

Reference guide

Reference guide

10.07.2024

Reference guide. Version 1.0

Document build date: 10.07.2024

This volume is a part of Yandex technical documentation.

© 2008—2024 Yandex LLC. All rights reserved.

Copyright Disclaimer

Yandex (and its applicable licensor) has exclusive rights for all results of intellectual activity and equated to them means of individualization, used for development, support, and usage of the service . It may include, but not limited to, computer programs (software), databases, images, texts, other works and inventions, utility models, trademarks, service marks, and commercial denominations. The copyright is protected under provision of Part 4 of the Russian Civil Code and international laws.

You may use or its components only within credentials granted by the Terms of Use of or within an appropriate Agreement.

Any infringements of exclusive rights of the copyright owner are punishable under civil, administrative or criminal Russian laws.

Contact information

Yandex LLC

<https://www.yandex.com>

Tel.: +7 495 739 7000

Email: pr@yandex-team.ru

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

Contents

AdRequest.Builder	5
Method description.....	5
AdRequest	7
Property description.....	7
AdSize	8
Method description.....	9
Property description.....	10
Field description.....	10
Banner	11
Method description.....	11
Event description.....	12
Gender	13
Constant description.....	13
Interstitial	13
Method description.....	14
Event description.....	15
Location.Builder	16
Method description.....	16
Location	17
Property description.....	18
MobileAds	18
Method description.....	18
RewardedAd	19
Method description.....	20
Event description.....	21
Reward	22
Method description.....	22
Field description.....	22
ScreenUtils	23
Method description.....	23

AdPosition.....	23
Enumeration description.....	23
AdSizeType.....	24
Enumeration description.....	24

Builder WithContextTags(List<string> contextTags)

```
public Builder WithContextTags(List<string> contextTags)
```

Sets a list of tags that describe the current user context. The list depends on which screen the user is on.

Parameters:

contextTags A list of tags produced by the app.

Returns:

An object of the AdRequest.Builder class with a list of assigned tags.

Builder WithGender(string gender)

```
public Builder WithGender(string gender)
```

Sets the user's gender in the app.

Parameters:

gender The gender of the user.

Returns:

An object of the AdRequest.Builder class with the specified gender.

Builder WithLocation(Location location)

```
public Builder WithLocation(Location location)
```

Sets the user location.

Parameters:

location User location.

Returns:

An object of the AdRequest.Builder class with the specified user location.

Builder WithParameters(Dictionary<string, string> parameters)

```
public Builder WithParameters(Dictionary<string, string> parameters)
```

Specifies a list of additional parameters.

Parameters:

parameters Additional parameters.

Returns:

An object of the AdRequest.Builder class with a list of additional parameters specified.

Builder WithAdRequest(AdRequest adRequest)

```
public Builder WithAdRequest(AdRequest adRequest)
```

Sets the AdRequest object.

Parameters:

adRequest An object of the AdRequest class.

Returns:

An object of the `AdRequest.Builder` class that was created based on the `AdRequest` object passed.

AdRequest Build()

```
public AdRequest Build()
```

Creates an `AdRequest` class object with data for a targeted ad request.

Parameters:

`adRequest` An object of the `AdRequest` class.

Returns:

An object of the `AdRequest` class including the data for a targeted ad request.

AdRequest class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

This class contains settings for getting a targeted ad. To create a request, use the [AdRequest.Builder](#) class.

Properties

<code>string Age { get }</code>	The age that the user entered in the app.
<code>string ContextQuery { get }</code>	The search query that the user entered in the app.
<code>List<string> ContextTags { get }</code>	A list of tags produced by the app.
<code>string Gender { get }</code>	The gender that the user entered in the app.
<code>Location Location { get }</code>	User location.
<code>Dictionary<string, string> Parameters { get }</code>	List of additional parameters.

Property description

string Age { get }

```
public string Age { get }
```

The age that the user entered in the app.

string ContextQuery { get }

```
public string ContextQuery { get }
```

The search query that the user entered in the app.

List<string> ContextTags { get }

```
public List<string> ContextTags { get }
```

A list of tags produced by the app.

string Gender { get }

```
public string Gender { get }
```

The gender that the user entered in the app.

Location Location { get }

```
public Location Location { get }
```

User location.

Dictionary<string, string> Parameters { get }

```
public Dictionary<string, string> Parameters { get }
```

List of additional parameters.

AdSize class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

This class is responsible for the banner size.

Methods

[static AdSize StickySize \(int width\)](#)

Creates an AdSize class object with the specified width of a “sticky” banner. Banner height depends on what types of ad units are selected in the Partner interface.

[static AdSize FlexibleSize \(int width, int height\)](#)

Creates an AdSize class object with the specified maximum height and width of the banner. Banner height depends on what types of ad units are selected in the Partner interface.

Properties

[int Width { get }](#)

Width of the banner.

.

[int Height { get }](#)

Height of the banner.

[AdSizeType AdSizeType { get }](#)

The size of the banner.

Fields

[public static readonly AdSize BANNER_240x400 \(deprecated\)](#)

A banner of the 240x400 size in dp (density-independent pixels).

.

<code>public static readonly AdSize BANNER_300x250</code> (deprecated)	A banner of the 300x250 size in dp (density-independent pixels).
<code>public static readonly AdSize BANNER_300x300</code> (deprecated)	A banner of the 300x300 size in dp (density-independent pixels).
<code>public static readonly AdSize BANNER_320x50</code> (deprecated)	A banner of the 320x50 size in dp (density-independent pixels).
<code>public static readonly AdSize BANNER_320x100</code> (deprecated)	A banner of the 320x100 size in dp (density-independent pixels).
<code>public static readonly AdSize BANNER_400x240</code> (deprecated)	A banner of the 400x240 size in dp (density-independent pixels).
<code>public static readonly AdSize BANNER_728x90</code> (deprecated)	A banner of the 728x90 size in dp (density-independent pixels).

Method description

StickySize(int width)

```
static AdSize YandexMobileAds.Base.AdSize.StickySize (int width)
```

Creates an AdSize class object with the specified width of a “sticky” banner. Banner height depends on what types of ad units are selected in the Partner interface.

Parameters:

width The width of a “sticky” banner.

Returns:

An object of the AdSize class with a given width of a “sticky” banner.

FlexibleSize (int width, int height)

```
static AdSize YandexMobileAds.Base.AdSize.FlexibleSize (int width, int height)
```

Creates an AdSize class object with the specified maximum height and width of the banner. Banner height depends on what types of ad units are selected in the Partner interface.

Parameters:

width The maximum width of the banner.
height The maximum height of the banner.

Returns:

An object of the AdSize class with the specified maximum width and height of the banner.

Property description

int Width { get }

```
public int Width { get }
```

Width of the banner.

int Height { get }

```
public int Height { get }
```

Height of the banner.

int Height { get }

```
public AdSizeType AdSizeType { get }
```

The size of the banner.

Field description

public static readonly AdSize BANNER_240x400 (deprecated)

```
public static readonly AdSize BANNER_240x400 = new AdSize {Width = 240, Height = 400, AdSizeType = AdSizeType.Fixed}
```

A banner of the 240x400 size in dp (density-independent pixels).

public static readonly AdSize BANNER_300x250 (deprecated)

```
public static readonly AdSize BANNER_300x250 = new AdSize {Width = 300, Height = 250, AdSizeType = AdSizeType.Fixed}
```

A banner of the 300x250 size in dp (density-independent pixels).

public static readonly AdSize BANNER_300x300 (deprecated)

```
public static readonly AdSize BANNER_300x250 = new AdSize {Width = 300, Height = 300, AdSizeType = AdSizeType.Fixed}
```

A banner of the 300x300 size in dp (density-independent pixels).

public static readonly AdSize BANNER_320x50 (deprecated)

```
public static readonly AdSize BANNER_320x50 = new AdSize {Width = 320, Height = 50, AdSizeType = AdSizeType.Fixed}
```

A banner of the 320x50 size in dp (density-independent pixels).

public static readonly AdSize BANNER_320x100 (deprecated)

```
public static readonly AdSize BANNER_320x100 = new AdSize {Width = 320, Height = 100, AdSizeType = AdSizeType.Fixed}
```

A banner of the 320x100 size in dp (density-independent pixels).

public static readonly AdSize BANNER_400x240 (deprecated)

```
public static readonly AdSize BANNER_400x240 = new AdSize {Width = 400, Height = 240, AdSizeType = AdSizeType.Fixed}
```

A banner of the 400x240 size in dp (density-independent pixels).

public static readonly AdSize BANNER_728x90 (deprecated)

```
public static readonly AdSize BANNER_728x90 = new AdSize {Width = 728, Height = 90, AdSizeType = AdSizeType.Fixed}
```

A banner of the 728x90 size in dp (density-independent pixels).

Banner class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

A class for displaying banner ads.

Methods

<code>Banner(string blockId, AdSize adSize, AdPosition position)</code>	Creates an object of the Banner class with the specified parameters.
<code>void LoadAd(AdRequest request)</code>	Starts ad loading in the background.
<code>void Hide()</code>	Hides the object of the Banner class from the screen.
<code>void Show()</code>	Shows the object of the Banner class on the screen.
<code>void Destroy()</code>	Destroys an object of the Banner class and clears all the resources occupied by it.

Events

<code>event EventHandler<EventArgs> OnAdLoaded</code>	Called when the ad is loaded.
<code>event EventHandler<AdFailureEventArgs> OnAdFailedToLoad</code>	Called if ad loading fails.
<code>event EventHandler<EventArgs> OnReturnedToApplication</code>	Called when the user returns to the app after the click.
<code>event EventHandler<EventArgs> OnLeftApplication</code>	Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).
<code>event EventHandler<EventArgs> OnAdClicked</code>	It's called when the user clicks on your ad.
<code>event EventHandler<ImpressionData> OnImpression</code>	It's called when an ad impression has been counted.

Method description

Banner(string blockId, AdSize adSize, AdPosition position)

```
public Banner(string blockId, AdSize adSize, AdPosition position)
```

Creates an object of the Banner class with the specified parameters.

Parameters:

<code>blockId</code>	A unique identifier of the advertising space that is set in the Partner interface in the format demo-banner-yandex.
----------------------	---

`adSize` Banner ad sizes from the `AdSize` class.
`position` The banner's position on the screen.

Returns:

An object of the `Banner` class with the specified parameters.

void LoadAd(AdRequest request)

```
public void LoadAd(AdRequest request)
```

Starts ad loading in the background.

Parameters:

`request` An object of the `AdRequest` class including the data for a targeted ad request.

void Hide()

```
public void Hide()
```

Hides the object of the `Banner` class from the screen.

void Show()

```
public void Show()
```

Shows the object of the `Banner` class on the screen.

void Destroy()

```
public void Destroy()
```

Destroys an object of the `Banner` class and clears all the resources occupied by it.

Event description

event EventHandler<EventArgs> OnAdLoaded

```
public event EventHandler<EventArgs> OnAdLoaded
```

Called when the ad is loaded.

event EventHandler<AdFailureEventArgs> OnAdFailedToLoad

```
public event EventHandler<AdFailureEventArgs> OnAdFailedToLoad
```

Called if ad loading fails.

event EventHandler<EventArgs> OnReturnedToApplication

```
public event EventHandler<EventArgs> OnReturnedToApplication
```

Called when the user returns to the app after the click.

event EventHandler<EventArgs> OnLeftApplication

```
public event EventHandler<EventArgs> OnLeftApplication
```

Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).

event EventHandler<EventArgs> OnAdClicked

```
public event event EventHandler<EventArgs> OnAdClicked
```

It's called when the user clicks on your ad.

event EventHandler<ImpressionData> OnImpression

```
public event event EventHandler<ImpressionData> OnImpression
```

It's called when an ad impression has been counted.

Gender class



Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

The gender of the user.

Constants

`const string MALE`

Male.

.

`const string FEMALE`

Female.

.

Constant description

`const string MALE`

```
public const string MALE = "male"
```

Male.

`const string FEMALE`

```
public const string FEMALE = "female"
```

Female.

Interstitial class



Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

A class for displaying interstitial ads.

Methods

Interstitial(string blockId)	Creates an object of the Interstitial class with the specified parameters.
void LoadAd(AdRequest request)	Starts ad loading in the background.
void IsLoaded()	Returns the result of ad loading.
void Show()	Displays the ad if it was loaded.
void Destroy()	Destroys an object of the Interstitial class and clears its occupied resources.

Events

event EventHandler<EventArgs> OnInterstitialLoaded	Called when the ad is loaded.
event EventHandler<AdFailureEventArgs> OnInterstitialFailedToLoad	Called if ad loading fails.
event EventHandler<EventArgs> OnReturnedToApplication	Called when the user returns to the app after the click.
event EventHandler<EventArgs> OnLeftApplication	Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).
event EventHandler<EventArgs> OnAdClicked	It's called when the user clicks on your ad.
event EventHandler<EventArgs> OnInterstitialShown	Called when an interstitial ad is displayed.
event EventHandler<EventArgs> OnInterstitialDismissed	Called when an interstitial ad is closed.
event EventHandler<ImpressionData> OnImpression	It's called when an ad impression has been counted.
event EventHandler<AdFailureEventArgs> OnInterstitialFailedToShow	Called when the ad can't be shown.

Method description**Interstitial(string blockId)**

```
public Interstitial(string blockId)
```

Creates an object of the Interstitial class with the specified parameters.

Parameters:

blockId	A unique identifier of the advertising space that is set in the Partner interface in the format demo-banner-yandex.
---------	---

Returns:

An object of the `Interstitial` class with the specified parameters.

void LoadAd(AdRequest request)

```
public void LoadAd(AdRequest request)
```

Starts ad loading in the background.

Parameters:

request

An object of the `AdRequest` class including the data for a targeted ad request.

bool isLoaded()

```
public bool isLoaded()
```

Returns the result of ad loading.

Returns:

`true` when the ad is loaded and ready to be displayed. `false` if the ad is not loaded or not ready to be displayed.

void Show()

```
public void Show()
```

Displays the ad if it was loaded.

void Destroy()

```
public void Destroy()
```

Destroys an object of the `Interstitial` class and clears its occupied resources.

Event description

event EventHandler<EventArgs> OnInterstitialLoaded

```
public event EventHandler<EventArgs> OnInterstitialLoaded
```

Called when the ad is loaded.

event EventHandler<AdFailureEventArgs> OnInterstitialFailedToLoad

```
public event EventHandler<AdFailureEventArgs> OnInterstitialFailedToLoad
```

Called if ad loading fails.

event EventHandler<EventArgs> OnReturnedToApplication

```
public event EventHandler<EventArgs> OnReturnedToApplication
```

Called when the user returns to the app after the click.

event EventHandler<EventArgs> OnLeftApplication

```
public event EventHandler<EventArgs> OnLeftApplication
```

Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).

event EventHandler<EventArgs> OnAdClicked

```
public event EventHandler<EventArgs> OnAdClicked
```

It's called when the user clicks on your ad.

event EventHandler<EventArgs> OnInterstitialShown

```
public event EventHandler<EventArgs> OnInterstitialShown
```

Called when an interstitial ad is displayed.

event EventHandler<EventArgs> OnInterstitialDismissed

```
public event EventHandler<EventArgs> OnInterstitialDismissed
```

Called when an interstitial ad is closed.

event EventHandler<ImpressionData> OnImpression

```
public event EventHandler<ImpressionData> OnImpression
```

It's called when an ad impression has been counted.

event EventHandler<AdFailureEventArgs> OnInterstitialFailedToShow

```
public event EventHandler<AdFailureEventArgs> OnInterstitialFailedToShow
```

Called when the ad can't be shown.

Location.Builder class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

A class for building the user location.

Methods

Builder SetLatitude(double latitude)	Sets the latitude.
Builder SetLongitude(double longitude)	Sets the longitude.
Builder SetHorizontalAccuracy(double horizontalAccuracy)	Sets the horizontal accuracy.
Location Build()	Creates an object of the Location class that includes the data about the user location.

Method description

Builder SetLatitude(double latitude)

```
public Builder SetLatitude(double latitude)
```

Sets the latitude.

Parameters:

latitude	Latitude.
----------	-----------

Returns:

An object of the Location.Builder class with the specified latitude.

Property description

double Latitude { get }

```
public double Latitude { get }
```

Latitude.

double Longitude { get }

```
public double Longitude { get }
```

Longitude.

double HorizontalAccuracy { get }

```
public double HorizontalAccuracy { get }
```

Horizontal accuracy.

MobileAds class



Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

This class contains general Mobile Ads SDK settings.

Methods

[void SetUserConsent\(bool consent\)](#)

Sets a value that indicates whether a user from the GDPR region permits the collection of personal data that will be used for analytics and ad targeting. User data will not be collected until data collection is permitted. If the user once permitted or prohibited data collection, this value must be passed each time the app is launched.

[void SetLocationConsent\(bool consent\)](#)

It indicates if it's allowed to use a location to load targeted ads. By default, location data is not collected. In addition to getting the user's consent to collecting the location, you also need to request permission to access the location.

[void SetAgeRestrictedUser\(bool ageRestrictedUser\)](#)

The SDK automatically collects personal data if the user didn't restrict them. By default restriction is disabled.

Method description

void SetUserConsent(bool consent)

```
public static void SetUserConsent(bool consent)
```

Sets a value that indicates whether a user from the GDPR region permits the collection of personal data that will be used for analytics and ad targeting. User data will not be collected until data collection is permitted. If the user once permitted or prohibited data collection, this value must be passed each time the app is launched.

Parameters:

event EventHandler<EventArgs> OnLeftApplication	Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).
event EventHandler<EventArgs> OnAdClicked	It's called when the user clicks on your ad.
event EventHandler<EventArgs> OnRewardedAdShown	Called when the ad has been shown.
event EventHandler<EventArgs> OnRewardedAdDismissed	Called when the ad has been closed.
event EventHandler<ImpressionData> OnImpression	It's called when an ad impression has been counted.
event EventHandler<AdFailureEventArgs> OnRewardedAdFailedToShow	Called when the ad can't be shown.
event EventHandler<Reward> OnRewarded	Called when a user should be rewarded for viewing an ad (impression counted).

Method description

RewardedAd(string blockId)

```
public RewardedAd(string blockId)
```

Creates an object of the RewardedAd class with the specified parameters.

Parameters:

`blockId` A unique identifier of the advertising space that is set in the Partner interface in the format demo-banner-yandex.

Returns:

An object of the RewardedAd class with the specified parameters.

void LoadAd(AdRequest request)

```
public void LoadAd(AdRequest request)
```

Starts ad loading in the background.

Parameters:

`request` An object of the AdRequest class including the data for a targeted ad request.

bool isLoading()

```
public bool isLoading()
```

Returns the result of ad loading.

Returns:

`true` when the ad is loaded and ready to be displayed. `false` if the ad is not loaded or not ready to be displayed.

void Show()

```
public void Show()
```

Displays the ad if it was loaded.

void Destroy()

```
public void Destroy()
```

Destroys an object of the RewardedAd class and clears all its occupied resources.

Event description

event EventHandler<EventArgs> OnRewardedAdLoaded

```
public event EventHandler<EventArgs> OnRewardedAdLoaded
```

Called when the ad is loaded.

event EventHandler<AdFailureEventArgs> OnRewardedAdFailedToLoad

```
public event EventHandler<AdFailureEventArgs> OnRewardedAdFailedToLoad
```

Called if ad loading fails.

event EventHandler<EventArgs> OnReturnedToApplication

```
public event EventHandler<EventArgs> OnReturnedToApplication
```

Called when the user returns to the app after the click.

event EventHandler<EventArgs> OnLeftApplication

```
public event EventHandler<EventArgs> OnLeftApplication
```

Called when the app becomes inactive because the user has clicked on an ad and is about to switch to a different app (for example, a browser).

event EventHandler<EventArgs> OnAdClicked

```
public event event EventHandler<EventArgs> OnAdClicked
```

It's called when the user clicks on your ad.

event EventHandler<EventArgs> OnRewardedAdShown

```
public event event EventHandler<EventArgs> OnRewardedAdShown
```

Called when the ad has been shown.

event EventHandler<EventArgs> OnRewardedAdDismissed

```
public event event EventHandler<EventArgs> OnRewardedAdDismissed
```

Called when the ad has been closed.

event EventHandler<ImpressionData> OnImpression

```
public event event EventHandler<ImpressionData> OnImpression
```

It's called when an ad impression has been counted.

event EventHandler<AdFailureEventArgs> OnRewardedAdFailedToShow

```
public event event EventHandler<IAdFailureEventArgs> OnRewardedAdFailedToShow
```

Called when the ad can't be shown.

event EventHandler<Reward> OnRewarded

```
public event EventHandler<Reward> OnRewarded
```

Called when a user should be rewarded for viewing an ad (impression counted).

Reward class



Warning:

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

This class contains information about rewards for the user.

Methods

[Reward\(int amount, string type\)](#)

Creates an object of the Reward class with the specified parameters.

Fields

[string amount](#)

The reward amount in any units.

.

[string type](#)

Type of reward.

.

Method description

Reward(int amount, string type)

```
public Reward(int amount, string type)
```

Creates an object of the Reward class with the specified parameters.

Parameters:

amount

The reward amount in any units.

type

Type of reward.

Field description

string amount

```
public readonly string amount
```

The reward amount in any units.

string type

```
public readonly string type
```

Type of reward.

ScreenUtils class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

This class is responsible for converting pixels to density-independent pixels.

Methods

[ConvertPixelsToDp\(int pixels\)](#)

Converts the width in pixels to density-independent pixels.

Method description

ConvertPixelsToDp(int pixels)

```
public static int ConvertPixelsToDp(int pixels)
```

Converts the width in pixels to density-independent pixels.

Parameters:

pixels

The number of pixels.

AdPosition enumeration

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

The position of the banner.

Enumerations

[public enum AdPosition](#)

The position of the banner.

Enumeration description

public enum AdPosition

```
public enum AdPosition
{
    TopLeft,
    TopCenter,
    TopRight,
    CenterLeft,
    Center,
    CenterRight,
    BottomLeft,
    BottomCenter,
    BottomRight
}
```

The position of the banner.

TopLeft

At the top-left corner.

TopCenter

In the top center.

TopRight

In the top-right corner.

CenterLeft

Centered in the left corner.

Center

In the center.

CenterRight

Centered in the right corner.

BottomLeft

In the bottom-left corner.

BottomCenter

In the bottom center.

BottomRight

In the bottom-right corner.

AdSizeType class

**Warning:**

This is an archived version of the documentation. Actual documentation for all platforms can be found [here](#).

The size of the banner.

Enumerations

`public enum AdSizeType`

The size of the banner.

Enumeration description

public enum AdSizeType

```
public enum AdSizeType
{
    Fixed, // deprecated
    Flexible, // deprecated
    Sticky,
    Inline
}
```

The size of the banner.

Fixed (deprecated)

A banner with a specified maximum height and width.

Flexible (deprecated)

A banner with a specified maximum height and width.

Sticky

A sticky banner with the specified width.

Inline

A adaptive inline banner with a specified maximum height and width.