Yandex Mobile Ads

Integration

10.07.2024

Yandex

Yandex Mobile Ads. Integration. Version 2.0

Document build date: 10.07.2024

This volume is a part of Yandex technical documentation.

Copyright Disclaimer

Contact information

Yandex LLC

https://www.yandex.com

Тел.: +7 495 739 7000

Email: pr@yandex-team.ru

16 L'va Tolstogo St., Moscow, Russia 119021

Contents

Integrating the Mobile Ads SDK

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Note:

1. To load ads of any type, you need iOS 12.0 or later.
2. To ensure that iOS 14 (or higher) runs correctly, keep in mind the additional steps.

Attention:

The SDK requires enabling the resources located in `YandexMobileAdsBundle.bundle` from `YandexMobileAds.framework`. When you integrate the SDK via CocoaPods, these resources are enabled automatically.

When using a customized `YandexMobileAds.framework` integration, make sure that `YandexMobileAds.bundle` is copied to the project resources.

The library is compatible with the following dependency management systems:

Swift Package Manager

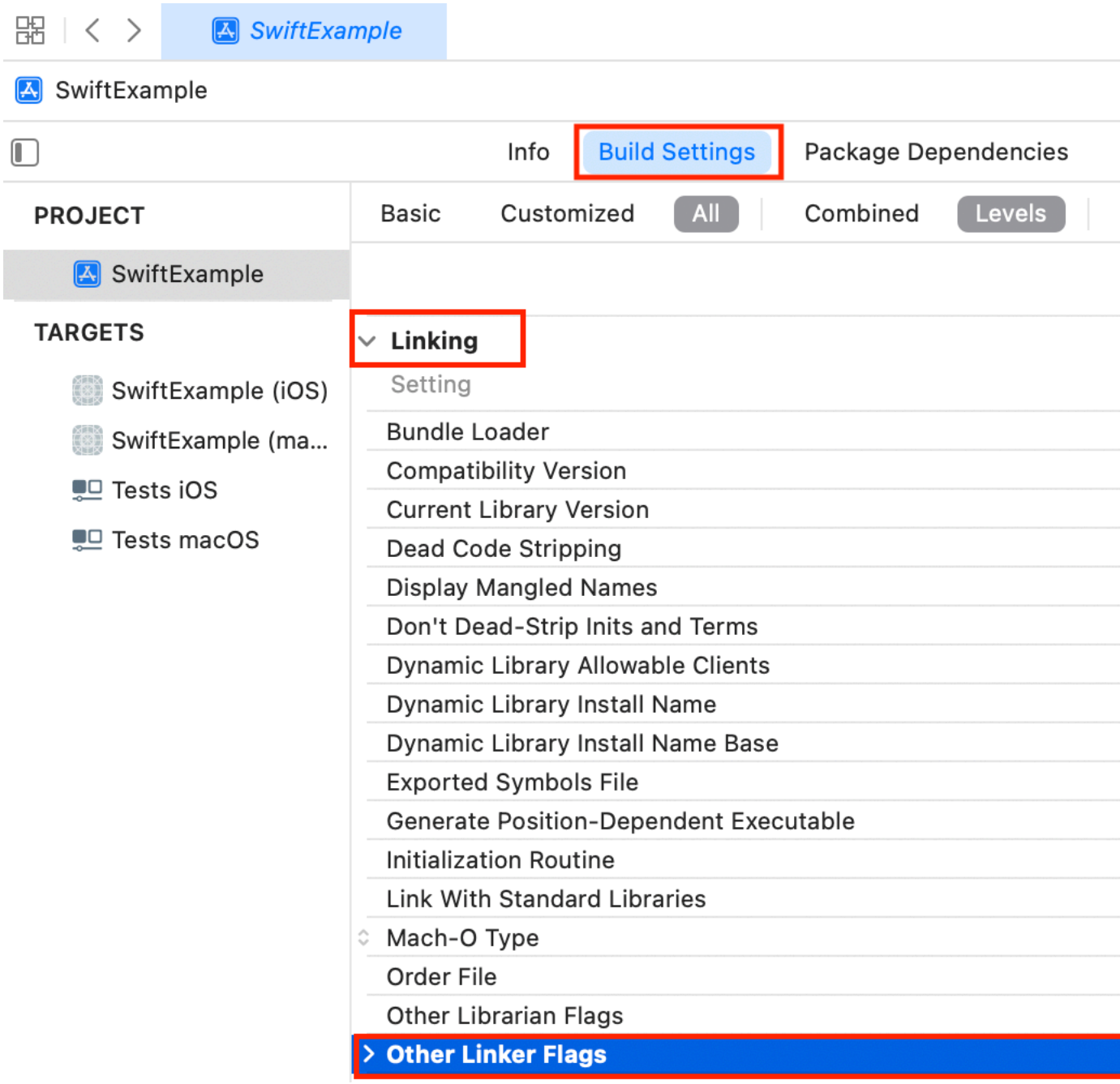
Note:

Mediation adapters aren't supported by the Swift Package Manager integration. If you use mediation, use the integration via CocoaPods.

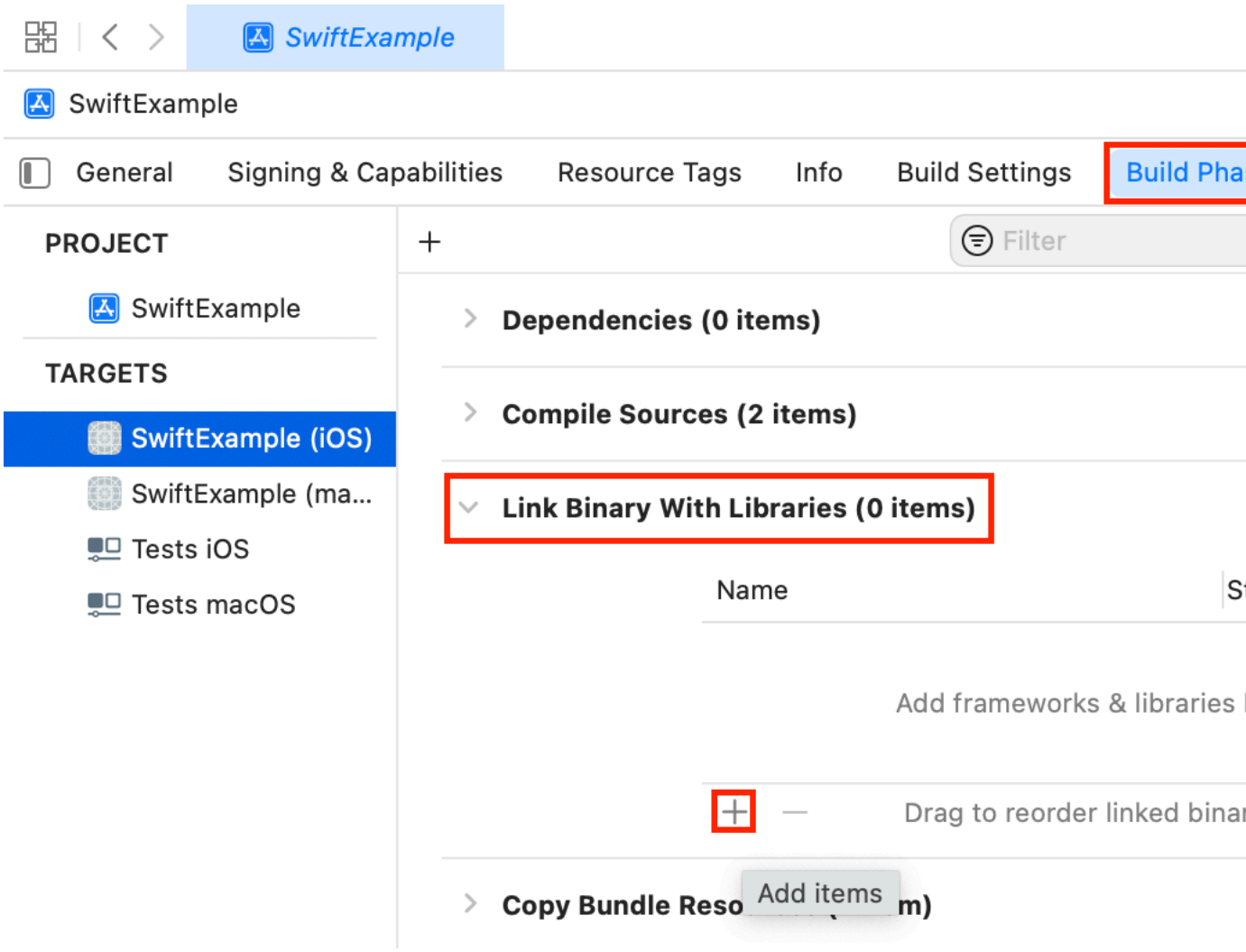
To add the library, follow these steps:

1. In Xcode, add a dependency to your project using **File** → **Add Packages**.
2. Specify the repository URL `https://github.com/yandexmobile/yandex-ads-sdk-swift`: it includes the Swift package.

3. In **Build Settings**, under **Linking**, add the `Other Linker Flags = -ObjC` parameter value.

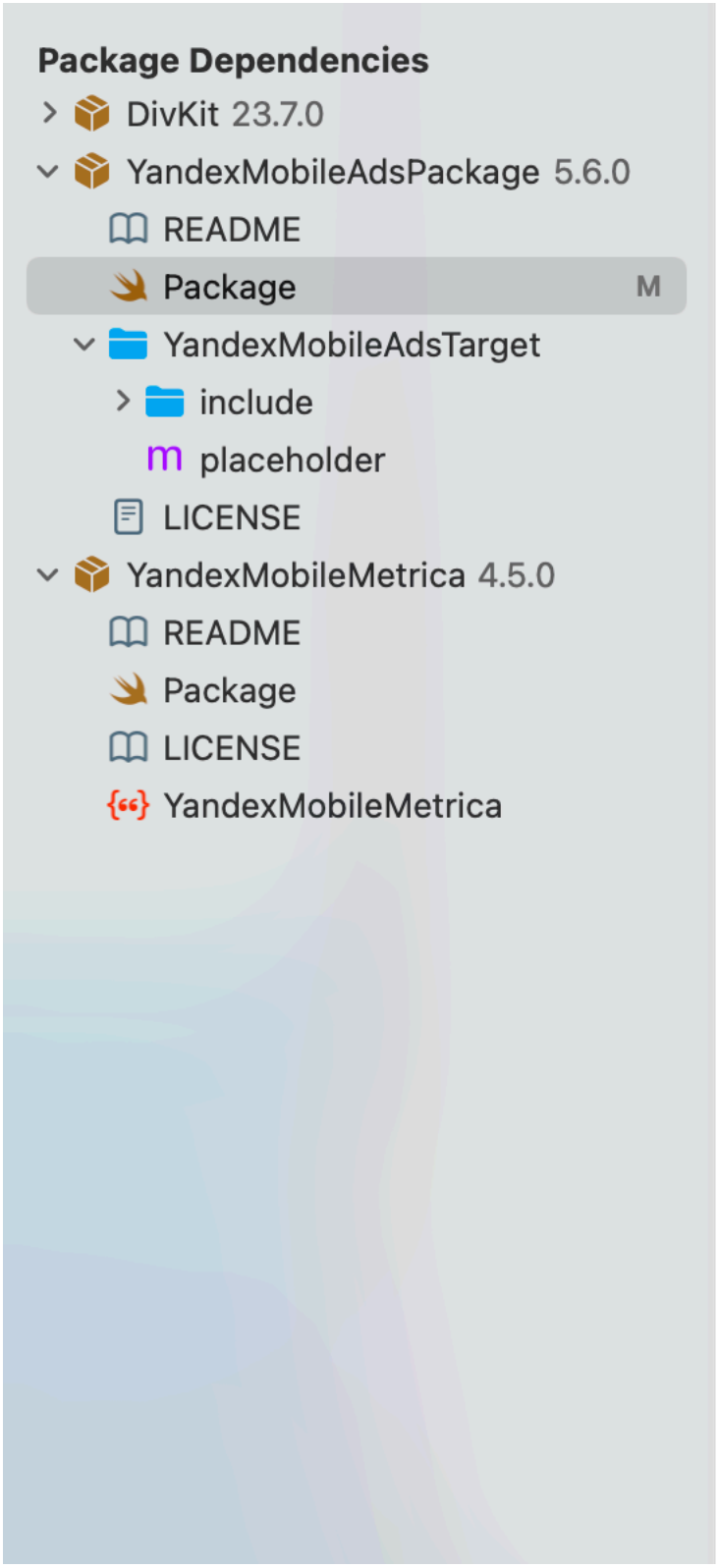


4. Check that the target is linked to YandexMobileAdsPackage. If the target is not linked, add a link to the library under **Link Binary With Libraries** using the **+** icon.

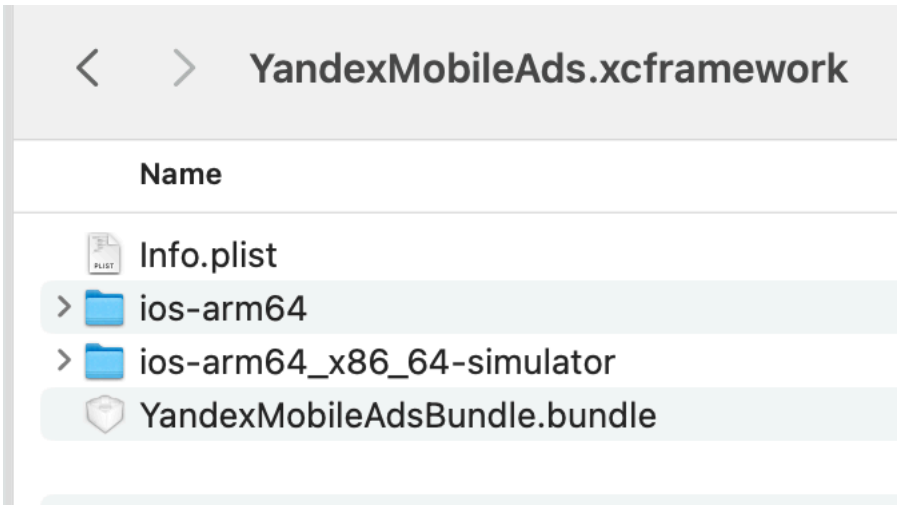


5. Add YandexMobileAdsBundle.bundle from YandexMobileAds. To do this:

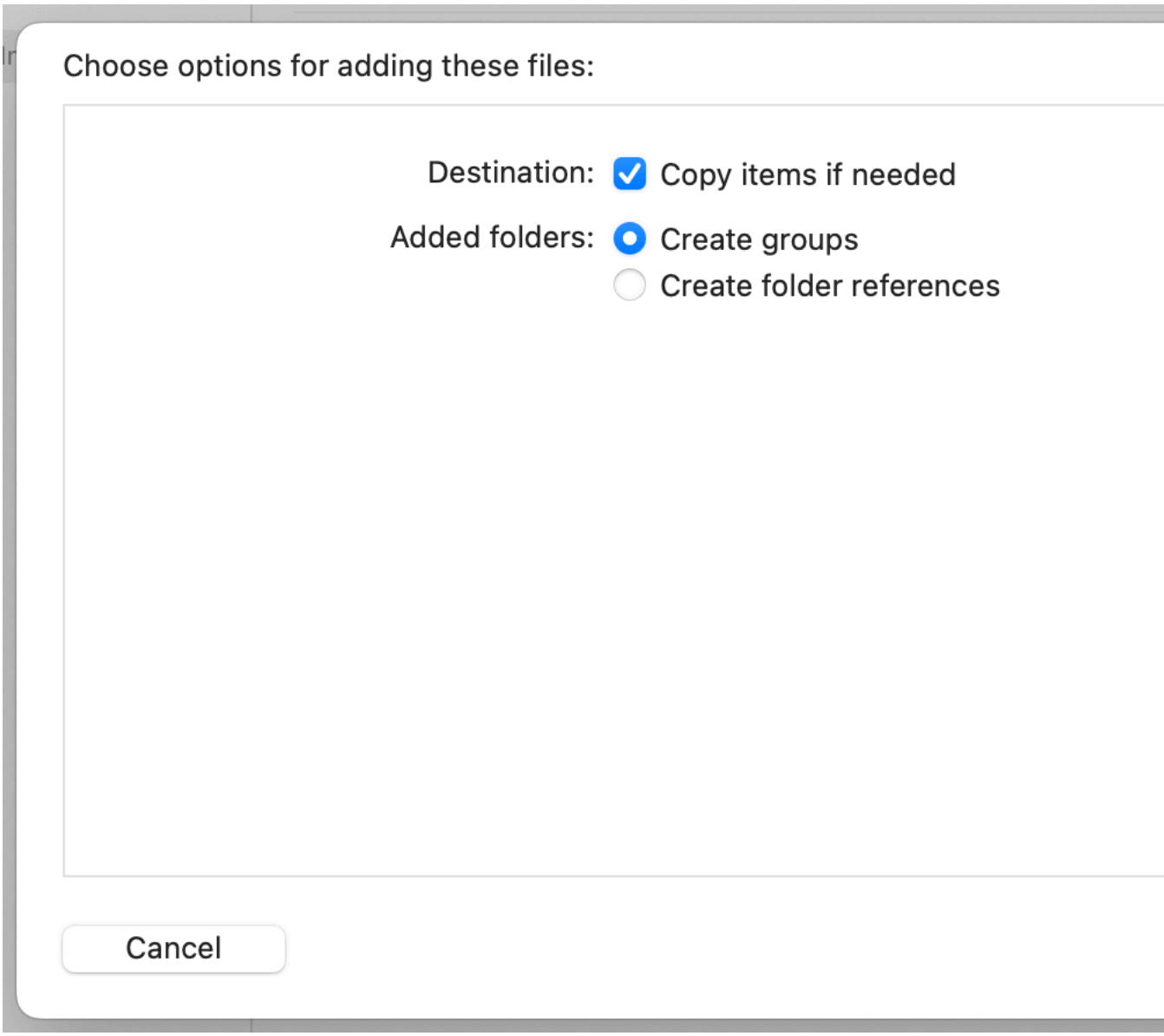
   a. Follow the link for **YandexMobileAds** under **.binaryTarget** (**Package Dependencies → YandexMobileAdsPackage → click on Package.swift → find the link for YandexMobileAds under .binaryTarget**).

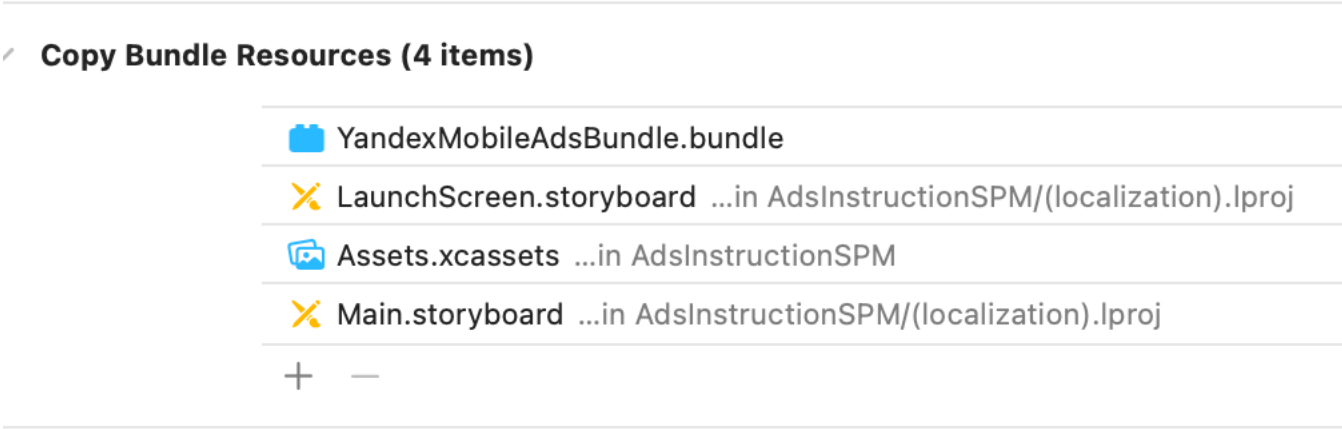


   b. Open the downloaded archive.



c. Add YandexMobileAdsBundle.bundle to the Copy Bundle Resource phase (**Your target → Build Phases → Copy Bundle Resource → + button → Add other → select YandexMobileAdsBundle.bundle → select Copy items if needed and Create groups**).





CocoaPods

The Yandex Mobile Ads SDK library has been adapted to work with the CocoaPods dependency management system and supports a static integration method. To connect the library, add the following dependencies to the project's `Podfile` (see the example for a static framework):

```
pod 'YandexMobileAds', '5.9.1'
pod 'YandexMobileAdsInstream', '0.18.0'
```

SKAdNetwork support

Note:

SKAdNetwork is supported for SDK version 4.1.2 and higher.

Mobile Ads SDK supports tracking of app installations using the SKAdNetwork framework. Installation tracking works for any device, even if no access to IDFA was granted.

To enable this feature, add the IDs of the supported ad networks to the `Info.plist` file of your application.

```
<key>SKAdNetworkItems</key>
<array>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zq492l623r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>633vhxswh4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>tmhh9296z4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vcra2ehyfk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zh3b7bxvad.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xmn954pzmp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>79w64w269u.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>488r3q3dtq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>d7g9azk84q.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>nzq8sh4pbs.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>866k9ut3g3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2q884k2j68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x8jxxk4ff5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gfat3222tu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pd25vrrwzn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>lr83yxwka7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cp8zw746q7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pwdxu55a5a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c6k4g5qg8m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>s39g8k73mm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wg4vff78zm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g28c52eehv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>523jb4fst2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>294l99pt4k.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3qy4746246.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a8cz6cu7e5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ggvn48r87g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y755zyxw56.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qlbq5gtkt8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mls7yz5dvl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>67369282zy.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>899vrgt9g8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mj797d8u6f.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>3sh42y64q3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f38h382jlk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>24t9a8vw3c.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mp6xlyr22a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x44k69ngh6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>88k8774x49.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hs6bdukanm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t3b3f7n3x8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>prcb7njmu6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c7g47wypnu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>52fl2v3hgk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9vvzujtq5s.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m8dbw4sv7c.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9g2aggbj52.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m5mvw97r93.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>z5b3gh5ugf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dd3a75yxkv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9nlqeag3gk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cj5566h2ga.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>h5jmj969g5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dr774724x4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t7ky8fmwkd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>fz2k2k5tej.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>u679fj5vs4.skadnetwork</string>
    </dict>
```

```
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cs644xg564.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9b89h5y424.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>w28pnjg2k4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2rq3zucswp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a7xqa6mtl2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g2y4y55b64.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vc83br9sjg.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cstr6suwn9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>eqhxz8m8av.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7k3cvf297u.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>w9q455wk68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nu4557a4je.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v4nxqhlyqp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wzmmz9fp6w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7fmhfwg9en.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>su67r6k2v3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>yclnxrl5pm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7tnzynbdc7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>l6nv3x923s.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>h8vml93bkz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>uzqba5354d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8qiegk9qfv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v79kvwwj4g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xx9sdjej2w.skadnetwork</string>
```

```
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>au67k4efj4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t38b2kh725.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7ug5zh24hu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>rx5hdcabgc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5lm9lj6jb7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qqp299437r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zmvfpc5aq8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9rd848q2bz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>79pbpufp6p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dmv22haz9p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y5ghdn5j9k.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n6fk4nfna4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7rz58n8ntl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v9wttpbfk9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n38lu8286q.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>feyaarzu9v.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7fbxrn65az.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>47vhws6wlr.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ejvt5qm6ak.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>b55w3d8y8z.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v7896pgt74.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5ghnmfs3dh.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>275upjj5gd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>627r9wr2y5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>kbd757ywx3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>sczv5946wb.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8w3np9l82g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hb56zgv37p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9t245vhmpl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nrt9jy4kw9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7953jerfzd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dn942472g5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6v7lgmsu45.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cad8qz2s3j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n9x2a789qt.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>r26jy69rpl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>eh6m2bh4zr.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>jb7bn6koa5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>fkak3gfpt6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a2p9lx4jpn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>97r2b46745.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>22mmun2rn5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>238da6jt44.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>44jx6755aq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>b9bk5wbcq9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>k674qkevps.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>tl55sbb4fm.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>24zw6aqk47.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4468km3ulz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2tdux39lx8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2u9pt9hc89.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8s468mfl3y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3cgn6rq224.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>glqzh8vgby.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>av6w8kgt66.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>klf5c3l5u5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nfqy3847ph.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dticjx1a9i.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ppxm28t8ap.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9wsyqb3ku7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>74b6s63p6l.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xy9t38ct57.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>424m5254lk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qu637u8glc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f73kdq92p3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>44n7hlldy6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>kbmxgpxpgc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5l3tpt7t6e.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ecpz2srf59.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x5854y7y24.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f7s53z58qe.skadnetwork</string>
    </dict>
```

```
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x8uqf25wch.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>uw77j35x4d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6964rsfnh4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gvmwg8q7h5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6yxyv74ff7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>84993kbrcf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>54nzkqm89y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pwa73g5rt2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mlmmfzh3r3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9yg77x724h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n66cz3y3bx.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>578prtvx9j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4dzt52r2t5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>bvpn9ufa9b.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6qx585k4p6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mtkv5xtk9e.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>l93v5h6a4m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>rvh3l7un93.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gta9lk7p23.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5tjdwbrq8w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>r45fhb6rf7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>32z4fx6l9h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>e5fvkxwrpn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8c4e2ghe7u.skadnetwork</string>
```

```
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>axh5283zss.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3rd42ekr43.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5mv394q32t.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3qcr597p9d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v72qych5uu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ydx93a7ass.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4pfyvq9l8r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5a6flpkh64.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4fzdc2evr5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4w7y6s5ca2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>252b5q8x7y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2fnua5tdw4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3l6bd9hu43.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4mn522wn87.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6g9af3uyq4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6p4ks3rnbw.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6xzpu9s2p8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>737z793b9f.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>89z7zv988g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8m87ys6875.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8r8llnkz5a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>bxvub5ada5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c3frkrj4fj.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>cg4yq2srnc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dbu4b84rxf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dkc879ngq3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dzg6xy7pwj.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gta8lk7p23.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hdw39hrw9y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hjevpa356n.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>krvm3zuq6h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ln5gz23vtd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ludvb6z3bs.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m297p6643m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>p78axxw29g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pu4na253f3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>s69wq72ugq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t6d3zquu66.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vutu7akeur.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x2jnk7ly8j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x5l83yy675.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y45688jllp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>yrqqpx2mcb.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>z4gj7hsk7h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wzmmZ9fp6w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4pfyvq9L8r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>V72QYCH5UU.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>2U9PT9HC89.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3RD42EKR43.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4FZDC2EVR5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7953JERFZD.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7UG5ZH24HU.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9RD848Q2BZ.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9T245VHMPL.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>CJ5566H2GA.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>F38H382JLK.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>KLF5C3L5U5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>M8DBW4SV7C.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>W9Q455WK68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>WZMMZ9FP6W.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>XY9T38CT57.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>YCLNXRL5PM.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4PFYVQ9L8R.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>TL55SBB4FM.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>BD757YWX3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>33r6p7g8nc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g69uk9uh2b.skadnetwork</string>
    </dict>
</array>
```

For more information, see Configuring a Source App in the Apple documentation.

See the SDK usage examples.

Ad formats

Enabling banner ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

A *banner* is a configurable ad that covers part of the screen and reacts to clicks.

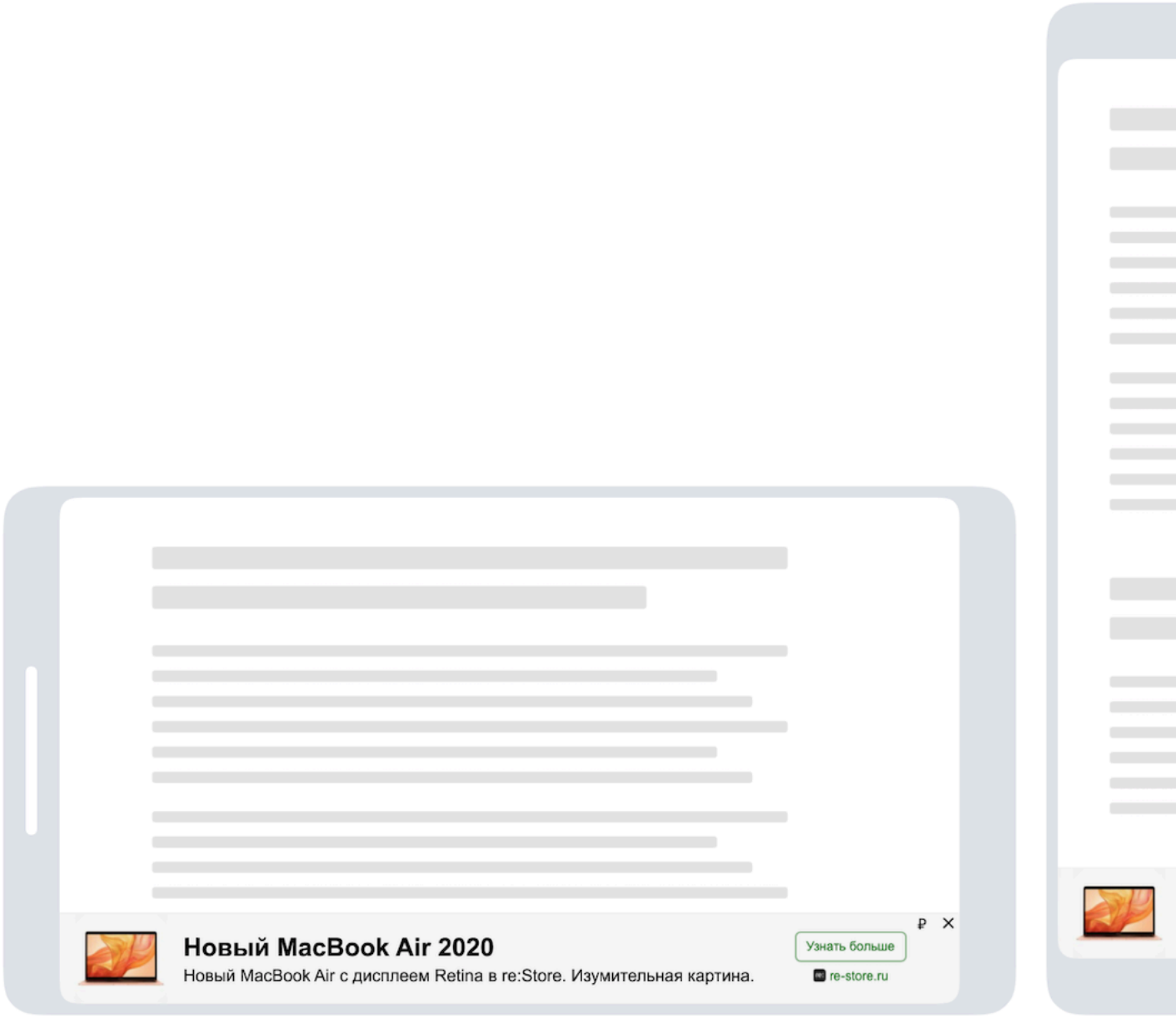
Banner types

Sticky banner

Features:

1. The specified banner width is used. The height is selected automatically.
2. The width of banners is set using the +stickySizeWithContainerWidth: method.
3. The banner height shouldn't exceed 15% of the device height and should be at least 50 dp.

Examples of displaying banners:

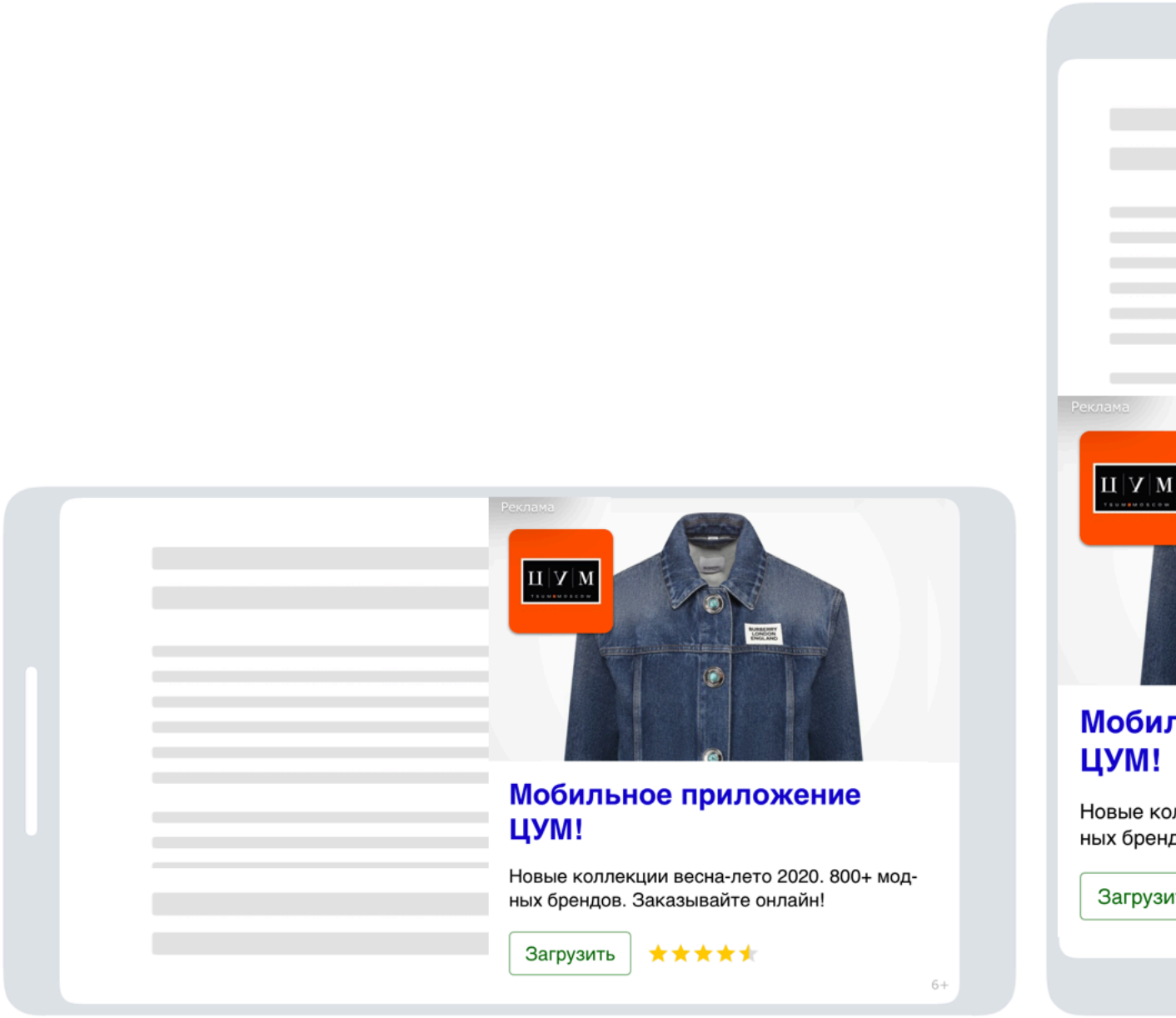


Flex banner

Features:

1. A banner fills up the entire unit using the set maximum sizes.
2. The width and height of banners is set using the +flexibleSizeWithCGSize: method.
3. The SDK imposes `NSLayoutConstraint` on the banner height.

Examples of displaying banners:



Enabling a banner

To enable the banner:

1. Add the import:

Swift

```
import YandexMobileAds
```

Objective-C

```
#import <YandexMobileAds/YandexMobileAds.h>
```

2. Create @property, where the link to the banner ad will be stored:

   Swift

   ```
   var adView: YMAAdView!
   ```

   Objective-C

   ```
   @property (nonatomic, strong) YMAAdView *adView;
   ```

3. Create a banner:

   Sticky banner

   To set the width of a banner, call the +stickySizeWithContainerWidth: method.

   Swift

   ```
   let adSize = YMAAdSize.stickySize(withContainerWidth: width)
   let adView = YMAAdView(adUnitID: "", adSize: adSize)
   adView.delegate = self
   ```

   Objective-C

   ```
   YMAAdSize adSize = [YMAAdSize stickySizeWithContainerWidth:width];
   YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:<AdUnitID> adSize:adSize];
   adView.delegate = self;
   ```

   Flex banner

   To set the width and height of a banner, call the +flexibleSizeWithCGSize: method.

   Swift

   ```
   let adSize = YMAAdSize.flexibleSize(with: size)
   let adView = YMAAdView(adUnitID: "", adSize: adSize)
   adView.delegate = self
   ```

   Objective-C

   ```
   YMAAdSize adSize = [YMAAdSize flexibleSizeWithCGSize:size];
   YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:<AdUnitID> adSize:adSize];
   adView.delegate = self;
   ```

   Restriction: Banner size requirements when displaying video ads

   Minimum size of a banner that supports video playback is 300x160 or 160x300.

   `AdUnitId` is a unique identifier in R-M-XXXXXX-Y format, which is assigned in the Partner Interface.

   In addition, `self` must conform to the YMAAdViewDelegate protocol. If the delegate implements the -viewControllerForPresentingModalView method, links open in the in-app browser. Otherwise, links open in the browser installed on the device.

   To find out why ads aren't working correctly, use the -adViewDidFailLoading:error: method.

   For error descriptions, see YMAAdErrorCode.

4. Display a banner. There are two ways to place a banner:

   - Using `autolayout constraints`.

     Add the banner to `UIView`. Then add `autolayout constraints` so the banner is displayed in the desired location.

     Swift

     ```
     view.addSubview(adView)
     adView.translatesAutoresizingMaskIntoConstraints = false
     var adViewConstraints = [
         adView.leadingAnchor.constraint(equalTo: adView.superview!.leadingAnchor),
         adView.trailingAnchor.constraint(equalTo: adView.superview!.trailingAnchor)
     ]
     let bottomDistance: CGFloat = 8
     if #available(iOS 11.0, *) {
         adViewConstraints.append(
             adView.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant:
      bottomDistance)
         )
     } else {
         adViewConstraints.append(
     ```

```
        adView.bottomAnchor.constraint(equalTo: view.bottomAnchor, constant: bottomDistance)
    )
}
NSLayoutConstraint.activate(adViewConstraints)
```

Objective-C

```
UIView *adView = self.adView;
[self.view addSubview:adView];
adView.translatesAutoresizingMaskIntoConstraints = NO;

NSMutableArray *adViewConstraints = [NSMutableArray arrayWithArray:@[
    [adView.leadingAnchor constraintEqualToAnchor:adView.superview.leadingAnchor],
    [adView.trailingAnchor constraintEqualToAnchor:adView.superview.trailingAnchor]
]];
int bottomDistance = 8;
if (@available(iOS 11.0, *)) {
    UILayoutGuide *guide = self.view.safeAreaLayoutGuide;
    [adViewConstraints addObject:[adView.bottomAnchor constraintEqualToAnchor:guide.bottomAnchor
                                                                     constant:bottomDistance]];
} else {
    [adViewConstraints addObject:[adView.bottomAnchor
 constraintEqualToAnchor:adView.superview.bottomAnchor
                                                                     constant:bottomDistance]];
}
[NSLayoutConstraint activateConstraints:adViewConstraints];
```

- Using the following methods:

  Swift

  ```
  displayAtTop(in:)
  displayAtBottom(in:)
  ```

  Objective-C

  ```
  -displayAtTopInView:;
  -displayAtBottomInView:;
  ```

  In both cases, banners are centered horizontally.

5. Load the banner. Optionally, you can use the YMAAdRequest class to transmit the data for targeting.

   Swift

   ```
   func loadAd(with request: YMAAdRequest?)
   ```

   Objective-C

   ```
   - (void)loadAdWithRequest:(YMAAdRequest *)request;
   ```

   When the banner loads, the delegate is notified.
6. You can optionally enable logging by using the +enableLogging method. If an impression wasn't registered, a message appears in the console.
7. Optionally, you can set up notifications about the end of video playback in banner ads.

   Usage example

   Swift

   ```
   // Getting an instance of YMAVideoController using VideoController.
   let videoController = adView.videoController

   // Setting up a delegate that implements the YMAVideoDelegate protocol.
   videoController.delegate = self

   // Implementing the YMAVideoDelegate protocol method.
   // MARK: - YMAVideoDelegate
   func videoControllerDidFinishPlayingVideo(_ videoController: YMAVideoController) {
       print("Video complete");
   }
   ```

   Objective-C

   ```
   // Getting an instance of YMAVideoController using videoController.
   YMAVideoController *videoController = self.adView.videoController;

   // Setting up a delegate that implements the YMAVideoDelegate method.
   videoController.delegate = self;

   // Implementing the YMAVideoDelegate protocol method.
   #pragma mark - YMAVideoDelegate
   - (void)videoControllerDidFinishPlayingVideo:(YMAVideoController *)videoController
   ```

```
{
    NSLog(@"%@", @"Video complete");
}
```

If an ad is integrated this way, the banner appears after the app starts.

To see how the banner ad will be displayed in the app, use a demo AdUnitId:

- demo-banner-yandex

Related information

Ad example

Example of working with banners

The following code demonstrates creating and configuring the AdView object, registering a listener, and loading a banner.

Swift

```
override func viewDidLoad() {
    super.viewDidLoad()

    // Setting the ad width
    let adSize = YMAAdSize.stickySize(withContainerWidth: containerWidth())

    // Creating an adView instance     adView = YMAAdView(adUnitID: "R-M-XXXXX-YY", adSize: adSize)
    adView.delegate = self
    addAdView()

    // Loading the ad     adView.loadAd()
}

private func addAdView() {
    view.addSubview(adView)
    adView.translatesAutoresizingMaskIntoConstraints = false
    var adViewConstraints = [
        adView.leadingAnchor.constraint(equalTo: adView.superview!.leadingAnchor),
        adView.trailingAnchor.constraint(equalTo: adView.superview!.trailingAnchor)
    ]
    let bottomDistance: CGFloat = 8
    if #available(iOS 11.0, *) {
        adViewConstraints.append(
            adView.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant:
 bottomDistance)
        )
    } else {
        adViewConstraints.append(
            adView.bottomAnchor.constraint(equalTo: view.bottomAnchor, constant: bottomDistance)
        )
    }
    NSLayoutConstraint.activate(adViewConstraints)
}

private func containerWidth() -> CGFloat {
    var containerWidth = view.frame.width
    if #available(iOS 11, *) {
        containerWidth = view.frame.inset(by: view.safeAreaInsets).width
    }
    return containerWidth
}
```

Objective-C

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    // Setting the ad width
    YMAAdSize *adSize =
        [YMAAdSize stickySizeWithContainerWidth:[self containerWidth]];

    // Creating an adView instance     self.adView = [[YMAAdView alloc] initWithAdUnitID:@"R-M-XXXXX-YY"
 adSize:adSize];
    self.adView.delegate = self;
    [self addAdView];

    // Loading the ad     [self.adView loadAd];
}

- (void)addAdView
{
    UIView *adView = self.adView;
    [self.view addSubview:adView];
    adView.translatesAutoresizingMaskIntoConstraints = NO;
```

```
    NSMutableArray *adViewConstraints = [NSMutableArray arrayWithArray:@[
        [adView.leadingAnchor constraintEqualToAnchor:adView.superview.leadingAnchor],
        [adView.trailingAnchor constraintEqualToAnchor:adView.superview.trailingAnchor]
    ]];
    int bottomDistance = 8;
    if (@available(iOS 11.0, *)) {
        UILayoutGuide *guide = self.view.safeAreaLayoutGuide;
        [adViewConstraints addObject:[adView.bottomAnchor constraintEqualToAnchor:guide.bottomAnchor
                                                                         constant:bottomDistance]];
    } else {
        [adViewConstraints addObject:[adView.bottomAnchor constraintEqualToAnchor:adView.superview.bottomAnchor
                                                                         constant:bottomDistance]];
    }
    [NSLayoutConstraint activateConstraints:adViewConstraints];
}

- (CGFloat)containerWidth
{
    CGFloat containerWidth = self.view.frame.size.width;
    if (@available(iOS 11.0, *)) {
        containerWidth =
            UIEdgeInsetsInsetRect(self.view.frame, self.view.safeAreaInsets).size.width;
    }
    return containerWidth;
}
```

Adaptive banners

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Adaptive banners are banners that fit seamlessly into user-defined unit sizes. Depending on how an adaptive banner is integrated, the optimal height is determined for the given width or the specified size of ad placement is used.

Note:

You can read about creating an ad unit for an adaptive banner in the Advertising Network Help.

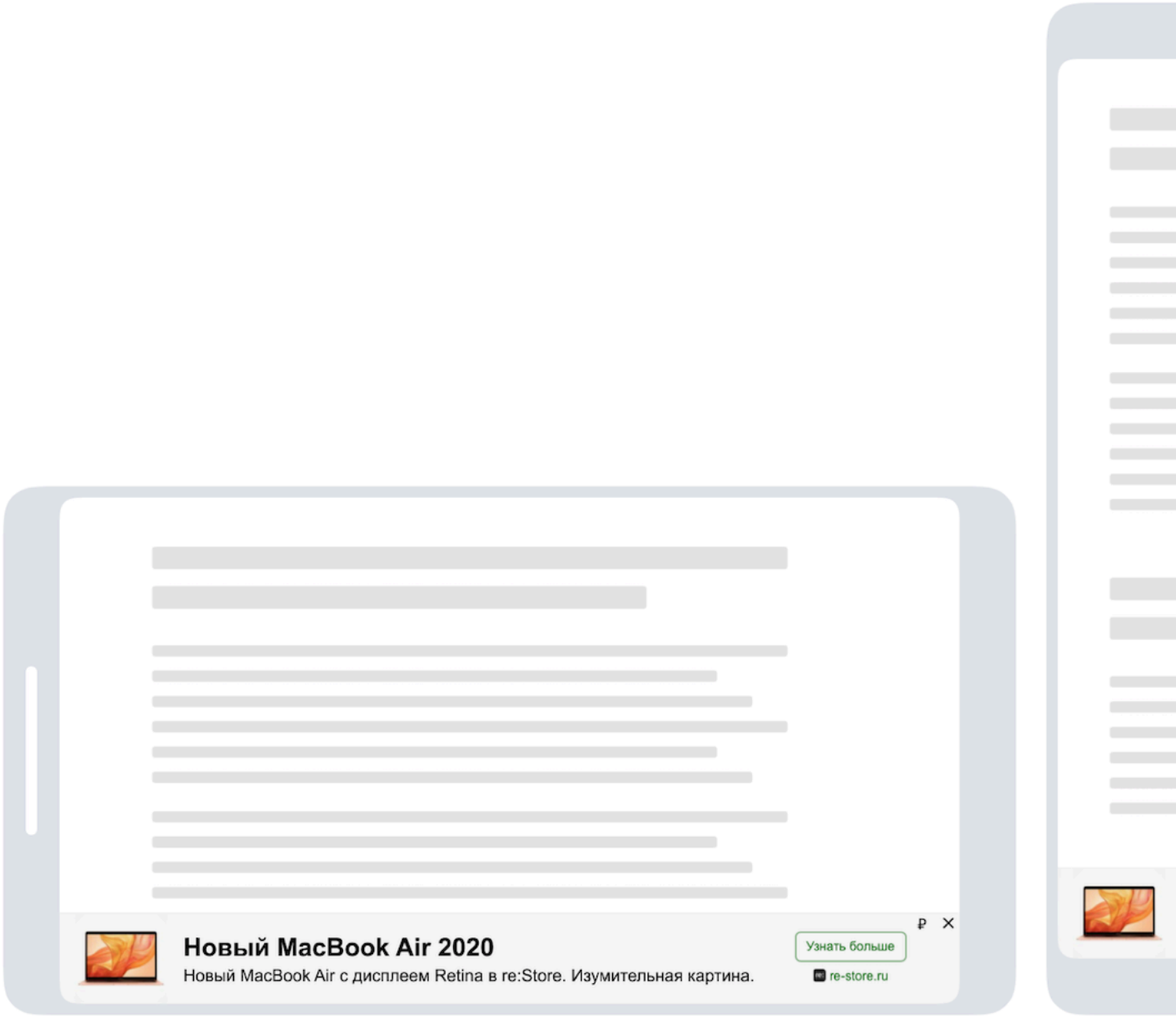
Types of adaptive banners

Banner with set width

Features:

1. An alternative to 320x50 banners (when determining the banner height, the aspect ratio of 320x50 is maintained).
2. The banner is fixed in place at the top or bottom of the screen (set up in the app).
3. The given banner width is used instead of the device screen width. This lets you take into account the display's features.
4. The width of adaptive banners is set using the +stickySizeWithContainerWidth: method.

Examples of displaying adaptive banners:

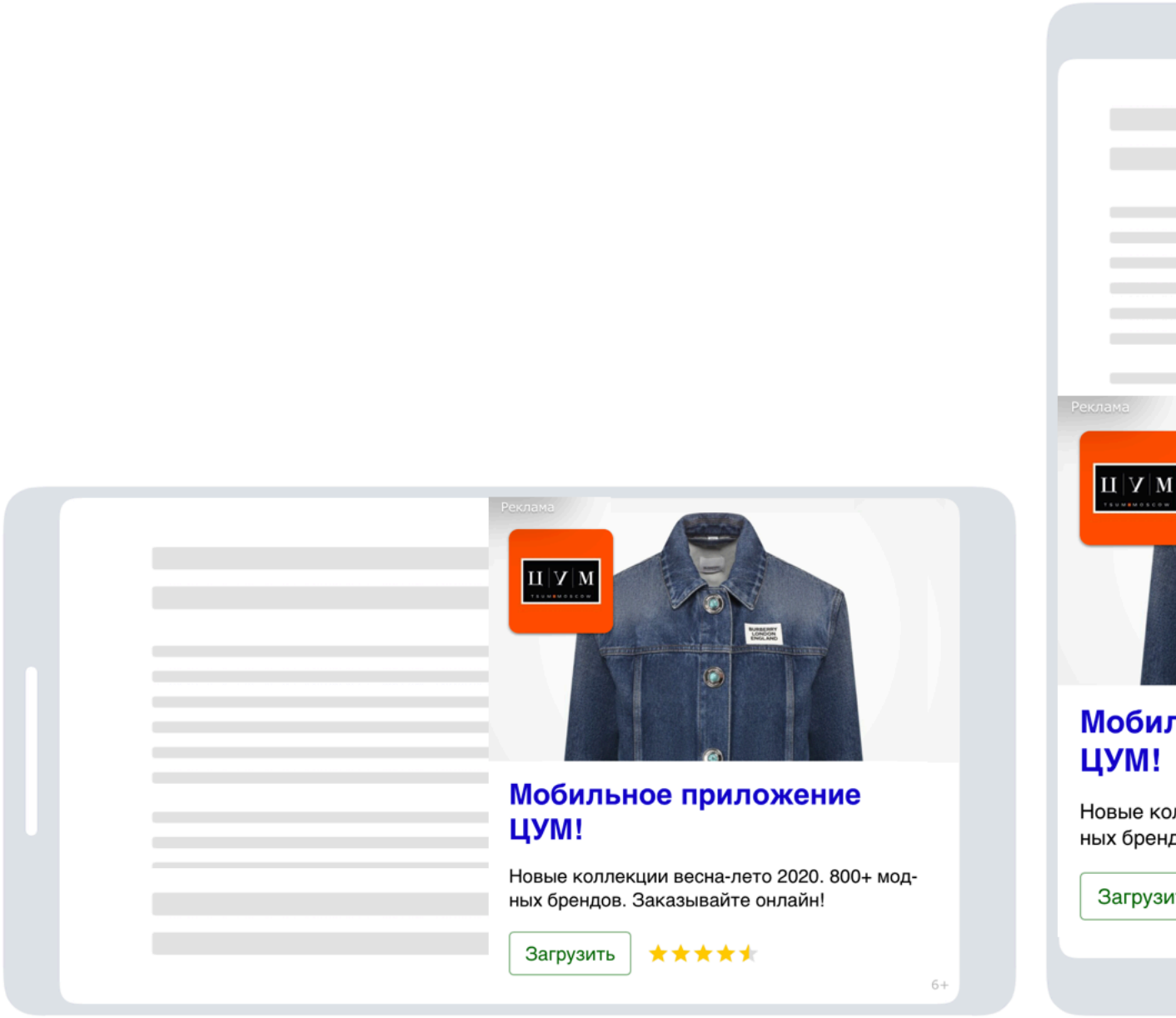


Banner with set width and height

Features:

1. An adaptive banner fills up the entire unit using the given width and height.
2. The width and height of adaptive banners is set using the +flexibleSizeWithCGSize: method.

Examples of displaying adaptive banners:



Enabling banner ads

1. Add the import:

   Swift

   ```
   import YandexMobileAds
   ```

   Objective-C

   ```
   #import <YandexMobileAds/YandexMobileAds.h>
   ```

2. Create @property, where the link to the banner ad will be stored:

   Swift

   ```
   var adView: YMAAdView!
   ```

   Objective-C

   ```
   @property (nonatomic, strong) YMAAdView *adView;
   ```

3. Specify the banner size and initialize the banner in the app.

   Banner with set width

   To set the width of an adaptive banner, call the +stickySizeWithContainerWidth: method.

   ```
   YMAAdSize adSize = [YMAAdSize stickySizeWithContainerWidth:width];
   YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:<AdUnitID> adSize:adSize];
   adView.delegate = self;
   ```

   Banner with set width and height

   To set the width and height of an adaptive banner, call the +flexibleSizeWithCGSize: method.

   ```
   YMAAdSize adSize = [YMAAdSize flexibleSizeWithCGSize:size];
   YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:<AdUnitID> adSize:adSize];
   adView.delegate = self;
   ```

   `AdUnitId` is a unique identifier in R-M-XXXXXX-Y format, which is assigned in the Partner interface.

   In addition, `self` must conform to the YMAAdViewDelegate protocol. If the delegate implements the -viewControllerForPresentingModalView method, links open in the in-app browser. Otherwise, links open in the browser installed on the device.

4. Display a banner. You can place a banner, for example, using `autolayout constraints`.

   Usage example

   Add the banner to `UIView`. Then add `autolayout constraints` so the banner is displayed in the desired location.

   ```
   [self.view addSubview:self.adView];
    self.adView.translatesAutoresizingMaskIntoConstraints = false;
       UIView *adView = self.adView;
       NSDictionary<NSString *, UIView *> *views =
           NSDictionaryOfVariableBindings(adView);
       NSArray<NSLayoutConstraint *> *horizontal =
          [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[adView]|"
                                                  options:0
                                                  metrics:nil
                                                    views:views];
       NSArray<NSLayoutConstraint *> *vertical =
          [NSLayoutConstraint constraintsWithVisualFormat:@"V:[adView]-|"
                                                  options:0
                                                  metrics:nil
                                                    views:views];
       [self.view addConstraints:horizontal];
       [self.view addConstraints:vertical];
   ```

5. Load the banner. Optionally, you can use the YMAAdRequest class to transmit the data for targeting.

   Swift

   ```
   func loadAd(with request: YMAAdRequest?)
   ```

   Objective-C

   ```
   - (void)loadAdWithRequest:(YMAAdRequest *)request;
   ```

   When the banner loads, the delegate is notified.

Example of working with adaptive banners

The following code demonstrates creating and configuring the `AdView` object, registering a listener, and loading an adaptive banner:

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    // Setting the ad width
    YMAAdSize *adSize =
        [YMAAdSize stickySizeWithContainerWidth:[self containerWidth]];

    // Creating an adView instance
    self.adView = [[YMAAdView alloc] initWithAdUnitID:@"R-M-XXXXX-YY" adSize:adSize];
    self.adView.delegate = self;
    [self addAdView];

    // Loading ads     [self.adView loadAd];
}
```

```
- (void)addAdView
{
    UIView *adView = self.adView;
    adView.translatesAutoresizingMaskIntoConstraints = NO;
    NSDictionary<NSString *, UIView *> *views = NSDictionaryOfVariableBindings(adView);
    [self.view addSubview:adView];

    NSArray<NSLayoutConstraint *> *horizontal =
       [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[adView]|"
                                               options:0
                                               metrics:nil
                                                 views:views];
    NSArray<NSLayoutConstraint *> *vertical =
       [NSLayoutConstraint constraintsWithVisualFormat:@"V:[adView]-|"
                                               options:0
                                               metrics:nil
                                                 views:views];
    [self.view addConstraints:horizontal];
    [self.view addConstraints:vertical];
}

- (CGFloat)containerWidth
{
    CGFloat containerWidth = self.view.frame.size.width;
    if (@available(iOS 11.0, *)) {
        containerWidth =
            UIEdgeInsetsInsetRect(self.view.frame, self.view.safeAreaInsets).size.width;
    }
    return containerWidth;
}
```

Related information

Ad example

Classes and protocols for working with banner ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Classes

- YMAAdSize
- YMAAdView
- YMAVideoController

Protocols

- YMAAdViewDelegate
- YMAVideoDelegate

Enabling interstitial ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

An interstitial ad is a configurable ad that covers the entire screen and responds to clicks.

To enable advertising:

1. Add the import:

   Swift

   ```
   import YandexMobileAds
   ```

   Objective-C

   ```
   #import <YandexMobileAds/YandexMobileAds.h>
   ```

2. Create `@property` where the link to the banner ad will be stored:

   Swift

   ```
   var interstitialAd: YMAInterstitialAd!
   ```

   Objective-C

   ```
   @property (nonatomic, strong) YMAInterstitialAd *interstitialAd;
   ```

3. Perform initialization and pre-loading of the ad. This must be performed after displaying the app interface.

   Optionally, you can use the YMAAdRequest class to transmit the data for targeting. For an example of using the class, see Interstitial ads.

   Swift

   ```
   interstitialAd = YMAInterstitialAd(adUnitID: "<AdUnitID>")
   interstitialAd.delegate = self
   interstitialAd.load()
   ```

   Objective-C

   ```
   self.interstitialAd = [[YMAInterstitialAd alloc] initWithAdUnitID:<your unique AdUnitId>];
   self.interstitialAd.delegate = self;
   [self.interstitialAd load];
   ```

   `AdUnitId` is a unique identifier in R-M-XXXXXX-Y format, which is assigned in the Partner Interface.

   In addition, `self` must conform to the YMAInterstitialAdDelegate protocol. The ad must be pre-loaded in the same orientation as it will be shown (otherwise, the ad won't be shown because the banner size doesn't match the screen size).

4. Start displaying ads by using this method:

   Swift

   ```
   func interstitialAdDidLoad(_ interstitialAd: YMAInterstitialAd) {
       interstitialAd.present(from: self)
   }
   ```

   Objective-C

   ```
   - (void)interstitialAdDidLoad:(YMAInterstitialAd *)interstitialAd
   {
       [interstitialAd presentFromViewController:self];
   }
   ```

5. You can optionally enable logging by using the +enableLogging method. If an impression wasn't registered, a message appears in the console.

To find out why ads aren't working correctly, use the methods

Swift

```
func interstitialAdDidFail(toLoad interstitialAd: YMAInterstitialAd, error: Error)
func interstitialAdDidFail(toPresent interstitialAd: YMAInterstitialAd, error: Error)
```

Objective-C

```
- (void)interstitialAdDidFailToLoad:(YMAInterstitialAd *)interstitialAd error:(NSError *)error;
- (void)interstitialAdDidFailToPresent:(YMAInterstitialAd *)interstitialAd error:(NSError *)error;
```

For error descriptions, see YMAAdErrorCode.

To see how the ad will be displayed in the app, use a demo `AdUnitId`:

- `demo-interstitial-yandex`

Related information

Ad example

Classes and protocols for working with interstitial ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Classes

- YMAInterstitialAd

Protocols

- YMAInterstitialAdDelegate

Enabling rewarded ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

A *rewarded ad* is a configurable full-screen ad. The user gets a reward for viewing the ad.

To enable advertising:

1. Add the import:

   Swift

   ```
   import YandexMobileAds
   ```

   Objective-C

   ```
   #import <YandexMobileAds/YandexMobileAds.h>
   ```

2. Create `@property` for storing the link to the ad:

   Swift

   ```
   var rewardedAd: YMARewardedAd!
   ```

   Objective-C

   ```
   @property (nonatomic, strong) YMARewardedAd *rewardedAd;
   ```

3. Initialize the ad:

   Swift

   ```
   rewardedAd = YMARewardedAd(adUnitID: “<AdUnitID>”)
   rewardedAd.delegate = self
   ```

   Objective-C

   ```
   self.rewardedAd = [[YMARewardedAd alloc] initWithAdUnitID:<your unique AdUnitID>];
   self.rewardedAd.delegate = self;
   ```

   `AdUnitId` is a unique identifier in R-M-XXXXXX-Y format, which is assigned in the Partner Interface.

   In addition, `self` must conform to the YMARewardedAdDelegate protocol.

4. Load an ad.

   Swift

   ```
   rewardedAd.load()
   ```

   Objective-C

   ```
   [self.rewardedAd load];
   ```

   Optionally, you can use the YMAAdRequest class to transmit the data for targeting. For an example of using the class, see Interstitial ads.

   The ad must be pre-loaded in the same orientation as it will be shown (otherwise, the ad will not be shown because the banner size won't match the screen size).

5. Start displaying ads by using this method:

   Swift

   ```
   func rewardedAdDidLoad(_ rewardedAd: YMARewardedAd) {
       rewardedAd.present(from: self)
   }
   ```

   Objective-C

   ```
   - (void)rewardedAdDidLoad:(YMARewardedAd *)rewardedAd
   {
       [rewardedAd presentFromViewController:self];
   }
   ```

6. If you are using the “client-side reward” mechanism, implement the - rewardedAd:didReward: delegate method. It is called when the impression is registered and the user can be rewarded for viewing the ad. Use this chance to give the reward to the app user.
7. You can optionally enable logging by using the +enableLogging method. If an impression wasn't registered, a message appears in the console.

   To find out why ads aren't working correctly, use the methods

   Swift

   ```
   func rewardedAdDidFail(toLoad rewardedAd: YMARewardedAd, error: Error)
   func rewardedAdDidFail(toPresent rewardedAd: YMARewardedAd, error: Error)
   ```

   Objective-C

   ```
   - (void)rewardedAdDidFailToLoad:(YMARewardedAd *)rewardedAd error:(NSError *)error;
   - (void)rewardedAdDidFailToPresent:(YMARewardedAd *)rewardedAd error:(NSError *)error;
   ```

   For error descriptions, see YMAAdErrorCode.

To see how the ad will be displayed in the app, use a demo `AdUnitId`:

- `demo-rewarded-yandex`

Classes and protocols for working with rewarded video ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Classes

- YMARewardedAd

Protocols

- YMAReward
- YMARewardedAdDelegate

Native ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Yandex Mobile Ads SDK lets you render ads using your own visual assets.

Native ads can adapt to the features and design of the app they are displayed in. The layout of a native ad matches the environment it is integrated into. This type of ad looks natural and contributes useful information to the app.

The SDK also provides a set of ready-made customizable visual assets (templates) that let you enjoy all the benefits of rendering from the platform's native tools without creating your own design.

The Yandex Mobile Ads SDK supports several types of advertising: App Install, Content, Image.

To enable advertising:

1. Read the advertising requirements.
2. Load an ad.
3. Configure the ad's design.

Related information

Ad example

MediaView integration guide

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Note:

Integration is described using the example of an `ImageAd`.

1. Enable the Mobile Ads SDK.

   Add dependencies to the `Podfile` project:

   ```
   pod 'YandexMobileAds', '5.9.1'
   pod 'YandexMobileAdsInstream', '0.18.0'
   ```

2. Load the native ad.

   a. Create an object of the YMANativeAdLoader class with the configuration YMANativeAdRequestConfiguration.
   b. Implement the YMANativeAdLoaderDelegate protocol and the -nativeAdLoader:didLoadAd: method.
   c. Set a `delegate` for the created `YMANativeAdLoader` object.
   d. Call the -loadAdWithRequestConfiguration: method.

3. Display the native ad.

   Restriction: Requirements for the size of mediaView when displaying video ads

   Minimum size of an instance of the YMANativeMediaView class, which supports video playback: 300x160 or 160x300.

   To support video playback in native ads, we recommend setting the width for `mediaView` to at least 300. To calculate the appropriate `mediaView` height value, use the aspectRatio property value. When displaying

ads using templates, the correct height for `mediaView` will be calculated automatically based on the width to height ratio.

a. For an easy way to display an ad, use the templates.

1. Create an object of the YMANativeBannerView class.
2. Set the loaded ad object for it.

Swift

```
override func viewDidLoad() {
    super.viewDidLoad()
    adLoader = YMANativeAdLoader()
    adLoader.delegate = self
    let requestConfiguration = YMANativeAdRequestConfiguration(adUnitID: “<AdUnitID>”)
    adLoader.loadAd(with: requestConfiguration)
}
```

Objective-C

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.adLoader = [[YMANativeAdLoader alloc] init];
    self.adLoader.delegate = self;
    YMANativeAdRequestConfiguration *requestConfiguration =
        [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:@"your_AdUnitID"];
    [self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
}done
```

For more information about native advertising, see Native ads.

b. You can also use the manual method to display native ads. This allows you to adapt native ads to your main content as closely as possible. Call the -bindWithAdView:error: method.

Swift

```
func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd) {
    ad.delegate = self
    do {
        try ad.bind(with: adView)
    } catch {
        print("Error: \(error)")
    }
}
func configureView(for ad: YMANativeAd) {
    let assets = ad.adAssets()

    if let media = assets.media {
        //you can use the aspect ratio if you need it to determine the size of media view.
        print(String(format: "Media aspect ratio: %.2f", media.aspectRatio))
    }
}
```

Objective-C

```
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
    ad.delegate = self;
    NSError *error = nil;
    BOOL result = [ad bindWithAdView:self.adView error:error];
    if (error != nil) {
        NSLog(@"Error: %@", error);
    }
}
- (void)configureViewForAd:(id<YMANativeAd>)ad
{
    YMANativeAdAssets *assets = [ad adAssets];

    YMANativeAdMedia *media = assets.media;
    if (media != nil) {
        //you can use the aspect ratio if you need it to determine the size of media view.
        NSLog(@"Media aspect ratio: %.2f", media.aspectRatio);
    }
}
```

For more information about native advertising, see Layout without a template.

Related information

Ad example

Advertising requirements

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Basic requirements

The ad impression won't be registered if even one of the following conditions isn't met:

1. The ad `View`, along with each of its `superviews`, must meet the following conditions:
   - The `View` must be visible.
   - The `View` must be non-transparent.
2. All required assets must meet the following conditions:
   - The asset, as well as its `superview`, must be fully visible.
   - The asset must be completely inside the ad `View`.
   - The asset must be within the hierarchy of the ad `View`.
   - The asset must be non-transparent.
   - The content of the asset must be truthful.
3. The app must be active (not running in the background).
4. At a single point in time, a loaded ad can only be served in one `View`. Simultaneous display of a single ad in multiple `Views` may result in a lost impression.

Note: If all the conditions are met but the impression was not registered, there will be another attempt to register the impression later.

Restrictions on changing assets

1. Don't edit the text content of assets.
2. Don't edit the content of images.
3. If you stretch an image, you must maintain the aspect ratio.
4. Don't crop an image by more than 20%. Permitted: masking, container resizing.

Native ad assets

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

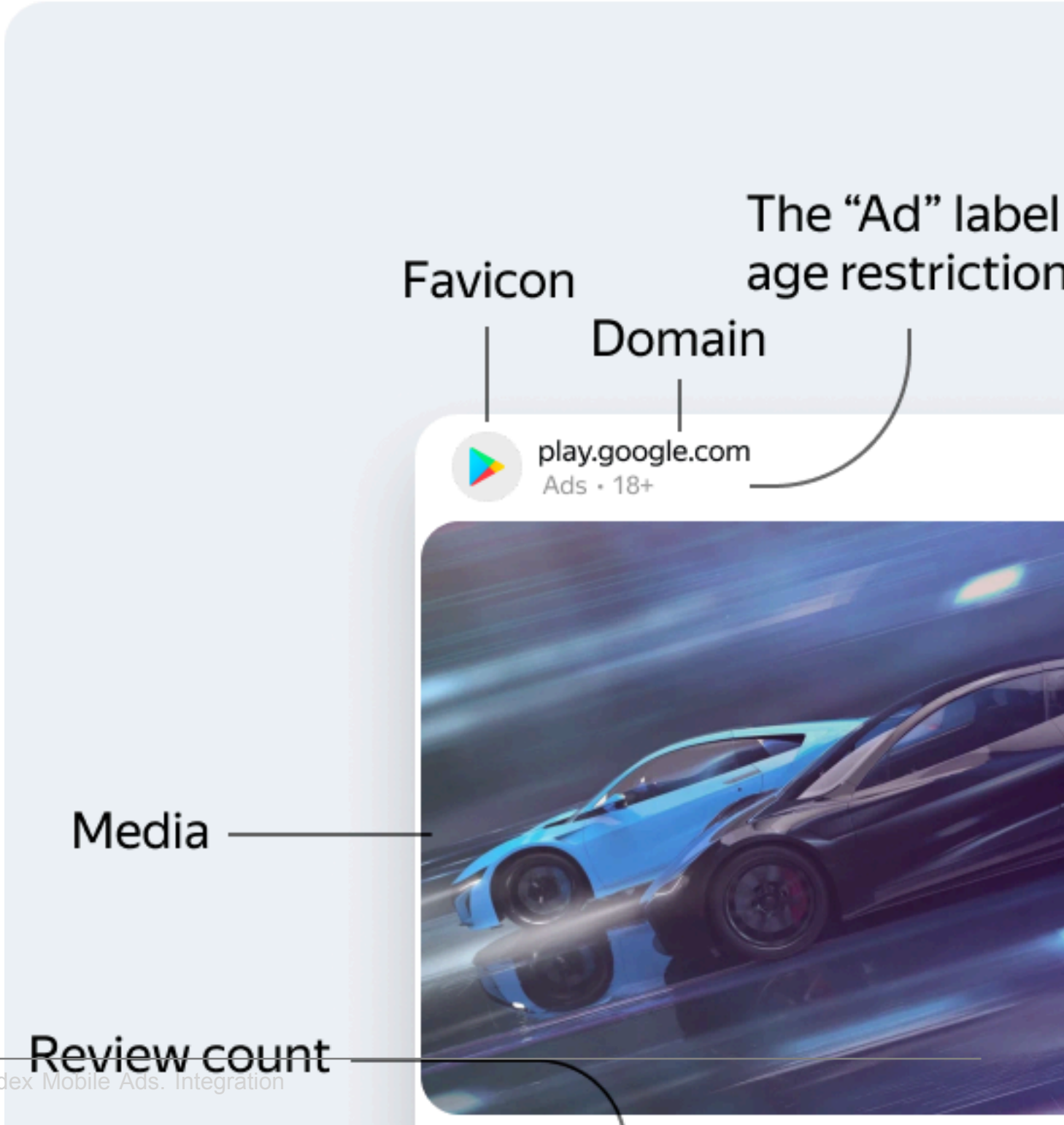
Note:

Read the advertising requirements.

Asset list

The library includes both the required and optional assets. According to the advertising rules, it's enough that you render only the required assets. In practice, layouts that use a complete set of assets are more clickable. That's why we recommend that you use the design based on a complete set of supported assets.

App ads


The “Ad” label
age restriction
Favicon
Domain
play.google.com
Ads · 18+
Media
Review count

Ad element	Asset	Type	Required
Title	titleLabel	UILabel	Yes
Domain	domainLabel	UILabel	Yes
Disclamer	warningLabel	UILabel	Yes
The “Ad” label and the age restriction label	sponsoredLabel	UILabel	Yes
Menu icon	feedbackButton	UIButton	Yes
Action button	callToActionButton	UIButton	Yes
Media	mediaView	YMANativeMediaView	Yes
App icon	iconImageView	UIImageView	Yes, for Ads for Mobile Apps
Price	priceLabel	UILabel	Yes, for Ads for Mobile Apps
Favicon	faviconImageView	UIImageView	No
Review count	reviewCountLabel	UILabel	No
Rating	ratingView	UIView<YMARating>	No
Text	bodyLabel	UILabel	No

The list of required assets defines the set of data for which, if those assets are present, a View must be provided for rendering.

Tip:

We recommend that you use a layout that can render a complete set of both required and optional ad assets.

Asset layout

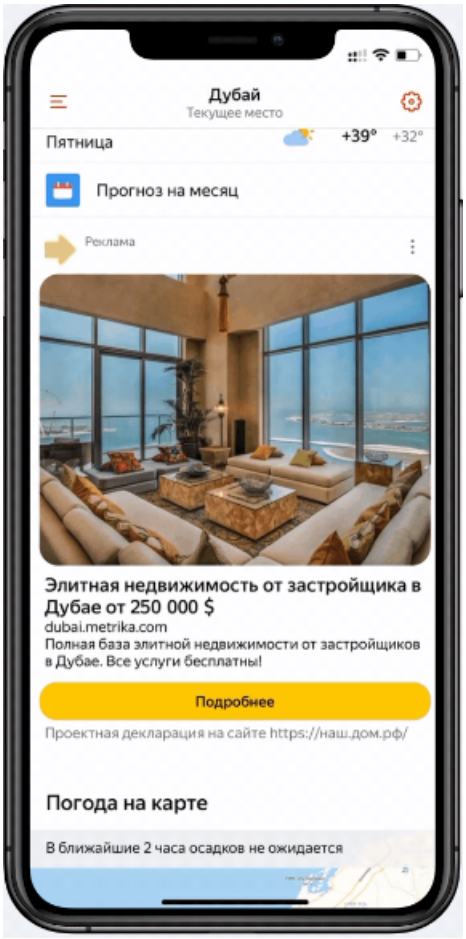
feedback

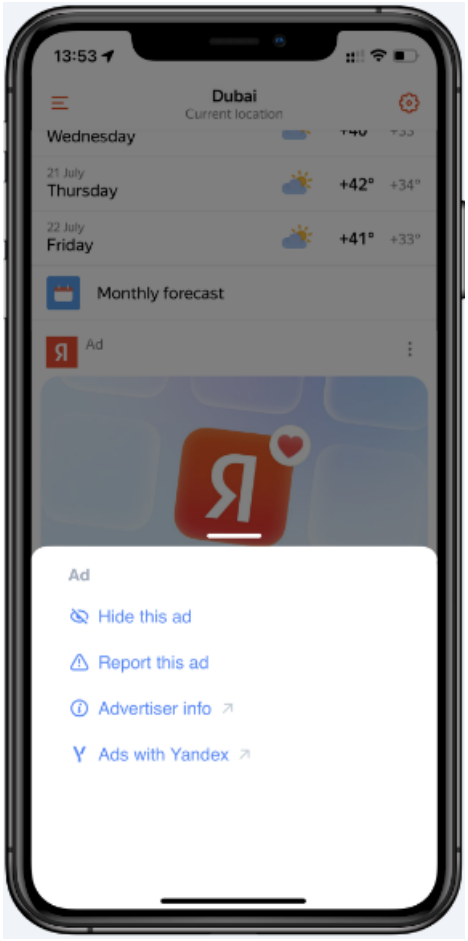
Menu icon. Required asset. The user can click the menu icon to hide or report an ad.

The menu icon is added to the upper-right corner of the ad.

Note:

The developer must define what to do with the ad after the reason for closing it is chosen (for example, hide the ad or show a text). If there is no further action defined, the SDK will register the reason for closing, but the ad will not be hidden.





Layout options:

feedback_dark_dots_with_background

White menu icon, with dark dots and a translucent background. Default value.

feedback_light_dots

Menu icon without a background, with light dots.

Code example:

```
YMANativeAdLoader *adLoader = [[YMANativeAdLoader alloc] init];
YMAMutableNativeAdRequestConfiguration *requestConfiguration =
            [[YMAMutableNativeAdRequestConfiguration alloc] initWithAdUnitID:@"demo-native-content-yandex"];
requestConfiguration.parameters = @{ @"feedback_image": "feedback_light_dots" };
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
```

feedback_dark_dots

Menu icon without a background, with dark dots.

Code example:

```
YMANativeAdLoader *adLoader = [[YMANativeAdLoader alloc] init];
YMAMutableNativeAdRequestConfiguration *requestConfiguration =
            [[YMAMutableNativeAdRequestConfiguration alloc] initWithAdUnitID:@"demo-native-content-yandex"];
requestConfiguration.parameters = @{ @"feedback_image": "feedback_dark_dots" };
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
```

media

`Subview` for media content (image or video).

How media content is displayed in mediaView: if the response to the ad request contains media content, mediaView displays it after buffering.

Tip:

1. To check for the presence of media content, use the media property of the YMANativeAdAssets object. If the response contains media content, the property returns a nonzero YMANativeAdMedia object.
2. To check for the presence of an image, use the image property of the YMANativeAdAssets object. If the response to the ad request contains an image, the property returns a nonzero YMANativeAdImage object.

Loading ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Loading ads

1. Create an instance of the YMANativeAdLoader class to get native ads.
2. Create a configuration for the `nativeAdRequestConfiguration` request using the YMANativeAdRequestConfiguration class. As the request parameters, you can use the ad unit ID, method for loading images, age, gender, and other data that might improve the quality of ad selection.
3. Set a delegate for retrieving an ad that implements the YMANativeAdLoaderDelegate protocol:
4. To track the ad loading process, implement the YMANativeAdLoaderDelegate protocol methods: -nativeAdLoader:didFailLoadingWithError: and -nativeAdLoader:didLoadAd:.
5. To load the ad, send the `loadAdWithRequestConfiguration:` message to the loader.

   Swift

   ```
   adLoader.loadAd(with: requestConfiguration)
   ```

   Objective-C

   ```
   [self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
   ```

6. If the ad loaded, the following method is called:

   Swift

   ```
   func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd)
   ```

   Objective-C

   ```
   - (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
   ```

7. If the ad didn't load, the following method is called:

   Swift

   ```
   func nativeAdLoader(_ loader: YMANativeAdLoader, didFailLoadingWithError error: Error)
   ```

   Objective-C

   ```
   - (void)nativeAdLoader:(YMANativeAdLoader *)loader didFailLoadingWithError:(NSError *)error
   ```

   For more information about possible errors, see YMANativeErrorCode.

General ad request

Swift

```
// Creating a loader
adLoader = YMANativeAdLoader()
adLoader.delegate = self

// Creating a request configuration
let requestConfiguration = YMANativeAdRequestConfiguration(adUnitID: "<AdUnitID>")

// Passing the request configuration to the loader
adLoader.loadAd(with: requestConfiguration)

// Implementing delegate methods
....

func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd) {
```

```
    // Render the ad
}
```

Objective-C

```
// Creating a loader self.adLoader = [[YMANativeAdLoader alloc] init];
self.adLoader.delegate = self;

// Creating a request configuration
YMANativeAdRequestConfiguration *requestConfiguration =
        [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:@"your_AdUnitID"];

// Passing the request configuration to the loader
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];

// Implementing delegate methods
....
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
    // Render the ad
}
```

Examples with a demo AdUnitId

To see how the ad will be displayed in the app, use a demo AdUnitId:

- For App Install ads: demo-native-app-yandex.
- For Content ads: demo-native-content-yandex.
- For video ads: demo-native-video-yandex.

Loading multiple ads

Yandex Mobile Ads SDK provides the option to load multiple ads in a single request (up to nine ads).

1. Create an instance of the YMANativeBulkAdLoader class to get native ads.
2. Create a configuration for the `nativeAdRequestConfiguration` request using the YMANativeAdRequestConfiguration class. As the request parameters, you can use the ad unit ID, method for loading images, age, gender characteristics, and other data that might improve the quality of ad selection.
3. Set a delegate for retrieving an ad that implements the YMANativeBulkAdLoaderDelegate protocol.
4. To track the ad loading process, implement the YMANativeBulkAdLoaderDelegate protocol methods: -nativeBulkAdLoader:didLoadAds: and -nativeBulkAdLoader:didFailLoadingWithError:.
5. Send to the loader the request configuration and count of ads requested (the `adsCount` parameter).

Swift

```
adLoader.loadAds(with: requestConfiguration, adsCount: adsCount)
```

Objective-C

```
[self.adLoader loadAdsWithRequestConfiguration:requestConfiguration adsCount:adsCount];
```

Swift

```
// Creating a request configuration
let requestConfiguration = YMAMutableNativeAdRequestConfiguration(adUnitID: AdUnitID)

// Creating a loader
adLoader = YMANativeBulkAdLoader()
adLoader.delegate = self

// Passing the request configuration and the number of requested ads to the loader
adLoader.loadAds(with: requestConfiguration, adsCount: adsCount)

// Implementing delegate methods

func nativeBulkAdLoader(_ nativeBulkAdLoader: YMANativeBulkAdLoader, didLoad ads: [YMANativeAd]) {
    ...
    // Working with each id<YMANativeAd> object separately. For more information, see "Customizing the ad
 design".
}
```

Objective-C

```
// Creating a request configuration
YMAMNativeAdRequestConfiguration *requestConfiguration =
    [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:AdUnitID];

// Creating a loader
```

```
self.adLoader = [[YMANativeBulkAdLoader alloc] init];
self.adLoader.delegate = self;

// Passing the request configuration and count of requested ads to the loader
[self.adLoader loadAdsWithRequestConfiguration:requestConfiguration adsCount:adsCount];

// Implementing delegate methods

- (void)nativeBulkAdLoader:(YMANativeBulkAdLoader *)nativeBulkAdLoader didLoadAds:(NSArray<idYMANativeAd> *)ads
{
    ...
    // Working with each id<YMANativeAd> object separately. For more information, see "Customizing the ad
 design".
}
```

Note:

Yandex Mobile Ads SDK doesn't guarantee that the requested number of ads will be loaded. The resulting array will contain from 0 to adsCount NativeAd objects. You can render all the received ad objects separately from each other using the above methods for laying out native ads.

Ways to load images

Automatic loading

If the app simultaneously stores links to just one or a small number of ads, we recommend using automatic loading.

When creating the configuration, set shouldLoadImagesAutomatically to YES:

Swift

```
let requestConfiguration = YMAMutableNativeAdRequestConfiguration(adUnitID: AdUnitID)
requestConfiguration.shouldLoadImagesAutomatically = true
```

Objective-C

```
YMAMutableNativeAdRequestConfiguration *requestConfiguration =
    [[YMAMutableNativeAdRequestConfiguration alloc] initWithAdUnitID:AdUnitID];
    requestConfiguration.shouldLoadImagesAutomatically = YES;
```

The resulting ad will have the images ready. They are stored in the device memory until the ad is destroyed.

Manual loading

If the app simultaneously stores links to a large number of ads, we recommend using manual image loading.

When creating the configuration, set shouldLoadImagesAutomatically to NO:

Swift

```
let requestConfiguration = YMAMutableNativeAdRequestConfiguration(adUnitID: AdUnitID)
requestConfiguration.shouldLoadImagesAutomatically = false
```

Objective-C

```
YMAMutableNativeAdRequestConfiguration *requestConfiguration =
    [[YMAMutableNativeAdRequestConfiguration alloc] initWithAdUnitID:AdUnitID];
    requestConfiguration.shouldLoadImagesAutomatically = NO;
```

The resulting ad will only have the image sizes. To load images in the ad, call the loadImages method on the resulting ad.

Attention:

All images are cached, but they can be deleted at any time, so you need to call the loadImages() method before every ad impression.

Swift

```
func showAd() {
    // Show ad: custom native view or template
    view.addSubview(adView)
    adView.ad?.loadImages()
}
```

Objective-C

```
- (void)showAd
{
    // Show ad: custom native view or template
    [self.view addSubview:self.adView];
    [self.adView.ad loadImages];
```

```
}
```

Notifications about image loading

Restriction: You can only get notifications when loading images manually.

Enabling notifications

To enable notifications that are sent when an image is loaded, call the -addImageLoadingObserver: method.

Swift

```
ad?.add(self)

...
func nativeAdDidFinishLoadingImages(_ ad: YMANativeAd) {
    print("Finished loading images")
}
```

Objective-C

```
[ad addImageLoadingObserver:self];
...
- (void)nativeAdDidFinishLoadingImages:(id<YMANativeAd>)ad
{
    NSLog(@"Finished loading images");
}
```

Disabling notifications

To disable notifications that are sent when an image is loaded, call the -removeImageLoadingObserver: method.

Swift

```
ad?.remove(self)
```

Objective-C

```
[ad removeImageLoadingObserver:self];
```

Ad slider

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Yandex Mobile Ads SDK lets you render a slider containing related ads. For more information about the slider, see this post.

The slider implements the native advertising paradigm. Publishers can adjust the ad layout based on its features and the design of their apps that render the slider.

Loading the slider

1. Create an instance of the YMANativeAdLoader class to fetch ads in the slider.
2. Create a configuration for the `nativeAdRequestConfiguration` request using the YMANativeAdRequestConfiguration class. As the request parameters, you can use the ad unit ID, method for loading images, age, gender, and other data that might improve the quality of ad selection.
3. Set a delegate for retrieving an ad that implements the YMANativeAdLoaderDelegate protocol:
4. To track the ad loading process, implement the YMANativeAdLoaderDelegate protocol methods: -nativeAdLoader:didFailLoadingWithError: and -nativeAdLoader:didLoadAd:.
5. To load the ad, send the `loadAdWithRequestConfiguration:` message to the loader.

Swift

```
adLoader.loadAd(with: requestConfiguration)
```

Objective-C

```
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
```

6. If the ad loaded, the following method is called:

   Swift

   ```
   func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd)
   ```

   Objective-C

   ```
   - (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
   ```

7. If the ad didn't load, the following method is called:

   Swift

   ```
   func nativeAdLoader(_ loader: YMANativeAdLoader, didFailLoadingWithError error: Error)
   ```

   Objective-C

   ```
   - (void)nativeAdLoader:(YMANativeAdLoader *)loader didFailLoadingWithError:(NSError *)error
   ```

   For more information about possible errors, see YMANativeErrorCode.

General ad request

Swift

```
// Creating a loader
adLoader = YMANativeAdLoader()
adLoader.delegate = self

// Creating a request configuration
let requestConfiguration = YMANativeAdRequestConfiguration(adUnitID: "<AdUnitID>")

// Passing the request configuration to the loader
adLoader.loadAd(with: requestConfiguration)

// Implementing delegate methods
....

func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd) {
    // Render the ad
}
```

Objective-C

```
// Creating a loader self.adLoader = [[YMANativeAdLoader alloc] init];
self.adLoader.delegate = self;

// Creating a request configuration
YMANativeAdRequestConfiguration *requestConfiguration =
        [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:@"your_AdUnitID"];

// Passing the request configuration to the loader
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];

// Implementing delegate methods
....
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
     // Render the ad
}
```

Displaying the slider

When the slider is loaded, you must render all its assets.

A successfully loaded slider contains one or more native ads with the same context. Ads in the slider must be displayed within the same shared container: otherwise, ad impressions won't count.

Layout without a template

When the template settings aren't enough to get the desired effect, you can configure native ads manually.

With this method, you can lay out native ads yourself by positioning ad elements in respect of each other. Your ad may contain both mandatory and optional display assets. You can find their full list in Native ad assets.

Tip:

We recommend that you use a layout that includes the complete set of possible assets. Experience has shown that layouts with a complete set of assets are more clickable.

Call the bindAdToSliderView method and pass a container for the ad slider to it.

Each ad in the slider is laid out using a standard native advertising layout method.

Swift

```
    func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd) {
        self.ad = ad
        ad.delegate = self
        // Проверяем наличие вложенных объявлений
        if ad.ads.count != 0 {
            // Создаем контейнер для слайдера; Вместо YMANativeAdView должен быть ваш наследник этого класса
            let sliderAdView = YMANativeAdView()

            // Вызываем метод bindAd(toSliderView: _) и передаем в него контейнер
            do {
                try ad.bindAd(toSliderView: sliderAdView)
            } catch {
                // Проверяем сообщение об ошибке и исправляем проблему
                return
            }

            for subAd in ad.ads {
                // Подписываемся на делегат
                subAd.delegate = self

                // Создаем рекламное view для объявления
                // Вместо YMANativeAdView должен быть ваш наследник этого класса
                let subAdView = YMANativeAdView()

                // Вызываем метод bind(with: subAdView) для объявления
                do {
                    try subAd.bind(with: subAdView)
                } catch {
                    // Проверяем сообщение об ошибке и исправляем проблему
                    return
                }

                // Добавляем объявление в контейнер
                sliderAdView.addSubview(subAdView)
                // Располагаем объявление в контейнере
            }

        } else {
            // Обработать как обычную нативную рекламу
            // https://yandex.ru/dev/mobile-ads/doc/ios/quick-start/config.html
        }
    }
```

Objective-C

```
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
    self.ad = ad;
    self.ad.delegate = self;
    // Проверяем наличие вложенных объявлений
    if (ad.ads.count != 0) {
        // Создаем контейнер для слайдера
        YMANativeAdView *sliderAdView = [[YMANativeAdView alloc] init];
        // Вызываем метод bindAdToSliderView и передаем в него контейнер
        NSError *error = nil;
        [ad bindAdToSliderView:sliderAdView error:&error];

        // Проверяем успешность привязки
        if (error != nil) {
            // Проверяем сообщение об ошибке и исправляем проблему
            return;
        }

        for (id<YMANativeAd> subAd in ad.ads) {
            // Подписываемся на делегат
            subAd.delegate = self;
            // Создаем рекламное view для объявления
            // Вместо YMANativeAdView должен быть ваш наследник этого класса
            YMANativeAdView *subAdView = [[YMANativeAdView alloc] init];
            NSError *error = nil;
            // Вызываем метод bindWithAdView и передаем в него рекламное view
            [subAd bindWithAdView:subAdView error:&error];

            // Проверяем успешность привязки
            if (error != nil) {
                // Проверяем сообщение об ошибке и исправляем проблему
                continue;
            }

            // Добавляем объявление в контейнер
            [sliderAdView addSubview:subAdView];
            // Располагаем объявление в контейнере
```

```
        }

    } else {
        // Обработать как обычную нативную рекламу
        // https://yandex.ru/dev/mobile-ads/doc/ios/quick-start/config.html
    }
}
```

Configuring the ad design

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

There are two ways to configure the layout of an ad:

1. Layout using a template.

   The easiest way to work with native ads is to use a standard template for layout. The template already has the set of required assets, and defines their arrangement relative to each other. The template works with any supported type of ad.
2. Layout without a template.

   Create a layout without using a template when the template settings don't give you enough configuration options to make the ad look organic and match the design of the app.

Layout using a template

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

To customize the layout and design, you can use a standard template or create your own design based on a standard template.

Using the standard layout template

Note:

After setting up the layout, set the position and ad size relative to the device screen.

1. Create an instance of the YMANativeBannerView class and set the loaded ad for it:

   Swift

   ```
   let bannerView = YMANativeBannerView()
   bannerView.ad = ad
   view.addSubview(bannerView)
   ```

   Objective-C

   ```
   YMANativeBannerView *bannerView = [[YMANativeBannerView alloc] init];
   bannerView.ad = ad;
   [self.view addSubview:bannerView];
   ```
2. To receive notifications about user interactions with the ad (opening the ad or exiting the app), assign it the YMANativeAdDelegate, which implements the methods:

   - -nativeAd:didDismissScreen:
   - -nativeAd:willPresentScreen:
   - -nativeAdWillLeaveApplication:
   - -viewControllerForPresentingModalView

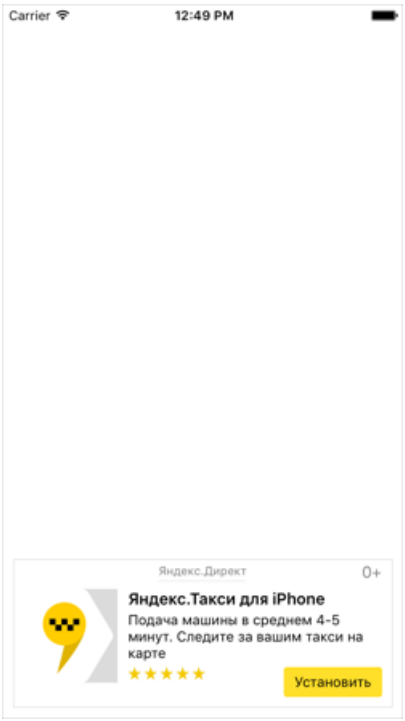
   Swift

   ```
   ad.delegate = self
   ```

   Objective-C

   ```
   ad.delegate = self;
   ```

3. Example of using the standard layout template:



Note: If the standard layout doesn't fit your app, you can change it. For more information, see Creating your own template-based layout.

Creating your own template-based layout

Note:

After setting up the layout, set the position and ad size relative to the device screen.

1. Create an instance of the YMANativeBannerView class and set the loaded ad for it:

   Swift

   ```
   let bannerView = YMANativeBannerView()
   bannerView.ad = ad
   view.addSubview(bannerView)
   ```

   Objective-C

   ```
   YMANativeBannerView *bannerView = [[YMANativeBannerView alloc] init];
   bannerView.ad = ad;
   [self.view addSubview:bannerView];
   ```

2. To receive notifications about user interactions with the ad (opening the ad or exiting the app), assign it the YMANativeAdDelegate, which implements the methods:

   - -nativeAd:didDismissScreen:
   - -nativeAd:willPresentScreen:
   - -nativeAdWillLeaveApplication:
   - -viewControllerForPresentingModalView

   Swift

   ```
   ad.delegate = self
   ```

   Objective-C

   ```
   ad.delegate = self;
   ```

3. Request the settings for the standard layout template:

Swift

```
let appearance = YMAMutableNativeTemplateAppearance.default()
```

Objective-C

```
YMAMutableNativeTemplateAppearance *appearance = [[YMANativeTemplateAppearance defaultAppearance]
 mutableCopy];
```

4. Set the preferred settings.
5. To apply the settings to the template, call the -applyAppearance: method:

Swift

```
bannerView.apply(appearance)
```

Objective-C

```
[bannerView applyAppearance:appearance];
```

Example of layout configuration

Swift

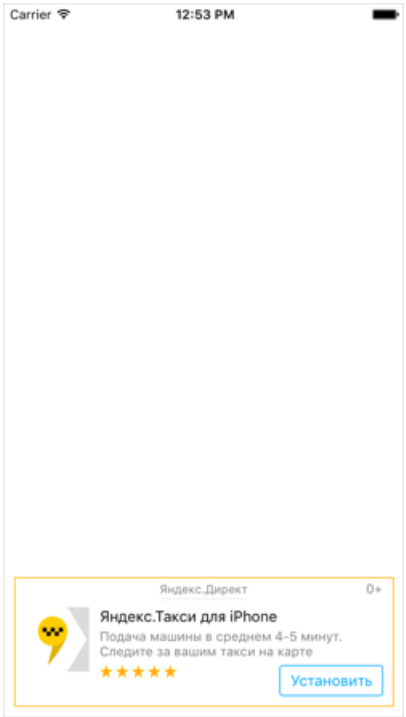
```
// Defining a custom color scheme.
let orangeColor = UIColor(red: 1, green: 176.0/255, blue: 32.0/255, alpha: 1)
let blueColor = UIColor(red: 0, green: 170.0/255, blue: 1, alpha: 1)

// Creating a copy with the settings of the native design template.
let appearance = YMAMutableNativeTemplateAppearance.default()

// Starting to change the native template settings.

// Setting the color for the ad frame.
appearance.borderColor = orangeColor

// Creating a copy with rating settings.
let ratingAppearance = appearance.ratingAppearance?.mutableCopy() as? YMAMutableRatingAppearance

// Setting the color for filled stars in the rating.
ratingAppearance?.filledStarColor = orangeColor
appearance.ratingAppearance = ratingAppearance

// Setting the font color and size for the action button label.
let callToActionTextAppearance = YMALabelAppearance(font: .systemFont(ofSize: 14), textColor: blueColor)

// Setting the button color for the normal state and clicked state, along with the color and thickness of the
 button border.
let callToActionAppearance = YMAButtonAppearance(
    textAppearance: callToActionTextAppearance,
    normalColor: .clear,
    highlightedColor: .gray,
    borderColor: blueColor,
    borderWidth: 1
)
appearance.callToActionAppearance = callToActionAppearance

// Setting the font size and color for the age restriction label.
appearance.ageAppearance = YMALabelAppearance(font: .systemFont(ofSize: 12), textColor: .gray)

// Setting the font size and color for the ad title.
appearance.titleAppearance = YMALabelAppearance(font: .systemFont(ofSize: 14), textColor: .black)

// Setting the font size and color for the main ad text.
appearance.bodyAppearance = YMALabelAppearance(font: .systemFont(ofSize: 12), textColor: .gray)

// Setting the image width and the sizing rule.
let imageConstraint = YMASizeConstraint(type: .fixed, value: 60)

// Applying the settings to the image.
appearance.imageAppearance = YMAImageAppearance(widthConstraint: imageConstraint)
```

Objective-C

```
// Define custom colors.
UIColor *orangeColor =
  [UIColor colorWithRed:255.f / 255.f green:176.f / 255.f blue:32.f / 255.f alpha:1.f];
UIColor *blueColor =
  [UIColor colorWithRed:0.f / 255.f green:170.f / 255.f blue:255.f / 255.f alpha:1.f];

// Creating a copy with the settings of the standard layout template.
YMAMutableNativeTemplateAppearance *appearance = [[YMANativeTemplateAppearance defaultAppearance] mutableCopy];
```

```
// Starting to change the standard template settings.

// Setting the color for the ad frame.
appearance.borderColor = orangeColor;

// Creating a copy with rating settings.
YMAMutableRatingAppearance *ratingAppearance = [appearance.ratingAppearance mutableCopy];

// Setting the color for filled stars in the rating.
ratingAppearance.filledStarColor = orangeColor;
appearance.ratingAppearance = ratingAppearance;

// Setting the font color and size for the action button label.
YMALabelAppearance *callToActionTextAppearance =
    [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:14.f]
                                 textColor:blueColor];

// Setting the button color for the normal state and the clicked state, along with the color and thickness of
 the button border.
YMAButtonAppearance *callToActionAppearance =
    [YMAButtonAppearance appearanceWithTextAppearance:callToActionTextAppearance
                                          normalColor:[UIColor clearColor]
                                     highlightedColor:[UIColor grayColor]
                                          borderColor:blueColor
                                          borderWidth:1.f];
appearance.callToActionAppearance = callToActionAppearance;

// Setting the font size and color for the age restriction label.
appearance.ageAppearance =
  [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:12.f]
                               textColor:[UIColor grayColor]];

// Setting the font size and color for the ad title.
appearance.titleAppearance =
  [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:14.f]
                               textColor:[UIColor blackColor]];

// Setting the font size and color for the main text of the ad.
appearance.bodyAppearance =
  [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:12.f]
                               textColor:[UIColor grayColor]];

// Setting the image width and sizing constraint.
YMASizeConstraint *imageConstraint =
  [YMASizeConstraint constraintWithType:YMASizeConstraintTypeFixed value:60.f];

// Applying the settings to the image.
appearance.imageAppearance =
  [YMAImageAppearance appearanceWithWidthConstraint:imageConstraint];
```

We get our custom design based on the template:



Setting the position and the ad size

Restriction: mediaView size requirements when displaying video ads

Minimum size of an instance of the YMANativeMediaView class, which supports video playback: 300x160 or 160x300.

To support video playback in native ad templates, we recommend setting the width for YMANativeBannerView to at least 300. The correct height for `mediaView` will be calculated automatically based on the width to height ratio.

There are two ways to control size and location:

1. Using the system's `AutoLayout` mechanism.

   Note:

   When using `AutoLayout`, set `constraint` for the `UIView` width. The height is determined automatically based on the specified width.
2. By manually setting all the sizes.

Setting sizes manually

Note:

We don't recommend setting an object width greater than 420 logical pixels.

Set the width and height for the `YMANativeBannerView` object. The height is determined using the +heightWithAd:width:appearance: method which must be passed the ad, the width, and the `YMANativeTemplateAppearance` object (used for setting the ad's appearance).

Swift

```
// Setting the margins on the left and right relative to the screen.
let inset: CGFloat = 50

// Setting the width.
let width = view.frame.width - 2 * inset

// Setting the height.
let height = YMANativeBannerView.height(with: ad, width: width, appearance: nil)

// Setting the vertical position.
let y = view.frame.maxY - height - inset

// Setting the coordinates and sizes for the frame.
bannerView.frame = CGRect(x: inset, y: y, width: width, height: height)
```

Objective-C

```
// Set the margins on the left and right relative to the screen.
CGFloat inset = 50.f;

// Setting the width.
CGFloat width = CGRectGetWidth(self.view.frame) - 2 * inset;

// Setting the height.
CGFloat height = [YMANativeBannerView heightWithAd:ad width:width appearance:nil];

// Setting the vertical position.
CGFloat y = CGRectGetMaxY(self.view.frame) - height - inset;

//  Setting the coordinates and sizes for the frame.
bannerView.frame = CGRectMake(inset, y, width, height);
```

Layout without a template

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

To display native ads, `YMANativeAd` must be bound to a specific `View`. This `View` must be defined as the root element for the `subview` set that the data contained in the native ad will be bound to.

Restriction: mediaView size requirements when displaying video ads

Minimum size of an instance of the YMANativeMediaView class, which supports video playback: 300x160 or 160x300.

To support video playback in native ads, we recommend setting the width for `mediaView` to at least 300. To calculate the appropriate `mediaView` height value, use the aspectRatio property value.

Layout of native ads

1. See the list of required and optional subviews for native ads.
2. Set the values for all required subviews. You can do this using `nib` or directly in the code. Create an instance of YMANativeAdView and define the `subview` values in `initWithFrame:`

Example of registering a set of `subviews`:

Swift

```
override init(frame: CGRect) {
    super.init(frame: frame)
    let titleLabel = createLabel()
    let bodyLabel = createLabel()
    let ageLabel = createLabel()
    let warningLabel = createLabel()
    let sponsoredByLabel = createLabel()
    let priceLabel = createLabel()
    let starRatingView = createStarRatingView()
    let button = createButton()
    let iconImageView = createIconAssetImageView()
    let mediaView = createMediaAssetView()
    addSubview(titleLabel)
    addSubview(bodyLabel)
    addSubview(ageLabel)
    addSubview(warningLabel)
    addSubview(sponsoredByLabel)
    addSubview(priceLabel)
    addSubview(starRatingView)
    addSubview(button)
    addSubview(iconImageView)
    addSubview(mediaView)
    self.titleLabel = titleLabel
    self.bodyLabel = bodyLabel
    self.ageLabel = ageLabel
    self.warningLabel = warningLabel
    self.sponsoredLabel = sponsoredByLabel
    self.callToActionButton = callToActionButton
    self.priceLabel = priceLabel
    self.ratingView = ratingView
    self.iconImageView = iconImageView
    self.mediaView = mediaView
}
```

Objective-C

```
- (instancetype)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self != nil) {
        UILabel *titleLabel = [self label];
        UILabel *bodyLabel = [self label];
        UILabel *ageLabel = [self secondaryLabel];
        UILabel *warningLabel = [self label];
        UILabel *sponsoredByLabel = [self label];
        UILabel *priceLabel = [self label];
        StarRatingView *ratingView = [self starRatingView];
        UIButton *callToActionButton = [self button];
        UIImageView *iconImageView = [self iconAssetImageView];
        YMANativeMediaView *mediaView = [self mediaAssetView];
        [self addSubview:titleLabel];
        [self addSubview:bodyLabel];
        [self addSubview:ageLabel];
        [self addSubview:warningLabel];
        [self addSubview:sponsoredByLabel];
        [self addSubview:callToActionButton];
        [self addSubview:priceLabel];
        [self addSubview:ratingView];
        [self addSubview:iconImageView];
        [self addSubview:mediaView];
        self.titleLabel = titleLabel;
        self.bodyLabel = bodyLabel;
        self.ageLabel = ageLabel;
        self.warningLabel = warningLabel;
        self.sponsoredLabel = sponsoredByLabel;
        self.callToActionButton = callToActionButton;
        self.priceLabel = priceLabel;
        self.ratingView = ratingView;
        self.iconImageView = iconImageView;
        self.mediaView = mediaView;
    }
    return self;
}
```

3. To receive notifications about user interactions with the ad (opening or closing the ad or exiting the app), assign it the YMANativeAdDelegate, which implements the methods:
   - -closeNativeAd:;
   - -nativeAd:didDismissScreen:
   - -nativeAd:willPresentScreen:
   - -nativeAdWillLeaveApplication:
   - -viewControllerForPresentingModalView
4. Request the values for native ad assets using the -adAssets method. This will help you calculate the position and sizes of these assets in advance.

   Swift

   ```
   let assets = ad.adAssets()
   ```

   Objective-C

   ```
   YMANativeAdAssets *assets = [ad adAssets];
   ```

   Example of getting the image size and aspect ratio of the ad title text and media

   Swift

   ```
   let image = assets.image
   let title = assets.title
   let media = assets.media
   print("Image size: \(image?.size ?? .zero)")
   print("Title: \(title ?? "")")
   print(String(format: "Media aspect ratio: %.2f", media?.aspectRatio ?? 0))
   ```

   Objective-C

   ```
   YMANativeAdImage *image = assets.image;
   NSString *title = assets.title;
   YMANativeAdMedia *media = assets.media;
   NSLog(@"Image size: %@", NSStringFromCGSize(image.size));
   NSLog(@"Title: %@", title);
   NSLog(@"Media aspect ratio: %.2f", media.aspectRatio);
   ```

5. Call the -bindWithAdView:error: method to bind the content to the native ad object.

   Swift

   ```
   // ...
   adView = YMANativeAdView(frame: frame)
   //configure content ad view
   // ...
   func nativeAdLoader(_ loader: YMANativeAdLoader, didLoad ad: YMANativeAd) {
       ad.delegate = self
       do {
           try ad.bind(with: contentAdView)
       } catch {
           print("Error: \(error)")
       }
   }
   ```

   Objective-C

   ```
   // ...
   self.adView = [[YMANativeAdView alloc] initWithFrame:frame];
   //configure content ad view
   // ...
   - (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
   {
       ad.delegate = self;
       NSError *error = nil;
       BOOL result = [ad bindWithAdView:self.contentAdView error:error];
       if (error != nil) {
           NSLog(@"Error: %@", error);
       }
   }
   ```

   Note:

   If a required element of a native ad has the corresponding `YMANativeAdView` property set to `nil`, binding doesn't occur and the ad isn't displayed. See details in the `error`.

Debugging

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Logging

You can optionally enable logging by using the +enableLogging method. If an impression wasn't registered, a message appears in the console.

Invalid integration indicator for native ads

If an error was made when integrating native ads, an indicator appears on top of the ad in simulator mode. Click on the indicator to see a message with debugging information to help you understand the reason for the error. Click the indicator again to hide the message.

Attention:

By default, the indicator is only shown in simulator mode (device type `YMADeviceTypeSimulator`). You can view device types in YMADeviceType.

To enable the indicator for real devices as well, pass the value `YMADeviceTypeHardware | YMADeviceTypeSimulator` in the enableVisibilityErrorIndicatorFordevicetype: method:

Swift

```
YMAMobileAds.enableVisibilityErrorIndicator(for: [.hardware, .simulator])
```

Objective-C

```
[YMAMobileAds enableVisibilityErrorIndicatorForDeviceType:YMADeviceTypeHardware | YMADeviceTypeSimulator]
```

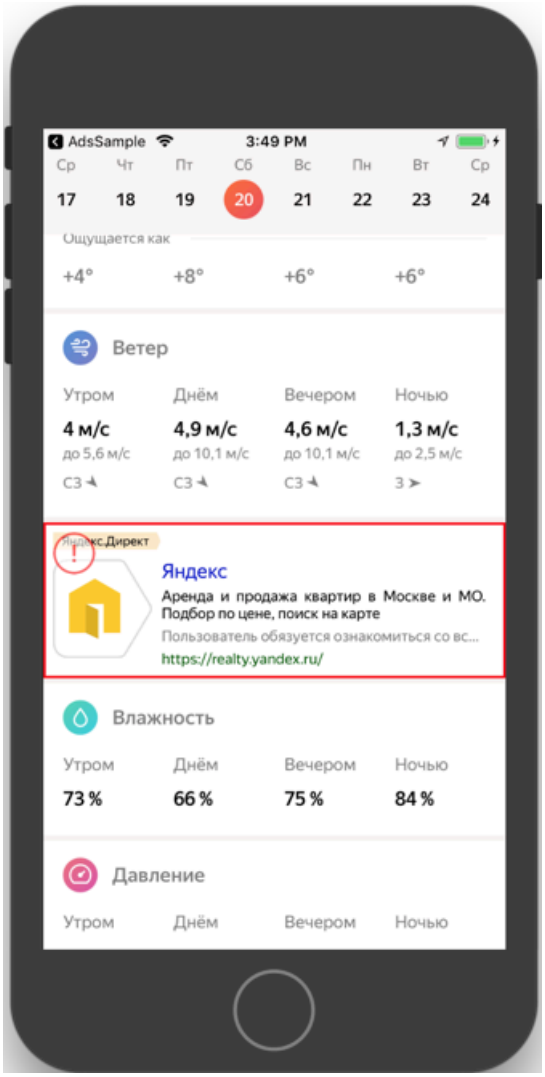
To turn off the indicator, pass the value `YMADeviceTypeNone` in the enableVisibilityErrorIndicatorForDeviceType: method:

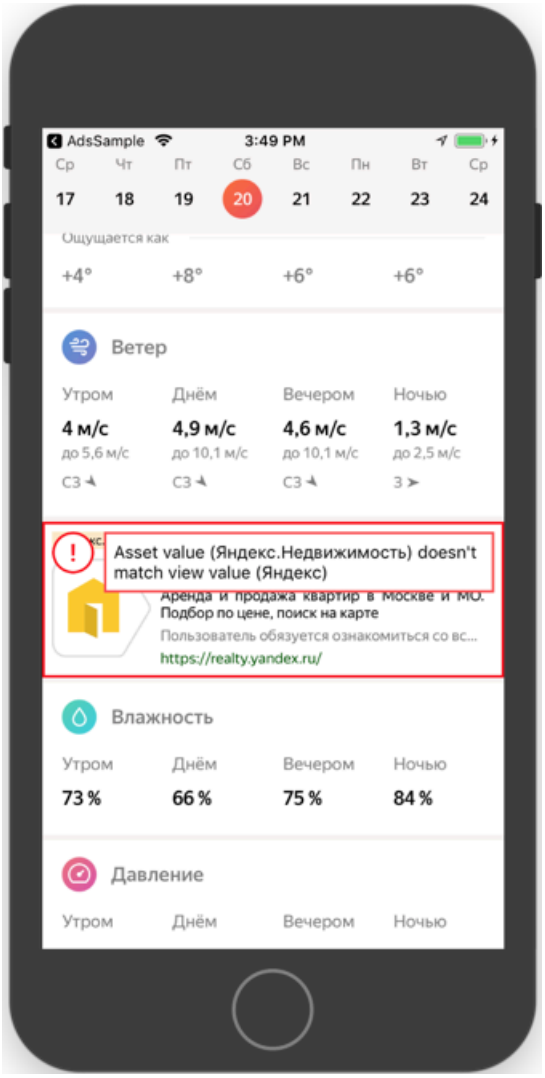
Swift

```
YMAMobileAds.enableVisibilityErrorIndicator(for: [])
```

Objective-C

```
[YMAMobileAds enableVisibilityErrorIndicatorForDeviceType:YMADeviceTypeNone]
```





Classes and protocols for working with native ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Classes

- YMAButtonAppearance
- YMAImageAppearance
- YMALabelAppearance
- YMANativeAdAssets
- YMANativeAdImage
- YMANativeAdLoader
- YMANativeAdMedia
- YMANativeAdView
- YMANativeBannerView
- YMANativeMediaView
- YMANativeTemplateAppearance
- YMARatingAppearance
- YMASizeConstraint

Protocols

- YMANativeAd
- YMANativeAdDelegate
- YMANativeAdImageLoadingObserver
- YMANativeAdLoaderDelegate
- YMARating

InStream ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

InStream is an ad format that lets you monetize your app by serving ads while video content is playing.

An InStream ad consists of a scenario with multiple video units. The type of video unit in an InStream scenario determines how the video ad should be played relative to the main video content.

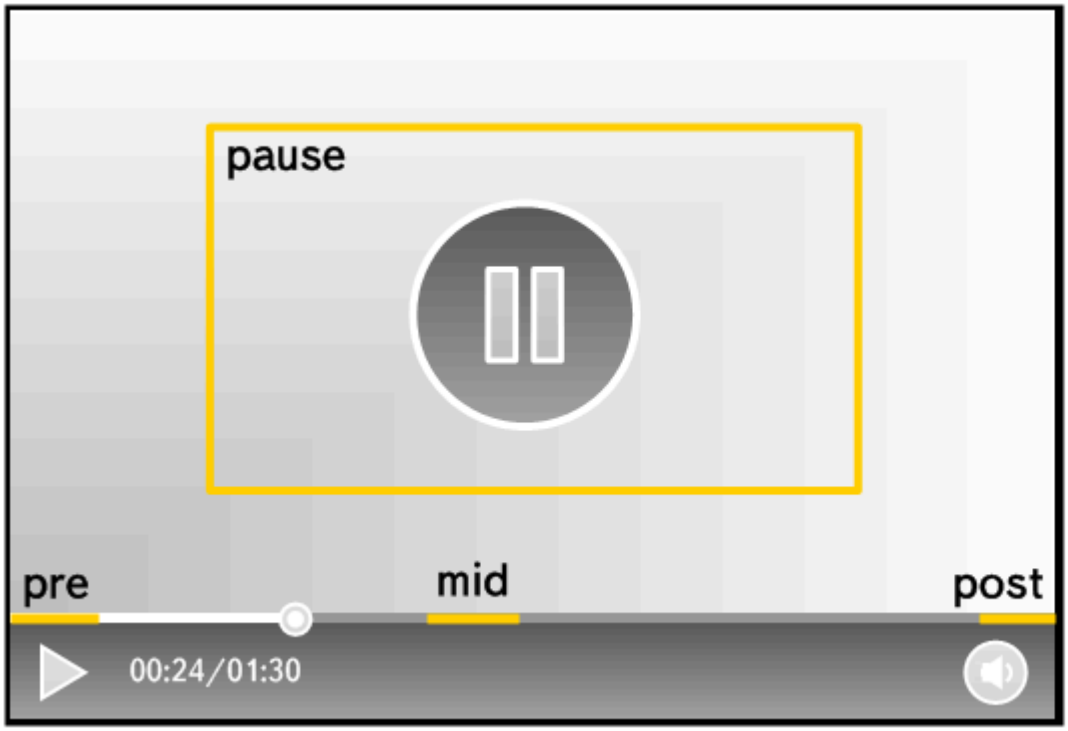
About InStream

To set up an InStream scenario, create a video resource in the Partner interface. Once created, the video resource is assigned a unique identifier (`Page ID`). This ID should be used in the Mobile Ads SDK.

To increase ad revenue, you can set up playing multiple ads within a single ad placement: an AdPod. You can set up the AdPod for an ad placement in the Partner interface.

Types of video units supported by the Mobile Ads SDK:

- Pre-Roll — A video ad is played before the main content.
- Mid-Roll — A video ad is played at a certain time within the main content.
- Post-Roll — A video ad is played after the main content.
- Pause-Roll — A video ad is played when the user clicks the pause button.
- In-Roll — A video ad is played anywhere in the video when a certain point is reached.



1. In-roll

API for working with InStream ads

An API for integrating InStream ads lets you support playing any type of ad break : Pre-Roll, Mid-Roll, Post-Roll, In-Roll , and Pause-Roll.

Pre-Roll, Mid-Roll, and Post-Roll ad breaks are played using the InstreamAdBinder API. To play In-Roll and Pause-Roll ad breaks, use the In-Roll API and Pause-Roll API, respectively.

Note:

You can also use the InstreamAdBinder API, In-Roll API, and Pause-Roll API concurrently if you:

1. Use different instances of the ad player.
2. Don't start the Pause-Roll and In-Roll APIs for playing ads if the main video was paused using the InStreamAdBinder API.

Integrating the InStream API

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

The InStream API is an API for setting up and managing the way InStream ads are loaded and played. It lets you support playing any type of ad break: Pre-Roll, Mid-Roll, Post-Roll, In-Roll, and Pause-Roll.

Pre-Roll, Mid-Roll, and Post-Roll ad breaks are played using the InstreamAdBinder API. To play In-Roll and Pause-Roll ad breaks, use the In-Roll API and Pause-Roll API, respectively.

Note:

You can also use the InstreamAdBinder API, In-Roll API, and Pause-Roll API concurrently if you:

1. Use different instances of the ad player.
2. Don't start the Pause-Roll and In-Roll APIs for playing ads if the main video was paused using the InStreamAdBinder API.

How it works

A loaded InStream ad object contains a schedule for playing ad breaks. Each ad break is described by an InstreamAdBreak object. An ad break may have one of the following types: Pre/Mid/Post/In/Pause-Roll. You can play Pre/MidPost-Roll ad breaks using the InstreamAdBinder API. To play In/Pause-Roll ad breaks, use the In-Roll API and Pause-Roll API, respectively.

The VideoPlayer protocol is used to interact with the main video content. To play an ad break inside an ad placement, use the InstreamAdPlayer protocol.

Playing Pre/Mid/Post-Roll video ads

InstreamAdBinder tracks the progress of playing the main video and automatically shows ad breaks based on their schedule from a video resource in the Partner interface.

InstreamAdBinder does not directly control the rendering of a video ad in PlayerView. Video ads must be played on the app side based on signals from player interfaces transmitted to InstreamAdBinder. InstreamAdBinder signals the start of playing an ad break by calling VideoPlayer.pauseVideo() and the end by calling VideoPlayer.resumeVideo().

When calling VideoPlayer.pauseVideo() on the app side, it's necessary to hide the main video controls, pause the main video, and start playing the video ad. On the ad SDK side, after calling the method, advertising controls are displayed inside the InstreamAdView container and the InstreamAdPlayer.playAd() method is called to start playing the video ad.

When calling VideoPlayer.resumeVideo() on the app side, it's necessary to return the main video controls and resume playing the main video. On the ad SDK side, ad controls inside the InstreamAdView container are removed before calling the method.

Playing In/Pause-Roll video ads

The In/Pause-Roll API doesn't directly control the rendering of a video ad in PlayerView. Video ads must be played on the app side based on signals from player interfaces transmitted to In/Pause-Roll. In/Pause-Roll signals the

start of playing an ad break by calling InstreamAdBreakDelegate.instreamAdBreakDidStart() and the end by calling InstreamAdBreakDelegate.instreamAdBreakDidComplete() or InstreamAdBreakDelegate.instreamAdBreakDidError().

When calling InstreamAdBreakDelegate.instreamAdBreakDidStart() on the app side, it's necessary to hide the main video controls and pause the main video. On the ad SDK side, after calling the method, advertising controls are displayed inside the InstreamAdView container and the InstreamAdPlayer.playAd() method is called to start playing the video ad.

When calling InstreamAdBreakDelegate.instreamAdBreakDidComplete() or InstreamAdBreakDelegate.instreamAdBreakDidError() on the app side, it's necessary to return the main video controls and resume playing the main video. On the ad SDK side, ad controls are removed from the InstreamAdView container before calling the methods.

Loading ads

1. Create an instance of the InstreamAdLoader class to get InStream ads.
2. To get notifications (ad loaded successfully or failed with an error), subscribe to ad load events. To do this, set a delegate that conforms to the InstreamAdLoaderDelegate protocol.
3. Create a configuration for the `instreamAdRequestConfiguration` request using the InstreamAdRequestConfiguration class. Pass the page identifier (`Page ID`) from the Partner interface as a request parameter.
4. Load ads using the InstreamAdLoader.loadInstreamAd method and pass `instreamAdRequestConfiguration` to it.

To test the integration, use the demo Page ID: demo-instream-vmap-yandex.

```
adLoader = InstreamAdLoader()
adLoader.delegate = self
let configuration = InstreamAdRequestConfiguration(pageID: PAGE_ID)
adloader.loadInstreamAd(configuration: configuration)
```

Rendering ads

Playing Pre/Mid/Post-Roll video ads

1. Implement the InstreamAdPlayer and VideoPlayer interfaces.

   The reference provides detailed information about the methods and their implementation. Additionally, see a test implementation example.

   Warning:

   To make implementation easier, we recommend using different instances of players to play video ads and content.
2. Add InstreamAdView to the View hierarchy of the app. InstreamAdView must contain PlayerView to play video ads in.

   Restriction:

   A container must be at least 300dp x 160dp in size.
3. Create an InstreamAdBinder object: pass the loaded InstreamAd object and the InstreamAdPlayer and VideoPlayer implementations to the builder.

   Set up notifications about the ad's progress (ready to play the video ad, the video ad played or failed to play): set a delegate that conforms to the InstreamAdBinderDelegate protocol.

   ```
   adBinder = InstreamAdBinder(ad: ad, adPlayer: adPlayer, videoPlayer:
   contentPlayer)
   adBinder.delegate = self
   ```
4. To start playing a Pre-Roll ad break faster, preload it in advance by calling the InstreamAdBinder.prepareAd() method.

   ```
   func preparePrerollAd(adBinder: InstreamAdBinder) {
       adBinder.delegate = self
       adBinder.prepareAd()
   }

   extension InstreamViewController: InstreamAdBinderDelegate {
       func instreamAdBinder(_ binder: InstreamAdBinder, didPrepare instreamAd: InstreamAd) {
           addInstreamAdBinderToPreloadedAdQueue(binder)
       }
       …
   }
   ```

5. Call the InstreamAdBinder.bind(with adView: InstreamAdView) method for the created InstreamAdBinder object. Pass `InstreamAdView`, which was previously added to the hierarchy, as a parameter. After that, the InStream SDK starts to automatically track the progress of playing the main video and manage the way video ads are played.

```
adBinder.bind(with: instreamAdView)
```

6. When playing InStream ads in the list, use the InStreamBinder.unbind() method when the cell with the ad is invalidated in the list. To implement a reused pool of players for scrolling, call InstreamAdbinder.invalidateAdPlayer() when reusing the ad player linked to InstreamAdBinder and InstreamAdBinder.invalidateVideoPlayer() when reusing the main content player.
7. When you stop using InStreamAdBinder, reset the state.

```
deinit {
    adBinder.unbind()
    adBinder.invalidateVideoPlayer()
    adBinder.invalidateAdPlayer()
}
```

Playing In/Pause-Roll video ads

Note:

Setting up playing Pause-Roll ad breaks is similar to In-Roll. To do this, replace In-Roll classes/methods with Pause-Roll ones.

1. Implement the InstreamAdPlayer interface.

   The reference provides detailed information about the methods and their implementation. Additionally, see a test implementation example.

   Warning:

   To make implementation easier, we recommend using different instances of players to play video ads and content.
2. Add InstreamAdView to the View hierarchy of the app. InstreamAdView must contain PlayerView to play video ads in.

   Restriction:

   A container must be at least 300dp x 160dp in size.
3. Use the InstreamAdLoader to load the InstreamAd object using the page ID (`Page ID`) from the Partner interface.
4. The InstreamAd object contains a set of different types of ad breaks. To get In-Roll ad breaks, use InrollQueueProvider. The InrollQueueProvider queue lets you receive In-Roll objects in the order required for display.

```
func instreamAdLoader(_ instreamAdLoader: InstreamAdLoader, didLoad ad:
InstreamAd) {
    inrollQueue = InrollQueueProvider(ad: ad).queue()
}
```

5. To launch the received In-Roll object, you need to prepare it. Unprepared In-Roll video ads won't start. To track if the In-Roll video ad is ready, set the InstreamAdBreakDelegate, call Inroll.prepare(with: adPlayer), and pass an instance of the created InstreamAdPlayer implementation to it.

```
private func prepareNextAd() {
    currentInroll = inrollQueue?.poll()
    currentInroll?.delegate = self
    currentInroll?.prepare(with: adPlayer)
}
```

6. Once the In-Roll video ad is prepared, InstreamAdBreakDelegate.instreamAdBreakDidPrepare() is called. The prepared In-Roll video ad is ready to play.

   Tip:

   Play video ads in the order they're received from the InrollQueue. If the received In-Roll video ads are played in a different order, this may lower your app's monetization.
7. To play the prepared In-Roll video ad, call Inroll.play(with adView: InstreamAdView) and pass InstreamAdView, which was previously added to the View hierarchy, as a parameter.

```
func instreamAdBreakDidPrepare(_ adBreak: InstreamAdBreak) {
    currentInroll?.play(instreamAdView)
}
```

8. After the ad break starts playing, the InstreamAdBreakDelegate.instreamAdBreakDidStart() method is called. After calling this method, pause the main video and hide its controls.

```
func instreamAdBreakDidStart(_ adBreak: InstreamAdBreak) {
```

```
    contentVideoPlayer?.pauseVideo()
}
```

9. Once the ad break is played, resume playing the main video. A video ad may play successfully or fail. Both situations need to be handled.

```
func instreamAdBreakDidComplete(_ adBreak: InstreamAdBreak) {
    handleAdBreakCompleted()
}
func instreamAdBreakDidError(_ adBreak: InstreamAdBreak) {
    handleAdBreakCompleted()
}
private fun handleAdBreakCompleted() {
    currentInroll = null
    contentVideoPlayer?.resumeVideo()
}
```

10. After the current In-Roll video ad finishes playing, check the play queue for the next In-Roll video ad in the InrollQueue.

```
private func prepareNextAd() {
    currentInroll = inrollQueue?.poll()
    currentInroll?.delegate = self
    currentInroll?.prepare(with: adPlayer)
}
```

11. When you stop using an In-Roll video ad, reset its state.

```
deinit {
    currentInroll?.invalidate()
}
```

Classes and protocols for working with InStream ads

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Classes

- InrollQueueProvider
- InstreamAdBinder
- InstreamAdBreakPosition
- InstreamAdBreakPositionType
- InstreamAdBreakType
- InstreamAdLoader
- InstreamAdPlayerError
- InstreamAdRequestConfiguration
- InstreamVideoAd
- MobileInstreamAds
- PauserollQueueProvider

Protocols

- AdPodInfo
- Inroll
- InrollQueue
- InstreamAd
- InstreamAdBinderDelegate
- InstreamAdBreak
- InstreamAdBreakDelegate
- InstreamAdLoaderDelegate
- InstreamAdPlayer
- InstreamAdPlayerDelegate
- InstreamAdSkipInfo
- InstreamCustomClickConnector

- InstreamCustomClickHandler
- InstreamCustomClickable
- InstreamVideoAdsProvider
- MediaFile
- PauseroII
- PauserollQueue
- SkipInfo
- VideoAd
- VideoPlayer
- VideoPlayerDelegate

GDPR

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

General information

The General Data Protection Regulation (GDPR) came into effect in the spring of 2018. GDPR regulates how information about citizens of the European Economic Area and Switzerland can be collected and processed. Its intention is to protect the privacy of confidential data and to ensure the transparency of all processes related to the collection, storage and processing of information on the internet.

The GDPR has an extraterritorial scope that applies to all companies that process the personal data of citizens of the European Economic Area and Switzerland, regardless of where the company is located.

Starting from version 2.11.0, the Yandex Mobile Ads SDK allows you to restrict the collection of data for users located in the European Economic Area and Switzerland who do not consent to data collection.

Quick guide

The user's consent for processing personal data must be sent to the SDK each time the application is launched.

1. Follow the instructions for connecting the Mobile Ads SDK.
2. Show a window where the user can accept the user agreement for personal data processing (for more information, see the example).

Attention:

This code is a sample, not a step-by-step guide to follow.

Swift

```
...
// The following code demonstrates creating a dialog.
func showGDPRDialog() {
    let alertController = UIAlertController(
        title: title,
        message: message,
        preferredStyle: .actionSheet)
    let acceptAction = UIAlertAction(
        title: "Accept",
        style: .default) { _ in
            self.setUserConsent(true)
        }
    alertController.addAction(acceptAction)
    let declineAction = UIAlertAction(
        title: "Decline",
        style: .default) { _ in
            self.setUserConsent(false)
        }
    alertController.addAction(declineAction)
    let openPrivacyPolicyAction = UIAlertAction(
```

```
        title: "View privacy policy",
        style: .default) { _ in
            UIApplication.shared.openURL(self.privacyPolicyURL)
        }
    alertController.addAction(openPrivacyPolicyAction)
    present(alertController, animated: true)
}

func setUserConsent(_ userConsent: Bool) {
    UserDefaults.standard.set(userConsent, forKey: kGDPRUserConsentKey)
}

func initializeAdsSDK() {
    let userConsent = UserDefaults.standard.bool(forKey: kGDPRUserConsentKey)
    YMAMobileAds.setUserConsent(userConsent)
}
```

Objective-C

```
...
// Example of creating a dialog window.
- (void)showGDPRDialog
{
    UIAlertController *alertController = [UIAlertController alertControllerWithTitle:title
                                                                              message:message

 preferredStyle:UIAlertControllerStyleActionSheet];
    UIAlertAction *acceptAction = [UIAlertAction actionWithTitle:@"Accept"
                                                           style:UIAlertActionStyleDefault
                                                         handler:^(UIAlertAction *action) {
                                                             [self setUserConsent:YES];
                                                         }];
    [alertController addAction:acceptAction];
    UIAlertAction *declineAction = [UIAlertAction actionWithTitle:@"Decline"
                                                           style:UIAlertActionStyleDefault
                                                         handler:^(UIAlertAction *action) {
                                                             [self setUserConsent:NO];
                                                         }];
    [alertController addAction:declineAction];
    UIAlertAction *openPrivacyPolicyAction = [UIAlertAction actionWithTitle:@"View privacy policy"
                                                                      style:UIAlertActionStyleDefault
                                                                    handler:^(UIAlertAction *action) {
                                                                        [[UIApplication sharedApplication]
 openURL:privacyPolicyURL];
                                                                    }];
    [alertController addAction:openPrivacyPolicyAction];
    [self presentViewController:alertController animated:YES completion:nil];
}

- (void)setUserConsent:(BOOL)userConsent
{
    [[NSUserDefaults standardUserDefaults] setBool:userConsent forKey:kGDPRUserConsentKey];
}

- (void)initializeAdsSDK
{
    BOOL userConsent = [[NSUserDefaults standardUserDefaults] boolForKey:kGDPRUserConsentKey];
    [YMAMobileAds setUserConsent:userConsent];
}
```

3. Use the + setUserConsent: method to pass the received value to the Mobile Ads SDK. For users from GDPR regions, the data will be processed only if they give their consent to it.

Support for iOS 14

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

SKAdNetwork support

Note:

SKAdNetwork is supported for SDK version 4.1.2 and higher. Follow the instructions for connecting the Mobile Ads SDK.

Mobile Ads SDK supports tracking of app installations using the SKAdNetwork framework. Installation tracking works for any device, even if no access to IDFA was granted.

To enable this feature, add the IDs of the supported ad networks to the `Info.plist` file of your application.

```
<key>SKAdNetworkItems</key>
<array>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zq492l623r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>633vhxswh4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>tmhh9296z4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vcra2ehyfk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zh3b7bxvad.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xmn954pzmp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>79w64w269u.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>488r3q3dtq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>d7g9azk84q.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nzq8sh4pbs.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>866k9ut3g3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2q884k2j68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x8jxxk4ff5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gfat3222tu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pd25vrrwzn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>lr83yxwka7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cp8zw746q7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pwdxu55a5a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c6k4g5qg8m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>s39g8k73mm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wg4vff78zm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g28c52eehv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>523jb4fst2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>294l99pt4k.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3qy4746246.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a8cz6cu7e5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ggvn48r87g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y755zyxw56.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qlbq5gtkt8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mls7yz5dvl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>67369282zy.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>899vrgt9g8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mj797d8u6f.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3sh42y64q3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f38h382jlk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>24t9a8vw3c.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mp6xlyr22a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x44k69ngh6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>88k8774x49.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hs6bdukanm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t3b3f7n3x8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>prcb7njmu6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c7g47wypnu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>52fl2v3hgk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9vvzujtq5s.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m8dbw4sv7c.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>9g2aggbj52.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m5mvw97r93.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>z5b3gh5ugf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dd3a75yxkv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9nlqeag3gk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cj5566h2ga.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>h5jmj969g5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dr774724x4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t7ky8fmwkd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>fz2k2k5tej.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>u679fj5vs4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cs644xg564.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9b89h5y424.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>w28pnjg2k4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2rq3zucswp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a7xqa6mtl2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g2y4y55b64.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vc83br9sjg.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cstr6suwn9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>eqhxz8m8av.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7k3cvf297u.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>w9q455wk68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nu4557a4je.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v4nxqhlyqp.skadnetwork</string>
    </dict>
```

```
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wzmmz9fp6w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7fmhfwg9en.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>su67r6k2v3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>yclnxrl5pm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7tnzynbdc7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>l6nv3x923s.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>h8vml93bkz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>uzqba5354d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8qiegk9qfv.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v79kvwwj4g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xx9sdjej2w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>au67k4efj4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t38b2kh725.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7ug5zh24hu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>rx5hdcabgc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5lm9lj6jb7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qqp299437r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>zmvfpc5aq8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9rd848q2bz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>79pbpufp6p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dmv22haz9p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y5ghdn5j9k.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n6fk4nfna4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7rz58n8ntl.skadnetwork</string>
```

```
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v9wttpbfk9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n38lu8286q.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>feyaarzu9v.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7fbxrn65az.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>47vhws6wlr.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ejvt5qm6ak.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>b55w3d8y8z.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v7896pgt74.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5ghnmfs3dh.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>275upjj5gd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>627r9wr2y5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>kbd757ywx3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>sczv5946wb.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8w3np9l82g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hb56zgv37p.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9t245vhmpl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nrt9jy4kw9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7953jerfzd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dn942472g5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6v7lgmsu45.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cad8qz2s3j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n9x2a789qt.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>r26jy69rpl.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>eh6m2bh4zr.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>jb7bn6koa5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>fkak3gfpt6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>a2p9lx4jpn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>97r2b46745.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>22mmun2rn5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>238da6jt44.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>44jx6755aq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>b9bk5wbcq9.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>k674qkevps.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>tl55sbb4fm.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>24zw6aqk47.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4468km3ulz.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2tdux39lx8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2u9pt9hc89.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8s468mfl3y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3cgn6rq224.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>glqzh8vgby.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>av6w8kgt66.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>klf5c3l5u5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>nfqy3847ph.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dticjx1a9i.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ppxm28t8ap.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9wsyqb3ku7.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>74b6s63p6l.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>xy9t38ct57.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>424m5254lk.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>qu637u8glc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f73kdq92p3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>44n7hlldy6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>kbmxgpxpgc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5l3tpt7t6e.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ecpz2srf59.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x5854y7y24.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>f7s53z58qe.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x8uqf25wch.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>uw77j35x4d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6964rsfnh4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gvmwg8q7h5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6yxyv74ff7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>84993kbrcf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>54nzkqm89y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pwa73g5rt2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mlmmfzh3r3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9yg77x724h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>n66cz3y3bx.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>578prtvx9j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4dzt52r2t5.skadnetwork</string>
    </dict>
```

```
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>bvpn9ufa9b.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6qx585k4p6.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>mtkv5xtk9e.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>l93v5h6a4m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>rvh3l7un93.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gta9lk7p23.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5tjdwbrq8w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>r45fhb6rf7.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>32z4fx6l9h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>e5fvkxwrpn.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8c4e2ghe7u.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>axh5283zss.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3rd42ekr43.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5mv394q32t.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3qcr597p9d.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>v72qych5uu.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ydx93a7ass.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4pfyvq9l8r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>5a6flpkh64.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4fzdc2evr5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4w7y6s5ca2.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>252b5q8x7y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2fnua5tdw4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3l6bd9hu43.skadnetwork</string>
```

```
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4mn522wn87.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6g9af3uyq4.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6p4ks3rnbw.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>6xzpu9s2p8.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>737z793b9f.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>89z7zv988g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8m87ys6875.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>8r8llnkz5a.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>bxvub5ada5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>c3frkrj4fj.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>cg4yq2srnc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dbu4b84rxf.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dkc879ngq3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>dzg6xy7pwj.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>gta8lk7p23.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hdw39hrw9y.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>hjevpa356n.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>krvm3zuq6h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ln5gz23vtd.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>ludvb6z3bs.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>m297p6643m.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>p78axxw29g.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>pu4na253f3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
```

```
        <string>s69wq72ugq.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>t6d3zquu66.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>vutu7akeur.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x2jnk7ly8j.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>x5l83yy675.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>y45688jllp.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>yrqqpx2mcb.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>z4gj7hsk7h.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>wzmmZ9fp6w.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4pfyvq9L8r.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>V72QYCH5UU.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>2U9PT9HC89.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>3RD42EKR43.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4FZDC2EVR5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7953JERFZD.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>7UG5ZH24HU.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9RD848Q2BZ.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>9T245VHMPL.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>CJ5566H2GA.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>F38H382JLK.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>KLF5C3L5U5.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>M8DBW4SV7C.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>W9Q455WK68.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>WZMMZ9FP6W.skadnetwork</string>
    </dict>
    <dict>
```

```
        <key>SKAdNetworkIdentifier</key>
        <string>XY9T38CT57.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>YCLNXRL5PM.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>4PFYVQ9L8R.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>TL55SBB4FM.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>BD757YWX3.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>33r6p7g8nc.skadnetwork</string>
    </dict>
    <dict>
        <key>SKAdNetworkIdentifier</key>
        <string>g69uk9uh2b.skadnetwork</string>
    </dict>
</array>
```

For more information, see Configuring a Source App in the Apple documentation.

Changes to iOS 14

As of January 2021, Apple has restricted access to IDFA on iOS 14. This means that Apple will no longer provide IDFA by default. IDFA (Identifier for Advertisers) is a unique device ID that is used for ad attribution and retargeting.

The Yandex Mobile Ads SDK uses IDFA to maximize publisher revenue by displaying more relevant ads.

To receive the IDFA, the app developer must now explicitly ask for the user's permission. You can do this using the AppTrackingTransparency framework.

It works similar to requests for sending push notifications.

Requesting permission to access IDFA

When starting the app, check the permission status using the trackingAuthorizationStatus property. Further recommendations depend on whether the user disabled access to the IDFA.

The user didn't disable access to the IDFA

The app can show the user an IDFA access permission request only once, so it's important to convince the user to grant the permission.

Strategy for working with users:

1. Before showing a system dialog, display a dialog box explaining:

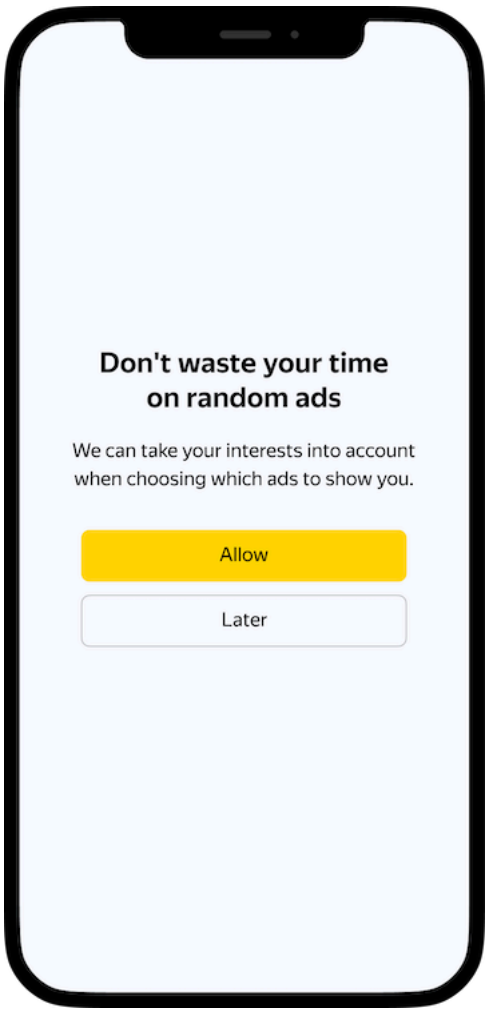
   - What exactly is requested.
   - How it will be used.
   - Why the user should allow access.

   You can add on-screen buttons like **Allow** and **Not Now**.

   Note:

   If the user agrees to grant permission in the app's dialog box with explanations, it's highly likely that they will give consent in the system dialog. If the user refuses to grant permission in the dialog box with explanations, you'll be able to display this dialog box again.

   Sample dialog box with explanations



2. Request access to the IDFA via App Tracking Transparency.

The user disabled access to the IDFA

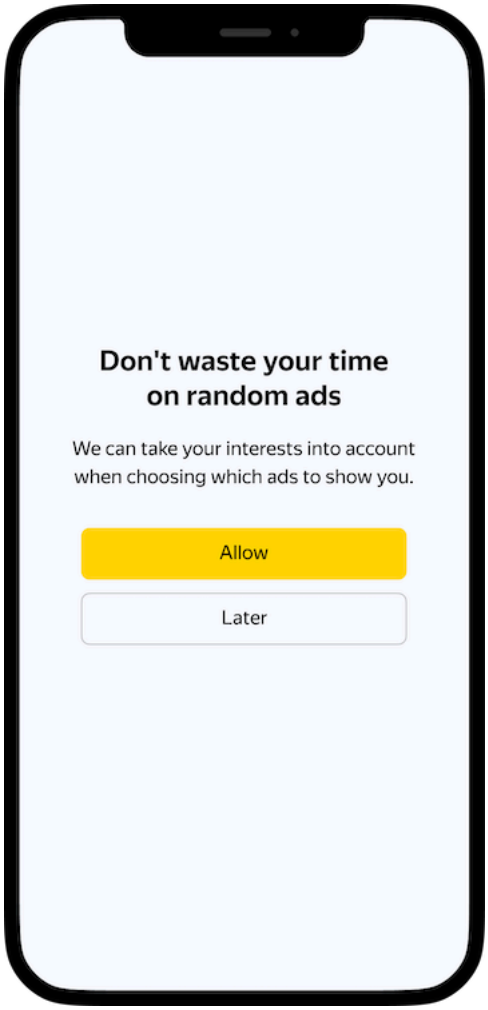
The user may have refused to grant permission to the IDFA in the system dialog.

Strategy for working with users:

1. In the app, display a dialog box explaining:
   - What exactly is requested.
   - How it will be used.
   - Why the user should allow access.

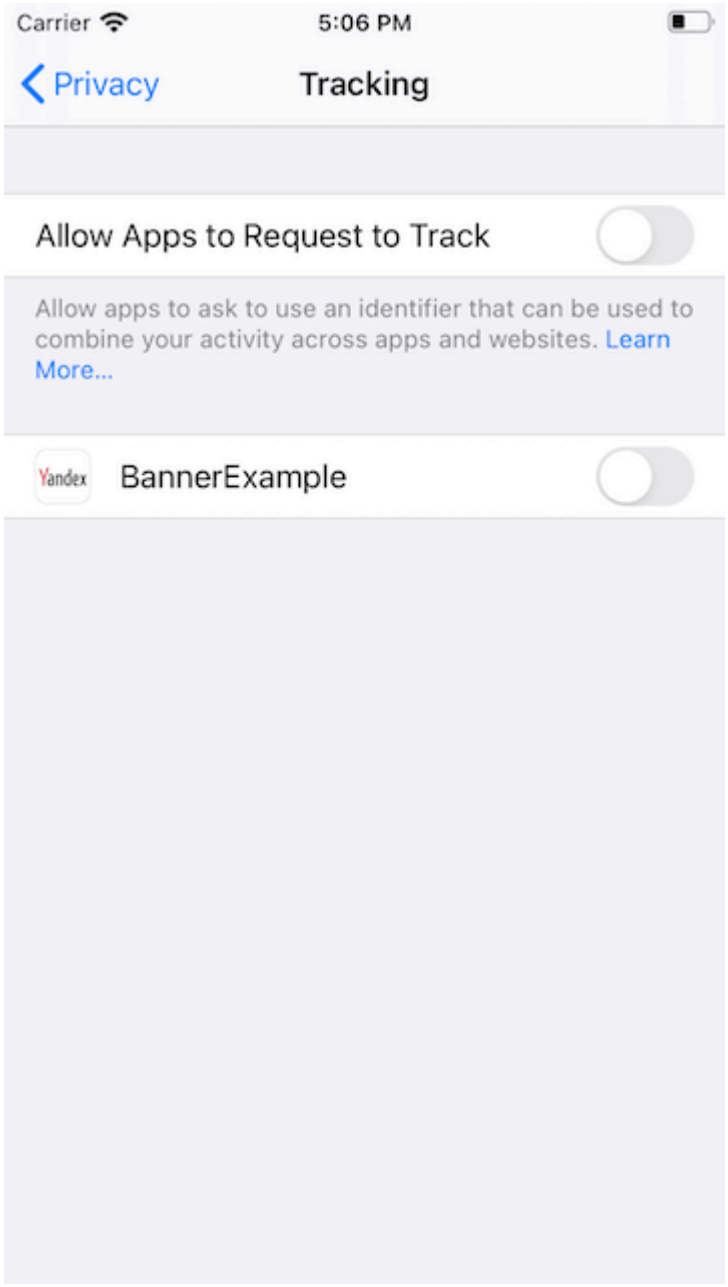
   You can add on-screen buttons like **Allow** and **Not Now**.

   Sample dialog box with explanations



2. Redirect the user to the settings via UIApplication.openSettingsUrlString.

Sample user settings with access disabled



Requesting permission to access the IDFA via App Tracking Transparency

In iOS 14, use a new framework named App Tracking Transparency to show the system dialog in your app. The user will choose to allow or deny access to the IDFA in it.

Note:

This system dialog can only be displayed once each time the app is installed. If the user selects **Ask App Not to Track**, you won't be able to show the dialog for this app again.

1. You can't change the system dialog, but you can add text with explanations to it. To do this, add the `NSUserTrackingUsageDescription` key to `Info.plist`. For example:

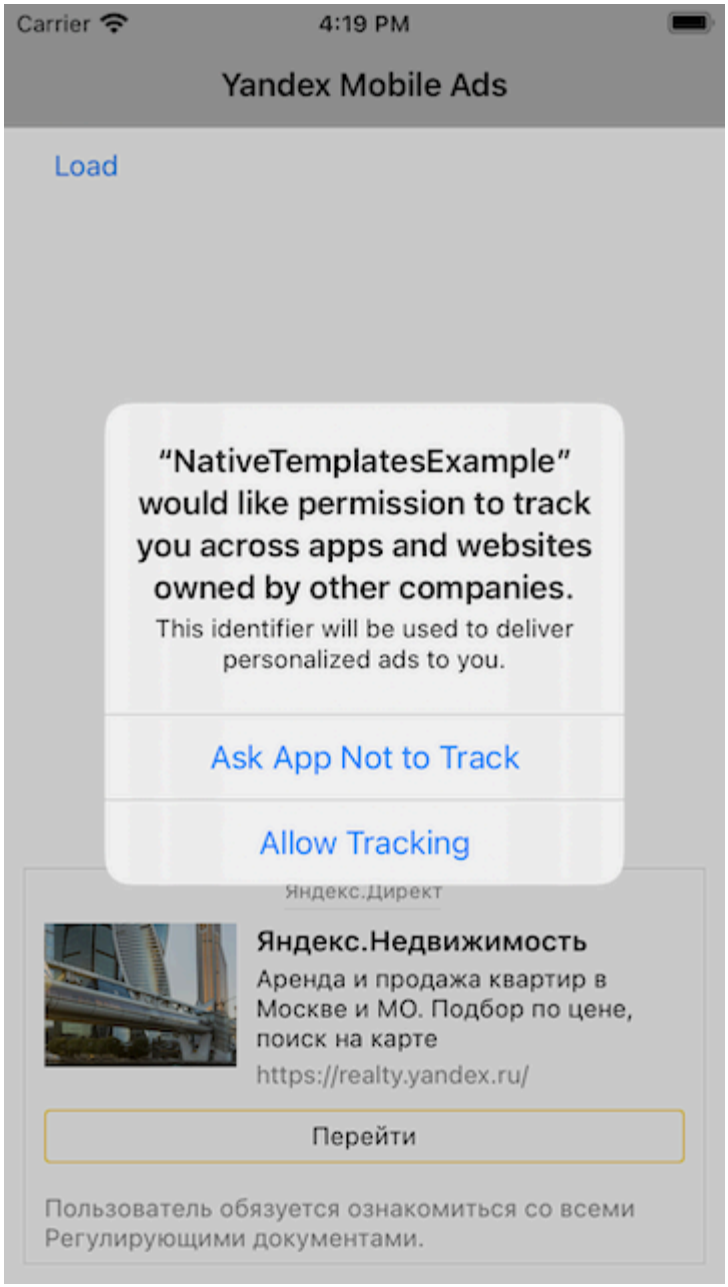
```
<key>NSUserTrackingUsageDescription</key>
```

```
<string>This identifier will be used to deliver personalized ads to you.</string>
```

Text from `Info.plist` will be shown to the user in the system dialog. In the text, explain to the user why the app requests permission to use the IDFA.

2. To display a dialog box requesting permission to access the IDFA, call the requestTrackingAuthorization(completionHandler:) method.

Sample system dialog box



3. Before loading ads, wait until a callback is received. The Yandex Mobile Ads SDK will then be able to use the IDFA in requests for ads.

Swift

```
import AppTrackingTransparency
...
func requestTrackingAuthorization() {
    ATTrackingManager.requestTrackingAuthorization { status in
        // Start ad loading
    }
```

```
}
```

Objective-C

```
#import <AppTrackingTransparency/AppTrackingTransparency.h>
...
- (void)requestTrackingAuthorization {
  [ATTrackingManager requestTrackingAuthorizationWithCompletionHandler:^(ATTrackingManagerAuthorizationStatus
 status) {
    // Start ad loading
  }];
}
```

4. To check the App Tracking Transparency authorization status, use the trackingAuthorizationStatus property.

Read more about App Tracking Transparency in the Apple documentation.

Ad targeting

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

These classes can be used to configure the SDK and ad targeting settings.

- YMAMobileAds
- YMAAdRequest

Tracking ad activity

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

The library lets you track events in the lifecycle of an ad (for example, unsuccessful queries and clicks) using a set of optional methods for the following protocols:

- YMAAdViewDelegate
- YYMAInterstitialAdDelegate
- YMANativeAdDelegate

Banner ads

When working with this type of ad, use the following methods:

Swift

```
extension YourClass: YMAAdViewDelegate {
    func adViewDidLoad(_ adView: YMAAdView) { }
    func adViewDidFailLoading(_ adView: YMAAdView, error: Error) { }
    func adViewWillLeaveApplication(_ adView: YMAAdView) { }
    func adView(_ adView: YMAAdView, willPresentScreen viewController: UIViewController?) { }
    func adView(_ adView: YMAAdView, didDismissScreen viewController: UIViewController?) { }
}
```

Objective-C

```
@protocol YMAAdViewDelegate <NSObject>

@optional

-(void)adViewDidLoad:(nonnull YMAAdView *)adView;
-(void)adViewDidFailLoading:(nonnull YMAAdView *)adView error:(nonnull NSError *)error;
-(void)adViewWillLeaveApplication:(nonnull YMAAdView *)adView;
-(void)adView:(nonnull YMAAdView *)adView willPresentScreen:(nullable UIViewController *)viewController;
-(void)adView:(nonnull YMAAdView *)adView didDismissScreen:(nullable UIViewController *)viewController;

@end
```

Interstitial ads

When working with this type of ad, use the following methods:

Swift

```
extension YourClass: YMAInterstitialAdDelegate {
    func interstitialAdDidLoad(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAdDidFail(toLoad interstitialAd: YMAInterstitialAd, error: Error) { }
    func interstitialAdWillLeaveApplication(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAdWillAppear(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAdDidAppear(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAdWillDisappear(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAdDidDisappear(_ interstitialAd: YMAInterstitialAd) { }
    func interstitialAd(_ interstitialAd: YMAInterstitialAd, willPresentScreen webBrowser: UIViewController?)
 { }
}
```

Objective-C

```
@protocol YMAInterstitialAdDelegate <NSObject>

@optional

-(void)interstitialAdDidLoad:(nonnull YMAInterstitialAd *)interstitialAd;
-(void)interstitialAdDidFailToLoad:(nonnull YMAInterstitialAd *)interstitialAd error:(nonnull NSError
 *)error;
-(void)interstitialAdWillLeaveApplication:(nonnull YMAInterstitialAd *)interstitialAd;
-(void)interstitialAdWillAppear:(nonnull YMAInterstitialAd *)interstitialAd;
-(void)interstitialAdDidAppear:(nonnull YMAInterstitialAd *)interstitialAd;
-(void)interstitialAdWillDisappear:(nonnull YMAInterstitialAd *)interstitialAd;
-(void)interstitialAdDidDisappear:(nonnull YMAInterstitialAd *)interstitialAd;
(void)interstitialAd:(nonnull YMAInterstitialAd *)interstitialAd willPresentScreen:(nullable UIViewController
 *)webBrowser;

@end
```

Native ads

When working with this type of ad, use the following methods:

Swift

```
extension YourClass: YMANativeAdDelegate {
    func nativeAdWillLeaveApplication(_ ad: YMANativeAd) { }
    func nativeAd(_ ad: YMANativeAd, willPresentScreen viewController: UIViewController?) { }
    func nativeAd(_ ad: YMANativeAd, didDismissScreen viewController: UIViewController?) { }
}
```

Objective-C

```
@protocol YMANativeAdDelegate <NSObject>

@optional

- (void)nativeAdWillLeaveApplication:(null_unspecified id)ad;
- (void)nativeAd:(null_unspecified id)ad willPresentScreen:(UIViewController *)viewController;
- (void)nativeAd:(null_unspecified id)ad didDismissScreen:(UIViewController *)viewController;

@end
```

Guide for migrating to version 5

Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found here.

Additions

YMAAdViewDelegate protocol

- Added the `-adViewDidClick:` method that notifies you that the user clicked on the banner.

YMAInterstitialAdDelegate protocol

- Added the `-interstitialAdDidClick:` method that notifies you that the user clicked on the ad.

YMANativeAdDelegate protocol

- Added the `-nativeAdDidClick:` method that notifies your that the user clicked on the ad.

YMARewardedAdDelegate protocol

- Added the `-rewardedAdDidClick:` method that notifies you that the user clicked on the ad.

Changes

YMAAdView class

- The `blockID` property was renamed `adUnitID`.
- The `-initWithBlockID:adSize:` method was renamed `-initWithAdUnitID:adSize:`.

YMAInterstitialAd class

- The `blockID` property was renamed `adUnitID`.
- The `-initWithBlockID:` method was renamed `-initWithAdUnitID:`.

YMANativeAdRequestConfiguration class

- The `blockID` property was renamed `adUnitID`.
- The `-initWithBlockID:` method was renamed `-initWithAdUnitID:`.

YMARewardedAd class

- The `blockID` property was renamed `adUnitID`.
- The `-initWithBlockID:` method was renamed `-initWithAdUnitID:`.

Removals

YMAAdSize class

Removed the following methods:

- `+flexibleSize`
- `+flexibleSizeWithContainerWidth:`

Recommendations

Version 4.X.Y	Version 5
`YMAAdSize *adSize = [YMAAdSize flexibleSizeWithContainerWidth:width];`	Removed, an alternative option is `YMAAdSize *adSize = [YMAAdSize fixedSizeWithCGSize:size];`
`YMAAdSize *adSize = [YMAAdSize flexibleSize:width];`	Removed, an alternative options is `YMAAdSize *adSize = [YMAAdSize fixedSizeWithCGSize:size];`
`YMAAdView *adView = [[YMAAdView alloc] initWithBlockID:<BlockID> adSize:adSize];`	The `BlockID` parameter is no longer supported, use the new parameter `AdUnitID:YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:<AdUnitID> adSize:adSize];`
`YMAInterstitialAd *interstitialAd = [[YMAInterstitialAd alloc] initWithBlockID:<BlockID>];`	The `BlockID` parameter is no longer supported, use the new parameter `AdUnitID:YMAInterstitialAd *interstitialAd = [[YMAInterstitialAd alloc] initWithAdUnitID:<AdUnitID>];`

Version 4.X.Y	Version 5
`YMARewardedAd *rewardedAd = [[YMARewardedAd alloc] initWithBlockID:<BlockID>];`	The `BlockID` parameter is no longer supported, use the new parameter `AdUnitID`:`YMARewardedAd *rewardedAd = [[YMARewardedAd alloc] initWithAdUnitID:<AdUnitID>];`
`YMANativeAdRequestConfiguration *requestConfiguration = [[YMANativeAdRequestConfiguration alloc] initWithBlockID:<BlockID>];`	The `BlockID` parameter is no longer supported, use the new parameter named `AdUnitId`: `YMANativeAdRequestConfiguration *requestConfiguration = [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:<AdUnitID>];`