

# Yandex Mobile Ads

Интеграция

10.07.2024

Yandex Mobile Ads. Интеграция. Версия 2.0

Дата подготовки документа: 10.07.2024

Этот документ является составной частью технической документации Яндекса.

© 2008—2024 ООО «ЯНДЕКС». Все права защищены.

## **Предупреждение об исключительных правах и конфиденциальной информации**

Исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации юридических лиц, товаров, работ, услуг и предприятий, которым предоставляется правовая охрана (интеллектуальную собственность), используемые при разработке, поддержке и эксплуатации службы Yandex Mobile Ads, включая, но не ограничиваясь, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования, принадлежат ООО «ЯНДЕКС» либо его лицензиарам.

Использование результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Yandex Mobile Ads, не допускается без получения предварительного согласия правообладателя. Настоящий документ содержит конфиденциальную информацию ООО «ЯНДЕКС». Использование конфиденциальной информации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Yandex Mobile Ads, а равно как и разглашение таковой, не допускается. При этом под разглашением понимается любое действие или бездействие, в результате которых конфиденциальная информация в любой возможной форме (устной, письменной, иной форме, в том числе с использованием технических средств) становится известной третьим лицам без согласия обладателя такой информации либо вопреки трудовому или гражданско-правовому договору.

Отношения ООО «ЯНДЕКС» с лицами, привлекаемыми для разработки, поддержки и эксплуатации службы Yandex Mobile Ads, регулируются законодательством Российской Федерации и заключаемыми в соответствии с ним трудовыми и/или гражданско-правовыми договорами (соглашениями). Нарушение требований об охране результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации, а равно как и конфиденциальной информации, влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с законодательством Российской Федерации.

## **Контактная информация**

ООО «ЯНДЕКС»

<https://www.yandex.ru>

Тел.: +7 495 739 7000

Email: [pr@yandex-team.ru](mailto:pr@yandex-team.ru)

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

# Содержание

Подключение Mobile Ads SDK.....	4
Поддержка SKAdNetwork.....	9
.....	19
<b>Форматы рекламы.....</b>	<b>20</b>
Баннерная реклама.....	20
Типы баннера.....	20
Подключение баннера.....	22
Пример работы с баннером.....	25
Классы и протоколы для работы.....	30
Полноэкранная реклама.....	30
Классы и протоколы для работы.....	32
Реклама с вознаграждением.....	32
Классы и протоколы для работы.....	33
Нативная реклама.....	34
Руководство по интеграции MediaView.....	34
Требования к размещению рекламы и рекламных компонентов.....	36
Компоненты нативной рекламы.....	36
Загрузка рекламы.....	40
Слайдер рекламных объявлений.....	43
Настройка внешнего оформления рекламы.....	46
Отладка.....	53
Классы и протоколы для работы.....	54
InStream реклама.....	55
Об InStream.....	55
API для работы с InStream.....	56
Интеграция InStream API.....	56
Классы и протоколы для работы.....	60
<b>GDPR.....</b>	<b>61</b>
Общие сведения.....	61
Краткое руководство.....	61
<b>Поддержка iOS 14.....</b>	<b>62</b>
Поддержка SKAdNetwork.....	62
Изменения в iOS 14.....	73
Запрос разрешения на доступ к IDFA.....	73
Запрос доступа к IDFA через App Tracking Transparency.....	76
<b>Таргетирование рекламы.....</b>	<b>78</b>
<b>Отслеживание работы рекламы.....</b>	<b>78</b>
<b>Руководство по миграции на версию 5.....</b>	<b>79</b>

---

# Подключение Mobile Ads SDK

**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

**Примечание:**

1. Для загрузки любого вида рекламы необходима версия iOS 13.0 и выше.
2. Для корректной работы на iOS 14 и выше обратите внимание на [дополнительные шаги](#).

**Внимание:**

Для работы SDK требуется подключение ресурсов, расположенных в `YandexMobileAdsBundle.bundle` из `YandexMobileAds.framework`. При подключении SDK через `CocoaPods` эти ресурсы подключаются автоматически.

При использовании кастомизированного подключения `YandexMobileAds.framework` обязательно убедитесь в том, что `YandexMobileAds.bundle` копируется в ресурсы проекта.

Библиотека может работать со следующими системами управления зависимостями:

**Swift Package Manager****Примечание:**

Адаптеры медиации недоступны для подключения через Swift Package Manager. Если вы используете медиацию, рекомендуется интеграция через `CocoaPods`.

Чтобы подключить библиотеку, выполните следующее:

1. В Xcode, в своем проекте добавьте зависимость через **File** → **Add Packages**.
2. Укажите URL репозитория `https://github.com/yandexmobile/yandex-ads-sdk-swift`, в нем находится Swift-пакет.

3. В настройках сборки **Build Settings**, в секции **Linking**, добавьте значение параметра Other Linker Flags = -ObjC.

The screenshot shows the Xcode interface for the SwiftExample project. The 'Build Settings' tab is selected and highlighted with a red box. In the left sidebar, the 'Linking' section is expanded and highlighted with a red box. The 'Other Linker Flags' setting at the bottom of the Linking section is highlighted with a blue bar and a red border.

PROJECT	Basic	Customized	All	Combined	Levels
SwiftExample					
TARGETS					
SwiftExample (iOS)					
SwiftExample (macOS)					
Tests iOS					
Tests macOS					

**Linking**

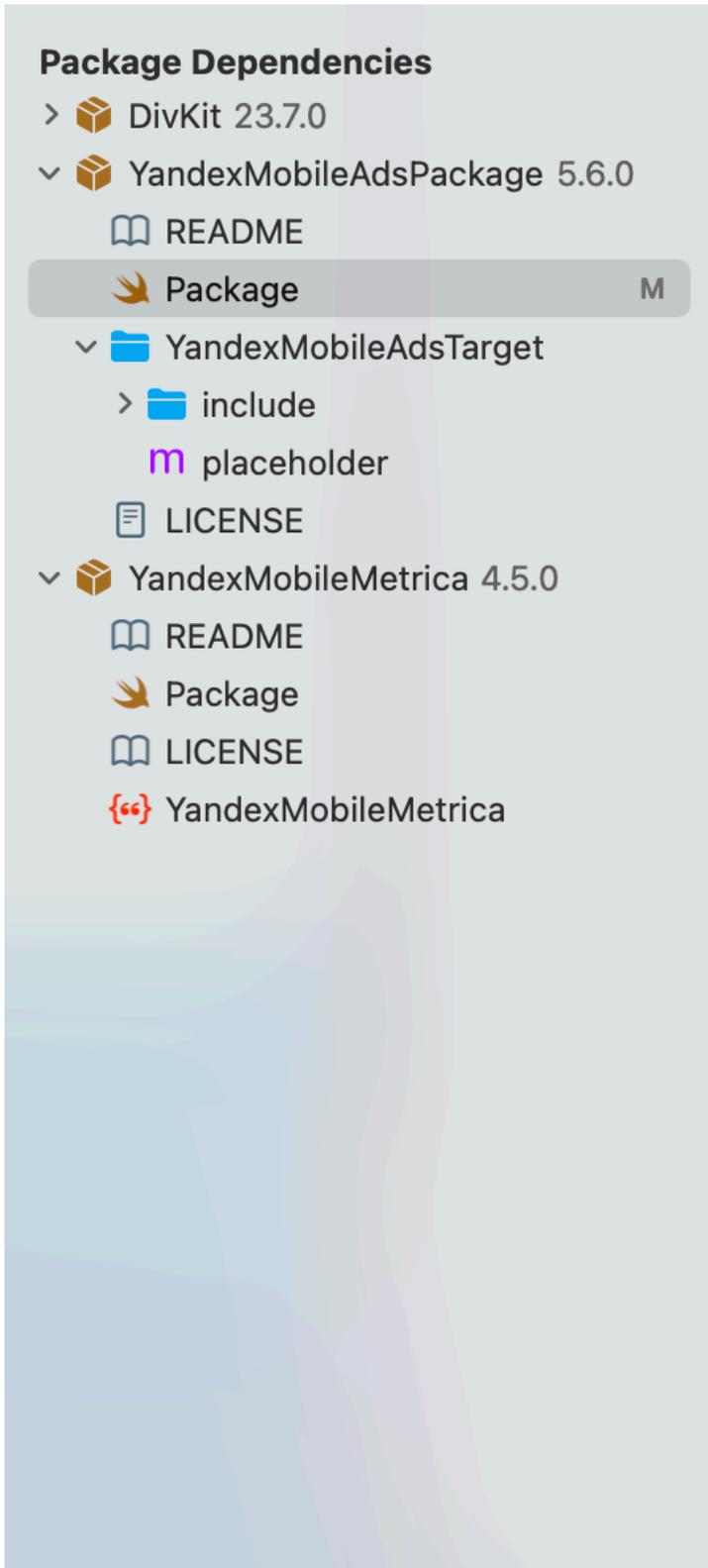
- Setting
- Bundle Loader
- Compatibility Version
- Current Library Version
- Dead Code Stripping
- Display Mangled Names
- Don't Dead-Strip Inits and Terms
- Dynamic Library Allowable Clients
- Dynamic Library Install Name
- Dynamic Library Install Name Base
- Exported Symbols File
- Generate Position-Dependent Executable
- Initialization Routine
- Link With Standard Libraries
- ◇ Mach-O Type
- Order File
- Other Librarian Flags
- > Other Linker Flags**

4. Проверьте, что таргет слинкован с YandexMobileAdsPackage. Если таргет не слинкован, добавьте связь с библиотекой в секции **Link Binary With Libraries** через значок +.

The screenshot shows the Xcode interface for a project named 'SwiftExample'. The 'TARGETS' list on the left includes 'SwiftExample (iOS)', which is selected. The main pane shows the 'Link Binary With Libraries' section, which is currently empty. A red box highlights the '+ Add items' button in this section. Other sections visible include 'Dependencies (0 items)', 'Compile Sources (2 items)', and 'Copy Bundle Resources (1 item)'. The 'Build Phases' tab is also visible at the top right.

5. Подключите YandexMobileAdsBundle.bundle из YandexMobileAds. Для этого:

- a. перейдите по ссылке для **YandexMobileAds** в секции **.binaryTarget (Package Dependencies → YandexMobileAdsPackage → кликните на Package.swift → найдите ссылку для YandexMobileAds в секции .binaryTarget)**;

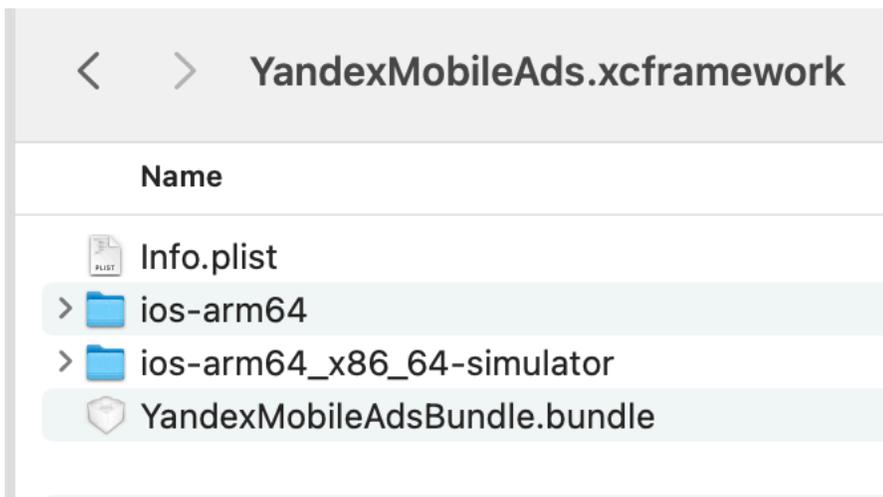


```

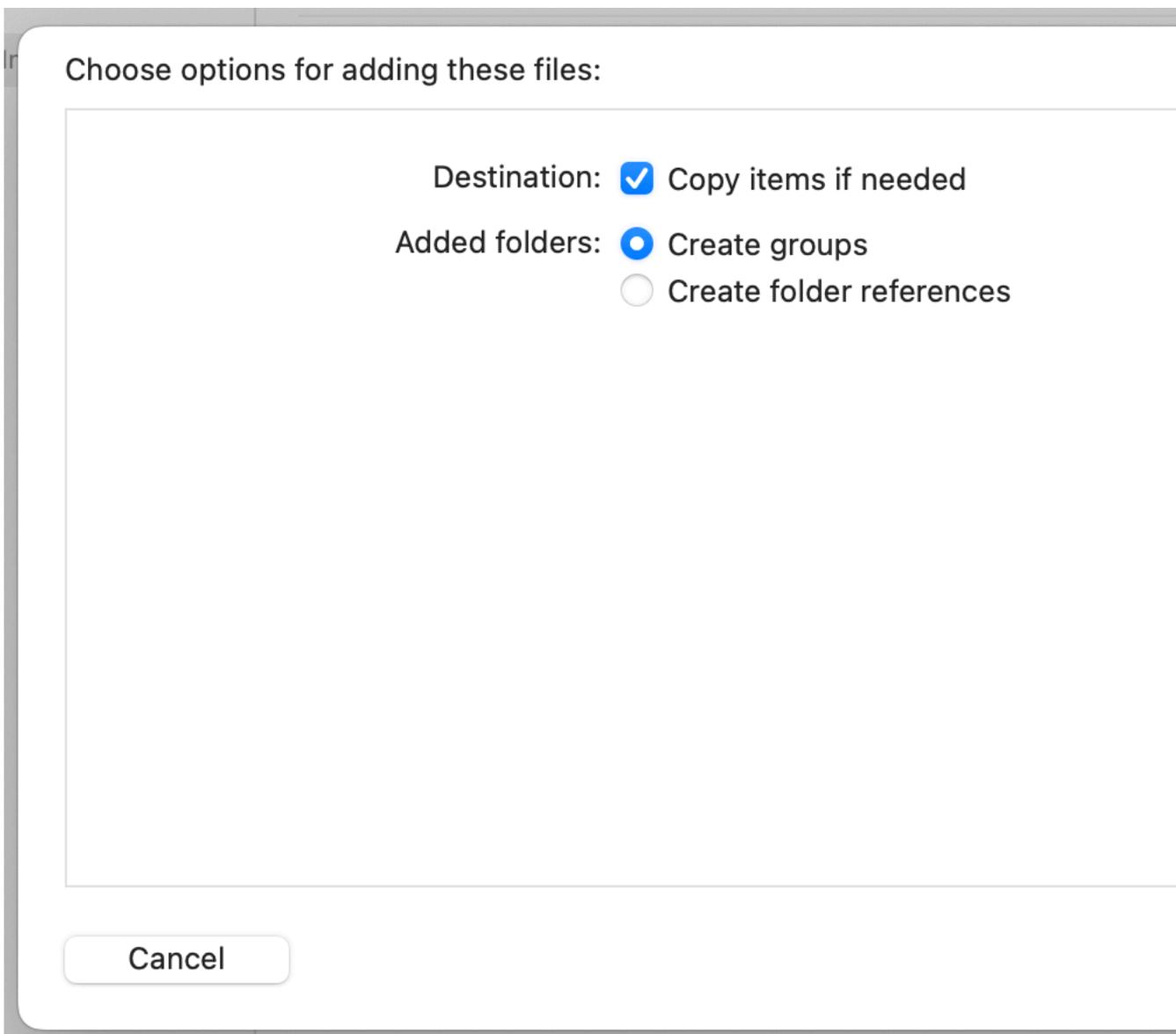
24 dependencies
25     .target(
26     .target(
27     .product
28     .product
29 ],
30 path: "Yande
31 linkerSettin
32     .linkedF
33     .linkedF
34     .linkedF
35     .linkedF
36     .linkedF
37     .linkedF
38     .linkedF
39     .linkedF
40     .linkedF
41     .linkedF
42     .linkedF
43     .linkedF
44     .linkedF
45     .linkedF
46     .linkedF
47     .linkedF
48 ]
49 ),
50 .binaryTarget(
51     name: "Yande
52     url:
53         "https:/
54         .0/Yande
55     checksum: "d
56 ),
57 .binaryTarget(
58     name: "Yande
59     url:

```

- b. откройте скачанный архив;



- с. подключите YandexMobileAdsBundle.bundle в Copy Bundle Resource фазу (**Ваш target** → **Build Phases** → **Copy Bundle Resource** → **Кнопка +** → **Add other** → **выберите YandexMobileAdsBundle.bundle** → **выберите пункты Copy items if needed и Create groups**).



### Copy Bundle Resources (4 items)

-  YandexMobileAdsBundle.bundle
  -  LaunchScreen.storyboard ...in AdsInstructionSPM/(localization).lproj
  -  Assets.xcassets ...in AdsInstructionSPM
  -  Main.storyboard ...in AdsInstructionSPM/(localization).lproj
- + -

### CocoaPods

Библиотека Yandex Mobile Ads SDK адаптирована для работы с системой управления зависимостями CocoaPods и поддерживает статический способ подключения. Чтобы подключить библиотеку, добавьте в Podfile проекта зависимости ([пример статического фреймворка](#)):

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

## Поддержка SKAdNetwork

### Примечание:

SKAdNetwork поддерживается для версии SDK 4.1.2 и выше.

Mobile Ads SDK поддерживает трекинг установок приложений с помощью фреймворка [SKAdNetwork](#). Трекинг установок работает для всех устройств, даже если доступ к IDFA отсутствует.

Чтобы включить функциональность, добавьте идентификаторы поддерживаемых рекламных сетей в файл Info.plist приложения.

```
<key>SKAdNetworkItems</key>  
<array>  
  <dict>  
    <key>SKAdNetworkIdentifier</key>  
    <string>zq492l623r.skadnetwork</string>  
  </dict>  
  <dict>  
    <key>SKAdNetworkIdentifier</key>  
    <string>633vhxsw4.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>  
</array>
```

```
<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>c6k4g5qq8m.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>294199pt4k.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>52f12v3hkg.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>9g2agbj52.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>yclnrxl15pm.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>8qiekg9qfv.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>7rz58n8nt1.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>8w3np9182g.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>r26jy69rp1.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>t155sbb4fm.skadnetwork</string>
</dict>
```

```
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>24zw6aqk47.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>4468km3ulz.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>2tdux391x8.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>klf5c315u5.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>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>578prt vx9j.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>193v5h6a4m.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>rvh317un93.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>r45fhh6rf7.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>32z4fx619h.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>6p4ks3rnw.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>b xvub5ada5.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>p78axw29g.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>
```

Для дополнительной информации ознакомьтесь с разделом [Configuring a Source App](#) документации Apple.

Ознакомьтесь с [примерами использования SDK](#).

# Форматы рекламы

## Подключение баннерной рекламы

**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

*Баннер* — настраиваемое объявление, которое занимает часть экрана и реагирует на нажатие.

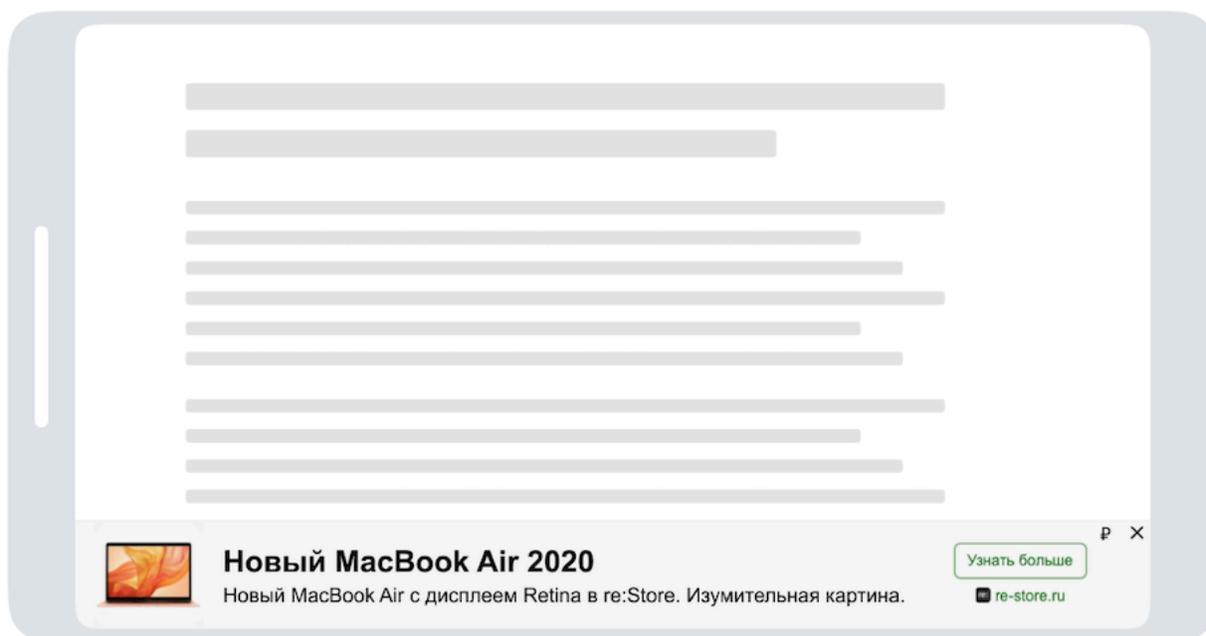
## Типы баннера

### Sticky баннер

Особенности:

1. Используется заданная ширина баннера. Высота подбирается автоматически.
2. Ширина баннера задается с помощью метода `+stickySizeWithContainerWidth`.
3. Высота баннера не должна превышать 15% высоты устройства и не должна быть меньше 50 dp.

Примеры отображения баннера:

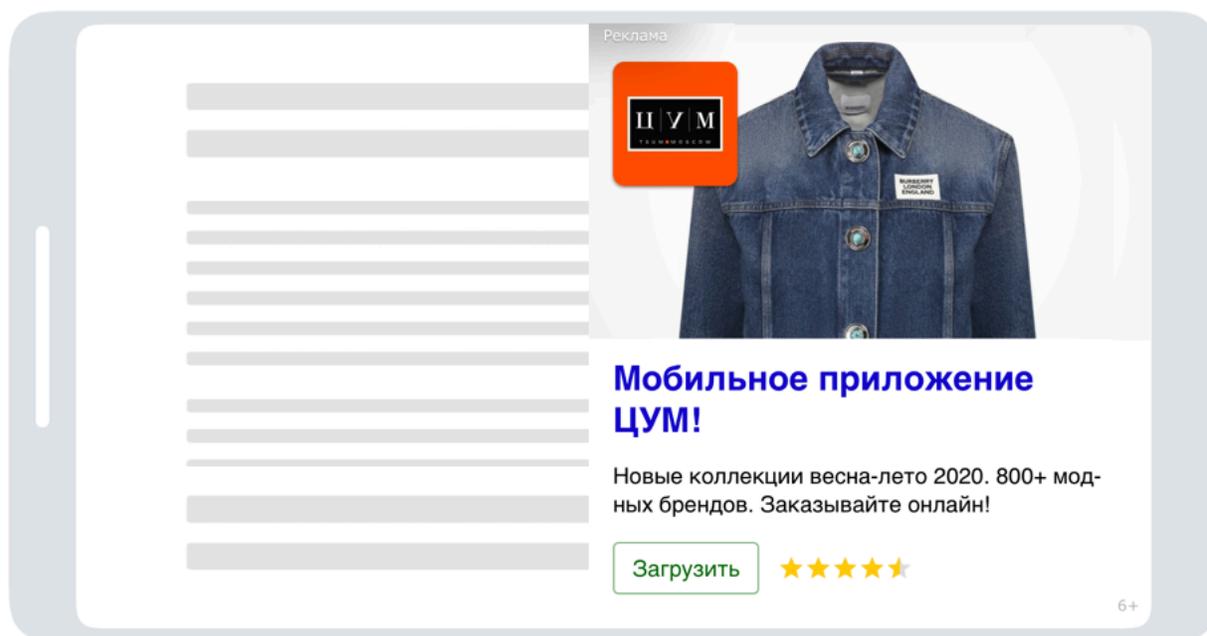


### Flex баннер

Особенности:

1. Баннер заполняет весь блок, используя заданные максимальные размеры.
2. Ширина и высота баннера задается с помощью метода `+flexibleSizeWithCGSize:`.
3. SDK накладывает `NSLayoutConstraint` на высоту баннера.

Примеры отображения баннера:



## Подключение баннера

Чтобы подключить баннер:

1. Добавьте импорт:

### Swift

```
import YandexMobileAds
```

### Objective-C

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

- Создайте `@property`, где будет храниться ссылка на баннер:

**Swift**

```
var adView: YMAAView!
```

**Objective-C**

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

- Создайте баннер:

**Sticky баннер**

Чтобы задать ширину баннера, вызовите метод [+stickySizeWithContainerWidth:](#).

**Swift**

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

**Objective-C**

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

**Flex баннер**

Чтобы задать ширину и высоту баннера, вызовите метод [+flexibleSizeWithCGSize:](#).

**Swift**

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

**Objective-C**

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

**Ограничение: Требования к размерам баннера при отображении видеорекламы**

Минимальный размер баннера, в котором поддерживается воспроизведение видео: 300x160 или 160x300.

`AdUnitId` — уникальный идентификатор рекламного места, который выдается в Партнерском интерфейсе и имеет вид: R-M-XXXXXX-Y.

При этом `self` должен удовлетворять протоколу [YMAAViewDelegate](#). В случае если делегат реализует метод [-viewControllerForPresentingModalView](#), ссылки будут открываться во встроенном браузере внутри приложения. Иначе ссылки будут открываться в браузере, установленном на устройстве.

Чтобы определить причины, по которым реклама работает некорректно, используйте метод [-adViewDidFailLoading:error:](#).

Описание ошибок доступно в разделе [YMAAErrorCode](#).

- Разместите баннер. Вы можете разместить баннер двумя способами:

- С помощью `autolayout constraints`.

Добавьте баннер в `UIView`. Затем добавьте `autolayout constraints`, чтобы баннер отображался в нужном месте.

**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];

```

- С помощью следующих методов:

**Swift**

```

displayAtTop(in:)
displayAtBottom(in:)

```

**Objective-C**

```

-displayAtTopInView;;
-displayAtBottomInView;;

```

В обоих случаях баннеры центрируются по горизонтали.

5. Загрузите баннер. Опционально, с помощью класса [YMAAdRequest](#), можно передать данные для таргетирования.

**Swift**

```

func loadAd(with request: YMAAdRequest?)

```

**Objective-C**

```

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

```

Когда баннер загрузится, делегат получит об этом уведомление.

6. Опционально можно включить логирование с помощью метода [+enableLogging](#). Если показ рекламы не был засчитан, в консоли появится сообщение.
7. Опционально можно настроить получение уведомлений об окончании проигрывания видео в баннерной рекламе.

**Пример реализации****Swift**

```

// Получение экземпляра YMAVideoController с помощью videoController.
let videoController = adView.videoController

// Установка делегата, который реализует протокол YMAVideoDelegate.
videoController.delegate = self

// Реализация метода протокола YMAVideoDelegate.
// MARK: - YMAVideoDelegate
func videoControllerDidFinishPlayingVideo(_ videoController: YMAVideoController) {
    print("Video complete");
}

```

**Objective-C**

```

// Получение экземпляра YMAVideoController с помощью videoController.

```

```

YMAVideoController *viewController = self.adView.videoController;

// Установка делегата, который реализует протокол YMAVideoDelegate.
viewController.delegate = self;

// Реализация метода протокола YMAVideoDelegate.
#pragma mark - YMAVideoDelegate
- (void)videoControllerDidFinishPlayingVideo:(YMAVideoController *)viewController
{
    NSLog(@"%@@", @"Video complete");
}

```

Если реклама подключена данным образом, после запуска приложения появится баннер.

Чтобы посмотреть, как баннерная реклама будет отображаться в приложении, используйте демонстрационный AdUnitID:

- demo-banner-yandex

### Узнайте больше

[Пример рекламы](#)

## Пример работы с баннером

Следующий код демонстрирует создание и настройку объекта AdView, регистрацию слушателя, а также загрузку баннера.

### Swift

```

override func viewDidLoad() {
    super.viewDidLoad()

    // Задание ширины рекламы
    let adSize = YMAAdSize.stickySize(withContainerWidth: containerWidth())

    // Создание экземпляра adView
    adView = YMAAdView(adUnitID: "R-M-XXXX-YY", adSize: adSize)
    adView.delegate = self
    addAdView()

    // Загрузка рекламы
    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];

    // Задание ширины рекламы
    YMAAdSize *adSize =
        [YMAAdSize stickySizeWithContainerWidth:[self containerWidth]];

    // Создание экземпляра adView
    self.adView = [[YMAAdView alloc] initWithAdUnitID:@"R-M-XXXX-YY" adSize:adSize];
    self.adView.delegate = self;
}

```

```
[self addAdView];

// Загрузка рекламы
[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;
}
```

## Адаптивный баннер

*Адаптивный баннер* — это баннер, органично вписывающийся в размеры блока, определенные пользователем. В зависимости от способа интеграции адаптивного баннера, для него подбирается оптимальная высота с заданной шириной, либо используются указанные размеры рекламного места.

### Примечание:

Про создание рекламного блока для адаптивного баннера можно прочитать в [справке Рекламной сети](#).

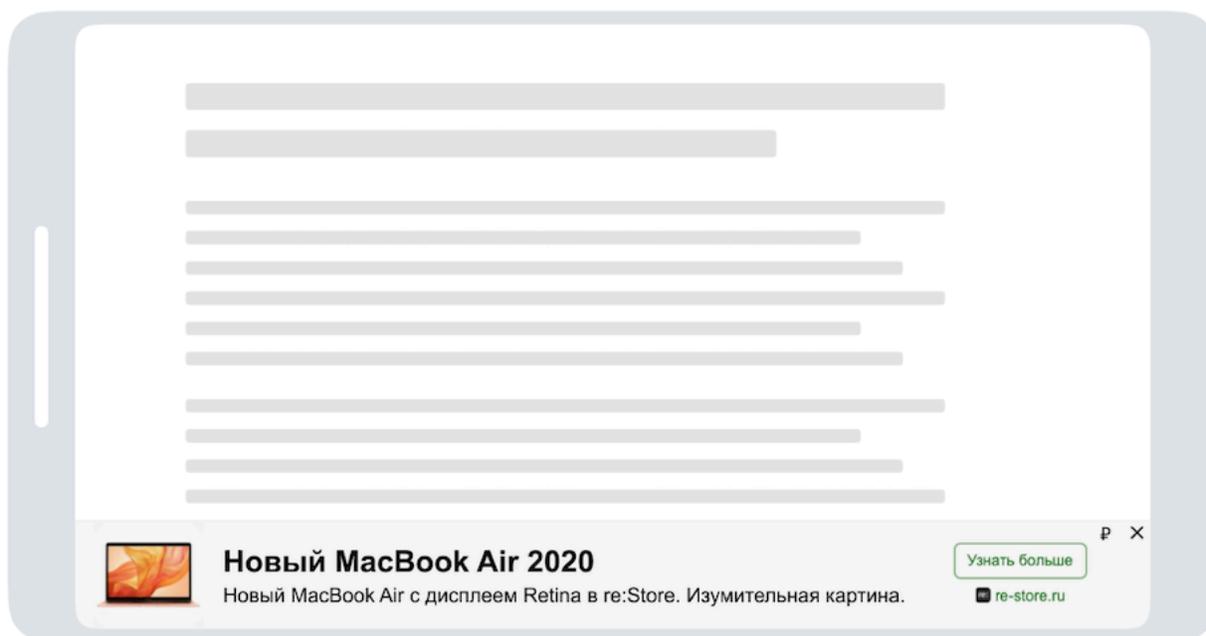
## Типы адаптивного баннера

### Баннер с заданной шириной

Особенности:

1. Альтернатива баннерам с размерами 320x50 (при определении высоты баннера сохраняется соотношение сторон 320x50).
2. Фиксация баннера в верхней или нижней части экрана (настраивается в приложении).
3. Использование заданной ширины баннера, а не ширины экрана устройства. Это позволяет учитывать особенности дисплея.
4. Ширина адаптивного баннера задается с помощью метода `+stickySizeWithContainerWidth:`.

Примеры отображения адаптивного баннера:

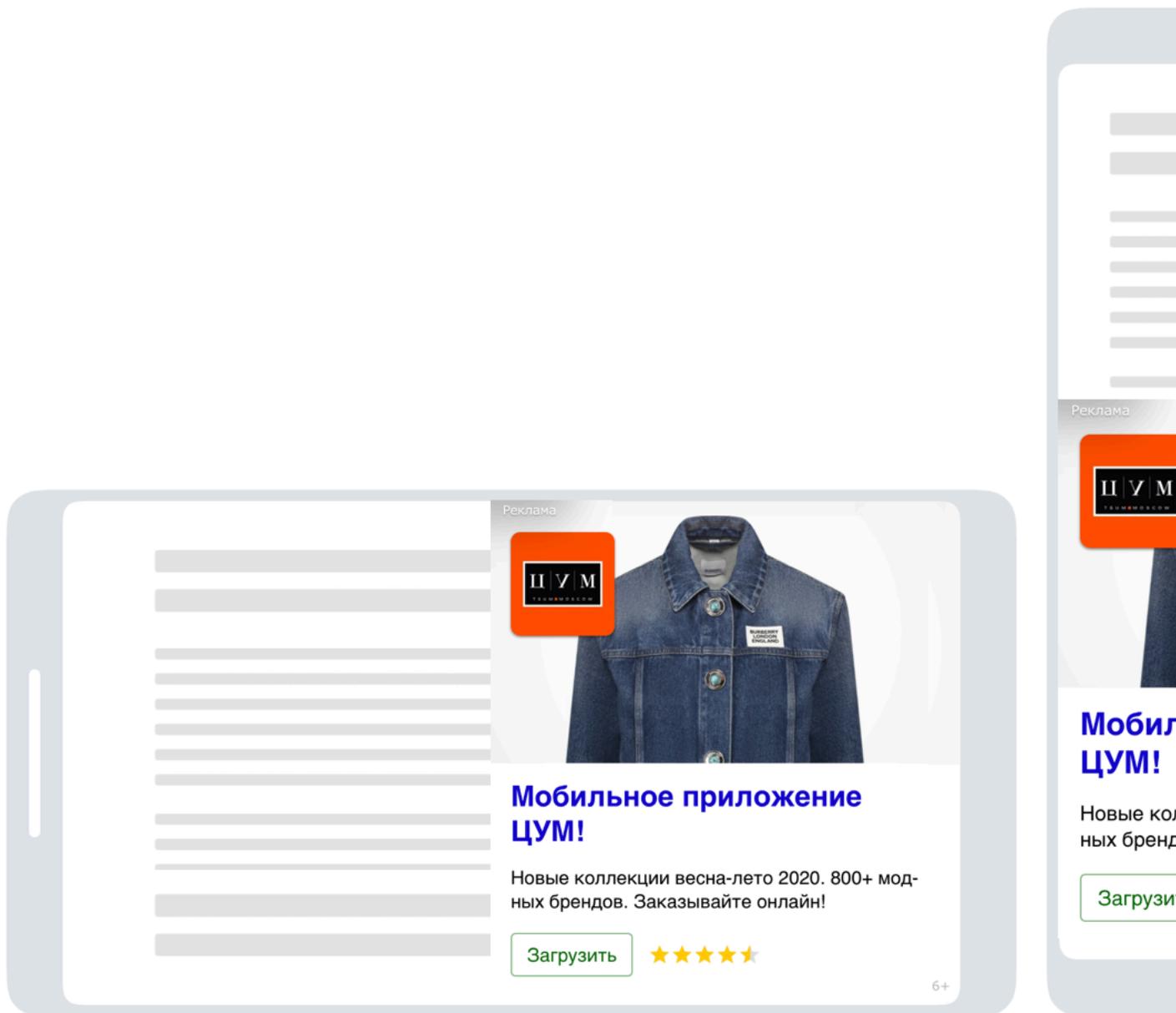


#### Баннер с заданной шириной и высотой

Особенности:

1. Адаптивный баннер заполняет весь блок, используя заданную ширину и высоту.
2. Ширина и высота адаптивного баннера задается с помощью метода `+flexibleSizeWithCGSize:`.

Примеры отображения адаптивного баннера:



### Подключение баннерной рекламы

1. Добавьте импорт:

**Swift**

```
import YandexMobileAds
```

**Objective-C**

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

2. Создайте @property, где будет храниться ссылка на баннер:

**Swift**

```
var adView: YMAAdView!
```

**Objective-C**

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

### 3. Укажите размер баннера и инициализируйте баннер в приложении.

#### Баннер с заданной шириной

Чтобы задать ширину адаптивного баннера, вызовите метод `+stickySizeWithContainerWidth:`.

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

#### Баннер с заданной шириной и высотой

Чтобы задать ширину и высоту адаптивного баннера, вызовите метод `+flexibleSizeWithCGSize:`.

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

`AdUnitId` — уникальный идентификатор рекламного места, который выдается в Партнерском интерфейсе и имеет вид: R-M-XXXXXX-Y.

При этом `self` должен удовлетворять протоколу `YMAAdViewDelegate`. В случае если делегат реализует метод `-viewControllerForPresentingModalView`, ссылки будут открываться во встроенном браузере внутри приложения. Иначе ссылки будут открываться в браузере, установленном на устройстве.

### 4. Разместите баннер. Вы можете это сделать, например, с помощью `autoLayout constraints`.

#### Пример реализации

Добавьте баннер в `UIView`. Затем добавьте `autoLayout constraints`, чтобы баннер отображался в нужном месте.

```
[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. Загрузите баннер. Опционально, с помощью класса `YMAAdRequest`, можно передать данные для таргетирования.

#### Swift

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

#### Objective-C

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

Когда баннер загрузится, делегат получит об этом уведомление.

### Пример работы с адаптивным баннером

Следующий код демонстрирует создание и настройку объекта `AdView`, регистрацию слушателя, а также загрузку адаптивного баннера:

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

    // Задание ширины рекламы
    YMAAdSize *adSize =
        [YMAAdSize stickySizeWithContainerWidth:[self containerView]];

    // Создание экземпляра adView
    self.adView = [[YMAAdView alloc] initWithAdUnitID:@"R-M-XXXXX-YY" adSize:adSize];
    self.adView.delegate = self;
    [self addAdView];
}
```

```
// Загрузка рекламы
[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 =
            UIRectInsetsInsetRect(self.view.frame, self.view.safeAreaInsets).size.width;
    }
    return containerWidth;
}
```

### Узнайте больше

[Пример рекламы](#)

## Классы и протоколы для работы с баннерной рекламой



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

### Классы

- [YMAAdSize](#)
- [YMAAdView](#)
- [YMAVideoController](#)

### Протоколы

- [YMAAdViewDelegate](#)
- [YMAVideoDelegate](#)

## Подключение полноэкранной рекламы



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

*Полноэкранная реклама (Interstitial)* — это настраиваемое объявление, отображаемое на весь экран и реагирующее на нажатие.

Чтобы подключить рекламу:

1. Добавьте импорт:

**Swift**

```
import YandexMobileAds
```

**Objective-C**

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

2. Создайте @property, где будет храниться ссылка на баннер:

**Swift**

```
var interstitialAd: YMAInterstitialAd!
```

**Objective-C**

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

3. Произведите инициализацию и предзагрузку рекламы. Это необходимо выполнить после отображения интерфейса приложения.

Опционально, с помощью класса [YMAAdRequest](#), можно передать данные для таргетирования. Пример использования класса см. в разделе [Полноэкранная реклама](#).

**Swift**

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

**Objective-C**

```
self.interstitialAd = [[YMAInterstitialAd alloc] initWithAdUnitID:<ваш уникальный AdUnitID>];
self.interstitialAd.delegate = self;
[self.interstitialAd load];
```

AdUnitId — уникальный идентификатор рекламного места, который выдается в Партнерском интерфейсе и имеет вид: R-M-XXXXXX-Y.

При этом self должен удовлетворять протоколу [YMAInterstitialAdDelegate](#). Производить предзагрузку рекламы необходимо в той же ориентации, в которой она будет показана (иначе реклама не будет показана из-за несоответствия размеров экрана и баннера).

4. Начните отображение объявления, используя данный метод:

**Swift**

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

**Objective-C**

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

5. Опционально можно включить логирование с помощью метода [+enableLogging](#). Если показ рекламы не был засчитан, в консоли появится сообщение.

Чтобы определить причины, по которым реклама работает некорректно, используйте методы

**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;
```

Описание ошибок доступно в разделе [YMAAdErrorCode](#).

Чтобы посмотреть, как реклама будет отображаться в приложении, используйте демонстрационный AdUnitID:

- demo-interstitial-yandex

### Узнайте больше

[Пример рекламы](#)

## Классы и протоколы для работы с полноэкранный рекламой



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

### Классы

- [YMAInterstitialAd](#)

### Протоколы

- [YMAInterstitialAdDelegate](#)

## Подключение рекламы с вознаграждением



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

*Реклама с вознаграждением (Rewarded Ad)* — настраиваемое объявление, отображаемое на весь экран. За просмотр такой рекламы пользователь получает вознаграждение.

Чтобы подключить рекламу:

1. Добавьте импорт:

#### Swift

```
import YandexMobileAds
```

#### Objective-C

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

2. Создайте @property, где будет храниться ссылка на рекламное объявление:

#### Swift

```
var rewardedAd: YMARewardedAd!
```

#### Objective-C

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

3. Произведите инициализацию рекламы:

#### Swift

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

#### Objective-C

```
self.rewardedAd = [[YMARewardedAd alloc] initWithAdUnitID:<ваш уникальный AdUnitID>];
```

```
self.rewardedAd.delegate = self;
```

AdUnitId — уникальный идентификатор рекламного места, который выдается в Партнерском интерфейсе и имеет вид: R-M-XXXXXX-Y.

При этом self должен удовлетворять протоколу [YMARewardedAdDelegate](#).

#### 4. Загрузите рекламу.

##### Swift

```
rewardedAd.load()
```

##### Objective-C

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

Опционально, с помощью класса [YMAAdRequest](#), можно передать данные для таргетирования. Пример использования класса см. в разделе [Полноэкранная реклама](#).

Производить предзагрузку рекламы необходимо в той же ориентации, в которой она будет показана (иначе реклама не будет показана из-за несоответствия размеров экрана и баннера).

#### 5. Начните отображение объявления, используя данный метод:

##### Swift

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

##### Objective-C

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

- Если вы используете механизм награждения на стороне клиента («client-side reward»), реализуйте метод делегата [-rewardedAd:didReward:](#). Он вызывается, когда показ засчитан и пользователь может быть награжден за просмотр рекламы. Используйте этот момент, чтобы выдать награду пользователю приложения.
- Опционально можно включить логирование с помощью метода [+enableLogging](#). Если показ рекламы не был засчитан, в консоли появится сообщение.

Чтобы определить причины, по которым реклама работает некорректно, используйте методы

##### 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;
```

Описание ошибок доступно в разделе [YMAAdErrorCode](#).

Чтобы посмотреть, как реклама будет отображаться в приложении, используйте демонстрационный AdUnitID:

- demo-rewarded-yandex

## Классы и протоколы для работы с видеорекламой с вознаграждением



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

**Классы**

- [YMARewardedAd](#)

**Протоколы**

- [YMAReward](#)
- [YMARewardedAdDelegate](#)

## Нативная реклама

**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Yandex Mobile Ads SDK предоставляет возможность показывать рекламу с использованием собственных визуальных компонентов.

Нативная реклама может изменяться в зависимости от функций и дизайна приложения, в котором она отображается. Оформление нативной рекламы соответствует среде, в которую она встраивается. Такая реклама выглядит органично и дополняет приложение полезной информацией.

SDK также предоставляет набор готовых настраиваемых визуальных компонентов (шаблонов), которые позволяют использовать все преимущества отрисовки нативными средствами платформы и не требуют создания собственного дизайна.

Yandex Mobile Ads SDK поддерживает несколько типов рекламы: App Install, Content, Image.

Чтобы подключить рекламу:

1. Ознакомьтесь с [требованиями к размещению рекламы и рекламных компонентов](#).
2. [Загрузите рекламу](#).
3. [Настройте внешнее оформление рекламы](#).

**Узнайте больше**

[Пример рекламы](#)

## Руководство по интеграции MediaView

**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

**Примечание:**

Интеграция описана на примере рекламы типа ImageAd.

1. Подключите Mobile Ads SDK.

Добавьте в Podfile проекта зависимость:

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

2. Загрузите нативную рекламу.
  - a. Создайте объект класса [YMANativeAdLoader](#) с конфигурацией [YMANativeAdRequestConfiguration](#).
  - b. Реализуйте протокол [YMANativeAdLoaderDelegate](#) и метод `-nativeAdLoader:didLoadAd:`.
  - c. Установите delegate для созданного объекта [YMANativeAdLoader](#).
  - d. Вызовите метод `-loadAdWithRequestConfiguration:`.

### 3. Отобразите нативную рекламу.

#### Ограничение: Требования к размерам `mediaView` при отображении видеорекламы

Минимальный размер экземпляра класса `YMANativeMediaView`, в котором поддерживается воспроизведение видео: 300x160 или 160x300.

Для поддержки воспроизведения видео в нативной рекламе рекомендуется выставить ширину для `mediaView` не менее 300. Чтобы вычислить соответствующее значение высоты `mediaView`, используйте значение свойства `aspectRatio`. При отображении рекламы с помощью шаблонов, корректная высота для `mediaView` будет вычислена автоматически, с учетом соотношения ширины и высоты.

а. Для простого способа отображения рекламы воспользуйтесь шаблонами.

1. Создайте объект класса `YMANativeBannerView`.
2. Установите для него объект загруженной рекламы.

#### 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
```

Подробнее о нативной рекламе читайте в разделе [Нативная реклама](#).

б. Также вы можете воспользоваться ручным способом отображения нативной рекламы. Такой способ позволяет отображать нативную рекламу максимально адаптивно под ваш основной контент. Вызовите метод `-bindWithAdView:error:`.

#### 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);  
}  
}
```

Подробнее о нативной рекламе читайте в разделе [Оформление без использования шаблона](#).

### Узнайте больше

[Пример рекламы](#)

## Требования к размещению рекламы и рекламных компонентов



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

### Основные требования

Показ рекламы не будет засчитан, если хотя бы одно из перечисленных условий не выполнено:

1. Рекламная View, а также каждая из ее `Superview` должны удовлетворять следующим условиям:
  - View должна быть видима;
  - View должна быть непрозрачна.
2. Все обязательные рекламные компоненты должны удовлетворять следующим условиям:
  - рекламный компонент, а также каждый из его `Superview` должны быть полностью видимы;
  - рекламный компонент должен полностью находиться в пределах рекламной View;
  - рекламный компонент должен находиться в иерархии рекламной View;
  - рекламный компонент должен быть непрозрачен;
  - содержимое рекламного компонента должно соответствовать действительности.
3. Приложение должно находиться в активном состоянии (не в фоне).
4. В единый момент времени загруженное рекламное объявление может быть показано только в одной View. Одновременное отображение одного рекламного объявления в нескольких View может привести к потере показа.

**Примечание:** Если все условия выполнены, а показ не засчитан, то попытка его засчитать будет произведена позже.

### Ограничения на изменение рекламных компонентов

1. Нельзя редактировать текстовое содержимое компонентов.
2. Нельзя редактировать содержимое изображения.
3. Растягивать изображение можно только с соблюдением пропорций.
4. Обрезать изображение можно не более чем на 20%. Допустимые инструменты: наложение маски, изменение размеров контейнера.

## Компоненты нативной рекламы



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

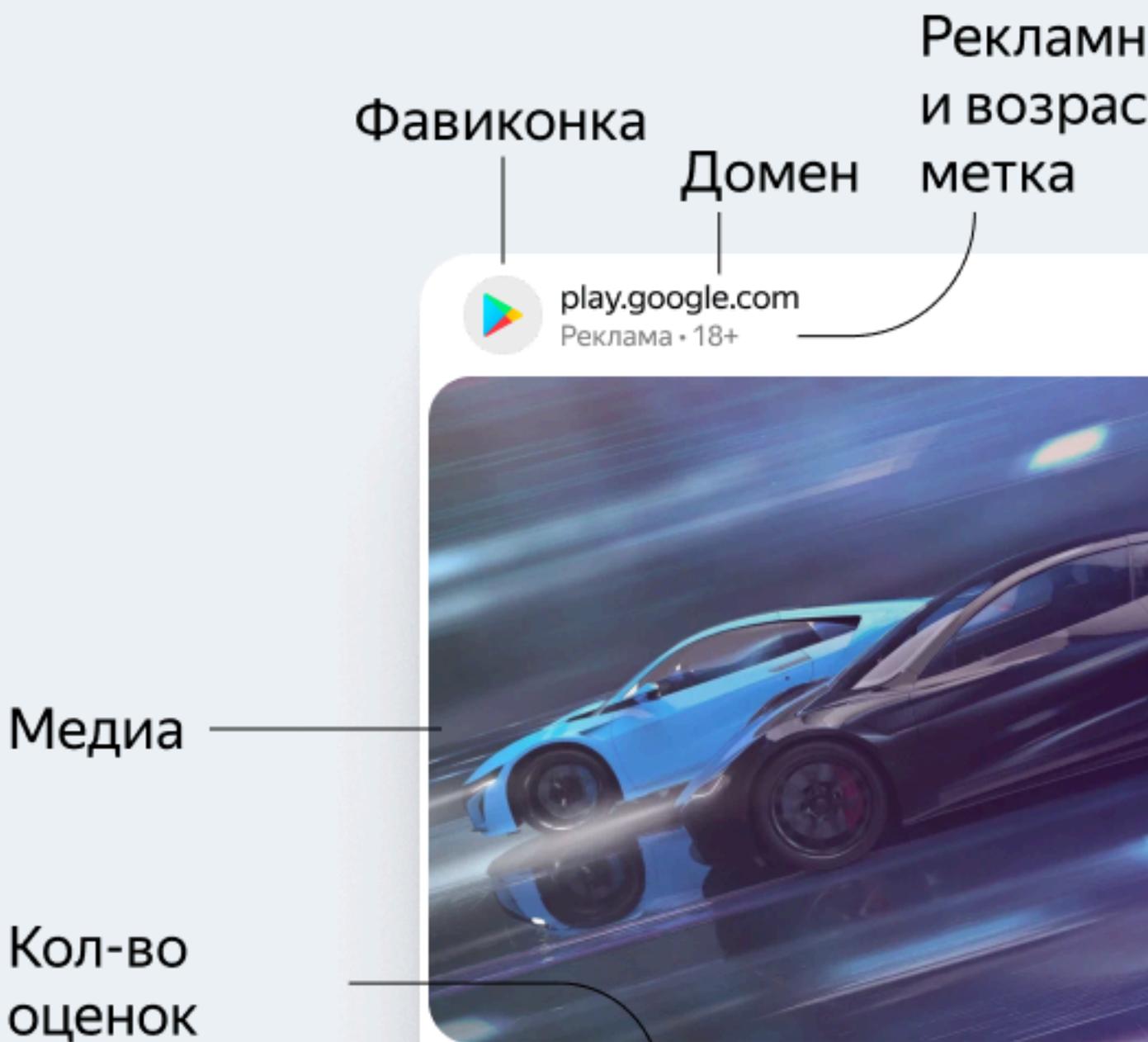
### Примечание:

Ознакомьтесь с [требованиями к размещению рекламы и рекламных компонентов](#).

### Список компонентов

В библиотеке есть обязательные и опциональные компоненты. Правила размещения рекламных объявлений определяют необходимость отображения только обязательных компонентов. На практике, макеты с полным набором компонентов более кликабельные. Поэтому рекомендуется использовать дизайн, который включает весь набор возможных компонентов.

# Реклама приложе



Элемент в объявлении	Компонент	Тип	Обязательность
Заголовок	titleLabel	UILabel	Да
Домен	domainLabel	UILabel	Да
Предупреждение	warningLabel	UILabel	Да
Рекламная и возрастная метка	sponsoredLabel	UILabel	Да
Значок меню	feedbackButton	UIButton	Да
Кнопка действия	callToActionButton	UIButton	Да
Медиа	mediaView	YMANativeMediaView	Да
Иконка приложения	iconImageView	UIImageView	Да, для РМП
Цена	priceLabel	UILabel	Да, для РМП
Фавиконка	faviconImageView	UIImageView	Нет
Количество оценок	reviewCountLabel	UILabel	Нет
Рейтинг	ratingView	UIView<YMARating>	Нет
Текст	bodyLabel	UILabel	Нет

Список обязательных компонентов описывает набор данных, для которых, в случае наличия этих компонентов, должна быть предоставлена View для отображения.

**Совет:**

Рекомендуется использовать макет, который сможет отобразить весь набор обязательных и опциональных компонентов рекламы.

**Оформление компонентов**

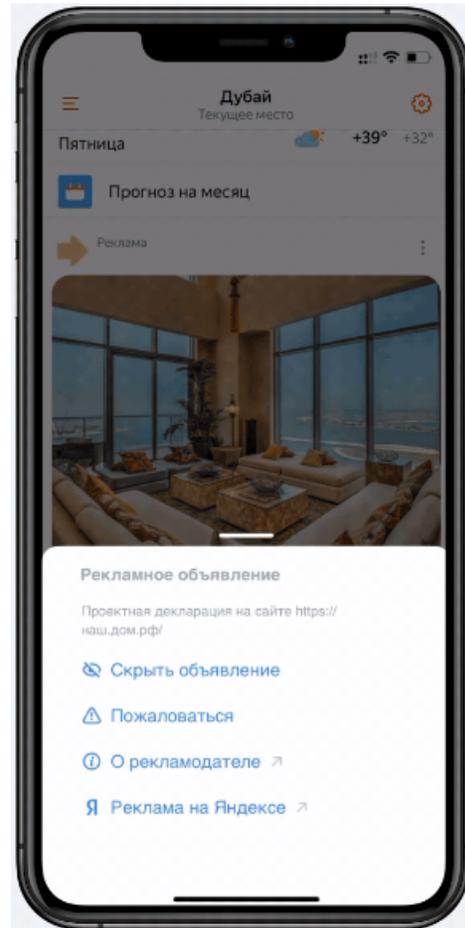
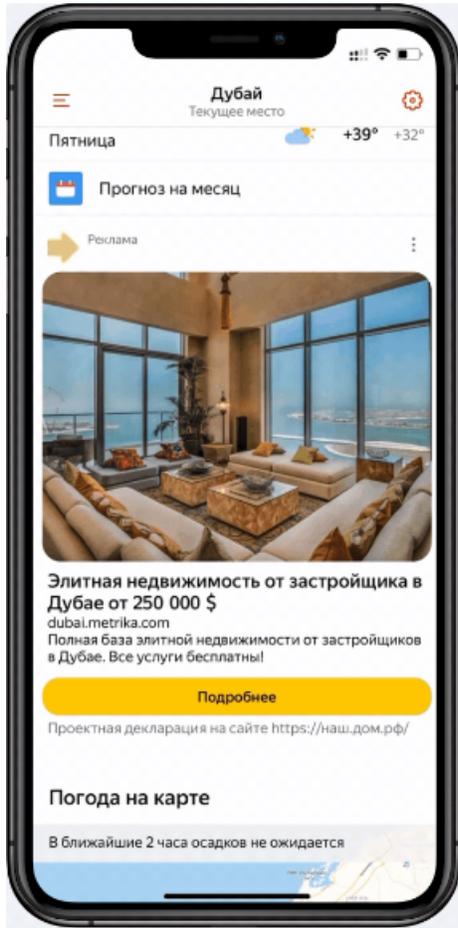
**feedback**

Значок меню. Обязательный компонент. С помощью значка меню  пользователь может скрыть объявление или пожаловаться на него.

Значок меню добавляется в правый верхний угол объявления.

**Примечание:**

Разработчику заранее необходимо определить что делать с объявлением после выбора причины закрытия (например, скрыть объявление или показать какую-то надпись). Если дальнейшее действие не определено, SDK учтет причину закрытия, но объявление не будет скрыто.



Варианты оформления:

**feedback\_dark\_dots\_with\_background**

Белый значок меню, с темными точками и полупрозрачной подложкой. Значение по умолчанию.

**feedback\_light\_dots**

Значок меню без фона, со светлыми точками.

Пример кода:

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

**feedback\_dark\_dots**

Значок меню без фона, с темными точками.

Пример кода:

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

**media**

Subview для медиаконтента (изображение или видео).

Логика показа медиаконтента в mediaView: если рекламный ответ содержит медиаконтент, mediaView отобразит его после буферизации.

**Совет:**

1. Проверить наличие медиаконтента можно через свойство `media` объекта `YMANativeAdAssets`. Если в рекламном ответе содержится медиаконтент, свойство вернет ненулевой объект `YMANativeAdMedia`.
2. Проверить наличие картинки можно через свойство `image` объекта `YMANativeAdAssets`. Если в рекламном ответе содержится картинка, свойство вернет ненулевой объект `YMANativeAdImage`.

**Загрузка рекламы****Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

**Загрузка рекламных объявлений**

1. Создайте экземпляр класса `YMANativeAdLoader` для получения нативных объявлений.
2. Создайте конфигурацию запроса `nativeAdRequestConfiguration` с помощью класса `YMANativeAdRequestConfiguration`. В качестве параметров запроса можно передать идентификатор рекламного блока, способ загрузки изображений, возраст, гендерные признаки и другие данные, способные улучшить качество подбора рекламы.
3. Установите делегат для получения рекламы, который реализует протокол `YMANativeAdLoaderDelegate`.
4. Чтобы отслеживать процесс загрузки рекламы, реализуйте методы протокола `YMANativeAdLoaderDelegate`: `-nativeAdLoader:didFailLoadingWithError:`, `-nativeAdLoader:didLoadAd:`.
5. Отправьте загрузчику сообщение `loadAdWithRequestConfiguration:`, чтобы загрузить рекламу.

**Swift**

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

**Objective-C**

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

6. Если реклама загрузилась, будет вызван метод:

**Swift**

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

**Objective-C**

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

7. Если реклама не загрузилась, будет вызван следующий метод:

**Swift**

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

**Objective-C**

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

Подробнее о возможных ошибках см. раздел [YMANativeErrorCode](#).

**Общий запрос за рекламой****Swift**

```
// Создание загрузчика
adLoader = YMANativeAdLoader()
adLoader.delegate = self

// Создание конфигурации запроса
let requestConfiguration = YMANativeAdRequestConfiguration(adUnitID: "<AdUnitID>")

// Передача загрузчику конфигурации запроса
adLoader.loadAd(with: requestConfiguration)

// Реализация методов делегата
....
```

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

### Objective-C

```
// Создание загрузчика
self.adLoader = [[YMANativeAdLoader alloc] init];
self.adLoader.delegate = self;

// Создание конфигурации запроса
YMANativeAdRequestConfiguration *requestConfiguration =
    [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:@"your_AdUnitID"];

// Передача загрузчику конфигурации запроса
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];

// Реализация методов делегата
...
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
    // Отобразите рекламу
}
```

### Примеры с демонстрационным AdUnitID

Чтобы посмотреть, как реклама будет отображаться в приложении, используйте демонстрационный AdUnitID:

- для рекламы типа App Install — demo-native-app-yandex;
- для рекламы типа Content — demo-native-content-yandex;
- для рекламы с видео — demo-native-video-yandex.

### Загрузка нескольких рекламных объявлений

Yandex Mobile Ads SDK предоставляет возможность загрузки нескольких рекламных объявлений одним запросом (до девяти объявлений).

1. Создайте экземпляр класса [YMANativeBulkAdLoader](#) для получения нативных объявлений.
2. Создайте конфигурацию запроса `nativeAdRequestConfiguration` с помощью класса [YMANativeAdRequestConfiguration](#). В качестве параметров запроса можно передать идентификатор рекламного блока, способ загрузки изображений, возраст, гендерные признаки и другие данные, способные улучшить качество подбора рекламы.
3. Установите делегат для получения рекламы, который реализует протокол [YMANativeBulkAdLoaderDelegate](#).
4. Чтобы отслеживать процесс загрузки рекламы, реализуйте методы протокола [YMANativeBulkAdLoaderDelegate](#): `-nativeBulkAdLoader:didLoadAds;`, `-nativeBulkAdLoader:didFailLoadingWithError;`.
5. Отправьте загрузчику конфигурацию запроса и количество запрашиваемых объявлений (параметр `adsCount`).

### Swift

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

### Objective-C

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

### Swift

```
// Создание конфигурации запроса
let requestConfiguration = YMANutableNativeAdRequestConfiguration(adUnitID: AdUnitID)

// Создание загрузчика
adLoader = YMANativeBulkAdLoader()
adLoader.delegate = self

// Передача загрузчику конфигурации запроса и количество запрашиваемых рекламных объявлений
adLoader.loadAds(with: requestConfiguration, adsCount: adsCount)

// Реализация методов делегата
func nativeBulkAdLoader(_ nativeBulkAdLoader: YMANativeBulkAdLoader, didLoad ads: [YMANativeAd]) {
```

```

    ...
    // Отдельная работа с каждым объектом id<YMANativeAd>. Подробности в разделе
    "Настройка внешнего оформления рекламы".
}

```

### Objective-C

```

// Создание конфигурации запроса
YMANativeAdRequestConfiguration *requestConfiguration =
    [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:AdUnitID];

// Создание загрузчика
self.adLoader = [[YMANativeBulkAdLoader alloc] init];
self.adLoader.delegate = self;

// Передача загрузчику конфигурации запроса и количество запрашиваемых рекламных объявлений
[self.adLoader loadAdsWithRequestConfiguration:requestConfiguration adsCount:adsCount];

// Реализация методов делегата
- (void)nativeBulkAdLoader:(YMANativeBulkAdLoader *)nativeBulkAdLoader didLoadAds:(NSArray<id<YMANativeAd>>
*)ads
{
    ...
    // Отдельная работа с каждым объектом id<YMANativeAd>. Подробности в разделе
    "Настройка внешнего оформления рекламы".
}

```

### Примечание:

Yandex Mobile Ads SDK не гарантирует, что будет загружено запрошенное количество объявлений. Полученный массив будет содержать от 0 до adsCount объектов NativeAd. Все полученные объекты рекламы можно показывать отдельно друг от друга, используя описанные выше способы внешнего оформления нативных объявлений.

## Способы загрузки изображений

### Автоматическая загрузка

Если приложение одновременно хранит ссылки на одно или небольшое количество объявлений, рекомендуется использовать автоматическую загрузку.

Передайте YES в качестве значения свойства [shouldLoadImagesAutomatically](#) при создании конфигурации:

#### Swift

```

let requestConfiguration = YMANutableNativeAdRequestConfiguration(adUnitID: AdUnitID)
requestConfiguration.shouldLoadImagesAutomatically = true

```

#### Objective-C

```

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

```

В полученном объявлении будут присутствовать готовые изображения. Они хранятся в памяти приложения до уничтожения объявления.

### Ручная загрузка

Если приложение одновременно хранит ссылки на большое количество объявлений, рекомендуется использовать ручную загрузку изображений.

Передайте NO в качестве значения свойства [shouldLoadImagesAutomatically](#) при создании конфигурации:

#### Swift

```

let requestConfiguration = YMANutableNativeAdRequestConfiguration(adUnitID: AdUnitID)
requestConfiguration.shouldLoadImagesAutomatically = false

```

#### Objective-C

```

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

```

В полученном объявлении будут присутствовать только размеры изображений. Чтобы загрузить сами изображения, вызовите метод [loadImages](#) у полученного объявления.



### Внимание:

Все изображения кэшируются, но могут быть удалены в любой момент, поэтому необходимо вызывать метод [loadImages](#) перед каждым показом объявления.

#### 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];
}
```

## Уведомления о процессе загрузки изображений

**Ограничение:** Получение уведомлений работает только при использовании ручной загрузки изображений.

### Включить уведомления

Чтобы включить уведомления о завершении загрузки изображений, вызовите метод [-addImageLoadingObserver](#):

#### 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");
}
```

### Отключить уведомления

Чтобы отключить уведомления о завершении загрузки изображений, вызовите метод [-removeImageLoadingObserver](#):

#### Swift

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

#### Objective-C

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

## Слайдер рекламных объявлений



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Yandex Mobile Ads SDK предоставляет возможность показывать слайдер из связанных между собой рекламных объявлений. Подробнее про слайдер можно узнать в [статье](#).

Слайдер реализован по принципу работы нативной рекламы. Внешний вид рекламы может настраиваться публишером в зависимости от функций и дизайна приложения, в котором будет отображаться слайдер.

## Загрузка слайдера

1. Создайте экземпляр класса [YMANativeAdLoader](#) для получения рекламных объявлений в рамках слайдера.
2. Создайте конфигурацию запроса `nativeAdRequestConfiguration` с помощью класса [YMANativeAdRequestConfiguration](#). В качестве параметров запроса можно передать идентификатор рекламного блока, способ загрузки изображений, возраст, гендерные признаки и другие данные, способные улучшить качество подбора рекламы.
3. Установите делегат для получения рекламы, который реализует протокол [YMANativeAdLoaderDelegate](#).
4. Чтобы отслеживать процесс загрузки рекламы, реализуйте методы протокола [YMANativeAdLoaderDelegate](#): `-nativeAdLoader:didFailLoadingWithError:`, `-nativeAdLoader:didLoadAd:`.
5. Отправьте загрузчику сообщение `loadAdWithRequestConfiguration:`, чтобы загрузить рекламу.

### Swift

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

### Objective-C

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

6. Если реклама загрузилась, будет вызван метод:

### Swift

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

### Objective-C

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

7. Если реклама не загрузилась, будет вызван следующий метод:

### Swift

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

### Objective-C

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

Подробнее о возможных ошибках см. раздел [YMANativeErrorCode](#).

## Общий запрос за рекламой

### Swift

```
// Создание загрузчика
adLoader = YMANativeAdLoader()
adLoader.delegate = self

// Создание конфигурации запроса
let requestConfiguration = YMANativeAdRequestConfiguration(adUnitID: "<AdUnitID>")

// Передача загрузчику конфигурации запроса
adLoader.loadAd(with: requestConfiguration)

// Реализация методов делегата
....

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

### Objective-C

```
// Создание загрузчика
self.adLoader = [[YMANativeAdLoader alloc] init];
self.adLoader.delegate = self;

// Создание конфигурации запроса
YMANativeAdRequestConfiguration *requestConfiguration =
    [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:@"your_AdUnitID"];

// Передача загрузчику конфигурации запроса
[self.adLoader loadAdWithRequestConfiguration:requestConfiguration];
```

```
// Реализация методов делегата
...
- (void)nativeAdLoader:(YMANativeAdLoader *)loader didLoadAd:(id<YMANativeAd>)ad
{
    // Отобразите рекламу
}
```

## Показ слайдера

После окончания загрузки слайдера, все его компоненты необходимо отобразить.

Успешно загруженный слайдер содержит одно или несколько связанных между собой нативных объявлений. Показ рекламных объявлений в рамках слайдера должен происходить в пределах одного общего контейнера, иначе показ объявлений не будет засчитан.

## Оформление без использования шаблона

Ручная настройка внешнего оформления нативной рекламы используется в тех случаях, когда возможностей настройки шаблона недостаточно для получения желаемого результата.

Данный способ позволяет самостоятельно сверстать макет нативной рекламы, определить расположение элементов рекламы относительно друг друга. В объявлении могут присутствовать как обязательные, так и опциональные для показа компоненты. Полный их перечень можно найти в разделе [Компоненты нативной рекламы](#).

### Совет:

Рекомендуется использовать макет, который включает весь набор возможных компонентов. Как показывает практика, макеты, включающие весь набор компонентов, более кликабельные.

Вызовите метод [bindAdToSliderView](#) и передайте в него контейнер для слайдера объявлений.

Внешнее оформление каждого объявления, входящего в слайдер, настраивается одним из [способов оформления](#) стандартной нативной рекламы.

### 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
    }
}

```

**Настройка внешнего оформления рекламы****Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Существует два способа настройки внешнего оформления рекламы:

**1. Оформление с помощью шаблона.**

Использование стандартного шаблона оформления — это самый простой способ работы с нативной рекламой. В шаблоне уже добавлен набор обязательных рекламных компонентов и настроено их расположение относительно друг друга. Шаблон работает с любым поддерживаемым типом рекламы.

**2. Оформление без использования шаблона.**

Оформление без шаблона используется в тех случаях, когда для настройки шаблона недостаточно функций, чтобы реклама выглядела органично и соответствовала дизайну приложения.

**Оформление с помощью шаблона****Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Для настройки внешнего оформления можно использовать стандартный шаблон оформления или создать свое оформление на основе стандартного шаблона.

**Использование стандартного шаблона оформления****Примечание:**

После настройки оформления задайте [местоположение и размеры рекламы](#) относительно экрана устройства.

1. Создайте объект — экземпляр класса [YMANativeBannerView](#) и установите для него загруженную рекламу:

**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. Чтобы получать уведомления о взаимодействии пользователя с рекламой (открытие рекламы, выход из приложения), установите для нее делегат [YMANativeAdDelegate](#), реализующий методы:

- [-nativeAd:didDismissScreen::](#);
- [-nativeAd:willPresentScreen::](#);
- [-nativeAdWillLeaveApplication::](#);
- [-viewControllerForPresentingModalView.](#)

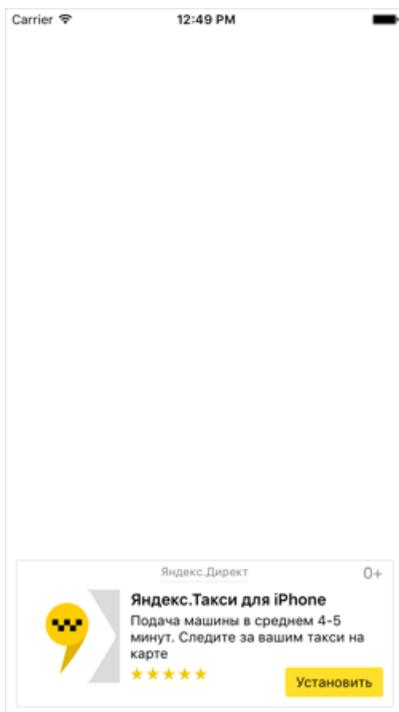
**Swift**

```
ad.delegate = self
```

**Objective-C**

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

3. Пример использования стандартного шаблона оформления:



**Примечание:** Если вашему приложению не подходит стандартное оформление, его можно изменить. Подробнее смотрите в разделе [Создание своего оформления на основе шаблона](#).

**Создание своего оформления на основе шаблона**

**Примечание:**

После настройки оформления задайте [местоположение и размеры рекламы](#) относительно экрана устройства.

1. Создайте объект — экземпляр класса [YMANativeBannerView](#) и установите для него загруженную рекламу:

#### 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. Чтобы получать уведомления о взаимодействии пользователя с рекламой (открытие рекламы, выход из приложения), установите для нее делегат [YMANativeAdDelegate](#), реализующий методы:

- [-nativeAd:didDismissScreen::](#);
- [-nativeAd:willPresentScreen::](#);
- [-nativeAdWillLeaveApplication::](#);
- [-viewControllerForPresentingModalView.](#)

#### Swift

```
ad.delegate = self
```

#### Objective-C

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

3. Запросите настройки стандартного шаблона оформления:

#### Swift

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

#### Objective-C

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

4. Задайте предпочитаемые настройки.
5. Чтобы применить настройки к шаблону, вызовите метод [-applyAppearance::](#)

#### Swift

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

#### Objective-C

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

## Пример настроек оформления

### Swift

```
// Определяем собственные цветовые решения для работы.
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)

// Создаем копию с настройками стандартного шаблона оформления.
let appearance = YMAMutableNativeTemplateAppearance.default()

// Начинаем изменять стандартные настройки шаблона.

// Задаем цвет для рамки рекламного объявления.
appearance.borderColor = orangeColor

// Создаем копию с настройками рейтинга.
let ratingAppearance = appearance.ratingAppearance?.mutableCopy() as? YMAMutableRatingAppearance

// Задаем цвет для закрасненных звезд в рейтинге.
```

```

ratingAppearance?.filledStarColor = orangeColor
appearance.ratingAppearance = ratingAppearance

// Задаем цвет и размер шрифта для надписи на кнопке с действием.
let callToActionTextAppearance = YMALabelAppearance(font: .systemFont(ofSize: 14), textColor: blueColor)

// Задаем цвет кнопки для обычного и нажатого состояния, цвет и толщину обводки кнопки.
let callToActionAppearance = YMAButtonAppearance(
    textAppearance: callToActionTextAppearance,
    normalColor: .clear,
    highlightedColor: .gray,
    borderColor: blueColor,
    borderWidth: 1
)
appearance.callToActionAppearance = callToActionAppearance

// Задаем размер и цвет шрифта для надписи с возрастным ограничением.
appearance.ageAppearance = YMALabelAppearance(font: .systemFont(ofSize: 12), textColor: .gray)

// Задаем размер и цвет шрифта для заголовка рекламного объявления.
appearance.titleAppearance = YMALabelAppearance(font: .systemFont(ofSize: 14), textColor: .black)

// Задаем размер и цвет шрифта для основного рекламного текста.
appearance.bodyAppearance = YMALabelAppearance(font: .systemFont(ofSize: 12), textColor: .gray)

// Задаем ширину изображения и правило формирования размера.
let imageConstraint = YMASizeConstraint(type: .fixed, value: 60)

// Применяем настройки к изображению.
appearance.imageAppearance = YMAImageAppearance(widthConstraint: imageConstraint)

```

### Objective-C

```

// Определяем собственные цветовые решения для работы.
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];

// Создаем копию с настройками стандартного шаблона оформления.
YMAMutableNativeTemplateAppearance *appearance = [[YMANativeTemplateAppearance defaultAppearance] mutableCopy];

// Начинаем изменять стандартные настройки шаблона.

// Задаем цвет для рамки рекламного объявления.
appearance.borderColor = orangeColor;

// Создаем копию с настройками рейтинга.
YMAMutableRatingAppearance *ratingAppearance = [appearance.ratingAppearance mutableCopy];

// Задаем цвет для покрашенных звезд в рейтинге.
ratingAppearance.filledStarColor = orangeColor;
appearance.ratingAppearance = ratingAppearance;

// Задаем цвет и размер шрифта для надписи на кнопке с действием.
YMALabelAppearance *callToActionTextAppearance =
    [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:14.f]
                                     textColor:blueColor];

// Задаем цвет кнопки для обычного и нажатого состояния, цвет и толщину обводки кнопки.
YMAButtonAppearance *callToActionAppearance =
    [YMAButtonAppearance appearanceWithTextAppearance:callToActionTextAppearance
                                     normalColor:[UIColor clearColor]
                                     highlightedColor:[UIColor grayColor]
                                     borderColor:blueColor
                                     borderWidth:1.f];
appearance.callToActionAppearance = callToActionAppearance;

// задаем размер и цвет шрифта для надписи с возрастным ограничением.
appearance.ageAppearance =
    [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:12.f]
                                     textColor:[UIColor grayColor]];

// Задаем размер и цвет шрифта для заголовка рекламного объявления.
appearance.titleAppearance =
    [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:14.f]
                                     textColor:[UIColor blackColor]];

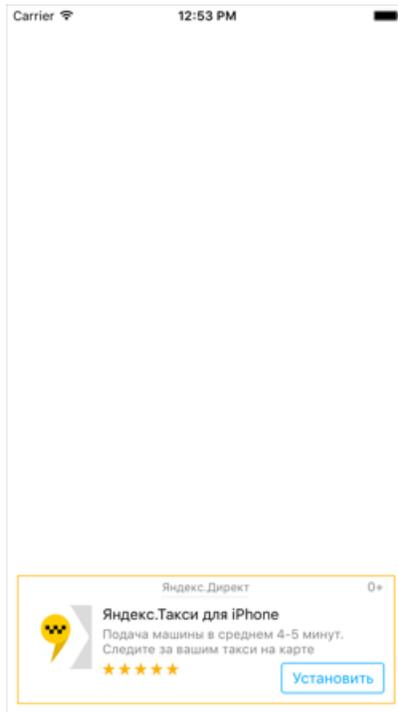
// Задаем размер и цвет шрифта для основного рекламного текста.
appearance.bodyAppearance =
    [YMALabelAppearance appearanceWithFont:[UIFont systemFontOfSize:12.f]
                                     textColor:[UIColor grayColor]];

// Задаем ширину изображения и правило формирования размера.
YMASizeConstraint *imageConstraint =
    [YMASizeConstraint constraintWithType:YMASizeConstraintTypeFixed value:60.f];

// Применяем настройки к изображению.
appearance.imageAppearance =
    [YMAImageAppearance appearanceWithWidthConstraint:imageConstraint];

```

Получаем собственный дизайн на основе шаблона:



### Настройка местоположения и размера рекламы

#### Ограничение: Требования к размерам `mediaView` при отображении видеорекламы

Минимальный размер экземпляра класса `YMANativeMediaView`, в котором поддерживается воспроизведение видео: 300x160 или 160x300.

Для поддержки воспроизведения видео в шаблонах нативной рекламы рекомендуется выставить ширину для `YMANativeBannerView` не менее 300. Корректная высота для `mediaView` будет вычислена автоматически, с учетом соотношения ширины и высоты.

Управлять размерами и местоположением можно одним из двух способов:

1. с помощью системного механизма `AutoLayout`;

#### Примечание:

При работе с `AutoLayout` задайте `constraint` для ширины `UIView`. Высота будет определена автоматически на основании указанной ширины.

2. вручную, задавая все размеры.

### Настройка размеров вручную

#### Примечание:

Не рекомендуется задавать ширину объекта более 420 логических пикселей.

Задайте для объекта `YMANativeBannerView` ширину и высоту. Высота определяется с помощью метода `+heightWithAd:width:appearance:`, в который необходимо передать рекламное объявление, ширину и объект `YMANativeTemplateAppearance` (используется для настроек внешнего вида рекламного объявления).

#### Swift

```
// Задаем отступы слева и справа относительно экрана.
let inset: CGFloat = 50

// Задаем ширину.
let width = view.frame.width - 2 * inset

// Задаем высоту.
let height = YMANativeBannerView.height(with: ad, width: width, appearance: nil)

// Задаем позицию по вертикали.
let y = view.frame.maxY - height - inset

// Задаем координаты и размеры для frame.
```

```
bannerView.frame = CGRect(x: inset, y: y, width: width, height: height)
```

### Objective-C

```
// Задаем отступы слева и справа относительно экрана.
CGFloat inset = 50.f;

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

// Задаем высоту.
CGFloat height = [YMANativeBannerView heightForAd:ad width:width appearance:nil];

// Задаем позицию по вертикали.
CGFloat y = CGRectGetMaxY(self.view.frame) - height - inset;

// Задаем координаты и размеры для frame.
bannerView.frame = CGRectMake(inset, y, width, height);
```

### Оформление без использования шаблона



#### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Для отображения нативной рекламы объекты `YMANativeAd` должны быть привязаны к определенной `View`. Эта `View` должна определяться как корневой элемент для набора `subview`, к элементам которого будут привязываться данные, содержащиеся в нативной рекламе.

#### Ограничение: Требования к размерам `mediaView` при отображении видеорекламы

Минимальный размер экземпляра класса `YMANativeMediaView`, в котором поддерживается воспроизведение видео: 300x160 или 160x300.

Для поддержки воспроизведения видео в нативной рекламе рекомендуется выставить ширину для `mediaView` не менее 300. Чтобы вычислить соответствующее значение высоты `mediaView`, используйте значение свойства `aspectRatio`.

### Оформление нативной рекламы

1. Ознакомьтесь со [списком](#) обязательных и опциональных `subview` для нативной рекламы.
2. Установите значения всех обязательных `subview`. Это можно сделать с помощью `initWithFrame` или непосредственно в коде. Для этого создайте объект класса `YMANativeAdView` и определите значения `subview` в `initWithFrame`:

#### Пример регистрации набора `subview`:

##### 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;
}
```

- Чтобы получать уведомления о взаимодействии пользователя с рекламой (открытие и закрытие рекламы, выход из приложения), установите для нее делегат [YMANativeAdDelegate](#), реализующий методы:
  - closeNativeAd;
  - nativeAd:didDismissScreen;
  - nativeAd:willPresentScreen;
  - nativeAdWillLeaveApplication;
  - viewControllerForPresentingModalView.
- Запросите значения компонентов нативной рекламы с помощью метода [-adAssets](#). Это поможет заранее рассчитать расположение и размеры этих компонентов.

**Swift**

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

**Objective-C**

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

**Пример получения размеров изображения, соотношения сторон медиа и текста заголовка объявления****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. Вызовите метод `-bindWithAdView:error:`, чтобы связать контент с объектом нативной рекламы.

#### 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);
    }
}
```

#### Примечание:

Если для обязательного элемента нативной рекламы соответствующее свойство `YMANativeAdView` принимает значение `nil`, привязка не осуществится — реклама не будет показана. Детали в этом случае можно получить из `error`.

## Отладка



#### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

## Логирование

Опционально можно включить логирование с помощью метода `+enableLogging`. Если показ рекламы не был засчитан, в консоли появится сообщение.

## Индикатор некорректной интеграции для нативной рекламы

Если при интеграции нативной рекламы допущена ошибка, то в режиме симулятора, поверх рекламного объявления, появится индикатор. По нажатию на индикатор вы увидите сообщение с отладочной информацией, по которой можно понять причину ошибки. Повторное нажатие на индикатор скрывает сообщение.



#### Внимание:

По умолчанию индикатор отображается только в режиме симулятора (тип устройства `YMADeviceTypeSimulator`). Типы устройства можно посмотреть в [YMADeviceType](#).

Чтобы включить индикатор также и для реальных устройств, передайте значение `YMADeviceTypeHardware | YMADeviceTypeSimulator` в методе `enableVisibilityErrorIndicatorForDeviceType::`

#### Swift

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

#### Objective-C

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

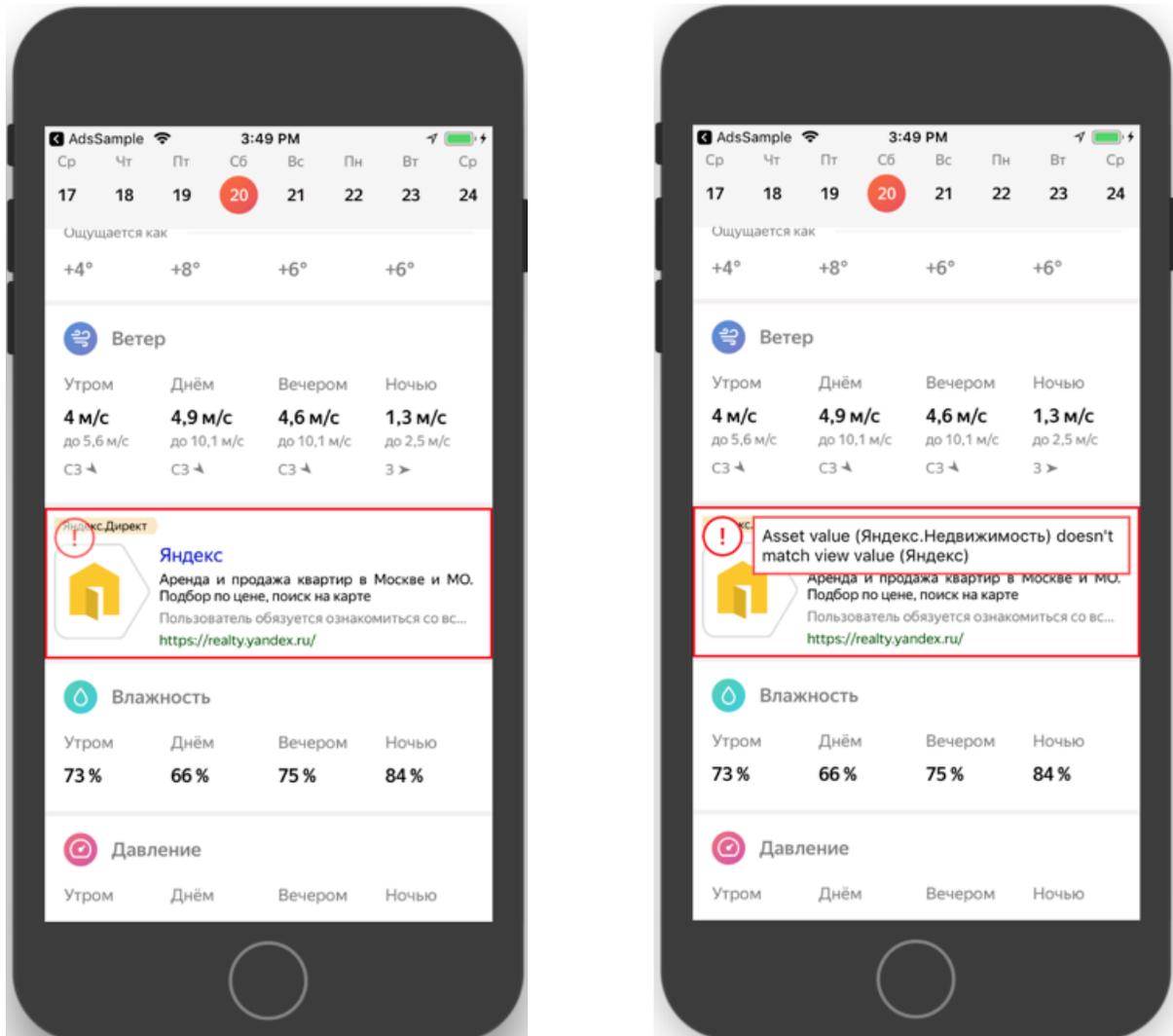
Чтобы выключить индикатор, передайте значение `YMADeviceTypeNone` в методе `enableVisibilityErrorIndicatorForDeviceType::`

**Swift**

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

**Objective-C**

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



**Классы и протоколы для работы с нативной рекламой**



**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

**Классы**

- [YMAButtonAppearance](#)
- [YMAImageAppearance](#)
- [YMALabelAppearance](#)
- [YMANativeAdAssets](#)
- [YMANativeAdImage](#)
- [YMANativeAdLoader](#)

- [YMANativeAdMedia](#)
- [YMANativeAdView](#)
- [YMANativeBannerView](#)
- [YMANativeMediaView](#)
- [YMANativeTemplateAppearance](#)
- [YMARatingAppearance](#)
- [YMASizeConstraint](#)

#### Протоколы

- [YMANativeAd](#)
- [YMANativeAdDelegate](#)
- [YMANativeAdImageLoadingObserver](#)
- [YMANativeAdLoaderDelegate](#)
- [YMARating](#)

## InStream реклама



#### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

*InStream* — формат рекламы, который позволяет монетизировать приложение через показ рекламы во время воспроизведения видеоконтента.

InStream объявление состоит из сценария с несколькими видеоблоками. Тип видеоблока в сценарии InStream определяет, как должен проигрываться рекламный ролик относительно основного видеоконтента.

## Об InStream

Чтобы настроить сценарий InStream, [создайте видеоресурс](#) в Партнерском интерфейсе. После создания видеоресурсу будет присвоен уникальный идентификатор (Page ID). Этот идентификатор необходимо использовать в Mobile Ads SDK.

Для увеличения рекламного дохода, есть возможность настроить показ нескольких рекламных объявлений в рамках одной рекламной вставки, так называемый AdPod. Настроить AdPod для рекламной вставки можно в Партнерском Интерфейсе.

Типы видеоблоков, которые поддерживает Mobile Ads SDK:

- Pre-Roll — рекламный ролик проигрывается перед основным контентом;
- Mid-Roll — рекламный ролик проигрывается по времени внутри основного контента;
- Post-Roll — рекламный ролик проигрывается после основного контента;
- Pause-Roll — рекламный ролик проигрывается при нажатии кнопки паузы;
- In-Roll — рекламный ролик проигрывается в любом месте видео при достижении определенной отметки.



#### 1. In-roll

### API для работы с InStream

API для [интеграции](#) InStream рекламы позволяет поддерживать проигрывание всех типов рекламных вставок: Pre-Roll, Mid-Roll, Post-Roll, In-Roll и Pause-Roll.

Для проигрывания вставок с типом Pre-Roll, Mid-Roll, Post-Roll используется InstreamAdBinder API. Для проигрывания вставок с типом In-Roll и Pause-Roll используются In-Roll API и Pause-Roll API соответственно.

#### Примечание:

Доступно одновременное использование InstreamAdBinder API, In-Roll API и Pause-Roll API при соблюдении определенных условий:

1. Используйте разные экземпляры рекламного плеера.
2. Не запускайте Pause-Roll и In-Roll API для воспроизведения, если через InStreamAdBinder API было приостановлено основное видео.

### Интеграция InStream API



#### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

InStream API — API для настройки, управления загрузкой и воспроизведением InStream рекламы. Позволяет поддерживать проигрывание всех типов рекламных вставок: Pre-Roll, Mid-Roll, Post-Roll, In-Roll и Pause-Roll.

Для проигрывания вставок с типом Pre-Roll, Mid-Roll, Post-Roll используется InstreamAdBinder API. Для проигрывания вставок с типом In-Roll и Pause-Roll используются In-Roll API и Pause-Roll API соответственно.

#### Примечание:

Доступно одновременное использование InstreamAdBinder API, In-Roll API и Pause-Roll API при соблюдении определенных условий:

1. Используйте разные экземпляры рекламного плеера.
2. Не запускайте Pause-Roll и In-Roll API для воспроизведения, если через InStreamAdBinder API было приостановлено основное видео.

## Принципы работы

Загруженный объект InStream рекламы содержит в себе расписание показа рекламных вставок. Каждая рекламная вставка описывается объектом InstreamAdBreak. Рекламная вставка может иметь один из следующих типов: Pre / Mid / Post / In / Pause-Roll. Показать Pre/MidPost-Roll вставки можно через InstreamAdBinder API. Показать In / Pause-Roll вставки можно через In-Roll API и Pause-Roll API соответственно.

Для взаимодействия с основным видеоконтентом используется протокол [VideoPlayer](#), а для воспроизведения рекламного видео внутри рекламной вставки — протокол [InstreamAdPlayer](#).

### Показ Pre / Mid / Post-Roll

InstreamAdBinder отслеживает ход воспроизведения основного видео и автоматически показывает рекламные вставки на основе расписания рекламных вставок из видеоресурса в Партнерском Интерфейсе.

InstreamAdBinder не управляет непосредственно отрисовкой рекламного видео в `PlayerView`. Воспроизводить рекламное видео необходимо со стороны приложения основываясь на сигналах от интерфейсов плееров, переданных в InstreamAdBinder. InstreamAdBinder сообщает о начале проигрывания рекламной вставки через вызов [VideoPlayer.pauseVideo\(\)](#) и об окончании проигрывания рекламной вставки через вызов [VideoPlayer.resumeVideo\(\)](#).

В момент вызова [VideoPlayer.pauseVideo\(\)](#) со стороны приложения необходимо скрыть контролы управления основным видео, приостановить основное видео и начать воспроизведение рекламного видео. Со стороны рекламного SDK после вызова метода будут показаны рекламные контролы внутри контейнера `InstreamAdView` и будет вызван метод [InstreamAdPlayer.playAd\(\)](#) для старта воспроизведения рекламного видео.

В момент вызова [VideoPlayer.resumeVideo\(\)](#) со стороны приложения необходимо вернуть контролы управления основным видео и продолжить воспроизведение основного видео. Со стороны рекламного SDK до вызова метода будут убраны рекламные контролы внутри контейнера `InstreamAdView`.

### Показ In / Pause-Roll

In / Pause-Roll API не управляет непосредственно отрисовкой рекламного видео в `PlayerView`. Воспроизводить рекламное видео необходимо со стороны приложения основываясь на сигналах от интерфейсов плееров, переданных в In / Pause-Roll. In / Pause-Roll сообщает о начале проигрывания рекламной вставки через вызов [InstreamAdBreakDelegate.instreamAdBreakDidStart\(\)](#) и об окончании проигрывания рекламной вставки через вызов [InstreamAdBreakDelegate.instreamAdBreakDidComplete\(\)](#) или [InstreamAdBreakDelegate.instreamAdBreakDidError\(\)](#).

В момент вызова [InstreamAdBreakDelegate.instreamAdBreakDidStart\(\)](#) со стороны приложения необходимо скрыть контролы управления основным видео, приостановить основное видео. Со стороны рекламного SDK после вызова метода будут показаны рекламные контролы внутри контейнера `InstreamAdView` и будет вызван метод [InstreamAdPlayer.playAd\(\)](#) для старта воспроизведения рекламного видео.

В момент вызова [InstreamAdBreakDelegate.instreamAdBreakDidComplete\(\)](#) или [InstreamAdBreakDelegate.instreamAdBreakDidError\(\)](#) со стороны приложения необходимо вернуть контролы управления основным видео и продолжить воспроизведение основного видео. Со стороны рекламного SDK до вызова методов будут убраны рекламные контролы из контейнера `InstreamAdView`.

## Загрузка рекламных объявлений

1. Создайте экземпляр класса `InstreamAdLoader` для получения InStream рекламы.
2. Чтобы получать уведомления (успешная загрузка рекламы или ошибка при загрузке рекламы) подпишитесь на события загрузки рекламы. Для этого установите делегат, который удовлетворяет протоколу [InstreamAdLoaderDelegate](#).
3. Создайте конфигурацию запроса `instreamAdRequestConfiguration` с помощью класса [InstreamAdRequestConfiguration](#). В качестве параметров запроса передайте [идентификатор страницы](#) (Page ID) из Партнерского интерфейса.
4. Загрузите рекламу с помощью метода [InstreamAdLoader.loadInstreamAd](#), передайте в него `instreamAdRequestConfiguration`.

Для тестирования корректности интеграции используйте демонстрационный Page ID: demo-instream-vmар-yandex.

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

## Показ рекламных объявлений

### Показ Pre / Mid / Post-Roll

1. Реализуйте интерфейсы [InstreamAdPlayer](#) и [VideoPlayer](#).

В справочнике приведена подробная информация по работе методов и их реализации. Дополнительно рекомендуется ориентироваться на [тестовый пример реализации](#).



#### Внимание:

Для упрощения реализации, рекомендуется использовать разные экземпляры плееров для воспроизведения рекламы и контентного видео.

2. Добавьте в иерархию View приложения [InstreamAdView](#). [InstreamAdView](#) должна содержать в себе [PlayerView](#), в которой будут проигрываться рекламные ролики.

#### Ограничение:

Размер контейнера должен быть не меньше 300dp x 160dp.

3. Создайте объект [InstreamAdBinder](#): передайте в конструктор объект загруженной рекламы [InstreamAd](#) и реализации плееров [InstreamAdPlayer](#), [VideoPlayer](#).

Настройте получение уведомлений о ходе воспроизведения рекламы (готовность к проигрыванию видеорекламы, завершение воспроизведения или ошибка в процессе воспроизведения): установите делегат, который удовлетворяет протоколу [InstreamAdBinderDelegate](#).

```
adBinder = InstreamAdBinder(ad: ad, adPlayer: adPlayer, videoPlayer:
contentPlayer)
adBinder.delegate = self
```

4. Для ускорения старта рекламной вставки с типом Pre-Roll, заранее предзагрузите ее через вызов метода [InstreamAdBinder.prepareAd\(\)](#).

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

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

5. Вызовите метод [InstreamAdBinder.bind\(with adView: InstreamAdView\)](#) у созданного объекта [InstreamAdBinder](#). В качестве параметра передайте [InstreamAdView](#), ранее добавленную в иерархию. После этого InStream SDK начнет автоматически отслеживать прогресс воспроизведения основного видео и возьмет на себя управление проигрыванием рекламных роликов.

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

6. При реализации проигрывания InStream рекламы в списке, необходимо использовать метод [InStreamBinder.unbind\(\)](#) при инвалидации ячейки с рекламой в списке. При реализации переиспользуемого пула плееров для скролла, необходимо вызывать [InstreamAdBinder.invalidateAdPlayer\(\)](#) при переиспользовании плеера рекламы привязанного к [InstreamAdBinder](#), а при переиспользовании плеера основного контента [InstreamAdBinder.invalidateVideoPlayer\(\)](#).
7. При остановке использования [InStreamAdBinder](#) необходимо очищать состояние.

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

### Показ In / Pause-Roll

#### Примечание:

Показ рекламной вставки Pause-Roll настраивается по аналогии с In-Roll. Для этого замените In-Roll классы / методы на Pause-Roll.

1. Реализуйте интерфейс [InstreamAdPlayer](#).

В справочнике приведена подробная информация по работе методов и их реализации. Дополнительно рекомендуется ориентироваться на [тестовый пример реализации](#).



**Внимание:**

Для упрощения реализации, рекомендуется использовать разные экземпляры плееров для воспроизведения рекламы и контентного видео.

2. Добавьте в иерархию View приложения `InstreamAdView`. `InstreamAdView` должна содержать в себе `PlayerView`, в которой будут проигрываться рекламные ролики.

**Ограничение:**

Размер контейнера должен быть не меньше 300dp x 160dp.

3. Загрузите через `InstreamAdLoader` объект `InstreamAd` с помощью [идентификатора страницы](#) (Page ID) из Партнерского интерфейса.

4. `InstreamAd` содержит в себе набор рекламных вставок разных типов. Чтобы получить рекламные вставки In-Roll, используйте [InrollQueueProvider](#). Очередь `InrollQueueProvider` позволяет получать объекты In-Roll в необходимом для показа порядке.

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

5. Чтобы запустить полученный объект In-Roll, необходимо его подготовить. Без подготовки In-Roll не запустится. Для отслеживания окончания подготовки In-Roll установите делегат [InstreamAdBreakDelegate](#), вызовите [Inroll.prepare\(with: adPlayer\)](#), и передайте в него инстанс созданной реализации `InstreamAdPlayer`.

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

6. По окончании подготовки In-Roll будет вызван [InstreamAdBreakDelegate.instreamAdBreakDidPrepare\(\)](#). Подготовленный In-Roll готов к показу.

**Совет:**

Показывайте объявления в порядке их получения из `InrollQueue`. Показ полученных In-Roll в другом порядке может снизить монетизацию вашего приложения.

7. Чтобы показать подготовленный In-Roll, вызовите [Inroll.play\(with adView: InstreamAdView\)](#) и передайте в качестве параметра `InstreamAdView`, ранее добавленный в иерархию view.

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

8. После начала проигрывания рекламной вставки, будет вызван метод [InstreamAdBreakDelegate.instreamAdBreakDidStart\(\)](#). По вызову этого метода необходимо поставить на паузу проигрывание основного видео и скрыть контролы управления основным видео.

```
func instreamAdBreakDidStart(_ adBreak: InstreamAdBreak) {
    contentVideoPlayer?.pauseVideo()
}
```

9. После проигрывания рекламной вставки необходимо продолжить воспроизведение основного видео. Проигрывание рекламной вставки может завершиться успешно или с ошибкой, эти две ситуации необходимо обрабатывать.

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

10. После завершения проигрывания текущего InRoll проверьте очередь воспроизведения на наличие следующего In-Roll в InrollQueue.

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

11. При остановке использования In-Roll необходимо очищать его состояние.

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

## Классы и протоколы для работы с InStream рекламой



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

### Классы

- [InrollQueueProvider](#)
- [InstreamAdBinder](#)
- [InstreamAdBreakPosition](#)
- [InstreamAdBreakPositionType](#)
- [InstreamAdBreakType](#)
- [InstreamAdLoader](#)
- [InstreamAdPlayerError](#)
- [InstreamAdRequestConfiguration](#)
- [InstreamVideoAd](#)
- [MobileInstreamAds](#)
- [PauserollQueueProvider](#)

### Протоколы

- [AdPodInfo](#)
- [Inroll](#)
- [InrollQueue](#)
- [InstreamAd](#)
- [InstreamAdBinderDelegate](#)
- [InstreamAdBreak](#)
- [InstreamAdBreakDelegate](#)
- [InstreamAdLoaderDelegate](#)
- [InstreamAdPlayer](#)
- [InstreamAdPlayerDelegate](#)
- [InstreamAdSkipInfo](#)
- [InstreamCustomClickConnector](#)
- [InstreamCustomClickHandler](#)
- [InstreamCustomClickable](#)
- [InstreamVideoAdsProvider](#)
- [MediaFile](#)
- [Pauseroll](#)
- [PauserollQueue](#)
- [SkipInfo](#)
- [VideoAd](#)
- [VideoPlayer](#)
- [VideoPlayerDelegate](#)

# GDPR

**Внимание:**

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

## Общие сведения

Весной 2018 года в силу вступил общий регламент по защите данных (General Data Protection Regulation, сокращенно GDPR). Регламент регулирует сбор и обработку информации о физических лицах — гражданах Европейской экономической зоны и Швейцарии. Он призван усилить защиту конфиденциальных данных и сделать прозрачными все элементы сбора, хранения и обработки информации в интернете.

GDPR имеет экстерриториальное действие и применяется ко всем компаниям, которые обрабатывают персональные данные граждан Европейской экономической зоны и Швейцарии, независимо от местонахождения такой компании.

Начиная с версии 2.11.0, Yandex Mobile Ads SDK позволяет ограничивать сбор данных пользователей, расположенных в Европейской экономической зоне и Швейцарии, при отсутствии их согласия на это.

## Краткое руководство

Согласие пользователя на обработку персональных данных необходимо передавать в SDK при каждом запуске приложения.

1. Подключите Mobile Ads SDK по [инструкции](#).
2. Отобразите для пользователя диалог, предлагающий принять пользовательское соглашение на обработку персональных данных (подробнее в [примере](#)).

**Внимание:**

Данный код является демонстрацией, а не руководством к действию.

**Swift**

```
...
// Код демонстрирует создание диалога.
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)
}
```

```
YMMobileAds.setUserConsent(userConsent)
}
```

### Objective-C

```
...
// Код демонстрирует создание диалога.
- (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:[NSURL URLWithString: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];
    [YMMobileAds setUserConsent:userConsent];
}
```

3. Передайте полученное значение в Mobile Ads SDK с помощью метода `+setUserConsent:`. Данные пользователей, находящихся в GDPR-регионе, будут обрабатываться только при наличии согласия пользователя на обработку данных.

## Поддержка iOS 14



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

## Поддержка SKAdNetwork

### Примечание:

SKAdNetwork поддерживается для версии SDK 4.1.2 и выше. Подключите Mobile Ads SDK по [инструкции](#).

Mobile Ads SDK поддерживает трекинг установок приложений с помощью фреймворка [SKAdNetwork](#). Трекинг установок работает для всех устройств, даже если доступ к IDFA отсутствует.

Чтобы включить функциональность, добавьте идентификаторы поддерживаемых рекламных сетей в файл `Info.plist` приложения.

```
<key>SKAdNetworkItems</key>
<array>
  <dict>
    <key>SKAdNetworkIdentifier</key>
    <string>zq4921623r.skadnetwork</string>
  </dict>
  <dict>
    <key>SKAdNetworkIdentifier</key>
    <string>633vhxsw4.skadnetwork</string>
  </dict>
</array>
```

```
</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>c6k4g5qq8m.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>294199pt4k.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>52f12v3hgg.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>yclnrxr15pm.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>8qiekg9qfv.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>v79kvwvj4g.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>r26jy69rp1.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>fkak3gftp6.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>t155sbb4fm.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>8s468mf13y.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>klf5c315u5.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>578prt vx9j.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>r45fhh6rf7.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>6p4ks3rnw.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>p78axw29g.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>pu4na253f3.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>s69wq72uqg.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>x5183yy675.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>y45688jllp.skadnetwork</string>
</dict>
<dict>
  <key>SKAdNetworkIdentifier</key>
  <string>yrrqpx2mcb.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>
```

Для дополнительной информации ознакомьтесь с разделом [Configuring a Source App](#) документации Apple.

## Изменения в iOS 14

С января 2021 года компания Apple [ограничивает доступ к IDFA на iOS 14](#). Это означает, что Apple перестанет отдавать IDFA по умолчанию. IDFA (Identifier for Advertisers) — уникальный идентификатор устройства, который используется для атрибуции и ретаргетинга рекламы.

Yandex Mobile Ads SDK использует IDFA, чтобы максимизировать доход площадок, за счёт отображения более релевантной рекламы.

Для получения IDFA разработчик приложения должен явно запросить разрешение у пользователя. Сделать это можно с помощью [AppTrackingTransparency framework](#).

Принцип работы аналогичен запросам на отправку push-уведомлений.

### Запрос разрешения на доступ к IDFA

При старте приложения проверьте статус разрешения с помощью [trackingAuthorizationStatus](#). Дальнейшие рекомендации зависят от того, установлен у пользователя запрет доступа к IDFA или нет.

#### **У пользователя не установлен запрет доступа к IDFA**

У приложения есть только одна возможность показать пользователю запрос на доступ к IDFA, поэтому важно убедить пользователя разрешить доступ.

Стратегия работы с пользователем:

1. Перед показом системного диалогового окна отобразите диалоговое окно с объяснением:

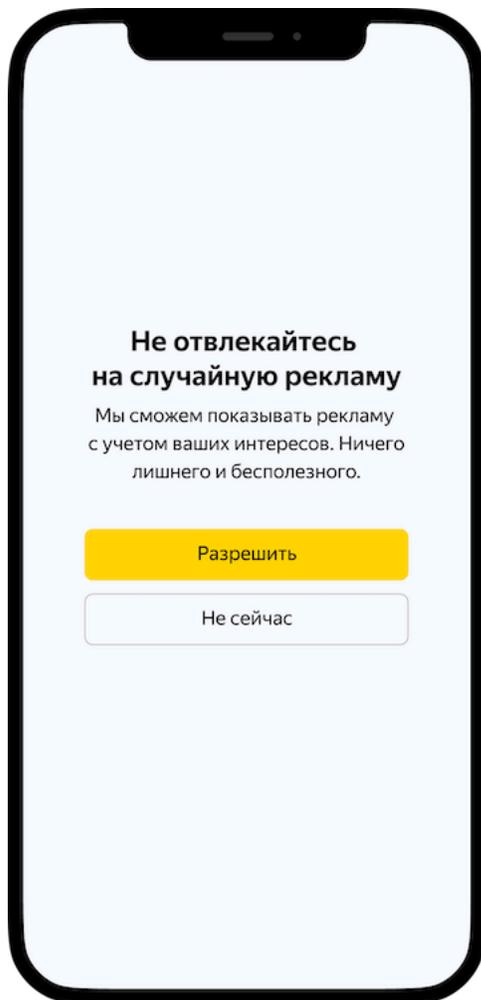
- что именно запрашивается;
- как это будет использоваться;
- почему стоит разрешить доступ.

На экран можно добавить кнопки **Разрешить** и **Не сейчас**.

**Примечание:**

Если пользователь согласился в поясняющем диалоговом окне приложения, с большой вероятностью он согласится в системном диалоговом окне. Если пользователь отказался в поясняющем диалоговом окне, будет возможность показать это диалоговое окно еще раз.

**Пример диалогового окна с объяснением**



2. Запросите доступ к IDFA через [App Tracking Transparency](#).

**У пользователя установлен запрет доступа к IDFA**

Возможна ситуация, когда пользователь отказался от предоставления доступа к IDFA в системном диалоговом окне.

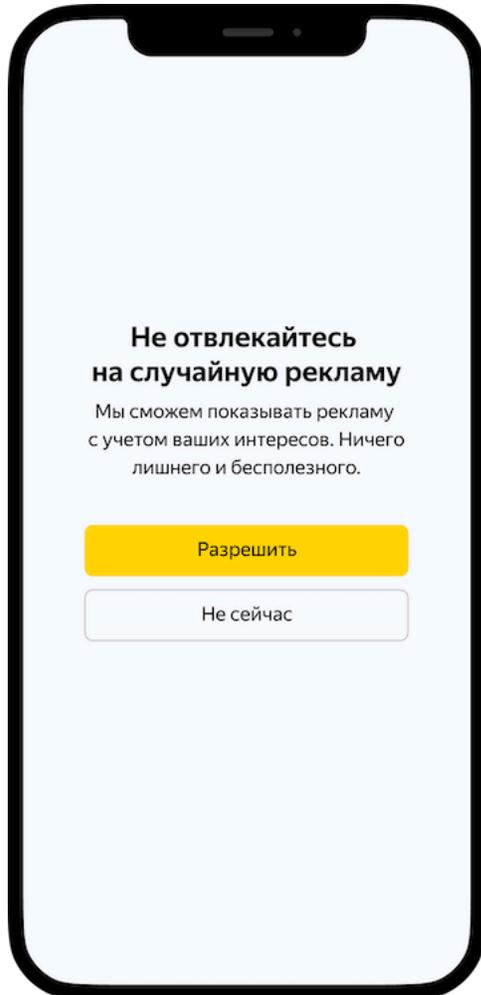
Стратегия работы с пользователем:

1. В приложении отобразите диалоговое окно с объяснением:

- что именно запрашивается;
- как это будет использоваться;
- почему стоит разрешить доступ.

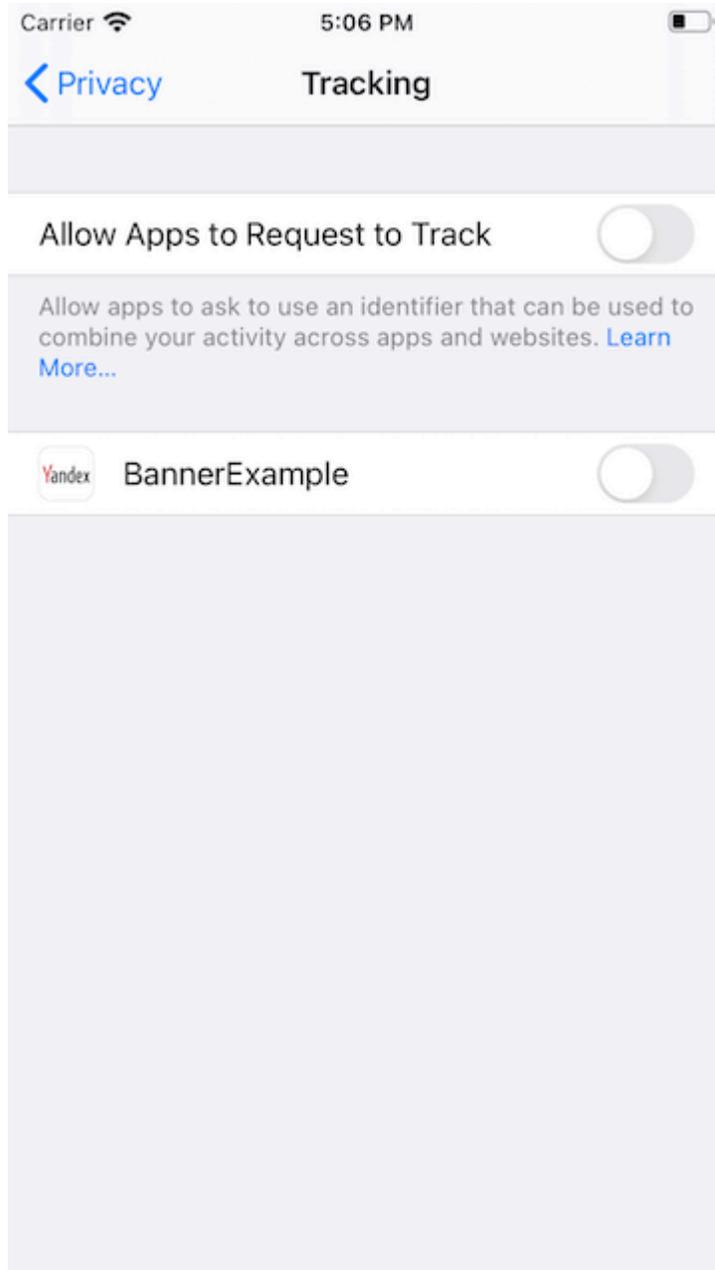
На экран можно добавить кнопки **Разрешить** и **Не сейчас**.

**Пример диалогового окна с объяснением**



2. Перенаправьте пользователя в настройки с помощью `UIApplication.openSettingsURLString`.

Пример настроек пользователя с установленным запретом доступа



## Запрос доступа к IDFA через App Tracking Transparency

При работе с iOS 14 используйте новый фреймворк [App Tracking Transparency](#). С его помощью отобразите системное диалоговое окно в своем приложении. В окне пользователь выберет разрешить или запретить доступ к IDFA.

### Примечание:

Системное диалоговое окно может отображаться только один раз при каждой установке приложения. Если пользователь выберет **Ask App Not to Track**, возможности показать это окно снова для этого приложения не будет.

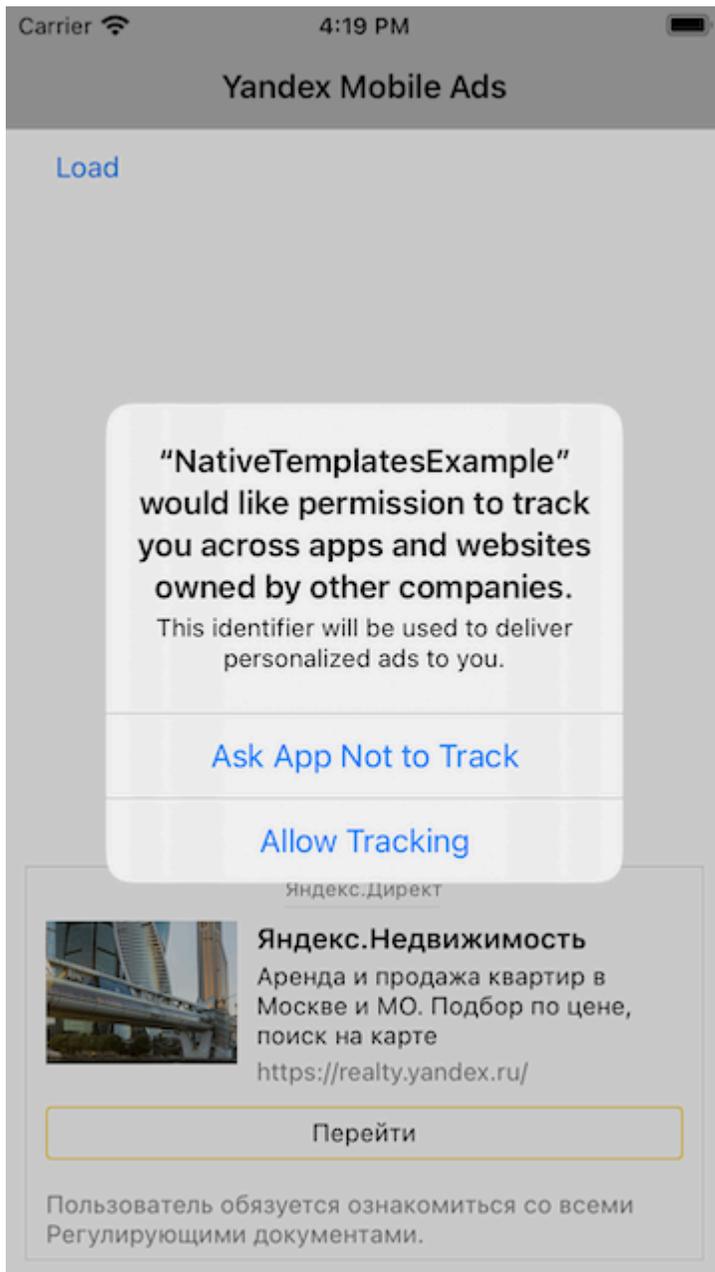
1. Системное диалоговое окно нельзя менять, но в него можно добавить текст с пояснением. Для этого добавьте ключ `NSUserTrackingUsageDescription` в `Info.plist`. Например:

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

Текст из `Info.plist` будет показан пользователю в системном диалоговом окне. В тексте объясните пользователю, почему приложение запрашивает разрешение на использование IDFA.

2. Чтобы отобразить диалоговое окно с запросом доступа к IDFA, вызовите метод `requestTrackingAuthorization(completionHandler:)`.

**Пример системного диалогового окна**



3. Прежде чем загружать рекламу дождитесь получения callback. Тогда Yandex Mobile Ads SDK сможет использовать IDFA в запросах за рекламой.

**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
    }]];
}

```

- Чтобы проверить статус авторизации App Tracking Transparency, используйте свойство `trackingAuthorizationStatus`.

Больше информации по App Tracking Transparency доступно в документации [Apple](#).

## Таргетирование рекламы



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

С помощью данных классов можно настраивать работу SDK и задавать настройки для таргетирования рекламы.

- [YMAAMobileAds](#)
- [YMAAdRequest](#)

## Отслеживание работы рекламы



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

Библиотека позволяет отслеживать события жизненного цикла рекламы (неудачные запросы, клики) с помощью ряда необязательных методов следующих протоколов:

- [YMAAdViewDelegate](#)
- [YYMAInterstitialAdDelegate](#)
- [YMANativeAdDelegate](#)

### Баннерная реклама (Banner)

При работе с данным видом рекламы используйте следующие методы:

#### 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
```

### Полноэкранная реклама (InterstitialAd)

При работе с данным видом рекламы используйте следующие методы:

#### 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
```

### Нативная реклама (NativeAd)

При работе с данным видом рекламы используйте следующие методы:

#### 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
```

## Руководство по миграции на версию 5



### Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится [здесь](#).

### Дополнения

#### Протокол YMAAdViewDelegate

- Новый метод `-adViewDidClick:`, который сообщает, что пользователь кликнул на баннер.

**Протокол YMAInterstitialAdDelegate**

- Новый метод `-interstitialAdDidClick:`, который сообщает, что пользователь кликнул по объявлению.

**Протокол YMANativeAdDelegate**

- Новый метод `-nativeAdDidClick:`, который сообщает, что пользователь кликнул по объявлению.

**Протокол YMARewardedAdDelegate**

- Новый метод `-rewardedAdDidClick:`, который сообщает, что пользователь кликнул по объявлению.

**Изменения****Класс YMAAdView**

- Свойство `blockID` переименовано в `adUnitID`.
- Метод `-initWithBlockID:adSize:` переименован в `-initWithAdUnitID:adSize:`.

**Класс YMAInterstitialAd**

- Свойство `blockID` переименовано в `adUnitID`.
- Метод `-initWithBlockID:` переименован в `-initWithAdUnitID:`.

**Класс YMANativeAdRequestConfiguration**

- Свойство `blockID` переименовано в `adUnitID`.
- Метод `-initWithBlockID:` переименован в `-initWithAdUnitID:`.

**Класс YMARewardedAd**

- Свойство `blockID` переименовано в `adUnitID`.
- Метод `-initWithBlockID:` переименован в `-initWithAdUnitID:`.

**Удаления****Класс YMAAdSize**

Удалены методы:

- `+flexibleSize`
- `+flexibleSizeWithContainerWidth:`

**Рекомендации**

Версия 4.X.Y	Версия 5
<code>YMAAdSize *adSize = [YMAAdSize flexibleSizeWithContainerWidth:width];</code>	Удалено, альтернативный вариант: <code>YMAAdSize *adSize = [YMAAdSize fixedSizeWithCGSize:size];</code>
<code>YMAAdSize *adSize = [YMAAdSize flexibleSize:width];</code>	Удалено, альтернативный вариант: <code>YMAAdSize *adSize = [YMAAdSize fixedSizeWithCGSize:size];</code>
<code>YMAAdView *adView = [[YMAAdView alloc] initWithBlockID:&lt;BlockID&gt; adSize:adSize];</code>	Параметр <code>BlockID</code> больше не поддерживается, используйте новый параметр <code>AdUnitID:YMAAdView *adView = [[YMAAdView alloc] initWithAdUnitID:&lt;AdUnitID&gt; adSize:adSize];</code>
<code>YMAInterstitialAd *interstitialAd = [[YMAInterstitialAd alloc] initWithBlockID:&lt;BlockID&gt;];</code>	Параметр <code>BlockID</code> больше не поддерживается, используйте новый параметр <code>AdUnitID:YMAInterstitialAd *interstitialAd = [[YMAInterstitialAd alloc] initWithAdUnitID:&lt;AdUnitID&gt;];</code>

Версия 4.X.Y	Версия 5
<pre>YMARewardedAd *rewardedAd = [[YMARewardedAd alloc] initWithBlockID:&lt;BlockID&gt;];</pre>	<p>Параметр BlockID больше не поддерживается, используйте новый параметр AdUnitID:YMARewardedAd</p> <pre>*rewardedAd = [[YMARewardedAd alloc] initWithAdUnitID:&lt;AdUnitID&gt;];</pre>
<pre>YMANativeAdRequestConfiguration *requestConfiguration = [[YMANativeAdRequestConfiguration alloc] initWithBlockID:&lt;BlockID&gt;];</pre>	<p>Параметр BlockID больше не поддерживается, используйте новый параметр AdUnitID:</p> <pre>YMANativeAdRequestConfiguration *requestConfiguration = [[YMANativeAdRequestConfiguration alloc] initWithAdUnitID:&lt;AdUnitID&gt;];</pre>