



How To Integrate Native Ads

Created by Smaato, last modified on Sep 07, 2015

- [Set up a Native Adspace](#)
- [SDK Users](#)
 - [iOS](#)
 - [Android](#)
- [API Users](#)
 - [Parameters](#)
 - [Example Native Ad Request](#)
 - [Example Native Ad Response](#)

Set up a Native Adspace

Set up an Adspace in your SPX inventory that you want to use for Native ads. Currently, we've got the following items available :

- Ad Title
- Ad Text
- Icon Image (e.g. 80x80 pixels)
- Main Image (e.g. 1200x627 pixels)
- Click to install text (e.g. Install, Read More etc...)
- Star rating (only for app ads)

Just create an Adspace in SPX as always, and select "Native" from the Ad Format dropdown.

SDK Users

iOS

For Native ads, the publishers just need to supply different UI components to the SDK to perform the final rendering.

However, it is **recommended** that all UI components should be added to a **dedicated UIView container** to handle the clicks by the SDK.

Only when the container is registered with the SDK, the ad will be visible, clicks are enabled and beacons are called.

Native Ads Integration

```
1 // Step 1: Declare a property in the view controller header:
2 @property SOMANativeAd* nativeAd;
3 // Step 2: Instantiate:
4 self.nativeAd = [[SOMANativeAd alloc] initWithPublisherId:@"0" adSpaceId:@"0"];
5 self.nativeAd.delegate = self;
6
7 // Step 3: Configure UI components
8 self.nativeAd.labelForTitle = self.title; // an UILabel instance to render the title
9 self.nativeAd.labelForDescription = self.text; // an UILabel instance to render the c
10 self.nativeAd.imageViewForIcon = self.icon; // an UIImageView instance to render the
11 self.nativeAd.imageViewForMainImage = self.photo; // an UIImageView instance to rende
12
13 // Step 4: Load the ad
14 [self.nativeAd load];
15
16 // Stop 5: Once the ad is loaded, make the final configuration and then register the
17 - (void)somaAdViewDidLoadAd:(SOMAAdView*)adview{
18 - (void)somaNativeAdDidLoad:(SOMANativeAd*)nativeAd{
19     self.callToAction.text = nativeAd.callToAction; // UILabel instance for the "call
20     [self updateRating:nativeAd.rating]; // Optional: native ad comes with a floating
21
22
23     // Finally, register the container. Make sure all the native ad UI components are
24     [nativeAd registerViewForUserInteraction:self.container];
25
26 }
27
28
29 // thats it!
```

Android

In order to start using native ads, you will need to go through following steps :

- Create all needed views (icon view, main image view, text views, rating bar etc...)
- Put your views together so that they look similar to your app content
- Pass the views to our SDK
- and start requesting

After you created all the views, please proceed by passing the views to our SDK by using the following code:

```
1 NativeAd nativeAd = new NativeAd(this);
2 nativeAd.setClickToActionButton(ctaButton).setIconImageView(iconImage)
3     .setMainImageView(mainImage).setTextView(nativeText)
4     .setTitleView(nativeTitle).setMainLayout(nativeRelativeLayout).setRati
```

The next and final step, will be that you start requesting ads by using the following method:

```
1 nativeAd.asyncLoadNewBanner();
```

Sure you can also get notified and know if we delivered an ad or not. To activate that, please let your activity implement our `AdListenerInterface`. Below is a sample code:

```
1 public class MainActivity implements AdListenerInterface
```

And of course, do not forget to register the listener by using following code :


```
1 nativeAd.setAdListener(MyActivity);
```

API Users

Parameters

In addition to the standard parameters that you need for calling an ad, use the following parameters to request Native ads:

Parameter	Mandatory	Description	Type	Set Value to:
apiver	yes	(You need to use at least version 5.0)	integer	500
nver	yes	1	integer	1
nsupport	no	The requested native ad component names separated by comma. If not set, at least a title and a click through url will be included in the ad response.	string	title,txt,icon,image,ctatext
format	yes	Use native for Native ads	string	native
width	yes	The main image width	integer	1200
height	yes	The main image height	integer	627

 You'll receive an XML response.

Example Native Ad Request

```
http://soma.smaato.net/oapi/reqAd.jsp?pub=0&adspace=3075&format=native&width=1200&height=62
```

Example Native Ad Response

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://soma.smaato.com/oapi/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  <sessionid>83E48A90C2565475700B60C3B92E57E7.soma-i-99a8e47d</sessionid>
  <status>success</status>
  <user>
    <id>900</id>
    <ownid />
  </user>
  <ads>
    <ad id="0" type="NATIVE">
      <log-id />
      <valid start="0" end="0" max="1" />
      <beacons>
        <beacon>http://soma.smaato.net/oapi/adspacer.gif</beacon>
        <beacon>http://52.17.141.198/oapi/getAd;jsessionId=83E48A90C2565475700B60C3B92E
      </beacons>
      <SNAST>
        <adtitle>Smaato</adtitle>
        <adtext>The Leading Global Mobile RTB Ad Exchange andi SSP</adtext>
        <iconimage>
          <icon width="80" height="80">http://52.17.141.198/oapi/getAd;jsessionId=83E4
        </iconimage>
        <mainimage>
          <image width="1200" height="627">http://52.17.141.198/oapi/getAd;jsessionId=
        </mainimage>
        <starrating>4.5</starrating>
        <ctatext>Read more</ctatext>
        <beacons />
        <clickurl>http://52.17.141.198/oapi/clickAd;jsessionId=83E48A90C2565475700B60C3
      </SNAST>
    </ad>
  </ads>
</response>
```