



Offers Website Widget & API



Introduction	3
Requirements	3
Standard Javascript Widget	3
Additional Parameters	3
limit	3
offset	4
hideexpired	4
hideexpiredtext	4
noclick	4
CSS Styling	4
HTTPS Feed Access	5
JSON Format	5
XML Format	6
Field Definitions	6

Introduction

Offers from OnBrand is a web platform for the aggregation and retrieval of retailer offers, allowing shopping centres and retail destinations to receive a relevant and up-to-date feed either via a Javascript widget or HTTPS-accessible JSON or XML.

This document describes the process of accessing a feed via all of the above methods.

Requirements

To access the Offers services you will require a valid Offers account and associated API ID number. This is available on request from OnBrand. Please email sayhello@onbrand.co.uk if you don't have this information available to you.

Standard Javascript Widget

The standard Offers widget produces a Pinterest style feed of retailer offers. The widget inserts plain HTML into any webpage, using JQuery and a custom plug-in. The widget is fully customisable by overriding the CSS.

The widget automatically makes use of all available horizontal space but can be problematic with fixed height web pages.

Please contact OnBrand if you have specific integration questions.

To call the widget insert the following code into your web page wherever you would like the offers to appear:

```
<link rel="stylesheet" href="https://offers.onbrand.co.uk/widgetstyle.css"
type="text/css" media="all" />

<script src="https://offers.onbrand.co.uk/yourfeed.v3.js"></script>

<div id="centrebuzz"></div>

<script>CBWidget({'cid' : XXX });</script>
```

The "XXX" in the widget call will need to be replaced with your Offers API ID.

Additional Parameters

There are some additional optional parameters that can be used with the widget to change the output of your offers feed.

limit

The optional '*limit*' parameter will limit the feed to a certain number of offers.

```
<script>CBWidget({'cid' : XXX, 'limit' : YYY });</script>
```

The “YYY” in the widget call will need to be replaced with your desired limit.

offset

The optional ‘*offset*’ parameter will offset the first offer by the given number.

```
<script>CBWidget({'cid' : XXX, 'offset' : YYY });</script>
```

The “YYY” in the widget call will need to be replaced with your desired offset number.

hideexpired

The optional ‘*hideexpired*’ parameter will hide any offers that have expired on your offers feed.

```
<script>CBWidget({'cid' : XXX, 'hideexpired' : YYY });</script>
```

The “YYY” in the widget call will need to be replaced with ever true or false, the default value is false.

hideexpiredtext

The optional ‘*hideexpiredtext*’ parameter will hide the expiry date of the offer.

```
<script>CBWidget({'cid' : XXX, 'hideexpiredtext' : YYY });</script>
```

The “YYY” in the widget call will need to be replaced with ever true or false, the default value is false.

noclick

The optional noclick parameter

```
<script>CBWidget({'cid' : XXX, 'noclick' : YYY });</script>
```

The “YYY” in the widget call will need to be replaced with ever true or false, the default value is false.

CSS Styling

The widget produces HTML as in the following example:

```
<div id="container">
  <div class="item">
    <span class="pitempicture"><img src="" /></a></span>
    <div class="pitemspacer"></div>
    <span class="pitemtitle"><a href=""></a></span>
    <span class="pitemcontent mcontent"></span>

  </div>
  ...
  <div class="item">
    <span class="pitempicture"><img src="" /></a></span>
    <div class="pitemspacer"></div>
    <span class="pitemtitle"><a href=""></a></span>
    <span class="pitemcontent mcontent"></span>

  </div>
</div>
```


The follow CSS selectors can be overridden to style the offers.

```
<style>
  #centrebuzz #container { }
  #centrebuzz .item { }
  #centrebuzz .pitempicture { }
  #centrebuzz .pitemimg { }
  #centrebuzz .spacer5 { }
  #centrebuzz .pitemtitle { }
  #centrebuzz .pitemtitle a { }
  #centrebuzz .pitemcontent { }
  #centrebuzz .mcontent { }
  #centrebuzz .expiry { }
</style>
```

To alter the CSS style of the feed, simply overload the above selectors in your sites CSS file or the head style tag of your web page.

You may then choose to remove the default widget CSS link from your web page.

```
<link rel="stylesheet" href="https://offers.onbrand.co.uk/widgetstyle.css"
type="text/css" media="all" />

<script src="https://offers.onbrand.co.uk/yourfeed.v3.js"></script>

<div id="centrebuzz"></div>

<script>CBWidget({'cid' : XXX });</script>
```

HTTPS Feed Access

Access to the raw feed data is via HTTPS. You can make an HTTPS GET call to the following URI to receive the data:

```
https://offers.onbrand.co.uk/yourfeed.php?cid=[APIID]&limit=[LIMIT]&format=[FORMAT]
```

In the above call, "APIID" will need to be replaced with your Offers API ID and FORMAT with one of the below valid format options. LIMIT can be completely excluded and will default to the most recent 30 offers if not specified.

JSON Format

To receive JSON data from the API, simply add json as the format type:

```
https://offers.onbrand.co.uk/yourfeed.php?cid=[APIID]&limit=[LIMIT]&format=json
```

You will receive the following data structure:

```
[
  {
    "id" : "Alphanumeric ID",
    "title" : "Item Title",
    "content" : "Item Content",
    "link" : "Item Link URI ",
    "photo" : "Item Photo URI",
    "expiry" : "Item expiry date",
    "retailerid" : "Retailer Aggregation ID"
  },
  ...
  {
    "id" : "Alphanumeric ID",
    "title" : "Item Title",
    "content" : "Item Content",
    "link" : "Item Link URI",
    "photo" : "Item Photo URI",
    "expiry" : "Item expiry date",
    "retailerid" : "Retailer Aggregation ID"
  }
]
```

Depending on your selected GET parameters, you will receive a larger or smaller array of JSON objects.

XML Format

To receive XML data from the API, simply add xml as the format type:

```
https://offers.onbrand.co.uk/yourfeed.php?cid=[APIID]&limit=[LIMIT]&format=xml
```

You will receive the following data structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<item>
  <id>[offer id]</id>
  <title>[title]</title>
  <content>[content]</content>
  <link>[link]</link>
  <photo>[photo url]</photo>
  <expiry>[expiry date]</expiry>
  <retailerid>[retailer id]</retailerid>
</item>
...
<item>
  <id>[offer id]</id>
  <title>[title]</title>
  <content>[content]</content>
  <link>[link]</link>
  <photo>[photo url]</photo>
  <expiry>[expiry date]</expiry>
  <retailerid>[retailer id]</retailerid>
</item>
```

Depending on your selected GET parameters, you will receive a larger or smaller XML document.

Field Definitions

Whether in XML or JSON format, the following fields will be returned by the GET call:

Field	Description
id	The unique alphanumeric ID number for the offer
title	The offer title in UTF-8
content	The text of the offer in UTF-8
link	The HTTP URI for the destination offer page
photo	The HTTP URI for the offer image
retailerid	The unique numeric ID for the offer source retailer
expiry	The expiry date for the offer