

Project:	Native Ads web
Title:	Configuration definition
Doc. Number:	008
Date:	16-11-2015



Version	Date	Author	Description
1.0	05-10-2015	Niek Kruse	Initial draft
1.1	16-11-2015	Niek Kruse	Added changes with additionalElements
1.2	17-11-2015	Niek Kruse	Added illustrative examples

## DESCRIPTION

This document contains the definition & structure of the JSON configuration file that the native ads JavaScript library uses for placing native ads on the right place, in the right format.

## STRUCTURE

The configuration is wrapped in a single JavaScript object. The configuration will make use of the Google JSON style guide: [http://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml#Property\\_Name\\_Format](http://google-styleguide.googlecode.com/svn/trunk/jsoncstyleguide.xml#Property_Name_Format)

## Base

Name	Description	Optional	Example value
affiliateID	The ID of the affiliate.	No	2234
adUnits	Array of adUnits. See adUnits section below	No	See adUnits Section
displayOverrides	Override the default widths for defining whether a device is mobile/tablet	Yes	See DisplayOverrides Section
additionalElementTemplates	HTML templates for the additional elements defined in adUnits.	No	See additionalElementTemplates section

## adUnits

An adUnit is an object that define what a native ad should look like and when and where it should be placed.

Name	Description	Example value	Optional	Default behavior
name	Name to identify the ad unit by	news_list	No	none
devices	String array of devices the adUnit should run on. Possible values: "mobile", "tablet"	["mobile"] (adunit will run on mobile devices)	No	
htmlTemplate	The HTML template of the ad Unit. It specifies the HTML structure of the adUnit	See the Ad-Template macros documentation.	No	
patterns	Array of patterns for this adUnit.  See patterns section	[{*pattern*}, ...]  See patterns section.	No	
additionalElements	Array of additionalElements that should be added to this adUnit. see AdditionalElement section below	[{*additionalElement*}, ...]	An additionalElement with name adUnit is required	
trigger	Indicates whether this adUnit should be triggered manually	true or false	Yes	false

## Patterns

The patterns section is used for specifying the patterns that should be matched to decide whether or not an adUnit should run on the current page.

When a the specified url pattern of an adUnit matches the pattern of the pathname of the page, the adUnit will run on that current page, and insert ads on the specified container

Name	Description	Example value	Optional
urlPatterns	<p>The URL patterns the page should match in order to display the ad.</p> <p>Matches against the current page's pathname.</p> <p>An asterisks (*) can be used as wildcard.</p> <p>Eg.</p> <p>"" – will run on home page "article/1234" will run on article 1234 "article/*" will run on all articles</p>	<pre>["/article/*", "", /post/*]</pre>	No
containers	Array of container	<pre>[{*container*}, ...]</pre> <p>See containers section</p>	No

## Containers

The containers section is used for specifying ad containers for a certain pattern. An adContainer is the parent element in which native ads can be inserted.

Name	Description	Example value	Optional
xPath	The xPath string to find the container the native ads should be inserted in on the page	<pre>//div[@id='newsList']</pre>	No
startIndex	The index after which child element in the container the first ad should be inserted	2	No

Interval	The interval after how many articles a new ad should be inserted	3	Yes  <b>Note: if not set, maxNumberOfAds should be set</b>
maxNumberOfAds	The max number of ads to show in the container (ads are distributed evenly on the childcontainers)	2	Yes  <b>NOTE: if not set, interval has to be set</b>

As an example, let's say we have the following HTML structure:

```
<ul id="adContainer">
  <li class="article"></li>
  <li class="article"></li>
  <li class="nativeAd"></li>
  <li class="article"></li>
  <li class="article"></li>
  <li class="nativeAd"></li>
</ul>
```

In this example, the ul with id "adContainer" is the adContainer. If we would specify a container for this HTML structure, it would look like this:

```
{
  "xPath":"//div[@id='adContainer']", //The ul with id 'adContainer'
  "startIndex":1, //start after element with index 1 (we start counting at 0)
  "interval":2 //insert a native ad after every 2 regular articles
}
```

### additionalElement

The additionalElement is part of the configuration for an adunit. An additional element specifies an element that is added on top of the native ad structure. Examples of additionalElements include the adlcon and the install button you see in native ads.

Name	Description	Optional?
templateName	The Name of the template for the additional element. For example: "adlcon".  The HTML structure of this template is specified in the additionalElementsTemplate section.	No
Position	The position object specifying the element placement, see additionalElement.position	No

### *additionalElement.position*

An additionalElement's placement is absolute, relative to the entire element of the native ad.

Name	Description	Optional?
verticalAnchor	The anchor to which the additional element anchors vertically. Possible values are: "top" "bottom"	No
verticalOffset	The vertical offset from the vertical anchor in pixels.	No
HorizontalAnchor	The anchor to which the additional element anchors horizontally. Possible values are: "left" "right"	No
horizontalOffset	The horizontal offset from the horizontal anchor	No
fixScreenSizeDifference	Boolean indicating whether the difference in screen sizes should be corrected for the horizontal offset.  Every configuration is written for the width of 375px. In the case that the adIcon anchors to the left and the icon is placed after dynamically sized, setting this Boolean will make sure the difference in screen size is calculated.	yes

For example, if we use the following configuration:

```
{
  "verticalAnchor": "top",
  "horizontalAnchor": "left",
  "verticalOffset": 10,
  "horizontalOffset": 20
}
```

this would generate the following output:

```
<div id="additionalElement" style="top:10px; left:20px; position:absolute;">
  {additionalElementTemplate}
</div>
```

Let's say the additionalElement template is a 10px by 10px green square, we'll leave the adUnit contents empty for now. This would generate the following HTML:

```
<div id="adUnit" style="width:120px; height:100px; border:1px solid black; position:relative">
  <div id="additionalElement" style="top:10px; left:20px; width:10px; height:10px; position:absolute;">
    <div style="width:10px; height:10px; background-color: green;"></div>
  </div>
</div>
```

Which would render like this:



If we change the position of the additionalElement like so:

```
{
  "verticalAnchor":"bottom",
  "horizontalAnchor":"right",
  "verticalOffset":10,
  "horizontalOffset":20
}
```

We would end up with this:



## additionalElementTemplates

Multiple additional elements can be added to any adUnit. It can happen occasionally that multiple adUnits make use of the same additional elements, like the adIcon and the installButton. In order to prevent duplication, the html template for these additionalElements are specified outside of the adUnits, in the additionalElementTemplates section.

The additionalElementsTemplates is an object consisting of other objects. The name of these objects should equal the name specified in the "templateName" property of an additionalElement.

There is the ability to make the templates platform independent by using the "independent" property, or make the template depend on the platform, by using the "ios" or "android" property. An example:

```
"additionalElementTemplates": {
  "greenPixel": {
    "independent": "<div style=\"width:10px; height:10px; background-color:green;\"></div>" //platform independent, same on every platform
  },
  "platformName":{
    "iOS":"<div>iOS</div>", //Shows "iOS" on iOS devices
    "Android":"<div>Android</div>" //Shows "Android" on Android devices
  }
}
```

## displayOverrides

The displayOverrides property can be used for overriding the default settings of when a site is seen as mobile and/or tablet by the ad library. If for example a website treats a device of 420 pixels as a smartphone, this can be told to the adLibrary by setting the displayOverrides.

The current default values are as follows:

```
{
  "mobile": {
    "minWidth": 0,
    "maxWidth": 415
  },
  "tablet": {
    "minWidth": 415,
    "maxWidth": 1024
  }
}
```

This can be overridden through the displayOverrides properties like so:

```
"displayOverrides": {
  "mobile": {
    "minWidth": 0,
    "maxWidth": 420
  },
  "tablet": {
    "minWidth": 420,
    "maxWidth": 1050
  }
}
```

## AD LIBRARY OVERRIDES

The ad library has the ability to have certain settings overridden through the **ADLIB\_OVERRIDES** object residing on the global scope of the page.

The ADLIB\_OVERRIDES can contain the following properties:

Name	Type	Description	Example value
configFileLocation	String	Manually override the location where the native ad library looks for the ad configuration. By default this is, use this override to change the location.  {hostname}/adconfig.json	/static/adconfig.json
currentPath	String	Manually override the current path of the page. This path is used for validating adUnit patterns against the current page pathname.	/article/123
platform	String	Manually override the detected platform.  Allowed values: "Android" "iOS"	iOS
version	String	Manually override the version number of the current device	8.1.0

formFactor	String	Manually override the formFactor of the device. The formFactor is used for determining which adUnits to display on the current page  Allowed values: "SmartPhone", "Tablet"	SmartPhone
fullDeviceName	String	Manually override the full device name.	"iPhone6"