Yandex Mobile Ads

Интеграция

10.07.2024

Яндекс

Yandex Mobile Ads. Интеграция. Версия 2.0

Дата подготовки документа: 10.07.2024

Этот документ является составной частью технической документации Яндекса.

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

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

Содержание

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 Android 4.1 и выше.
2. Видеореклама подбирается только на устройства с версией Android 5.0 и выше.

Библиотека Yandex Mobile Ads предоставляется в формате AAR. Чтобы подключить Mobile Ads SDK:

1. Добавьте зависимость от Yandex Mobile Ads в `build.gradle` файл модуля вашего приложения:

```
dependencies {
    ...
    implementation 'com.yandex.android:mobileads:5.10.0'
}
```

2. Обновите зависимость от Koltin Gradle Plugin в корневом `build.gradle` файле вашего проекта:

```
dependencies {
    ...
    classpath("org.jetbrains.kotlin:kotlin-gradle-plugin:1.7.10")
}
```

3. Добавьте поддержку Java 8 в `build.gradle` файл модуля вашего приложения:

```
android {

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

4. Настройте разрешение на использование рекламного идентификатора.

   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.

   Версия Mobile Ads SDK 4.5.0 и выше

   В Yandex Mobile Ads SDK, начиная с версии 4.5.0, появилось новое разрешение `com.google.android.gms.permission.AD_ID`. Оно прописано в файле `AndroidManifest.xml` библиотеки. Поэтому указывать его дополнительно в основном манифесте приложения не нужно. Разрешение позволяет использовать рекламный идентификатор для подбора релевантной рекламы от рекламных сетей.

   При необходимости вы можете удалить разрешение. Например, если какие-то политики, как Families Policy, не позволяют использовать идентификатор для подбора рекламы.

   Чтобы разрешение не попало в основной манифест приложения, добавьте в файл `AndroidManifest.xml` строку:

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

   Версия Mobile Ads SDK ниже 4.5.0

   Если ваше приложение использует версию Yandex Mobile Ads SDK ниже 4.5.0 и нет ограничений на использование рекламного идентификатора (например, Families Policy), добавьте разрешение в основной манифест приложения `AndroidManifest.xml`:

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

   Отсутствие разрешения и доступа к идентификатору может снизить релевантность рекламы и, как следствие, ваш доход.

Инициализация Mobile Ads SDK

Перед загрузкой рекламы инициализируйте библиотеку с помощью метода initialize(). Инициализация ускорит загрузку объявлений.

Примечание:

Это необходимо делать во время каждого нового запуска приложения. Поэтому рекомендуется добавить код инициализации в метод `onCreate` класса `Application`.

Пример инициализации:

```
public class YandexApplication extends Application {

    private static final String YANDEX_MOBILE_ADS_TAG = "YandexMobileAds";

    @Override
    public void onCreate() {
        super.onCreate();

        MobileAds.initialize(this, new InitializationListener() {
            @Override
            public void onInitializationCompleted() {
                Log.d(YANDEX_MOBILE_ADS_TAG, "SDK initialized");
            }
        });
    }
}
```

Ознакомьтесь с примерами использования SDK.

Форматы рекламы

Баннерная реклама

Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится здесь.

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

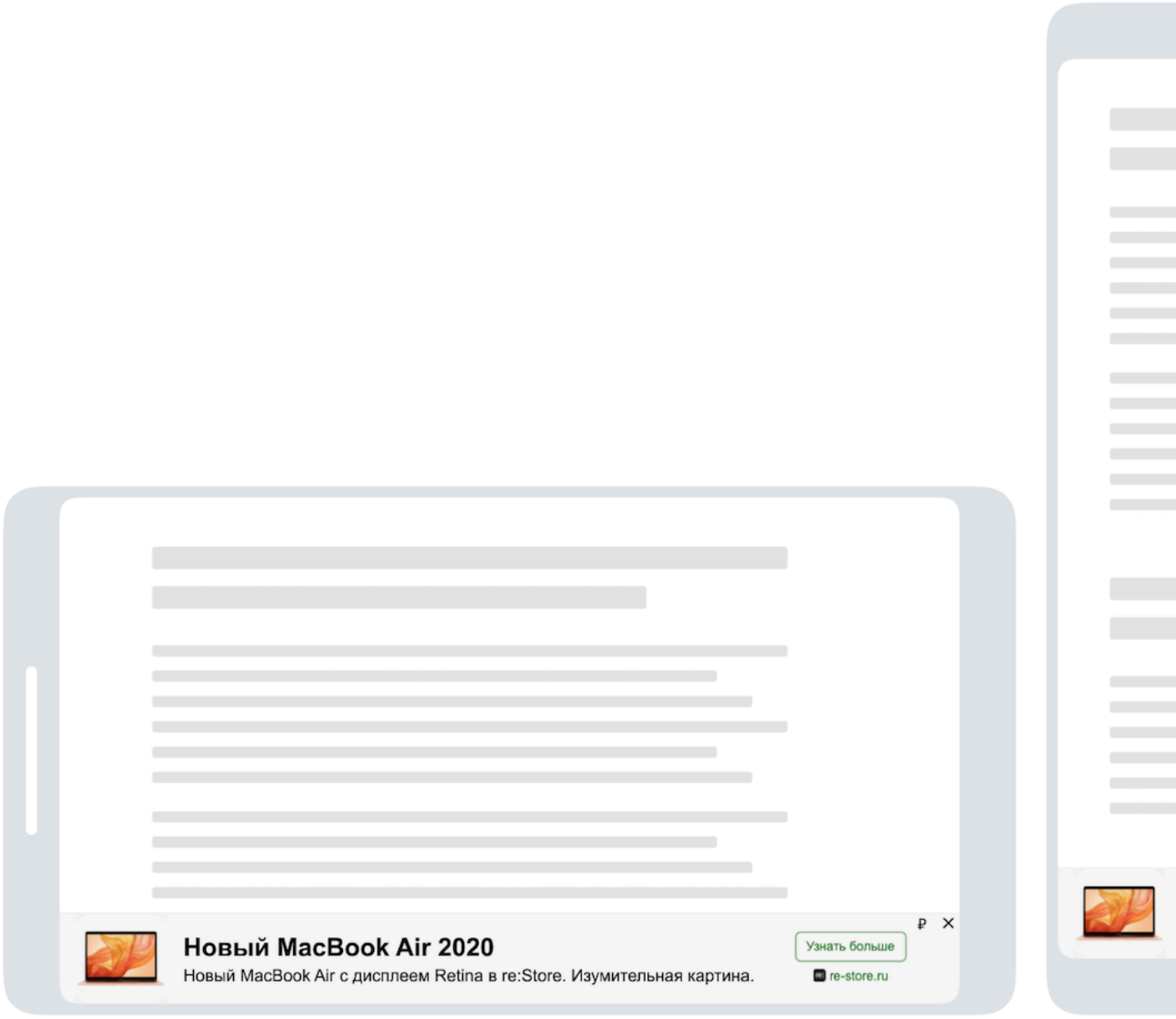
Типы баннера

Sticky баннер

Особенности:

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

Примеры отображения баннера:

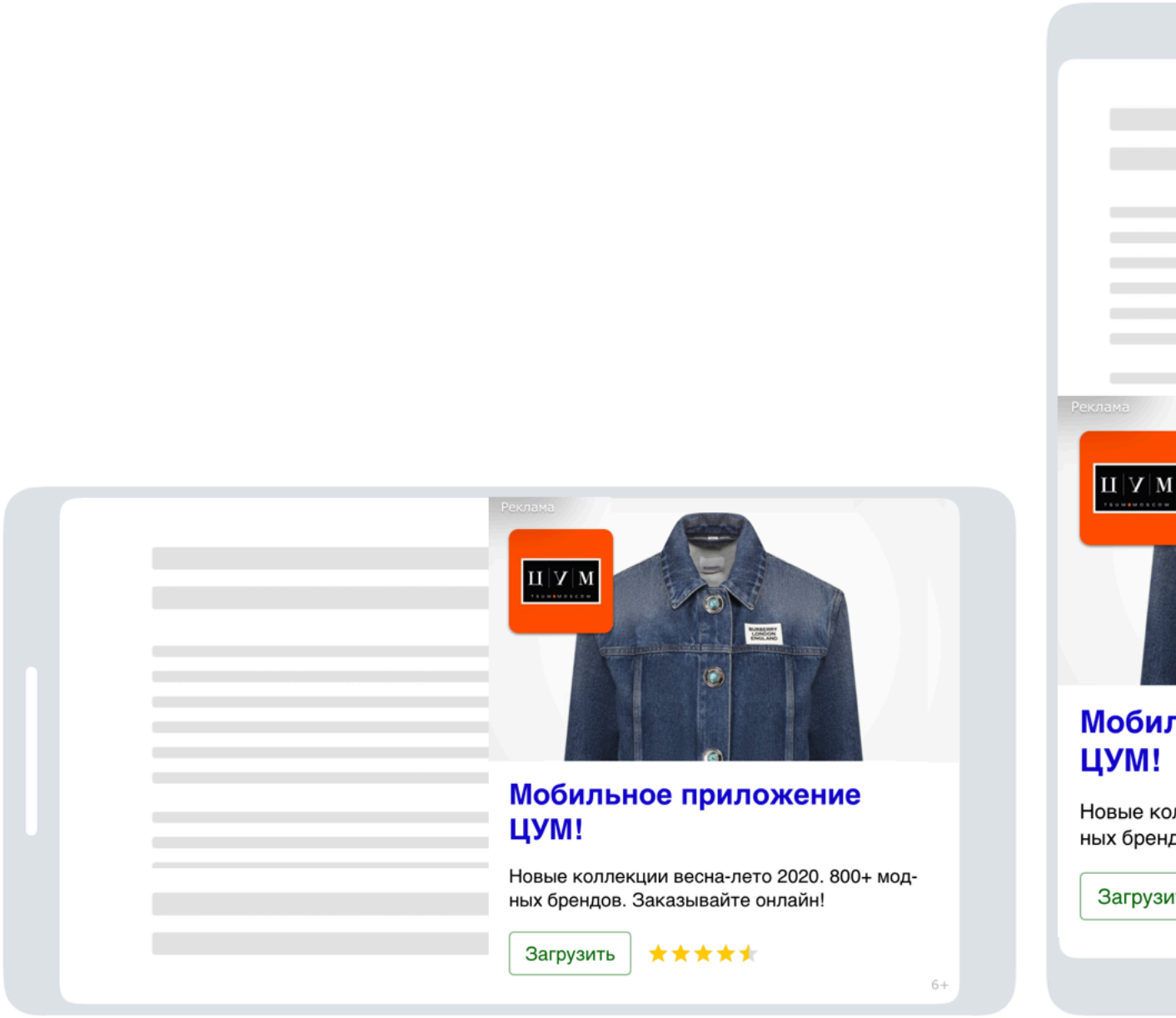


Flex баннер

Особенности:

1. Баннер заполняет весь блок, используя заданные максимальные размеры.
2. Ширина и высота баннера задается с помощью метода flexibleSize(int width, int height).

Примеры отображения баннера:



Подключение баннера

Создание BannerAdview

1. Добавьте объект класса `BannerAdView` в проект с помощью XML-файла или программно.

```
// Создание экземпляра mBannerAdView с помощью XML-файла.
        mBannerAdView = (BannerAdView) findViewById(R.id.banner_view);

// Создание экземпляра mBannerAdView программно.
        mBannerAdView = new BannerAdView(this);
```

2. Установите `AdUnitId`, используя метод setAdUnitId.

```
mBannerAdView.setAdUnitId(<AdUnitId>)
```

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

3. Установите размеры баннера, используя метод setAdSize.

   Sticky баннер

   Чтобы задать ширину баннера, вызовите метод stickySize(int width), где width — максимальная ширина баннера.

   ```
   mBannerAdView.setAdSize(AdSize.stickySize(width));
   ```

   Flex баннер

   Чтобы задать ширину и высоту баннера, вызовите метод flexibleSize(int width, int height).

   ```
   mBannerAdView.setAdSize(AdSize.flexibleSize(width, height));
   ```

   Ограничение: Требования к размерам баннера при отображении видеорекламы

   Минимальный размер баннера, в котором поддерживается воспроизведение видео: 300x160 или 160x300 в dp (density-independent pixels).
4. После создания и настройки объекта класса `BannerAdView` для отслеживания событий (открытие или закрытие рекламы, выход из приложения, успешная и неудачная загрузка рекламы) на объект рекламы можно установить слушатель AdEventListener.

Загрузка рекламы

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

После создания и настройки объекта класса `AdView`, рекламу необходимо загрузить.

Для загрузки рекламы используйте метод loadAd, принимающий в качестве параметра объект класса AdRequest:

```
mBannerAdView.loadAd(adRequest);
```

Особенности загрузки

С помощью объекта `AdRequest` передайте код, полученный в интерфейсе Adfox (подробнее смотрите в помощи по Adfox):

```
// Код из интерфейса Adfox для работы с прямыми кампаниями.
Map<String, String> parameters = new HashMap<String, String>();
parameters.put("adf_ownerid", "example");
parameters.put("adf_p1", "example");
parameters.put("adf_p2", "example");
parameters.put("adf_pfc", "example");
parameters.put("adf_pfb", "example");
parameters.put("adf_plp", "example");
parameters.put("adf_pli", "example");
parameters.put("adf_pop", "example");
parameters.put("adf_pt", "example");
parameters.put("adf_pd", "example");
parameters.put("adf_pw", "example");
parameters.put("adf_pv", "example");
parameters.put("adf_prr", "example");
parameters.put("adf_pdw", "example");
parameters.put("adf_pdh", "example");
parameters.put("adf_puid1", "example");

final AdRequest adRequest = AdRequest.builder().withParameters(parameters).build();
```

Пример работы с баннерной рекламой

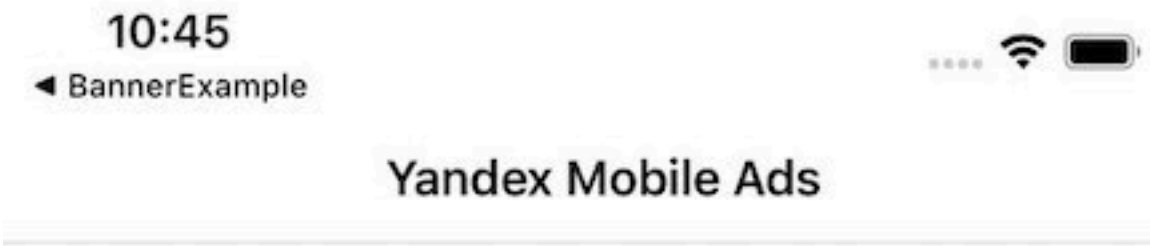
Следующий код демонстрирует создание и настройку объекта `BannerAdView`, регистрацию слушателя, а также загрузку баннера:

```
...
<LinearLayout>
    ...
    <com.yandex.mobile.ads.banner.BannerAdView
        android:id="@+id/banner_ad_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
```

```
</LinearLayout>
```

```
...
public class BannerExample extends Activity {
    ...
    private static final String adUnitId = "YOUR_adUnitId";
    private BannerAdView mBannerAdView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        ...
        // Создание экземпляра mAdView.
        mBannerAdView = (BannerAdView) findViewById(R.id.banner_ad_view);
        mBannerAdView.setAdUnitId(AdUnitId);
        mBannerAdView.setAdSize(AdSize.stickySize(AdSize.FULL_WIDTH));

        // Код из интерфейса Adfox для работы с прямыми кампаниями.
        Map<String, String> parameters = new HashMap<String, String>();
        parameters.put("adf_ownerid", "example");
        parameters.put("adf_p1", "example");
        parameters.put("adf_p2", "example");
        parameters.put("adf_pfc", "example");
        parameters.put("adf_pfb", "example");
        parameters.put("adf_plp", "example");
        parameters.put("adf_pli", "example");
        parameters.put("adf_pop", "example");
        parameters.put("adf_pt", "example");
        parameters.put("adf_pd", "example");
        parameters.put("adf_pw", "example");
        parameters.put("adf_pv", "example");
        parameters.put("adf_prr", "example");
        parameters.put("adf_pdw", "example");
        parameters.put("adf_pdh", "example");
        parameters.put("adf_puid1", "example");

        // Создание объекта таргетирования рекламы.
        final AdRequest adRequest = AdRequest.builder().withParameters(parameters).build();

        // Регистрация слушателя для отслеживания событий, происходящих в баннерной рекламе.
        mBannerAdView.setAdEventListener(new BannerAdEventListener() {
            @Override
            public void onAdLoaded() {
              ...
            }
        });

        // Загрузка объявления.
        mBannerAdView.loadAd(adRequest);
    }
}
```

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


10:45
BannerExample
Yandex Mobile Ads

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

- demo-banner-yandex

Понятия, связанные с данным

Классы и интерфейсы для работы с баннерной рекламой

Узнайте больше

Пример рекламы

Полноэкранная реклама

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

Чтобы подключить рекламу, необходимо:

Создать InterstitialAd
Загрузить рекламу
Отобразить рекламу

Создание InterstitialAd

1. Создайте объект класса InterstitialAd. Объект может быть создан только программно.

```
mInterstitialAd = new InterstitialAd(this);
```

2. Установите AdUnitId, используя метод setAdUnitId.

```
mInterstitialAd.setAdUnitId(AdUnitId);
```

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

3. После создания и настройки объекта класса InterstitialAd, для отслеживания событий (открытие или закрытие рекламы, выход из приложения, успешная и не успешная загрузка рекламы), на объект рекламы можно установить слушатель InterstitialEventListener интерфейса.

Загрузка рекламы

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

После создания и настройки объекта класса `InterstitialAd`, рекламу необходимо загрузить.

Для загрузки рекламы, используйте метод loadAd, принимающий в качестве параметра объект класса AdRequest:

```
mInterstitialAd.loadAd(adRequest);
```

Особенности загрузки

С помощью объекта `AdRequest` передайте код, полученный в интерфейсе Adfox (подробнее смотрите в помощи по Adfox):

```
// Код из интерфейса Adfox для работы с прямыми кампаниями.
Map<String, String> parameters = new HashMap<String, String>();
parameters.put("adf_ownerid", "example");
parameters.put("adf_p1", "example");
parameters.put("adf_p2", "example");
parameters.put("adf_pfc", "example");
parameters.put("adf_pfb", "example");
parameters.put("adf_plp", "example");
parameters.put("adf_pli", "example");
parameters.put("adf_pop", "example");
parameters.put("adf_pt", "example");
parameters.put("adf_pd", "example");
```

```
parameters.put("adf_pw", "example");
parameters.put("adf_pv", "example");
parameters.put("adf_prr", "example");
parameters.put("adf_pdw", "example");
parameters.put("adf_pdh", "example");
parameters.put("adf_puid1", "example");

final AdRequest adRequest = AdRequest.builder().withParameters(parameters).build();
```

Отображение рекламы

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

Рекомендуется предварительно проверить, что реклама действительно загружена. Для этого вызовите метод isLoaded.

Примечание:

Такая проверка не требуется, если метод show вызывается после срабатывания callback об окончании загрузки onAdLoaded.

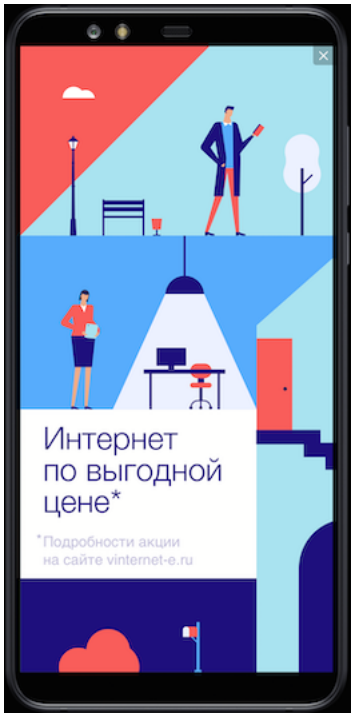
Пример работы с полноэкраннной рекламой

Следующий код демонстрирует создание и настройку объекта InterstitialAd, регистрацию слушателя, а также загрузку и отображение полноэкранной рекламы:

```
...
public class InterstitialExample extends Activity {
    ...
    private static final String adUnitAd = "YOUR_adUnitId";
    private InterstitialAd mInterstitialAd;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        ...
        // Создание экземпляра InterstitialAd.
        mInterstitialAd = new InterstitialAd(this);
        mInterstitialAd.setAdUnitId(adUnitId);

        // Код из интерфейса Adfox для работы с прямыми кампаниями.
        Map<String, String> parameters = new HashMap<String, String>();
        parameters.put("adf_ownerid", "example");
        parameters.put("adf_p1", "example");
        parameters.put("adf_p2", "example");
        parameters.put("adf_pfc", "example");
        parameters.put("adf_pfb", "example");
        parameters.put("adf_plp", "example");
        parameters.put("adf_pli", "example");
        parameters.put("adf_pop", "example");
        parameters.put("adf_pt", "example");
        parameters.put("adf_pd", "example");
        parameters.put("adf_pw", "example");
        parameters.put("adf_pv", "example");
        parameters.put("adf_prr", "example");
        parameters.put("adf_pdw", "example");
        parameters.put("adf_pdh", "example");
        parameters.put("adf_puid1", "example");

        // Создание объекта таргетирования рекламы.
        final AdRequest adRequest = AdRequest.builder().withParameters(parameters).build();

        // Регистрация слушателя для отслеживания событий, происходящих в рекламе.
         mInterstitialAd.setInterstitialAdEventListener(new InterstitialAdEventListener() {
            @Override
            public void onAdLoaded() {
                mInterstitialAd.show();
            }

            @Override
            public void onAdFailedToLoad(AdRequestError adRequestError) {
                ...
            }

            @Override
            public void onAdShown() {
                ...
            }

            @Override
            public void onAdDismissed() {
                ...
            }

            @Override
            public void onLeftApplication() {
                ...
```

```
            }

            @Override
            public void onReturnedToApplication() {
                ...
            }
        });

        // Загрузка объявления.
        mInterstitialAd.loadAd(adRequest);
    }
}
```

Если реклама подключена данным образом, после запуска приложения появится рекламный блок:



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

- demo-interstitial-yandex

Понятия, связанные с данным

Классы и интерфейсы для работы с полноэкранной рекламой

Узнайте больше

Пример рекламы

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:

Создать RewardedAd
Загрузить рекламу
Отобразить рекламу

Создание RewardedAd

1. Добавьте объект класса RewardedAd.

```
mRewardedAd = new RewardedAd(this);
```

2. Установите AdUnitId, используя метод setAdUnitId.

   `AdUnitId` — уникальный идентификатор рекламного места, который выдается в Партнерском интерфейсе и имеет вид: R-M-XXXXXX-Y.
3. Если вы используете механизм награждения на стороне приложения («client-side reward»), реализуйте метод интерфейса onRewarded. Он вызывается, когда показ засчитан и пользователь может быть награжден за просмотр рекламы. Используйте этот момент, чтобы выдать награду пользователю приложения.

После создания и настройки объекта класса RewardedAd, для отслеживания событий (открытие или закрытие рекламы, выход из приложения, успешная и не успешная загрузка рекламы), на объект рекламы необходимо установить слушатель RewardedAdEventListener интерфейса.

Загрузка рекламы

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

После создания и настройки объекта класса RewardedAd, рекламу необходимо загрузить. Для загрузки рекламы, используйте метод loadAd, принимающий в качестве параметра объект AdRequest (Builder, опционально принимающий данные для таргетирования рекламы).

Отображение рекламы

Загрузка рекламы с вознаграждением происходит в фоновом потоке сразу после вызова метода loadAd. Чтобы показать рекламу с вознаграждением необходимо вызвать метод show.

Рекомендуется предварительно проверить, что реклама действительно загружена. Для этого вызовите метод isLoaded.

Примечание:

Такая проверка не требуется, если метод show вызывается после срабатывания callback об окончании загрузки onAdLoaded.

Пример работы с рекламой с вознаграждением

Следующий код демонстрирует создание и настройку объекта RewardedAd, регистрацию слушателя, а также загрузку и отображение рекламы с вознаграждением:

```
...
public class RewardedAdExample extends Activity {
    ...
    private static final String AdUnitId = "YOUR_AdUnitId";
    private RewardedAd mRewardedAd;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        ...
        // Создание экземпляра RewardedAd.
        mRewardedAd = new RewardedAd(this);
        mRewardedAd.setAdUnitId(AdUnitId);

        // Создание объекта таргетирования рекламы.
        final AdRequest adRequest = new AdRequest.Builder().build();

        // Регистрация слушателя для отслеживания событий, происходящих в рекламе.
        mRewardedAd.setRewardedAdEventListener(new RewardedAdEventListener() {
            @Override
            public void onLoaded() {
                mRewardedAd.show();
            }

            @Override
            public void onRewarded(final Reward reward) {
                ...
            }
```

```
            @Override
            public void onAdFailedToLoad(final AdRequestError adRequestError) {
                ...
            }

            @Override
            public void onAdShown() {
                ...
            }

            @Override
            public void onAdDismissed() {
                ...
            }

            @Override
            public void onLeftApplication() {
                ...
            }

            @Override
            public void onReturnedToApplication() {
                ...
            }
        });

        // Загрузка объявления.
        mRewardedAd.loadAd(adRequest);
    }
}
```

Если реклама подключена данным образом, после запуска приложения появится рекламный блок.

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

- `demo-rewarded-yandex`

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.

Есть три типа загрузки рекламы:

- Загрузка одного объявления
- Загрузка нескольких объявлений
- Слайдер рекламных объявлений

Понятия, связанные с данным

Требования к размещению рекламы и рекламных компонентов

Классы и интерфейсы для работы с нативной рекламой

Загрузка и показ рекламных объявлений

Внимание:

Это архивная версия документации. Актуальная документация по всем платформам находится здесь.

Подготовка к работе

Для корректной работы Yandex Mobile Ads SDK выполните все шаги подключения рекламной библиотеки.

Загрузка рекламных объявлений

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

1. Создайте экземпляр класса NativeAdLoader для получения нативных рекламных объявлений.
2. Создайте конфигурацию запроса `nativeAdRequestConfiguration` с помощью класса NativeAdRequestConfiguration.Builder. В качестве параметров запроса можно передать идентификатор рекламного блока, способ загрузки изображений, возраст, гендерные признаки и другие данные, способные улучшить качество подбора рекламы.
3. Для получения уведомлений (успешная загрузка рекламы или ошибка при загрузке рекламы), создайте экземпляр NativeAdLoadListener и установите его в качестве слушателя событий загрузчика рекламных объявлений.
4. Запустите процесс загрузки рекламы.

Если на блоке включены медиа форматы рекламы, следует запрашивать рекламу с указанием размеров контейнера.

Общий запрос за рекламой

```
final NativeAdLoader nativeAdLoader = new NativeAdLoader(this);
    nativeAdLoader.setNativeAdLoadListener(new NativeAdLoadListener() {
        @Override
        public void onAdLoaded(@NonNull final NativeAd nativeAd) {
            //bind nativeAd
        }

        @Override
        public void onAdFailedToLoad(@NonNull final AdRequestError error) {
            //log error
        }
    });

final NativeAdRequestConfiguration nativeAdRequestConfiguration =
    new NativeAdRequestConfiguration.Builder(AdUnitId).build();
nativeAdLoader.loadAd(nativeAdRequestConfiguration);
```

Совет:

Рекомендуется держать сильную ссылку на рекламу и ее загрузчик на всем протяжении жизни экрана, в рамках которого происходит работа с данными компонентами.

Показ рекламных объявлений

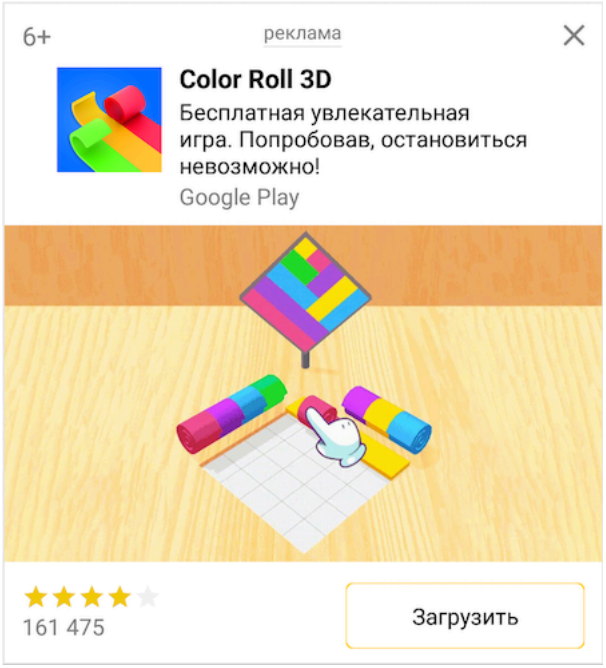
После окончания загрузки рекламы, все ее компоненты необходимо отобразить. Список доступных в объявлении компонентов можно получить из объекта рекламы NativeAd.

Существует два способа настройки внешнего оформления рекламы:

1. Оформление с помощью шаблона
2. Оформление без использования шаблона

Оформление с помощью шаблона

Пример шаблона нативной рекламы



Использование стандартного шаблона оформления — самый простой способ работы с нативной рекламой, который требует всего нескольких строк кода в базовом варианте.

В шаблоне уже добавлен весь набор необходимых рекламных компонентов и настроено их расположение относительно друг друга. Шаблон работает с любым поддерживаемым типом нативной рекламы.

```
final NativeBannerView nativeBannerView = new NativeBannerView(this);
nativeBannerView.setAd(nativeAd);
```

Шаблон нативной рекламы можно кастомизировать. Подробнее об этом в статье Пример оформления с помощью шаблона.

Оформление без использования шаблона

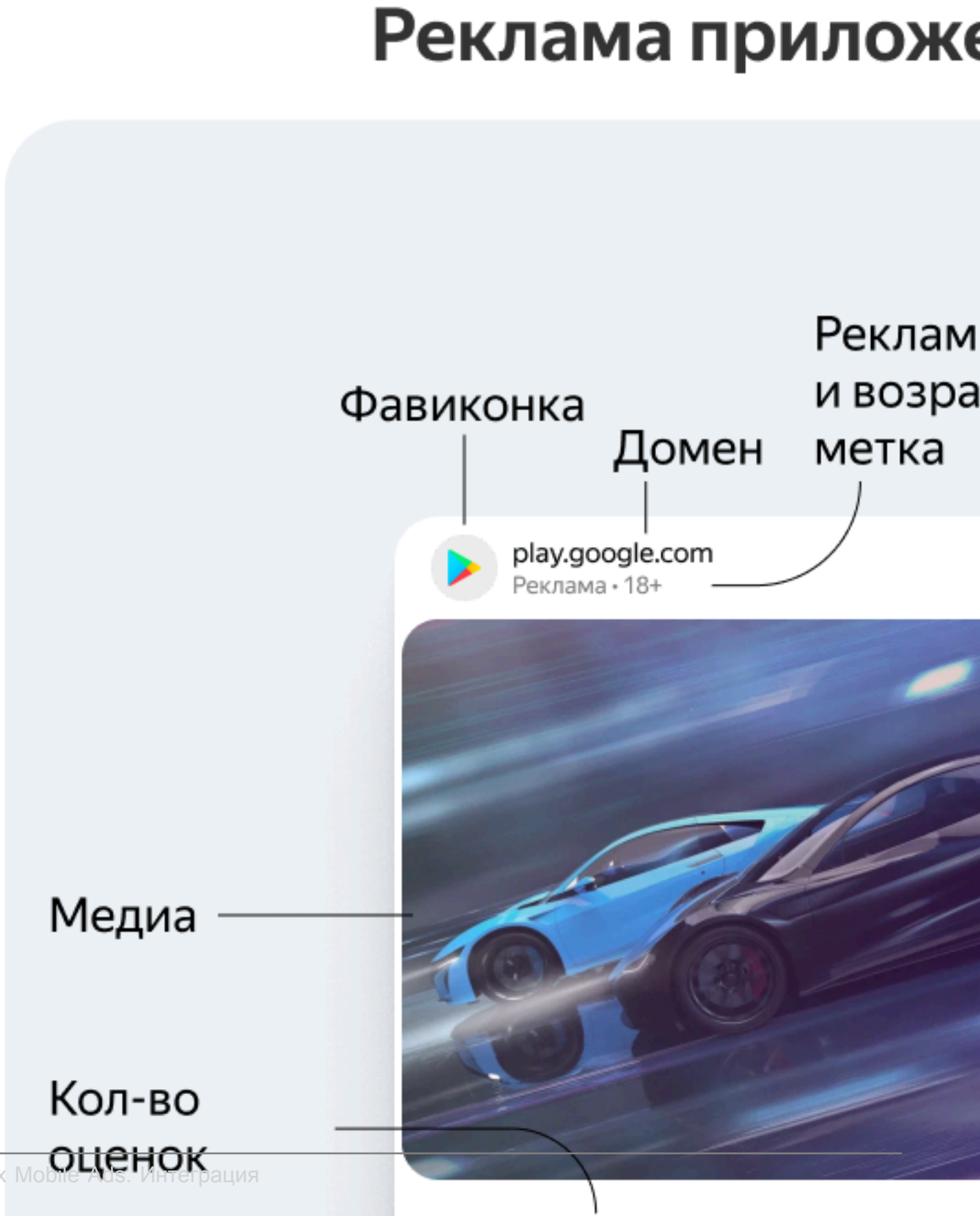
Ручная настройка внешнего оформления нативной рекламы используется в тех случаях, когда возможностей настройки шаблона недостаточно для получения желаемого результата.

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.

Пример оформления без использования шаблона



Для каждого компонента рекламного объявления предоставьте View через экземпляр класса NativeAdViewBinder.Builder. Класс принимает в качестве аргумента контейнер NativeAdView. Все компоненты рекламы должны быть определены как subview данного контейнера.

Сформированный макет рекламного объявления необходимо связать с объектом нативной рекламы NativeAd.

```
final NativeAdViewBinder nativeAdViewBinder = new NativeAdViewBinder.Builder(mNativeAdView)
                    .setAgeView((TextView) findViewById(R.id.age))
                    .setBodyView((TextView) findViewById(R.id.body))
                    .setCallToActionView((TextView) findViewById(R.id.call_to_action))
                    .setDomainView((TextView) findViewById(R.id.domain))
                    .setFaviconView((ImageView) findViewById(R.id.favicon))
                    .setFeedbackView((TextView) findViewById(R.id.feedback))
                    .setIconView((ImageView) findViewById(R.id.icon))
                    .setMediaView((MediaView) findViewById(R.id.media))
                    .setPriceView((TextView) findViewById(R.id.price))
                    .setRatingView((MyRatingView) findViewById(R.id.rating))
                    .setReviewCountView((TextView) findViewById(R.id.review_count))
                    .setSponsoredView((TextView) findViewById(R.id.sponsored))
                    .setTitleView((TextView) findViewById(R.id.title))
                    .setWarningView((TextView) findViewById(R.id.warning))
                    .build();

                    try {
                    nativeAd.bindNativeAd(nativeAdViewBinder);
                    mNativeAdView.setVisibility(View.VISIBLE);
                    } catch (final NativeAdException exception) {
                    //log exception
                    }
```

Примечание: Требования к размерам mediaView при отображении видеорекламы

Минимальный размер экземпляра класса MediaView, в котором поддерживается воспроизведение видео: 300x160 или 160x300 в dp (density-independent pixels).

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

Загрузка нескольких рекламных объявлений

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

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

Для балкового запроса за рекламой воспользуйтесь экземпляром класса NativeBulkAdLoader. Экземпляр класса принимает в методе loadAds аргумент COUNT, определяющий желаемое количество объявлений за один запрос.

```
final NativeBulkAdLoader nativeBulkAdLoader = new NativeBulkAdLoader(this);
    nativeBulkAdLoader.setNativeBulkAdLoadListener(new NativeBulkAdLoadListener() {
        @Override
        public void onAdsLoaded(@NonNull final List<NativeAd> nativeAds) {
            for (final NativeAd nativeAd : nativeAds) {
                //bind native ad
            }
        }

        @Override
        public void onAdsFailedToLoad(@NonNull final AdRequestError error) {
            //log error
        }
    });

    final NativeAdRequestConfiguration nativeAdRequestConfiguration =
    new NativeAdRequestConfiguration.Builder(AdUnitId).build();
    nativeBulkAdLoader.loadAds(nativeAdRequestConfiguration, COUNT);
```

Примечание:

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

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 паблишером в зависимости от функций и дизайна приложения, в котором будет отображаться слайдер.

Подготовка к работе

Для корректной работы Yandex Mobile Ads SDK выполните все шаги подключения рекламной библиотеки.

Загрузка слайдера

Примечание:

Все вызовы Mobile Ads SDK следует выполнять с главного потока.

1. Создайте экземпляр класса SliderAdLoader для получения рекламных объявлений в рамках слайдера.
2. Создайте конфигурацию запроса `nativeAdRequestConfiguration` с помощью класса NativeAdRequestConfiguration.Builder. В качестве параметров запроса можно передать идентификатор рекламного блока, способ загрузки изображений, возраст, гендерные признаки и другие данные, способные улучшить качество подбора рекламы.
3. Для получения уведомлений (успешная загрузка слайдера или ошибка при загрузке), создайте экземпляр SliderAdLoadListener и установите его в качестве слушателя событий загрузчика рекламных объявлений.
4. Запустите процесс загрузки слайдера.

Пример кода:

```
final SliderAdLoader sliderAdLoader = new SliderAdLoader(this);
sliderAdLoader.setSliderAdLoadListener(new SliderAdLoadListener() {
    @Override
    public void onSliderAdLoaded(@NonNull final SliderAd sliderAd) {
        //bind sliderAd
    }

    @Override
    public void onSliderAdFailedToLoad(@NonNull final AdRequestError error) {
        //log error
    }
});

final NativeAdRequestConfiguration nativeAdRequestConfiguration =
        new NativeAdRequestConfiguration.Builder(AdUnitId).build();
sliderAdLoader.loadSlider(nativeAdRequestConfiguration);
```

Совет:

Рекомендуется держать сильную ссылку на slider и его загрузчик на всем протяжении жизни экрана, в рамках которого происходит работа с данными компонентами.

Показ слайдера

После окончания загрузки слайдера, все его компоненты необходимо отобразить.

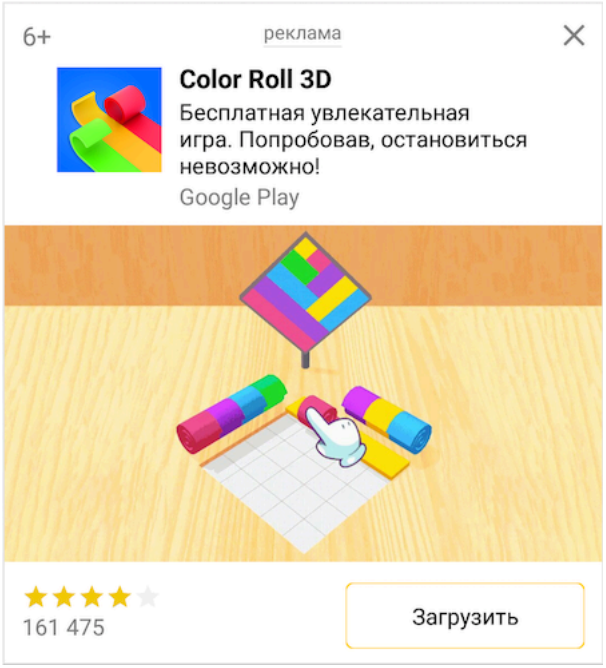
Успешно загруженный слайдер содержит одно или несколько связанных между собой нативных объявлений. Показ рекламных объявлений в рамках слайдера должен происходить в пределах одного общего контейнера, иначе показ объявлений не будет засчитан.

Существует два способа настройки внешнего оформления слайдера:

1. Оформление с помощью шаблона
2. Оформление без использования шаблона

Оформление с помощью шаблона

Пример шаблона нативной рекламы



Использование стандартного шаблона оформления — самый простой способ работы с нативной рекламой, который требует всего нескольких строк кода в базовом варианте.

В шаблоне уже добавлен весь набор необходимых рекламных компонентов и настроено их расположение относительно друг друга. Шаблон работает с любым поддерживаемым типом нативной рекламы.

Оформить слайдер с помощью шаблона очень просто. Привяжите объект SliderAd к корневому контейнеру NativeAdView, а все входящие в слайдер рекламные объявления свяжите с шаблоном:

```
@Override
public void onSliderAdLoaded(@NonNull final SliderAd sliderAd) {
    mNativeAdView = (NativeAdView) findViewById(R.id.native_slider_ad_container);
    try {
        final NativeAdViewBinder sliderViewBinder =
            new NativeAdViewBinder.Builder(mNativeAdView).build();
sliderAd.bindSliderAd(sliderViewBinder);
        final List<NativeAd> nativeAds = sliderAd.getNativeAds();
        for (final NativeAd nativeAd : nativeAds) {
            final NativeBannerView nativeBannerView = new NativeBannerView(this);
            nativeBannerView.setAd(nativeAd);
            mNativeAdView.addView(nativeBannerView);
        }
    } catch (final NativeAdException exception) {
        //log error
    }
}
```

Шаблон нативной рекламы можно кастомизировать. Подробнее об этом в статье Пример оформления с помощью шаблона.

Оформление без использования шаблона

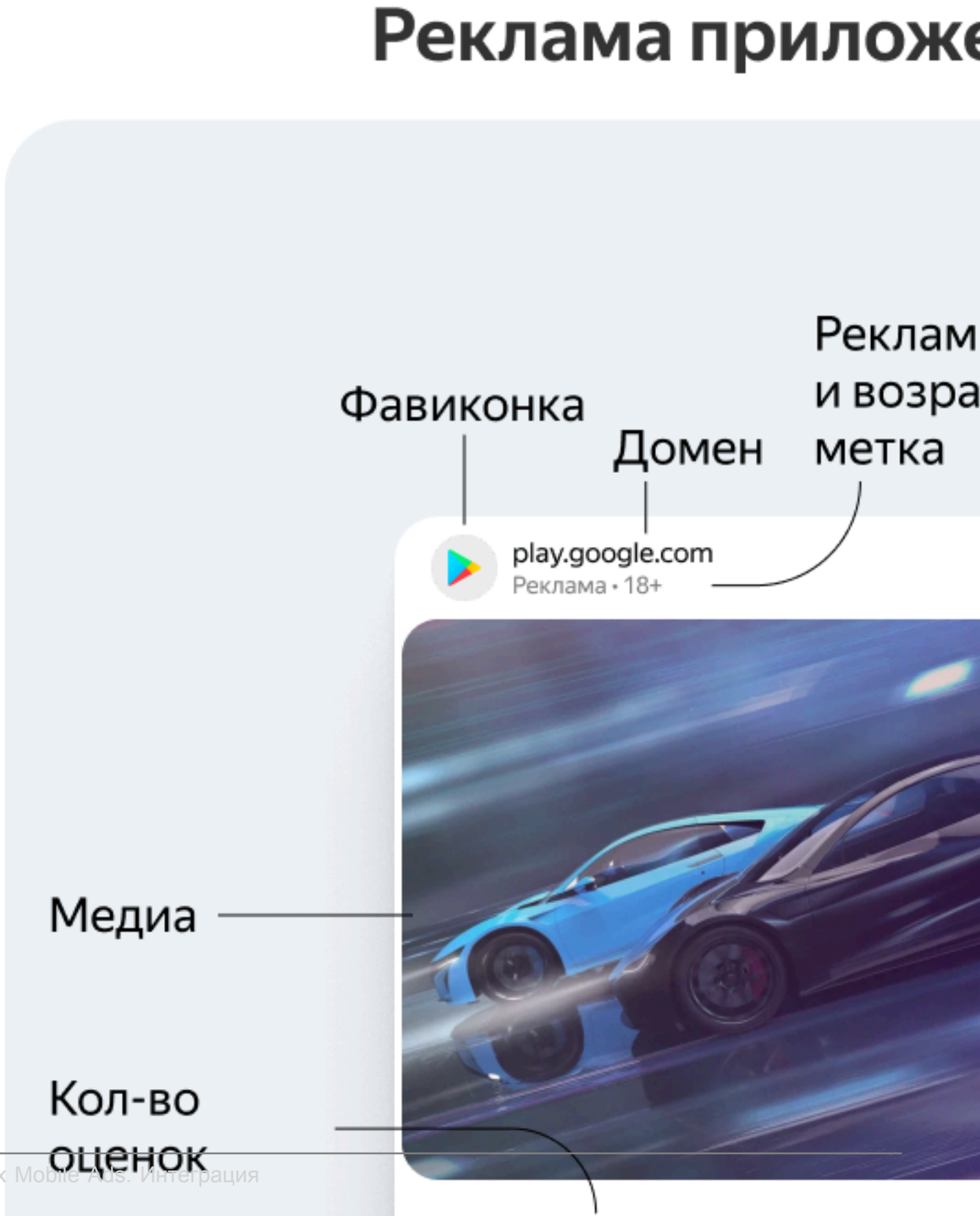
Ручная настройка внешнего оформления нативной рекламы используется в тех случаях, когда возможностей настройки шаблона недостаточно для получения желаемого результата.

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.

Пример оформления без использования шаблона



Привяжите объект SliderAd к корневому NativeAdView, а каждое объявление, входящее в слайдер, к дочернему NativeAdView.

Внешнее оформление каждого объявления, входящего в слайдер, настраивается одним из способов оформления стандартной нативной рекламы.

```
@Override
public void onSliderAdLoaded(@NonNull final SliderAd sliderAd) {
    mNativeAdView = (NativeAdView) findViewById(R.id.native_slider_ad_container);
    try {
        final NativeAdViewBinder sliderViewBinder =
            new NativeAdViewBinder.Builder(mNativeAdView).build();
sliderAd.bindSliderAd(sliderViewBinder);
        final List<NativeAd> nativeAds = sliderAd.getNativeAds();
        for (final NativeAd nativeAd : nativeAds) {
            final NativeAdView nativeAdView = mLayoutInflater.inflate(R.layout.widget_native_ad, mSliderAdView,
 false);
            final NativeAdViewBinder viewBinder = new NativeAdViewBinder.Builder(nativeAdView)
                    .setAgeView((TextView) nativeAdView.findViewById(R.id.age))
                    .setBodyView((TextView) nativeAdView.findViewById(R.id.body))
                    .setCallToActionView((Button) nativeAdView.findViewById(R.id.call_to_action))
                    .setDomainView((TextView) nativeAdView.findViewById(R.id.domain))
                    .setFaviconView((ImageView) nativeAdView.findViewById(R.id.favicon))
                    .setFeedbackView((Button) nativeAdView.findViewById(R.id.feedback))
                    .setIconView((ImageView) nativeAdView.findViewById(R.id.icon))
                    .setMediaView((MediaView) nativeAdView.findViewById(R.id.media))
                    .setPriceView((TextView) nativeAdView.findViewById(R.id.price))
                    .setRatingView((MyRatingView) nativeAdView.findViewById(R.id.rating))
                    .setReviewCountView((TextView) nativeAdView.findViewById(R.id.review_count))
                    .setSponsoredView((TextView) nativeAdView.findViewById(R.id.sponsored))
                    .setTitleView((TextView) nativeAdView.findViewById(R.id.title))
                    .setWarningView((TextView) nativeAdView.findViewById(R.id.warning))
                    .build();

            nativeAd.bindNativeAd(viewBinder);
            mNativeAdView.addView(nativeBannerView);
        }
    } catch (final NativeAdException exception) {
        //log error
    }
}
```

Примечание: Требования к размерам mediaView при отображении видеорекламы

Минимальный размер экземпляра класса MediaView, в котором поддерживается воспроизведение видео: 300x160 или 160x300 в dp (density-independent pixels).

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