

# Интеграция

## Интеграция

10.07.2024

Интеграция. Версия 1.0

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, 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, 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, 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, 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 плагина.....                        | 4  |
| Форматы рекламы.....                            | 5  |
| Баннерная реклама.....                          | 5  |
| Подключение баннера.....                        | 5  |
| Пример работы с баннерной рекламой.....         | 6  |
| Полноэкранная реклама.....                      | 7  |
| Создание и показ InterstitialAd.....            | 7  |
| Пример работы с полноэкранной рекламой.....     | 7  |
| Реклама с вознаграждением.....                  | 8  |
| Создание и показ RewardedAd.....                | 8  |
| Пример работы с рекламой с вознаграждением..... | 8  |
| GDPR.....                                       | 9  |
| Общие сведения.....                             | 9  |
| Краткое руководство.....                        | 9  |
| COPPA.....                                      | 9  |
| Общие сведения.....                             | 10 |
| Краткое руководство.....                        | 10 |

---

# Подключение Yandex Mobile Ads Flutter плагина



## Внимание:

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

## О плагине

Flutter — популярная кросс-платформенная технология разработки мобильных приложений. С ее помощью разработчики могут писать единый код, который можно применить для разных операционных систем — и для Android, и для iOS.

Теперь монетизировать Android- и iOS-приложения, созданные с помощью Flutter, будет проще. Всем партнерам Рекламной сети Яндекса доступен [плагин](#), с которым можно легко добавить в приложение монетизацию от Яндекса.

Плагин поддерживает загрузку и отображение рекламы:

- [баннерная](#) реклама;
- [полноэкранный](#) реклама;
- [реклама с вознаграждением](#).

## Технические требования

- Flutter 2.5.0 и выше.
- Android 4.1 и выше.

Видеореклама подбирается на устройства с версией Android 5.0 и выше.

- iOS 12.0 и выше.

Для корректной работы iOS 14 и выше обратите внимание на [дополнительные шаги](#).

Последняя версия Xcode с включенными инструментами командной строки.

- [Регистрация](#) аккаунта в Рекламной сети.

## Добавление плагина в проект

Чтобы интегрировать Mobile Ads SDK в приложение Flutter, используйте [Yandex Mobile Ads Flutter плагин](#).

Установите плагин Yandex Mobile Ads Flutter в свой проект. Из корня проекта вызовите команду:

```
flutter pub add yandex_mobileads
```

После добавления плагина в файле `pubspec.yaml` появится строка с зависимостью:

```
dependencies:  
  yandex_mobileads: ^X.X.X
```

X.X.X — номер [актуальной версии](#) плагина.

## Настройка для конкретной платформы

### Android

В Yandex Mobile Ads SDK уже добавлено разрешение `com.google.android.gms.permission.AD_ID`. Оно позволяет использовать `AD_ID` для подбора релевантной рекламы. При необходимости вы можете удалить разрешение.

### Подробнее о разрешении и рекламном идентификаторе

Рекламный идентификатор — уникальный идентификатор сервисов Google Play для показа рекламы пользователям, которые согласны видеть персонализированные объявления. Пользователь может отключить персонализацию рекламы или сбросить идентификатор в настройках. В таком случае рекламные сети не смогут использовать его для подбора релевантной рекламы.

### Как удалить разрешение

Если какие-то политики, например, [Google Play's Families Policy](#), не позволяют использовать рекламный идентификатор, удалите разрешение из файла `AndroidManifest.xml`.

```
<uses-permission android:name="com.google.android.gms.permission.AD_ID" tools:node="remove"/>
```

## iOS

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>
</array>
```

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

## Инициализация библиотеки

В `initState` виджета вашего приложения добавьте строку: `MobileAds.initialize();`

Ознакомьтесь с [примерами использования SDK](#).

### Понятия, связанные с данным

[Подключение плагина для Мобильной медиации](#)

[COPPA](#)

[GDPR](#)

# 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.

## Подключение баннера

1. Добавьте импорт:

```
import 'package:yandex_mobileads/mobile_ads.dart';
```

2. В конструкторе или `initState` виджета, в котором будет расположен баннер, создайте объект класса `BannerAd`:

```
final banner = BannerAd(
  adUnitId: 'R-M-XXXXXX-Y',
  // Flex-size
  adSize: AdSize.flexible(width: screenWidth, height: bannerHeight),
  // Sticky-size
  adSize: AdSize.sticky(width: screenWidth),
  adRequest: AdRequest(),
  onAdLoaded: () {
    /* Do something */
  },
  onAdFailedToLoad: (error) {
    /* Do something */
  },
);
```

3. После создания объекта класса BannerAd в методе build вашего виджета укажите виджет AdWidget, принимающий в себя объект баннера:

```
AdWidget(bannerAd: banner)
```

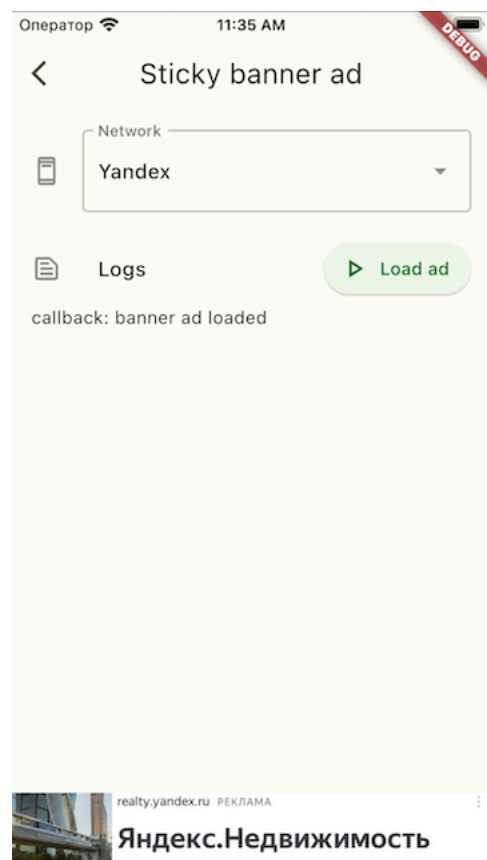
4. Загрузка рекламы произойдет автоматически. При необходимости можно повторно вызвать load объекта баннера после первого вызова build.

## Пример работы с баннерной рекламой

Код демонстрирует создание и настройку объекта BannerAd:

```
import 'package:yandex_mobileads/mobile_ads.dart';
// ...
class BannerAdPage extends StatelessWidget {
  final banner = BannerAd(
    adUnitId: 'demo-banner-yandex',
    adSize: AdSize.sticky(width: screenWidth),
    adRequest: AdRequest(),
    onAdLoaded: () {
      /* Do something */
    },
    onAdFailedToLoad: (error) {
      /* Do something */
    },
  );
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Align(
        alignment: Alignment.bottomCenter,
        child: AdWidget(bannerAd: banner),
      ),
    );
  }
}
```

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

## Подключение полноэкранной рекламы



### Внимание:

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

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

## Создание и показ InterstitialAd

1. Добавьте импорт:

```
import 'package:yandex_mobileads/mobile_ads.dart';
```

2. Создайте объект класса InterstitialAd. Объект может быть создан только асинхронно:

```
final ad = await InterstitialAd.create(  
  adUnitId: 'R-M-XXXXXX-Y',  
  onAdLoaded: () {  
    /* Do something */  
  },  
  onAdFailedToLoad: (error) {  
    /* Do something */  
  },  
);
```

3. После создания объекта класса InterstitialAd рекламу необходимо загрузить. Для загрузки рекламы используйте метод load, принимающий в себя опциональный объект AdRequest:

```
await ad.load(adRequest: AdRequest());
```

Можно дождаться загрузки рекламы асинхронно с помощью ключевого слова `await`. В случае ошибки загрузки произойдет вызов `onAdFailedToLoad` и ожидание будет прервано.

4. Загрузка полноэкранной рекламы происходит в фоновом потоке сразу после вызова метода load. Чтобы показать полноэкранную рекламу, необходимо вызвать метод show:

```
await ad.show();
```

Можно дождаться начала показа рекламы асинхронно.

5. Также можно дождаться конца показа рекламы с помощью асинхронного метода `waitForDismiss`:

```
await ad.waitForDismiss();
```

## Пример работы с полноэкранной рекламой

Код демонстрирует создание и настройку объекта InterstitialAd, загрузку и отображение полноэкранной рекламы:

```
Future<void> showInterstitialAd() async {  
  final ad = await InterstitialAd.create(  
    adUnitId: 'demo-interstitial-yandex',  
    onAdLoaded: () {  
      /* Do something */  
    },  
    onAdFailedToLoad: (error) {  
      /* Do something */  
    },  
  );  
  await ad.load(adRequest: AdRequest());  
  await ad.show();  
  await ad.waitForDismiss();  
}
```

Если реклама подключена данным образом, при вызове `showInterstitialAd` появится рекламный блок.

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

- `demo-interstitial-yandex`

## 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.

## Создание и показ RewardedAd

1. Добавьте импорт:

```
import 'package:yandex_mobileads/mobile_ads.dart';
```

2. Создайте объект класса `RewardedAd`. Объект может быть создан только асинхронно:

```
final ad = await RewardedAd.create(  
  adUnitId: 'R-M-XXXXXX-Y',  
  onAdLoaded: () {  
    /* Do something */  
  },  
  onAdFailedToLoad: (error) {  
    /* Do something */  
  },  
);
```

3. После создания объекта класса `RewardedAd` рекламу необходимо загрузить. Для загрузки рекламы используйте метод `load`, принимающий в себя опциональный объект `AdRequest`:

```
await ad.load(adRequest: AdRequest());
```

Можно дождаться загрузки рекламы асинхронно с помощью ключевого слова `await`.

4. Загрузка рекламы с вознаграждением происходит в фоновом потоке сразу после вызова метода `load`. Чтобы показать рекламу, необходимо вызвать метод `show`:

```
await ad.show();
```

Можно дождаться начала показа рекламы асинхронно.

5. Также можно дождаться конца показа рекламы и/или вознаграждения с помощью асинхронного метода `waitForDismiss`:

```
Reward? reward = await ad.waitForDismiss();
```

## Пример работы с рекламой с вознаграждением

Код демонстрирует создание и настройку объекта `RewardedAd`, загрузку и отображение рекламы с вознаграждением:

```
Future<void> showRewardedAd() async {  
  final ad = await RewardedAd.create(  
    adUnitId: 'demo-rewarded-yandex',  
    onAdLoaded: () {  
      /* Do something */  
    },  
    onAdFailedToLoad: (error) {  
      /* Do something */  
    },  
  );  
  await ad.load(adRequest: AdRequest());  
  await ad.show();  
  final reward = await ad.waitForDismiss();  
}
```



```
if (reward != null) debugPrint('got ${reward.amount} of ${reward.type}');  
}
```

Если реклама подключена данным образом, при вызове `showRewardedAd` появится рекламный блок.

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

- `demo-rewarded-yandex`

## 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.

Yandex Mobile Ads SDK Flutter позволит ограничить 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 Flutter по [инструкции](#).
2. Отобразите для пользователя диалог, предлагающий принять пользовательское соглашение на обработку персональных данных.
3. Передайте полученное значение в Mobile Ads SDK с помощью метода `setUserConsent`.

```
MobileAds.setUserConsent(true);
```

4. Данные пользователей, находящихся в GDPR-регионе, будут обрабатываться только при наличии согласия пользователя на обработку данных.

## COPPA



### Внимание:

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

## Общие сведения

21 апреля 2000 года в силу вступил Закон о защите конфиденциальности детей в Интернете (Children's Online Privacy Protection Act, сокращенно [COPPA](#)). Закон применяется к сбору персональной информации от детей младше 13 лет лицами или организациями под юрисдикцией США. Согласно закону, администрация приложения должна включить в политику конфиденциальности способы получения согласия родителей или опекунов и ответственности администрации за защиту конфиденциальности и безопасности детей в Интернете, включая ограничения в маркетинге.

Данные о возрасте пользователя необходимо передавать в SDK при каждом запуске приложения.

## Краткое руководство

Данные о возрасте пользователя необходимо передавать в SDK при каждом запуске приложения.

1. Подключите Mobile Ads SDK Flutter по [инструкции](#).
2. Отобразите для пользователя диалог, предлагающий принять пользовательское соглашение на обработку персональных данных.
3. Передайте полученное значение в Mobile Ads SDK с помощью метода `setAgeRestrictedUser`. По умолчанию считается, что пользователь не является ребенком младше 13 лет.

```
MobileAds.setAgeRestrictedUser(true);
```