**Flipp NativeX Integration Guide**

Version 2.3

Updated Feb 1st, 2024

**Table of Content**

[**1. Experience Overview 3**](#_d1lq887vtqaz)

[**2. Site Structure & Zones 4**](#_xqxzf3gmmq4q)

[**3. Placement 5**](#_3v10o5dw1szn)

[Standard Experience 5](#_z369gmvyw9j9)

[Compact Experience 6](#_uln4vl3xntee)

[**4. Implementation 7**](#_ta3htcqimout)

[Standard Experience 7](#_6nvffgjkujf0)

[Compact Experience 8](#_ymr52sail6ao)

[**5. Implementation Options 9**](#_yaa2eafqwo4b)

[Custom Header 9](#_x17ya3cvag4c)

[Viewable Impression Call-Back 10](#_p1pfxw614fa5)

[Slot Fill Result Call-Back 10](#_qly1ffewuh3l)

[Content Recommendation Feed Setup 11](#_rh5o71xvq1z6)

[Content Recommendation Setup 11](#_3ccoa05klz9l)

[Others Vendor Partners 11](#_r8vynjbmran8)

[Partner Provided User Key 12](#_1vmlr5aqbse7)

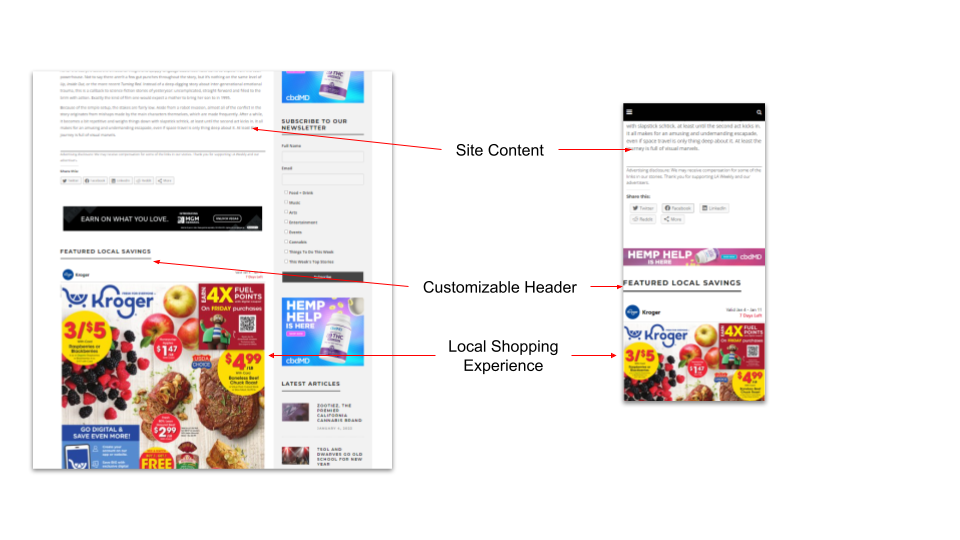
[**6. Testing 12**](#_hxmipg3q9vq0)

## 

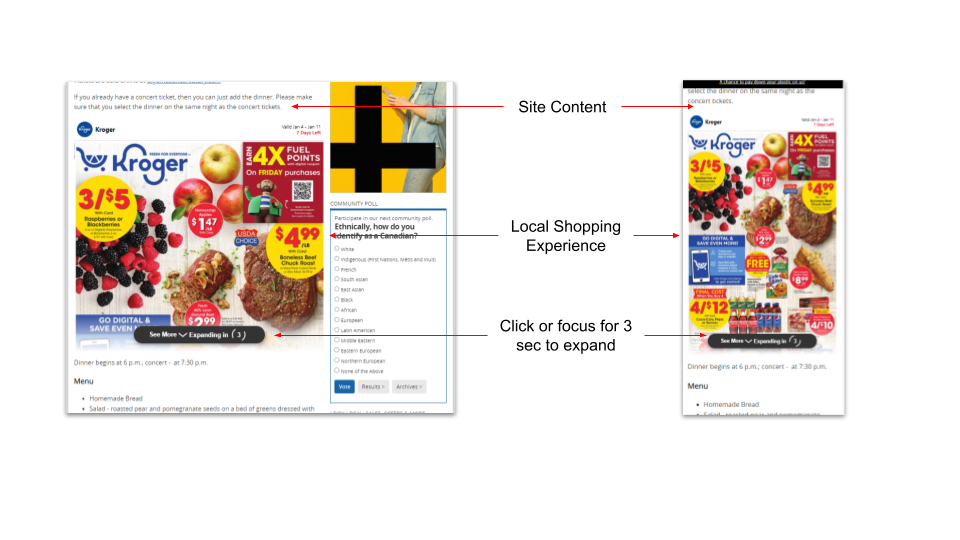
# 1. Experience Overview

This document will provide an implementation overview of the Flipp NativeX experience. Once implemented, it will allow a media partner to showcase and monetize relevant, local shopping content (i.e. circulars, weekly ads, coupons, etc.) automatically without additional operational effort, with daily performance data visible on Flipp’s reporting dashboard.

Flipp’s NativeX showcases locally relevant shopping experiences on both desktop, tablet and mobile platforms with a standard version meant to be placed after the page’s main content, and a compact version that can be integrated into the midst of the main content area. If there is no relevant content available, the NativeX experience can be programmed to hide itself 1, or show alternative content provided by the media partner (including another ad).



*Example of the NativeX Standard experience on desktop and mobile showing the customized header*



*Example of the NativeX Compact experience on desktop and mobile. This experience will expand to the full height after a 3 second animation if it is fully in view on the screen*

**Live examples of the experience can be seen here:**   
Standard experience: [Example 1](https://www.usatoday.com/story/graphics/2025/01/10/how-to-feed-birds-in-winter/77485983007/?flipp-content-code=publisher-test-staging)

Compact experience: [Example 2](https://www.greenfieldreporter.com/2025/01/10/tiktok-could-be-banned-this-month-heres-what-users-can-do-to-prepare/?flipp-content-code=publisher-test-staging)

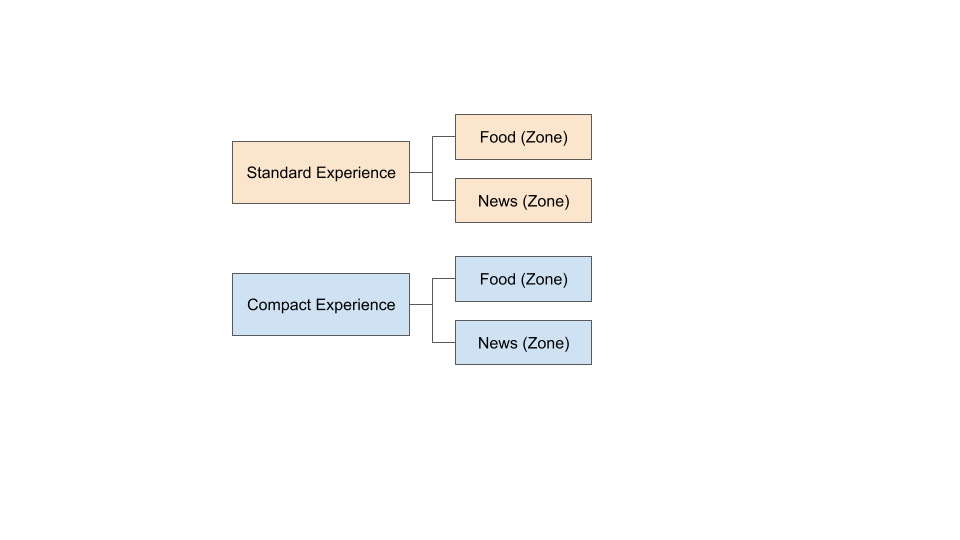
# 2. Site Structure, Zones, & Subdomains

For each media partner, Flipp will create a set of “zones” to facilitate billing and reporting. By default, there will be one zone for the standard experience, and another for the compact experience (if it is in use). Each zone will have their own integration code to be placed on sites using that experience.

We also have the option to create additional zones to better match content to the relevant context. For example, a media partner may want zones to target content and measure performance based on category ex. News, Science Blogs, Food. Aside from context, a media partner can also assign different zones to different sites to differentiate reporting for subsidiaries. If this is applicable to your situation, please let Flipp know so it can be set up in advance.

For media partners looking to gain deeper insights and improve segmentation in reporting, we offer the option to implement the mediaPartnerDescriptiveValueTag. This tag provides an additional layer of granularity, allowing partners to segment reporting data within the same zone\_id, ideal for those managing multiple instances or subdomains under a single zone.

Once the mediaPartnerDescriptiveValueTag is successfully implemented, an additional tab labeled Subdomain Reporting will appear within your reporting dashboard. This tab will show a detailed breakdown of data for each subdomain, allowing you to view specific metrics for each one while still keeping an overall view of your account's performance.



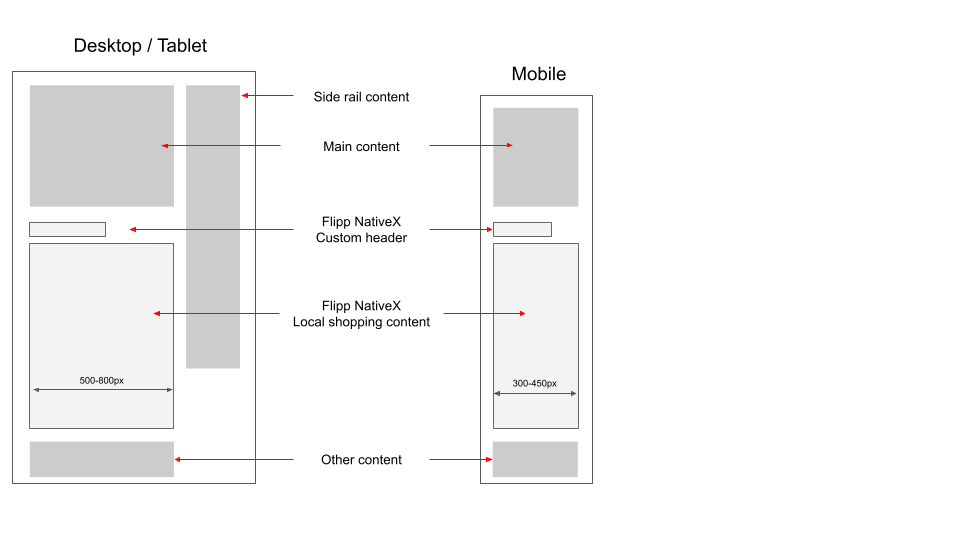
# 3. Placement

The media partner can use its own discretion when deciding the placement of Flipp’s NativeX experience in order to optimize both revenue and user experience in the context of the overall site. Nevertheless, the following guidelines should be observed.

## Standard Experience

This should be placed In the main content area/column of the page – never on the side rail. The experience should be placed after the main content or article of the page.

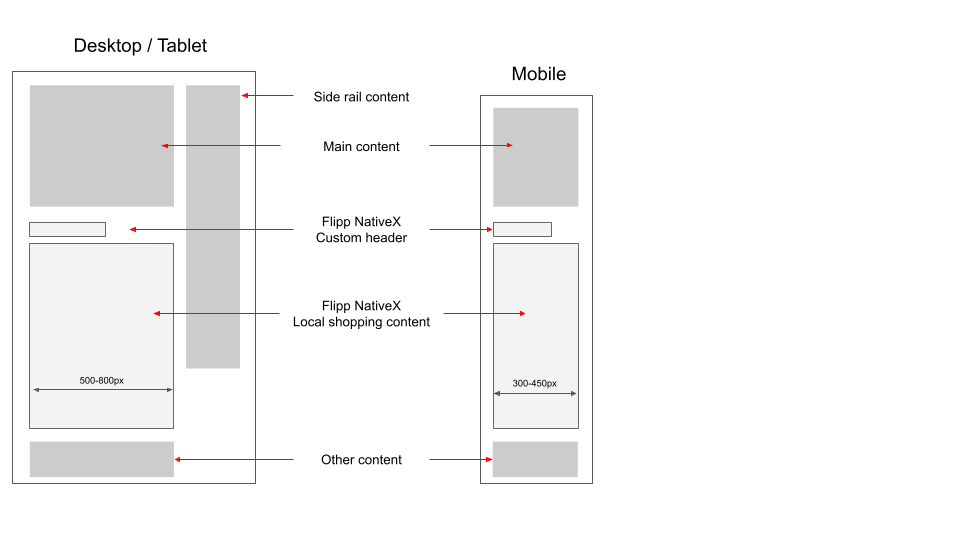
* **Width**: Same as the main content area of the page, with the same margin and white space as the main content of the page. 500-800px on desktop, 350-450px on mobile.
* **Height**: Automatically configured according to the aspect ratio of the content, generally 1800-2400px
* **Custom Header**: Recommended, use the same styling to denote a new section of the page



## Compact Experience

This should be in the main content area/column of the page – never on the side rail. The experience can be placed within the main article content, potentially replacing an existing ad unit, or just after the main article content. The higher on the page, the higher the volume and revenue potential, but it should not be so high that it is visible on page load without first scrolling.

* **Width**: Same as the main content area of the page, with the same margin and white space as the main content of the page. 500-800 px on desktop, 350-450 px on mobile
* **Height**: Automatically configured. The experience would default to approximately 600 px in height and expand to 1800-2400 px after a 3 second animation upon user focus
* **Custom Header**: Not recommended for mid content placement. Recommended if placed after the main content area



# 4. Implementation

The Flipp NativeX experience can be implemented to be in standard or compact mode. In standard mode, approximately 2 pages of the weekly ad is shown on load. In compact mode, only the 1st 600px is shown to start, if the user maintains focus on the shopping experience, an animation will play and the experience will expand to full size about 3 seconds later.

Both are implemented using a direct tag on page setup that consists of two parts: The controller and anchor. These tags are specific to each zone of the media partner and should be applied across all sites within that zone that are intended for the content module to appear.

The controller is the main script responsible for the logic and instrumentation and should be placed as high on the page as practical (e.g. head). The anchor controls the position of the actual experience and should be placed where you want the experience to appear.

## Standard Experience

The following is an example of a standard experience controller and anchor tag

**The controller tag should be placed at the head of the document**

<!-- Flipp Controller Tag-->

<!-- The ‘flipptag’ script is responsible for handling the logic behind ad requests, rendering iframes, positioning, firing beacons, etc. -->

<script async src="https://cdn-gateflipp.flippback.com/tag/js/flipptag.js?site\_id=1111111"></script>

<!-- window.flippxp is a global JavaScript object and supports actions like registering the ad slot, facilitating SDK communication, and managing bidding queues -->

<script>

window.flippxp = window.flippxp || {run: []};

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-123te4st, “wishabi-test-publisher”, 1111111, [ 2345678 ],

{

mediaPartnerDescriptiveValueTag: “SubDomain”,

}

{});

});

</script>

**The anchor tag should be placed where you want the shopping experience to appear**

<!-- Flipp Mobile Anchor Tag-->

<div id="flipp-ux-slot-123te4st">

<div class="flipp-ux-header" style="display: none">

<h3 class="cb-sidebar-widget-title cb-widget-title">Featured Local Savings</h3>

</div>

</div>

<!-- Flipp Desktop Anchor Tag-->

<div id="flipp-ux-slot-345st6te">

<div class="flipp-ux-header" style="display: none">

<h3 class="cb-sidebar-widget-title cb-widget-title">Featured Local Savings</h3>

</div>

</div>

## Compact Experience

The following is an example of a compact experience controller and anchor tag

**The controller tag should be placed at the head of the document**

<!-- Flipp Controller Tag→

<!-- The ‘flipptag’ script is responsible for handling the logic behind ad requests, rendering iframes, positioning, firing beacons, etc. -->

<script async

src="https://cdn-gateflipp.flippback.com/tag/js/flipptag.js?site\_id=1111111></script>

<script>

<!-- window.flippxp is a global JavaScript object and supports actions like registering the ad slot, facilitating SDK communication, and managing bidding queues -->

window.flippxp = window.flippxp || {run: []};

if (document.documentElement.clientWidth < 500) {

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-123te4st, “wishabi-test-publisher”, 1111111, [ 2345678 ],

{

startCompact:true,

dwellExpandable: true,

mediaPartnerDescriptiveValueTag: “SubDomain”

}

);});

} else {

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-345st6te”, "wishabi-test-publisher", 1111111, [ 2345678 ],

{

startCompact:true,

dwellExpandable: true,

mediaPartnerDescriptiveValueTag,

}

);});

}

</script>

**The anchor tag should be placed where you want the shopping experience to appear**

<!-- Flipp Mobile Anchor Tag-->

<div id="flipp-ux-slot-123te4st"> </div>

<!-- Flipp Desktop Anchor Tag-->

<div id="flipp-ux-slot-345st6te"> </div>

# 5. Implementation Options

## Custom Header

The basic anchor can be enhanced with a customizable header. This is highly recommended for post article placement for both the standard and custom experience. However, it is not recommended for mid-article placement of the compact experience. In the anchor example below, the black part is the mandatory anchor and you'll notice the unique ID matches the one that is registered in the controller. The blue component is the custom header container, this is set to display none by default and is unhidden by the controller script when there is weekly ad content to show. The red part is the actual custom header which should show the text “Featured Local Savings”, and can inherit any styles from the main page. The header can even vary for different sites within the same zone.

<!-- Flipp Mobile Anchor Tag-->

<div id="flipp-ux-slot-123te4st">

<div class="flipp-ux-header" style="display: none">

<h3 class="cb-sidebar-widget-title cb-widget-title">Featured Local Savings</h3>

</div>

</div>

<!-- Flipp Desktop Anchor Tag-->

<div id="flipp-ux-slot-345st6te">

<div class="flipp-ux-header" style="display: none">

<h3 class="cb-sidebar-widget-title cb-widget-title">Featured Local Savings</h3>

</div>

</div>

## Viewable Impression Call-Back

The publisher may configure an optional viewable impression call back function that will be called when a viewable impression is rendered. This can be used to facilitate publisher side instrumentation. To use this, define a call-back function of your choice and add it to the window.flippxp variable. See example below

<!-- Flipp Controller Tag-->

<script async

src="https://cdn-gateflipp.flippback.com/tag/js/flipptag.js?site\_id=1245511"></script>

<script>

const vimpcallback = () => {

console.log("Flipp Viewable Impression Event Happened");

};

window.flippxp = window.flippxp || {run: []};

window.flippxp.vimpTrackerFunc = vimpcallback;

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-a5c2e4", "sample", 1245511, [ 286625 ], ); });

</script>

## SlotFill Result Call-Back

The slotFillResultCallback is designed to provide publishers with immediate feedback regarding the status of ad slot fills. This functionality is particularly useful in scenarios where the targeted ad slot is not filled by a Flipp Experience. In such cases, publishers can now integrate their proprietary logic to execute alternative logic.

The function signature is `slotFillResultCallback?: (filled: boolean) => void;` it accepts a boolean value filled, which is set to `true` when a Flipp experience is rendered in the slot, and `false` when no Flipp experience is returned.

// Placeholder function that demonstrates callback

const slotFillResultCallbackFunc = function(filled) {

console.log("slot fill response callback: ", filled);

}

window.flippxp.run.push(function() {

window.flippxp.registerSlot(

"#flipp-ux-slot-123te4st",

"wishabi-test-publisher",

1181258,

[ 259136 ],

{

dwellExpandable: true,

slotFillResultCallback: slotFillResultCallbackFunc

}

);

## Content Recommendation Feed Setup

It is always recommended that the Flipp NativeX shopping experience be placed directly on the page, however there may be cases where it is logistically convenient for a content recommendation vendor to insert the content on the feed on behalf of the media partner.

The publisher should instruct the content recommendation vendor to place the code snippet appropriate for the platform at the appropriate location for the sites of each zone. We recommend placement after the 3rd card. The placement can be different between desktop/tablet and mobile.

### Content Recommendation Setup

Instruct them to allocate only a single pixel of height for this content in their i-frame with a small gap (e.g. 10px) ahead of it to separate it from other content, the Flipp experience will expand as needed. For OutBrain, no specific space allocation is required, though it would be good to have some gap before and after (e.g. 10px) to separate this from other content.

**The code should be placed where the experience would appear (e.g. after the 3rd content card)**

<script async src="https://cdn-gateflipp.flippback.com/tag/js/flipptag.js?site\_id=111111"></script>

<script>

window.flippxp = window.flippxp || {run: []};

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-123te4st", "wishabi-test-publisher", 1181258, [ 259136 ], { nestedIframe: true });

});

</script>

<div id="#flipp-ux-slot-123te4st"><div class='flipp-ux-header' style='display: none;padding: 0px 0px 8px 0px;font-size: 20px;line-height: 24px;color: #2A2A2A;font-family: Arial;font-weight: bold;'>Featured Local Savings</div></div>

Experience within content recommendation feeds can be enhanced with a customizable header. In the anchor examples above, the black part is the mandatory code snippet required for the experience. The red component is the custom header div container that is set to display=none (it would be unhidden when content is shown). Note that because Taboola encloses the experience in an i-frame, it may not be possible to inherit the styles on the page and thus styling will need to be explicitly defined.

Note that vendor experiences can only be used on mobile/desktop web, it is not compatible with Google AMP.

### Others Vendor Partners

Due to the varying nature of these implementations, we may need to provide a customized code snippet for integration. Please contact your Flipp partner technology manager to facilitate this process.

## Partner Provided User Key

By default, the Flipp provided integration code uses a cookie with a random generated anonymous ID to improve the user experience by limiting the frequency of content to the same ID or to prioritize content they are most likely to engage with. The default anonymous ID’s scope is limited to the specific domain.

If a media partner would rather send their own version of an ID to Flipp, say because they have a persistent logged in user ID that spans across multiple properties, they can pass that in as a dynamic string parameter in the integration code. If such a string is present, Flipp will use that as the user key instead. The optional userKey parameter is highlighted in red in the direction integration controller code example below. While any unique string can be used, we would require all potential PII data to be hashed to protect the user’s privacy.

<!-- Flipp Controller Tag-->

<script async

src="https://cdn-gateflipp.flippback.com/tag/js/flipptag.js?site\_id=1111111></script>

<script>

window.flippxp = window.flippxp || {run: []};

if (document.documentElement.clientWidth < 500) {

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-123te4st, “wishabi-test-publisher”, 1111111, [ 2345678 ], {userKey: "ddf3d8d8-0808-4879-9fdb-065374ee94de"});

});

} else {

window.flippxp.run.push(function() {

window.flippxp.registerSlot("#flipp-ux-slot-345st6te”, "wishabi-test-publisher", 1111111, [ 8765432 ], {userKey: "ddf3d8d8-0808-4879-9fdb-065374ee94de"});

});

}

</script>

# 6. Testing

The integration will not show any retailer content until it has been validated by Flipp. To facilitate testing, you can enter in the following URL parameter that would force the experience to render a sample shopping experience:

?flipp-content-code=publisher-test-staging

For example, if you want to force test content to show on a on “www.thiswebsite.com” with the controller and anchor implemented, you would put the following in the URL

www.thiswebsite.com?flipp-content-code=publisher-test-staging

This should only be used for internal implementation testing and public exposure should be minimized, as the sample content may not represent real offers in the user’s area.