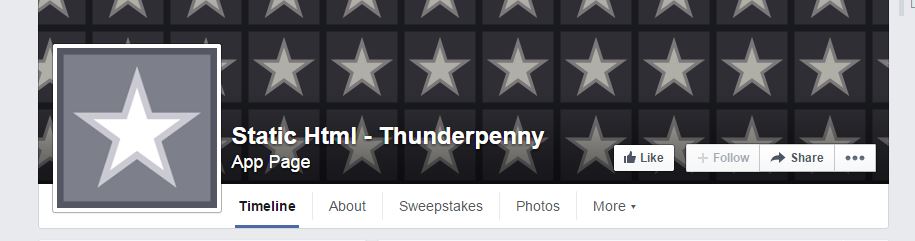
// ]]>

</script>

Step 2: Install an iframe app

Get the Thunderpenny iframe tab application on Facebook.

* Search for "Thunderpenny" in the Facebook search bar.
* Click on the ellipses on the right of the profile picture.



* Select “Add App to Page”.
* Chose your Facebook page, and click “Add Page Tab”.

Step 3: Setup the HTML Form

* Go to your fan page and locate the newly installed app in your tab menu. You may have to click "more" in order to see it. It is most likely named “Welcome”.
* Click on the “Edit tab” button.
* You will be redirected to Thunderpenny where you will be able to edit the iframe.
* Copy the HTML and place it in the index.html tab.
* If you mouse over the “index.html” tab, you will see more tabs appear. Click on the “script.js”.
* Delete everything that’s already there, then paste in the provided JavaScript.
* Click “Save & Publish”.
* The tab will now be working in Facebook.

# Step 8: Install Preference Center

To install the Listrak preference center form onto your site, simply copy the following code, and place it in the code file you wish to have the form.

Insert the Script

You can change the height and width values (in green) as needed. You shouldn't need to change any other values.

<script type="text/javascript">

var biJsHost = (("https:" == document.location.protocol) ? "https://" : "http://"); var cid = '<<Merchant Name>>';   
   document.write(unescape("%3Cscript src='" + biJsHost + "services.listrak.com/" + cid + "/scripts/ltkframe.min.js' type='text/javascript'%3E%3C/script%3E"));

</script>

<script type="text/javascript">

(new ltkframe({ cid: cid, passQueryString: true, width: '100%', minHeight: 700 })).Render();

</script>

Mini-Form

If you would like to add a mini-form somewhere on your site to link to the preference center, you can use the following code.

You will need to change [[YOUR PREFERENCE CENTER URL]] to the URL of the page where you placed the above script.

<form action="[[YOUR PREFERENCE CENTER URL]]" method="get">   
   Email: <input type="text" name="email" /><br />   
   <input type="submit" value="Submit" />   
</form>

Please visit http://services.listrak.com/<<Merchant Name>>/install.html for more information.

# Step 9: Install Refer a Friend

To install the Listrak Refer a Friend form onto your site, simply copy the following code, and place it in the code file you wish to have the form.

Insert the Script

You can change the height and width values (in green) as needed. You shouldn't need to change any other values.

<script type="text/javascript">

var biJsHost = (("https:" == document.location.protocol) ? "https://" : "http://"); var cid = '<<Merchant Name>>';   
   document.write(unescape("%3Cscript src='" + biJsHost + "services.listrak.com/" + cid + "/scripts/ltkframe.min.js' type='text/javascript'%3E%3C/script%3E"));

</script>

<script type="text/javascript">

(new ltkframe({ cid: cid, passQueryString: true, width: '100%', minHeight: 700 })).Render();

</script>

# Step 10: Install Sweepstakes

To install the Listrak Sweepstakes form onto your site, simply copy the following code, and place it in the code file you wish to have the form.

Insert the Script

You can change the height and width values (in green) as needed. You shouldn't need to change any other values.

<script type="text/javascript">

var biJsHost = (("https:" == document.location.protocol) ? "https://" : "http://"); var cid = '<<Sweeps Name>>';   
   document.write(unescape("%3Cscript src='" + biJsHost + "services.listrak.com/" + cid + "/scripts/ltkframe.min.js' type='text/javascript'%3E%3C/script%3E"));

</script>

<script type="text/javascript">

(new ltkframe({ cid: cid, passQueryString: true, width: '100%', minHeight: 700 })).Render();

</script>

# Step 11: File Import Specifications

Providing import files or “flat files” via FTP enables merchants to submit order history, customer information and product catalog data. This information is used throughout the Listrak Fusion Apps systems to support various solutions.

Accepted Files

There are four (4) import files that are currently accepted and imported by Listrak Fusion Apps. The import process has predetermined filenames that it will import automatically, but you may upload any file given you provide the filenames to your account/project manager. The four types of files and default filenames are:

* Orders.txt / Orders.csv / Orders.xlsx
* OrderItems.txt / OrderItems.csv / OrderItems.xlsx
* Customers.txt / Customers.csv / Customers.xlsx
* Products.txt / Products.csv / Products.xlsx

Formatting Requirements

To ensure proper processing of import files, there are several requirements that are imposed during processing. These requirements are:

* Accepted file format (.xlsx, .csv or .txt)
* Accepted encoding (UTF-7, ASCII, UTF-8, or Unicode)
* Header column definition row
* Remove all carriage returns (CR) / line feeds (LF, CRLF) from column data in plain text files
* Use text-qualifiers when necessary

File Extensions

File formats are flexible and allow for mapping of columns to Fusion Apps properties. The accepted file formats are:

* Microsoft Excel 2007 and later (.xlsx)
* Plain delimited text (.txt)
* Comma separated values (.csv)

FTP Setup

The information necessary to upload files is as follows:

* Server: [ftp.listrakbi.com](ftp://ftp.listrakbi.com)
* Port: 21
* Username: <<FTPUsername>>
* Password: <<FTPPassword>>

Upload Schedule

You may upload files as often as you need, though we typically like to see deliveries in 30 or 60 minute intervals. There is no need, however, to upload the same data more than once. For example, you may want to upload an entire product file every day, but you only need to upload customer information when it is changed in your database.

***Note:*** *When uploading product data, insure that any newly added products are live in your product catalog.*

File Specifications

The following information explains all of the various data fields contained within the import files. The first column shows the default header column name when imported. If you are not able to populate all columns, please consult your account manager or project manager.

Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Header Name** | | **Data Type** | **Required** | **Description** |
| **Email** | Alphanumeric | Yes | Email address of customer |
| **OrderNumber** | Alphanumeric | Yes | Unique order number |
| **DateEntered** | Datetime | Yes | Timestamp of order date (EST) |
| **OrderTotal** | Decimal | Yes | Total value of the order (remove currency symbols like $, £) |
| **ItemTotal** | Decimal | No | Total cost of items ordered (subtotal) |
| **TaxTotal** | Decimal | No | Total sales tax charged |
| **ShippingTotal** | Decimal | No | Total shipping costs |
| **HandlingTotal** | Decimal | No | Total handling costs |
| **Status** | Integer | Yes | Status indicator (see Appendix A) |
| **ShipDate** | Datetime | No | Timestamp when entire order shipped (EST) |
| **TrackingNumber** | Alphanumeric | No | Shipment tracking number |
| **ShippingMethod** | Alphanumeric | No | Shipping method (E.g. UPS Ground) |
| **CouponCode** | Alphanumeric | No | Coupon code used with order |
| **DiscountTotal** | Decimal | No | Total value of order discount |
| **Source** | Alphanumeric | No | Describes the source at which the order was placed (online, POS, etc.) |
| **Meta1** | Alphanumeric | No | Additional meta information |
| **Meta2** | Alphanumeric | No | Additional meta information |
| **Meta3** | Alphanumeric | No | Additional meta information |
| **Meta4** | Alphanumeric | No | Additional meta information |
| **Meta5** | Alphanumeric | No | Additional meta information |

Order Items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Header Name** | | **Data Type** | **Required** | **Description** |
| **OrderNumber** | Alphanumeric | Yes | Order number |
| **Sku** | Alphanumeric | Yes | Unique stock number of product |
| **Quantity** | Integer | Yes | Total number of units purchased |
| **Price** | Decimal | Yes | Price of one unit purchased |
| **Status** | Integer | No | Status indicator (see Appendix A) |
| **ShipDate** | Datetime | No | Timestamp when item shipped (EST) |
| **TrackingNumber** | Alphanumeric | No | Shipment tracking number |
| **ShippingMethod** | Alphanumeric | No | Shipping method (E.g. UPS Ground) |
| **DiscountedPrice** | Decimal | No | Total discounted cost of product |
| **ItemTotal** | Decimal | No | Total line item cost (quantity times price) |
| **Meta1** | Alphanumeric | No | Additional meta information |
| **Meta2** | Alphanumeric | No | Additional meta information |
| **Meta3** | Alphanumeric | No | Additional meta information |
| **Meta4** | Alphanumeric | No | Additional meta information |
| **Meta5** | Alphanumeric | No | Additional meta information |

Customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Header Name** | | **Data Type** | **Required** | **Description** |
| **Email** | Alphanumeric | Yes | Email address |
| **FirstName** | Alphanumeric | No | First name |
| **LastName** | Alphanumeric | No | Last name |
| **Gender** | Alphanumeric | No | Gender of customer (M/F, Male/Female) |
| **Birthday** | Date | No | Customer birthdate |
| **ZipCode** | Alphanumeric | No | Customer zip code |
| **CustomerNumber** | Alphanumeric | No | Internal customer number |
| **Registered** | Boolean | No | Indicates the customer registration state |
| **Meta1** | Alphanumeric | No | Additional meta information |
| **Meta2** | Alphanumeric | No | Additional meta information |
| **Meta3** | Alphanumeric | No | Additional meta information |
| **Meta4** | Alphanumeric | No | Additional meta information |
| **Meta5** | Alphanumeric | No | Additional meta information |

Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Header Name** | | **Data Type** | **Required** | **Description** |
| **Sku** | Alphanumeric | Yes | Unique stock number of product |
| **Title** | Alphanumeric | Yes | Product name |
| **ImageUrl** | Alphanumeric | Yes | URL for product image |
| **LinkUrl** | Alphanumeric | Yes | URL for product webpage |
| **Description** | Alphanumeric | Yes | Description of product |
| **Price** | Decimal | Yes | List price of product |
| **Brand** | Alphanumeric | Yes | Brand name of product |
| **Category** | Alphanumeric | Yes | Category or department |
| **SubCategory** | Alphanumeric | Yes | Sub-category or sub-department |
| **SalePrice** | Decimal | Yes | Sale price of product |
| **SaleStartDate** | Date | No | Start date time of sale |
| **SaleEndDate** | Date | No | End date time of sale |
| **OnSale** | Boolean | Yes | Explicit indicator that the item is on sale |
| **IsClearance** | Boolean | No | Explicit indicator that the item is a clearance item |
| **IsOutlet** | Boolean | No | Explicit indicator that the item is an outlet item |
| **QOH** | Integer | Yes | Quantity On Hand |
| **InStock** | Boolean | Yes | Explicit indicator that the item is in stock |
| **MasterSku** | Alphanumeric | Yes | Unique stock number of the master product |
| **ReviewProductID** | Alphanumeric | No | Unique identifier used by 3rd-party reviews provider |
| **ReviewUrl** | Alphanumeric | No | URL for review webpage |
| **Discontinued** | Boolean | Yes | Explicit indicator that the item has been discontinued |
| **Size** | Alphanumeric | No | Size of product (e.g. Small, M, 6 7/8”) |
| **Color** | Alphanumeric | No | Color of product (e.g. Green, Paisley, etc.) |
| **Style** | Alphanumeric | No | Style of product |
| **Gender** | Alphanumeric | No | Gender of product (if applicable) |
| **MSRP** | Decimal | No | Retail price of the product |
| **Meta1** | Alphanumeric | No | Additional meta information |
| **Meta2** | Alphanumeric | No | Additional meta information |
| **Meta3** | Alphanumeric | No | Additional meta information |
| **Meta4** | Alphanumeric | No | Additional meta information |
| **Meta5** | Alphanumeric | No | Additional meta information |

Note: Related Products are also importable via the products flat files, using the following Header Syntax:   
Related\_SKU\_N, Related\_TYPE\_N, Related\_RANK\_N, where N is an incremented number value.  
RANK can be inferred based on the incremented number value for SKU and TYPE

Appendix A

Status codes allow merchants to define what state an order is currently in. This is handy for denoting an order as shipped or canceled. The values for the status field must be numeric and are defined in the table below. The most common values will be: processing, shipped, returned and canceled.

|  |  |  |
| --- | --- | --- |
| **Status** | | **Value** |
| **Not Set** | 0 |
| **Misc** | 1 |
| **Pre-Order** | 2 |
| **Backorder** | 3 |
| **Pending** | 4 |
| **Hold** | 5 |
| **Processing** | 6 |
| **Shipped** | 7 |
| **Completed** | 8 |
| **Returned** | 9 |
| **Canceled** | 10 |
| **Unknown** | 11 |

Appendix B

Import fields marked as “Boolean” accept any of the following case insensitive import string values:

|  |  |  |
| --- | --- | --- |
| **Import Value** | | **Boolean Value** |
| **Yes** | True |
| **Y** | True |
| **True** | True |
| **1** | True |
| **No** | False |
| **N** | False |
| **False** | False |
| **0** | False |

Frequently Asked Questions

**What if I don’t know which encoding to use?**  
*Unless you are using a language with non-traditional characters like Japanese or Arabic, UTF-8 is typically sufficient.*

**How do I specify a delimiter other than comma, vertical bar, or tab?**  
*If you use a different delimiter, please notify your project manager to ensure your import settings are defined appropriately.*

**When do I need to use a text qualifier?**  
*Sometimes data within a column can contain the delimiter, such as a comma in a product description. Typically this is a double quote (“) in comma-separated value (CSV) files.*

**What are header column definitions?**  
*To make sure we import each column to the right location in our database, you must specify the headers as defined in the file definitions below. If you cannot name the headers, you can work with your account/project manager to make sure columns are mapped correctly.*

**When are the specifications for sending timestamps?**  
*To ensure the proper utilization of timestamps, they will need to be formatted in Eastern Standard Time(EST).This will ensure dates are documented correctly in Listrak and will be utilized properly when implementing automated campaigns.*

# Step 12: Website Recommendations

*<<Internal PM Notes – remember to remove!>>*

*If Listrak is completing the website implementation, please remove the following section.*

*<</Internal PM Notes>>*

Listrak’s Recommendation Engine can be easily used to display recommendations on your website using JavaScript and HTML. To do this, you will need to define an item template and consume that template where you want the recommendations to be displayed.

Define an Item Template

Templates allow for calling any number of Merchandise Blocks using a preconfigured block layout, defined using styles and html.

<script id="recommendation-template" type="text/html">

<li>

<img src="@Recommendation.ImageUrl" />

<strong title="@Recommendation.Title">@Recommendation.Title</strong>

<strong title="@Recommendation.Sku">@Recommendation.Sku</strong>

<em>@Recommendation.Price</em>

</li>

</script>

The @Recommendation object is used to render dynamic content from the items returned from the Listrak Recommendation Engine. By default, the following fields are available: Sku, Title, Price, ImageUrl and LinkUrl. Additional fields can be used by calling \_ltk.Recommender.AddField(“<<FIELD NAME>>”); before defining the item template.

<script type="text/javascript">

(function(d) { if (document.addEventListener) document.addEventListener('ltkAsyncListener', d);

else {e = document.documentElement; e.ltkAsyncProperty = 0; e.attachEvent('onpropertychange', function (e) {

if (e.propertyName == 'ltkAsyncProperty'){d();}});}})(function() {

/\*\*\*\*\*\*\*\*\*\* Begin Custom Code \*\*\*\*\*\*\*\*\*\*/

\_ltk.Recommender.AddField("Description");

/\*\*\*\*\*\*\*\*\*\* End Custom Code \*\*\*\*\*\*\*\*\*\*/

});

</script>

**Available Fields:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Meta1 | Meta2 | Meta3 | Meta4 |
| Meta5 | SalePrice | Brand | Category | Subcategory |
| Onsale | SaleStartDate | SaleEndDate | QuantityOnHand | InStock |
| MasterSku | Discontinued | IsViewable | IsClearance | IsOutlet |
| ReviewUrl | RecipeUID | ProductUID | RatingCount | StarRating |
| Rating | SalesLast7 | SalesLast30 | SalesLast90 | SalesLast365 |

Consume an Item Template

Once you are satisfied with the template, having it appear on a page is as simple as including that template script as html. Within this script, call @include() and pass the id of the item template script block that was created above.

<div data-ltk-merchandiseblock="blockUID">

<h3>Recently Viewed</h3>

<ul>

<script type="text/html">@include(recommendation-template)</script>

</ul>

</div>

In addition, you can include only a certain number of results when calling a block, and you can jump ahead to the later results of a block.​

These attributes are:   
**data-ltk-skip="#"** - skip the first N results returned by a block  
**data-ltk-take="#"** - take only the next N results returned by a block

These can be used individually or in conjunction, as shown in this example:

/\*\* show only the first four returned by the block \*\*/

<div data-ltk-merchandiseblock="blockUID" data-ltk-take="4">

    <h3>Recently Viewed</h3>

    <ul>

        <script type="text/html">@include(recommendation-template)</script>

    </ul>

</div>

/\*\* show items 5-8 returned by the block \*\*/

<div data-ltk-merchandiseblock="blockUID" data-ltk-skip="4" data-ltk-take="4">

    <h3>Just For You</h3>

    <ul>

        <script type="text/html">@include(recommendation-template)</script>

    </ul>

</div>

/\*\* show remaining items in the block beyond 8 \*\*/

<div data-ltk-merchandiseblock="blockUID" data-ltk-skip="8">

    <h3>Top Sellers</h3>

    <ul>

        <script type="text/html">@include(recommendation-template)</script>

    </ul>

</div>​

Additional Methods

Along with the code discussed below, there are additional JavaScript methods that can be utilized.

|  |  |  |
| --- | --- | --- |
| **Method** | | **Description** |
| **AddField()** | As exampled above, this allows the @Recommendation object to pull the field for display/usage.  This method can accept either a single value or a comma delimited list of values.  Example:  \_ltk.Recommender.AddField("Meta1"); \_ltk.Recommender.AddField("Description", "Category"); |
| **AddSku()** | Forces the recent activity of all viewers of a block to incorporate a specific SKU first, regardless of Activity type.  Example:  \_ltk.Recommender.AddSku(“SKU12345"); |
| **SetData()** | Forces the data for a given field to be set to the value provided, for all viewers.  Example:  \_ltk.Recommender.SetData("Category", "ASDF"); |
| **AddFilter()** | Allows the ability to hook in to and manipulate recommendations after they're generated but before they are rendered.  Example (stripping HTML from title in products being displayed):  \_ltk.Recommender.AddFilter(function (recommendations, callback) {      $(recommendations).each(function () {          if (this.Title) this.Title = this.Title.replace(/\<.\*?\>/g, "");      });      callback(recommendations);  });​ |
| **Render(function ())** | Registers an event handler for when recommendations are rendered, allowing control of additional actions by including jQuery.  Example (Fade in the displayed recommendations):  \_ltk.Recommendations.Render(function () {     $(this).fadeIn(3000); }); |
| **Render()** | This method needs to be called after all other recommendations scripts. This will initiate the loading and displaying of the recommendations.  Example:  \_ltk.Recommendations.Render(); |

Full Example

Below is a full example of Listrak website recommendations.

<html>

<head>

</head>

<body>

<script src="//code.jquery.com/jquery-1.11.0.min.js"></script>

<script src="//s1.listrakbi.com/scripts/script.js?m=asdfasdfasdf&v=1"></script>

<style type="text/css">

.recommendations {

display: none;

font-family: sans-serif;

}

.recommendations ul {

list-style-type: none;

padding: 0;

}

.recommendations li {

display: inline-block;

text-align: center;

width: 200px;

}

.recommendations li \* {

display: block;

margin: 15px;

}

.recommendations li strong {

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.recommendations li img {

margin: auto;

max-width: 100px;

}

</style>

<div class="recommendations" data-ltk-merchandiseblock="asdfasdf-asdf-asdf-asdf-asdfasdfasdf" data-ltk-take="4">

<h3>Recently Viewed</h3>

<ul>

<script type="text/html">@include(recommendation-template)</script>

</ul>

</div>

<div class="recommendations" data-ltk-merchandiseblock="asdfasdf-asdf-asdf-asdf-asdfasdfasdf" data-ltk-skip="4" data-ltk-take="4">

<h3>Just For You</h3>

<ul>

<script type="text/html">@include(recommendation-template)</script>

</ul>

</div>

<div class="recommendations" data-ltk-merchandiseblock="asdfasdf-asdf-asdf-asdf-asdfasdfasdf" data-ltk-skip="8">

<h3>Top Sellers</h3>

<ul>

<script type="text/html">@include(recommendation-template)</script>

</ul>

</div>

<script id="recommendation-template" type="text/html">

<li>

<img title="@(Recommendation.Description || 'No description available.')" src="@Recommendation.ImageUrl" />

<strong title="@Recommendation.Title">@Recommendation.Title</strong>

<em>@Recommendation.Price</em>

</li>

</script>

<script>

\_ltk.Recommender.AddField("Description");

\_ltk.Recommender.AddSku("ASDF123");

\_ltk.Recommender.SetData("category", "ASDF");

\_ltk.Recommender.AddFilter(function (recommendations, callback) {

$(recommendations).each(function () {

if (this.Title) this.Title = this.Title.replace(/\<.\*?\>/g, "");

});

callback(recommendations);

});

\_ltk.Recommender.Render(function () {

$(this).fadeIn(3000);

});

\_ltk.Recommender.Render();

</script>

</body>

</html>

*<<Internal PM Notes – remember to remove!>>*

*If Client is completing the website implementation, please remove the following section. For each website merchandise block, please append additional sections and update any highlighted text with the appropriate value.*

*Please restrict the merchandise block count to one for a single page. You may use multiple divs per page with the same merchandise block id. DO NOT REMOVE the style="display: none;" attribute, as that is needed to prevent a blank block from displaying should the recs call fail.*

*Below are the available parameters that should be configured per page*

|  |  |  |
| --- | --- | --- |
| **Parameter** | | **Description** |
| **<<PAGE>>** | The URL or type of page this block should go on. ie. Product Page or http://www.domain.com/some-url |
| **<<LOCATION>>** | The area of the site to place the div. ie Under the reviews section, beside the product options |
| **<<PARM>>** | Type of additional data being used for the recipes. You can have multiple <<PARM>> values per block and should be prepended with data-ltk- Ie. category, brand, search term |
| **<<PARM VALUE>>** | The value of the additional data being sent in <<PARM>>. Ie. Shirts, Nike, Mens |
| **<<OBJECT>>** | The type of page this block is on. This could match what is in <<PAGE>>. Ie. Product, Category Listing |
| **<<MERCH BLOCK>>** | This would be the UID of merchandise block. Ie. d10abebc-e678-4119-bb61-470ec55e05c9 |
| **<<SKU>>** | This would be the sku of the current product page this block is on. This can also be a comma delimited list of skus for search results, cart and product listing pages. The entire tag can be removed is no sku needs to be passed. |
| **<<TITLE>>** | This can be adjusted based on the design that creative or the client comes up with. |
| **<<ITEM DESIGN>>** | This is the title of the individual item design for that block. Multiple blocks can use the same design template. This value needs to be supplied to the developer. |

*<</Internal PM Notes>>*

Insert the Code

To implement website recommendations you will need to paste the following blocks of code on the corresponding pages, in the section of the page where the recommendations should be displayed. When this is completed notify your project manager to continue the integration.

PAGE

Place this block of HTML on your <<PAGE>> page in <<LOCATION>>. Please update <<PARM>> to the <<PARM VALUE>> of the current <<OBJECT>>.

<div class="recommendations" style="display: none;" data-ltk-merchandiseblock="<<MERCH BLOCK>>" data-ltk-<<PARM>>="<<PARM VALUE>>" data-ltk-sku="<<SKU>>">

<h3><<TITLE>></h3>

<ul>

<script type="text/html">@include(<<ITEM DESIGN>>-template)</script>

</ul>

</div>

# Appendix A: Function Reference

\_ltk.Order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property/Method** | | **Data Type** | **Description** | **Example** |
| **SetCustomer** | Method | Used to set customer specific data | SetCustomer(‘<emailAddress>’, ‘<firstName>’, ‘<lastName>’) |
| **- EmailAddress** | Alphanumeric | Email address of customer | ‘customer@domain.com’ |
| **- FirstName** | Alphanumeric | First name of customer | ‘John’ |
| **- LastName** | Alphanumeric | Last name of customer | ‘Smith’ |
| **OrderNumber** | Alphanumeric | Unique order number | ‘10-A9487-0024’ |
| **ItemTotal** | Decimal | Total cost of items ordered (subtotal) | ‘25.00’ |
| **ShippingTotal** | Decimal | Total shipping costs | ‘6.99’ |
| **TaxTotal** | Decimal | Total tax charges | ‘1.50’ |
| **HandlingTotal** | Decimal | Total handling costs | ‘0.50’ |
| **OrderTotal** | Decimal | Total cost of order | ‘33.99’ |
| **AddItem** | Method | Adds data for a single product ordered | AddItem(‘<productID>’, <quantity>, ‘<price>’) |
| **- ProductID** | Alphanumeric | Unique product ID or SKU | ‘SKU-123’ |
| **- Quantity** | Numeric | Quantity of product | 3 |
| **- Price** | Decimal | List price of a single product | ‘5.00’ |
| **Submit** | Method | Sends order data to Listrak | N/A |

\_ltk.SCA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property/Method** | | **Data Type** | **Description** | **Example** |
| **Stage** | Numeric | A number used to indicate checkout stage. The shopping cart is typically indicated as 1 and increments as the customer moves through the checkout process. | 1 |
| **AddItemWithLinks** | Method | Adds details about products currently in the customer’s shopping cart | AddItemWithLinks(‘<productID>’, <quantity>, ‘<price>’, ‘<productName>’, ‘<productImageUrl>’, ‘<productUrl>’) |
| **- ProductID** | Alphanumeric | Unique product ID or SKU | ‘BT-15267’ |
| **- Quantity** | Numeric | Quantity of product | 1 |
| **- Price** | Decimal | List price of a single product | ‘15.99’ |
| **- ProductName** | Alphanumeric | Product name or description | ‘8pk AAA alkaline batteries’ |
| **- ProductImageUrl** | Alphanumeric | Relative URL of product image | ‘/images/sd984kg.jpg’ |
| **- ProductUrl** | Alphanumeric | Relative URL of product page | ‘/prods/8pk-aaa-batteries/’ |
| **SetCustomer** | Method | Adds details about a customer | SetCustomer(‘<email>’, ‘<firstName>’, ‘<lastName>’) |
| **- Email** | Alphanumeric | Email address of customer | ‘customer@domain.com’ |
| **- FirstName** | Alphanumeric | First name of customer | ‘John’ |
| **- LastName** | Alphanumeric | Last name of customer | ‘Smith’ |
| **Submit** | Method | Sends order data to Listrak | N/A |

# Appendix B: Developer Tools

Chrome

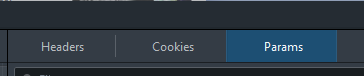
## *Hit ‘F12’, Ctrl + Shift + I, or select the menu options > Tools > Developer Tools.*

* Hit the ‘Network button’ 
* Hit the Filter icon 
* Type in ‘Listrak’ to help find the various network calls 
* Notice the red icon. Ensure this is red to capture network traffic.

FireFox

## *Hit ‘F12’, Ctrl + Shift + Q, or select the menu options > Developer > Network.*

* Hit the ‘Network button’ 
* Hit ‘HTML’ and ‘JS’ to help filter the results 
* Selecting a response will open the Headers window to the right. Click ‘Params’ for easier viewing.



IE

## *Hit ‘F12’, or select the gear icon > Developer > F12 Developer tools.*

* Select the Network icon 
* Make sure to hit the play button 
* Locate the request and double click to view detailed information.

