public class

Summary: Nested Classes | Inherited Constants | Fields | Ctors | Methods | Inherited Methods | [Expand All]

Added in API level 1

RemoteViews

extends Object
implements Parcelable LayoutInflater.Filter

java.lang.Object
↳android.widget.RemoteViews

Class Overview

A class that describes a view hierarchy that can be displayed in another process. The hierarchy is inflated from a layout resource file, and this class provides some basic operations for modifying the content of the inflated hierarchy.

Summary

Nested Classes

class RemoteViews.ActionException	Exception to send when something goes wrong executing an action
@interface RemoteViews.RemoteView	This annotation indicates that a subclass of View is alllowed to be used with the `RemoteViews` mechanism.

Inherited Constants [Expand]

▸ From interface android.os.Parcelable

Fields

public static final Creator<RemoteViews> CREATOR	Parcelable.Creator that instantiates RemoteViews objects

Public Constructors

RemoteViews (String packageName, int layoutId)
Create a new RemoteViews object that will display the views contained in the specified layout file.

RemoteViews (RemoteViews landscape, RemoteViews portrait)
Create a new RemoteViews object that will inflate as the specified landspace or portrait RemoteViews, depending on the current configuration.

RemoteViews (Parcel parcel)
Reads a RemoteViews object from a parcel.

Public Methods

void	addView (int viewId, RemoteViews nestedView) Equivalent to calling `addView(View)` after inflating the given `RemoteViews`.
View	apply (Context context, ViewGroup parent) Inflates the view hierarchy represented by this object and applies all of the actions.
RemoteViews	clone () Creates and returns a copy of this `Object`.

int	describeContents () Describe the kinds of special objects contained in this Parcelable's marshalled representation.
int	getLayoutId () Reutrns the layout id of the root layout associated with this RemoteViews.
String	getPackage ()
boolean	onLoadClass (Class clazz) Hook to allow clients of the LayoutInflater to restrict the set of Views that are allowed to be inflated.
void	reapply (Context context, View v) Applies all of the actions to the provided view.
void	removeAllViews (int viewId) Equivalent to calling `removeAllViews()`.
void	setBitmap (int viewId, String methodName, Bitmap value) Call a method taking one Bitmap on a view in the layout for this RemoteViews.
void	setBoolean (int viewId, String methodName, boolean value) Call a method taking one boolean on a view in the layout for this RemoteViews.
void	setBundle (int viewId, String methodName, Bundle value) Call a method taking one Bundle on a view in the layout for this RemoteViews.
void	setByte (int viewId, String methodName, byte value) Call a method taking one byte on a view in the layout for this RemoteViews.
void	setChar (int viewId, String methodName, char value) Call a method taking one char on a view in the layout for this RemoteViews.
void	setCharSequence (int viewId, String methodName, CharSequence value) Call a method taking one CharSequence on a view in the layout for this RemoteViews.
void	setChronometer (int viewId, long base, String format, boolean started) Equivalent to calling `Chronometer.setBase`, `Chronometer.setFormat`, and `Chronometer.start()` or `Chronometer.stop()`.
void	setContentDescription (int viewId, CharSequence contentDescription) Equivalent to calling View.setContentDescription(CharSequence).
void	setDisplayedChild (int viewId, int childIndex) Equivalent to calling `setDisplayedChild(int)`
void	setDouble (int viewId, String methodName, double value) Call a method taking one double on a view in the layout for this RemoteViews.
void	setEmptyView (int viewId, int emptyViewId) Equivalent to calling AdapterView.setEmptyView
void	setFloat (int viewId, String methodName, float value) Call a method taking one float on a view in the layout for this RemoteViews.
void	setImageViewBitmap (int viewId, Bitmap bitmap) Equivalent to calling ImageView.setImageBitmap
void	setImageViewResource (int viewId, int srcId) Equivalent to calling ImageView.setImageResource
void	setImageViewUri (int viewId, Uri uri) Equivalent to calling ImageView.setImageURI
void	setInt (int viewId, String methodName, int value) Call a method taking one int on a view in the layout for this RemoteViews.
void	setIntent (int viewId, String methodName, Intent value) Call a method taking one Intent on a view in the layout for this RemoteViews.

void setLabelFor (int viewId, int labeledId)
Equivalent to calling View.setLabelFor(int).

void setLong (int viewId, String methodName, long value)
Call a method taking one long on a view in the layout for this RemoteViews.

void setOnClickFillInIntent (int viewId, Intent fillInIntent)
When using collections (eg.

void setOnClickPendingIntent (int viewId, PendingIntent pendingIntent)
Equivalent to calling `setOnClickListener(android.view.View.OnClickListener)` to launch the provided `PendingIntent`.

void setPendingIntentTemplate (int viewId, PendingIntent pendingIntentTemplate)
When using collections (eg.

void setProgressBar (int viewId, int max, int progress, boolean indeterminate)
Equivalent to calling `ProgressBar.setMax`, `ProgressBar.setProgress`, and `ProgressBar.setIndeterminate` If indeterminate is true, then the values for max and progress are ignored.

void setRelativeScrollPosition (int viewId, int offset)
Equivalent to calling `smoothScrollToPosition(int, int)`.

void setRemoteAdapter (int viewId, Intent intent)
Equivalent to calling `setRemoteViewsAdapter(Intent)`.

void setRemoteAdapter (int appWidgetId, int viewId, Intent intent)
This method was deprecated in API level 14. This method has been deprecated. See `setRemoteAdapter(int, Intent)`

void setScrollPosition (int viewId, int position)
Equivalent to calling `smoothScrollToPosition(int, int)`.

void setShort (int viewId, String methodName, short value)
Call a method taking one short on a view in the layout for this RemoteViews.

void setString (int viewId, String methodName, String value)
Call a method taking one String on a view in the layout for this RemoteViews.

void setTextColor (int viewId, int color)
Equivalent to calling `setTextColor(int)`.

void setTextViewCompoundDrawables (int viewId, int left, int top, int right, int bottom)
Equivalent to calling `setCompoundDrawablesWithIntrinsicBounds(int, int, int, int)`.

void setTextViewCompoundDrawablesRelative (int viewId, int start, int top, int end, int bottom)
Equivalent to calling `setCompoundDrawablesRelativeWithIntrinsicBounds(int, int, int, int)`.

void setTextViewText (int viewId, CharSequence text)
Equivalent to calling TextView.setText

void setTextViewTextSize (int viewId, int units, float size)
Equivalent to calling `setTextSize(int, float)`

void setUri (int viewId, String methodName, Uri value)
Call a method taking one Uri on a view in the layout for this RemoteViews.

void setViewPadding (int viewId, int left, int top, int right, int bottom)
Equivalent to calling `setPadding(int, int, int, int)`.

void setViewVisibility (int viewId, int visibility)
Equivalent to calling View.setVisibility

void showNext(int viewId)
Equivalent to calling `showNext()`

void showPrevious(int viewId)
Equivalent to calling `showPrevious()`

void writeToParcel(Parcel dest, int flags)
Flatten this object in to a Parcel.

Inherited Methods [Expand]

- From class java.lang.Object
- From interface android.os.Parcelable
- From interface android.view.LayoutInflater.Filter

Fields

public static final Creator<RemoteViews> **CREATOR** Added in API level 1

Parcelable.Creator that instantiates RemoteViews objects

Public Constructors

public **RemoteViews** (String packageName, int layoutId) Added in API level 1

Create a new RemoteViews object that will display the views contained in the specified layout file.

Parameters

packageName	Name of the package that contains the layout resource
layoutId	The id of the layout resource

public **RemoteViews** (RemoteViews landscape, RemoteViews portrait) Added in API level 16

Create a new RemoteViews object that will inflate as the specified landspace or portrait RemoteViews, depending on the current configuration.

Parameters

landscape	The RemoteViews to inflate in landscape configuration
portrait	The RemoteViews to inflate in portrait configuration

public **RemoteViews** (Parcel parcel) Added in API level 1

Reads a RemoteViews object from a parcel.

Public Methods

public void **addView** (int viewId, RemoteViews nestedView) Added in API level 7

Equivalent to calling addView(View) (/reference/android/view/ViewGroup.html#addView(android.view.View)) after

inflating the given RemoteViews (/reference/android/widget/RemoteViews.html). This allows users to build "nested" RemoteViews (/reference/android/widget/RemoteViews.html). In cases where consumers of RemoteViews (/reference/android/widget/RemoteViews.html) may recycle layouts, use removeAllViews(int) (/reference/android/widget/RemoteViews.html#removeAllViews(int)) to clear any existing children.

Parameters

viewId	The id of the parent ViewGroup to add child into.
nestedView	RemoteViews that describes the child.

public View **apply** (Context context, ViewGroup parent)

Added in API level 1

Inflates the view hierarchy represented by this object and applies all of the actions.

Caller beware: this may throw

Parameters

context	Default context to use
parent	Parent that the resulting view hierarchy will be attached to. This method does **not** attach the hierarchy. The caller should do so when appropriate.

Returns

The inflated view hierarchy

public RemoteViews **clone** ()

Added in API level 1

Creates and returns a copy of this `Object`. The default implementation returns a so-called "shallow" copy: It creates a new instance of the same class and then copies the field values (including object references) from this instance to the new instance. A "deep" copy, in contrast, would also recursively clone nested objects. A subclass that needs to implement this kind of cloning should call `super.clone()` to create the new instance and then create deep copies of the nested, mutable objects.

Returns

a copy of this object.

public int **describeContents** ()

Added in API level 1

Describe the kinds of special objects contained in this Parcelable's marshalled representation.

Returns

a bitmask indicating the set of special object types marshalled by the Parcelable.

public int **getLayoutId** ()

Added in API level 1

Reutrns the layout id of the root layout associated with this RemoteViews. In the case that the RemoteViews has both a landscape and portrait root, this will return the layout id associated with the portrait layout.

Returns

the layout id.

public String **getPackage** ()

Added in API level 1

public boolean **onLoadClass** (Class clazz)

Added in API level 1

Hook to allow clients of the LayoutInflater to restrict the set of Views that are allowed to be inflated.

Parameters

clazz The class object for the View that is about to be inflated

Returns

True if this class is allowed to be inflated, or false otherwise

public void **reapply** (Context context, View v)

Added in API level 1

Applies all of the actions to the provided view.

Caller beware: this may throw

Parameters

v The view to apply the actions to. This should be the result of the `apply(Context, ViewGroup)` call.

public void **removeAllViews** (int viewId)

Added in API level 7

Equivalent to calling `removeAllViews()` `(/reference/android/view/ViewGroup.html#removeAllViews())`.

Parameters

viewId The id of the parent `ViewGroup` to remove all children from.

public void **setBitmap** (int viewId, String methodName, Bitmap value)

Added in API level 3

Call a method taking one Bitmap on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setBoolean** (int viewId, String methodName, boolean value)

Added in API level 3

Call a method taking one boolean on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setBundle** (int viewId, String methodName, Bundle value)

Added in API level 8

Call a method taking one Bundle on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setByte** (int viewId, String methodName, byte value)

Added in API level 3

Call a method taking one byte on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setChar** (int viewId, String methodName, char value)

Added in API level 3

Call a method taking one char on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setCharSequence** (int viewId, String methodName, CharSequence value)

Added in API level 3

Call a method taking one CharSequence on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setChronometer** (int viewId, long base, String format, boolean started)

Added in API level 1

Equivalent to calling `Chronometer.setBase` (/reference/android/widget/Chronometer.html#setBase(long)), `Chronometer.setFormat` (/reference/android/widget/Chronometer.html#setFormat(java.lang.String)), and `Chronometer.start()` (/reference/android/widget/Chronometer.html#start()) or `Chronometer.stop()` (/reference/android/widget/Chronometer.html#stop()).

Parameters

viewId	The id of the `Chronometer` to change
base	The time at which the timer would have read 0:00. This time should be based off of `SystemClock.elapsedRealtime()`.
format	The Chronometer format string, or null to simply display the timer value.
started	True if you want the clock to be started, false if not.

public void **setContentDescription** (int viewId, CharSequence contentDescription)

Added in API level 15

Equivalent to calling View.setContentDescription(CharSequence).

Parameters

viewId	The id of the view whose content description should change.
contentDescription	The new content description for the view.

public void **setDisplayedChild** (int viewId, int childIndex)

Added in API level 12

Equivalent to calling `setDisplayedChild(int)` `(/reference/android/widget/AdapterViewAnimator.html#setDisplayedChild(int))`

Parameters

viewId	The id of the view on which to call `setDisplayedChild(int)`

public void **setDouble** (int viewId, String methodName, double value)

Added in API level 3

Call a method taking one double on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setEmptyView** (int viewId, int emptyViewId)

Added in API level 11

Equivalent to calling AdapterView.setEmptyView

Parameters

viewId	The id of the view on which to set the empty view
emptyViewId	The view id of the empty view

public void **setFloat** (int viewId, String methodName, float value)

Added in API level 3

Call a method taking one float on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.

methodName	The name of the method to call.
value	The value to pass to the method.

public void **setImageViewBitmap** (int viewId, Bitmap bitmap)

Added in API level 3

Equivalent to calling ImageView.setImageBitmap

Parameters

viewId	The id of the view whose bitmap should change
bitmap	The new Bitmap for the drawable

public void **setImageViewResource** (int viewId, int srcId)

Added in API level 1

Equivalent to calling ImageView.setImageResource

Parameters

viewId	The id of the view whose drawable should change
srcId	The new resource id for the drawable

public void **setImageViewUri** (int viewId, Uri uri)

Added in API level 1

Equivalent to calling ImageView.setImageURI

Parameters

viewId	The id of the view whose drawable should change
uri	The Uri for the image

public void **setInt** (int viewId, String methodName, int value)

Added in API level 3

Call a method taking one int on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setIntent** (int viewId, String methodName, Intent value)

Added in API level 11

Call a method taking one Intent on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The `Intent` to pass the method.

public void **setLabelFor** (int viewId, int labeledId)

Added in API level 17

Equivalent to calling View.setLabelFor(int).

Parameters

viewId	The id of the view whose property to set.
labeledId	The id of a view for which this view serves as a label.

public void **setLong** (int viewId, String methodName, long value)

Added in API level 3

Call a method taking one long on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setOnClickFillInIntent** (int viewId, Intent fillInIntent)

Added in API level 11

When using collections (eg. `ListView` (/reference/android/widget/ListView.html), `StackView` (/reference/android/widget/StackView.html) etc.) in widgets, it is very costly to set PendingIntents on the individual items, and is hence not permitted. Instead a single PendingIntent template can be set on the collection, see `setPendingIntentTemplate(int, PendingIntent)` (/reference/android/widget/RemoteViews.html#setPendingIntentTemplate(int, android.app.PendingIntent)), and the individual on-click action of a given item can be distinguished by setting a fillInIntent on that item. The fillInIntent is then combined with the PendingIntent template in order to determine the final intent which will be executed when the item is clicked. This works as follows: any fields which are left blank in the PendingIntent template, but are provided by the fillInIntent will be overwritten, and the resulting PendingIntent will be used. of the PendingIntent template will then be filled in with the associated fields that are set in fillInIntent. See `fillIn(Intent, int)` (/reference/android/content/Intent.html#fillIn(android.content.Intent, int)) for more details.

Parameters

viewId	The id of the view on which to set the fillInIntent
fillInIntent	The intent which will be combined with the parent's PendingIntent in order to determine the on-click behavior of the view specified by viewId

public void **setOnClickPendingIntent** (int viewId, PendingIntent pendingIntent)

Added in API level 3

Equivalent to calling `setOnClickListener(android.view.View.OnClickListener)` (/reference/android/view/View.html#setOnClickListener(android.view.View.OnClickListener)) to launch the provided `PendingIntent` (/reference/android/app/PendingIntent.html). When setting the on-click action of items within collections (eg. `ListView` (/reference/android/widget/ListView.html), `StackView` (/reference/android/widget/StackView.html) etc.), this method will not work. Instead, use {@link RemoteViews#setPendingIntentTemplate(int, PendingIntent) in conjunction with RemoteViews#setOnClickFillInIntent(int, Intent).

Parameters

viewId	The id of the view that will trigger the `PendingIntent` when clicked
pendingIntent	The `PendingIntent` to send when user clicks

public void **setPendingIntentTemplate** (int viewId, PendingIntent pendingIntentTemplate)

Added in API level 11

When using collections (eg. `ListView` (/reference/android/widget/ListView.html), `StackView` (/reference/android/widget/StackView.html) etc.) in widgets, it is very costly to set PendingIntents on the individual items, and is hence not permitted. Instead this method should be used to set a single PendingIntent template on the collection, and individual items can differentiate their on-click behavior using `setOnClickFillInIntent(int, Intent)` (/reference/android/widget/RemoteViews.html#setOnClickFillInIntent(int, android.content.Intent)).

Parameters

viewId	The id of the collection who's children will use this PendingIntent template when clicked
pendingIntentTemplate	The `PendingIntent` to be combined with extras specified by a child of viewId and executed when that child is clicked

public void **setProgressBar** (int viewId, int max, int progress, boolean indeterminate)

Added in API level 1

Equivalent to calling `ProgressBar.setMax` (/reference/android/widget/ProgressBar.html#setMax(int)), `ProgressBar.setProgress` (/reference/android/widget/ProgressBar.html#setProgress(int)), and `ProgressBar.setIndeterminate` (/reference/android/widget/ProgressBar.html#setIndeterminate(boolean)) If indeterminate is true, then the values for max and progress are ignored.

Parameters

viewId	The id of the `ProgressBar` to change
max	The 100% value for the progress bar
progress	The current value of the progress bar.
indeterminate	True if the progress bar is indeterminate, false if not.

public void **setRelativeScrollPosition** (int viewId, int offset)

Added in API level 11

Equivalent to calling `smoothScrollToPosition(int, int)` (/reference/android/widget/AbsListView.html#smoothScrollToPosition(int, int)).

Parameters

viewId	The id of the view to change
offset	Scroll by this adapter position offset

public void **setRemoteAdapter** (int viewId, Intent intent) Added in API level 14

Equivalent to calling `setRemoteViewsAdapter(Intent)` (/reference/android/widget/AbsListView.html#setRemoteViewsAdapter(android.content.Intent)). Can only be used for App Widgets.

Parameters

viewId	The id of the `AdapterView`
intent	The intent of the service which will be providing data to the RemoteViewsAdapter

public void **setRemoteAdapter** (int appWidgetId, int viewId, Intent intent) Added in API level 11

> **This method was deprecated in API level 14.**
> This method has been deprecated. See `setRemoteAdapter(int, Intent)` (/reference/android/widget/RemoteViews.html#setRemoteAdapter(int, android.content.Intent))

Equivalent to calling `setRemoteViewsAdapter(Intent)` (/reference/android/widget/AbsListView.html#setRemoteViewsAdapter(android.content.Intent)).

Parameters

appWidgetId	The id of the app widget which contains the specified view. (This parameter is ignored in this deprecated method)
viewId	The id of the `AdapterView`
intent	The intent of the service which will be providing data to the RemoteViewsAdapter

public void **setScrollPosition** (int viewId, int position) Added in API level 11

Equivalent to calling `smoothScrollToPosition(int, int)` (/reference/android/widget/AbsListView.html#smoothScrollToPosition(int, int)).

Parameters

viewId	The id of the view to change
position	Scroll to this adapter position

public void **setShort** (int viewId, String methodName, short value) Added in API level 3

Call a method taking one short on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setString** (int viewId, String methodName, String value) Added in API level 3

Call a method taking one String on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setTextColor** (int viewId, int color)

Added in API level 3

Equivalent to calling `setTextColor(int)` (/reference/android/widget/TextView.html#setTextColor(int)).

Parameters

viewId	The id of the view whose text color should change
color	Sets the text color for all the states (normal, selected, focused) to be this color.

public void **setTextViewCompoundDrawables** (int viewId, int left, int top, int right, int bottom)

Added in API level 16

Equivalent to calling `setCompoundDrawablesWithIntrinsicBounds(int, int, int, int)` (/reference/android/widget/TextView.html#setCompoundDrawablesWithIntrinsicBounds(int, int, int, int)).

Parameters

viewId	The id of the view whose text should change
left	The id of a drawable to place to the left of the text, or 0
top	The id of a drawable to place above the text, or 0
right	The id of a drawable to place to the right of the text, or 0
bottom	The id of a drawable to place below the text, or 0

public void **setTextViewCompoundDrawablesRelative** (int viewId, int start, int top, int end, int bottom)

Added in API level 16

Equivalent to calling `setCompoundDrawablesRelativeWithIntrinsicBounds(int, int, int, int)` (/reference/android/widget/TextView.html#setCompoundDrawablesRelativeWithIntrinsicBounds(int, int, int, int)).

Parameters

viewId	The id of the view whose text should change
start	The id of a drawable to place before the text (relative to the

end	The id of a drawable to place after the text, or 0
bottom	The id of a drawable to place below the text, or 0

public void **setTextViewText** (int viewId, CharSequence text)

Added in API level 1

Equivalent to calling TextView.setText

Parameters

viewId	The id of the view whose text should change
text	The new text for the view

public void **setTextViewTextSize** (int viewId, int units, float size)

Added in API level 16

Equivalent to calling `setTextSize(int, float)` (/reference/android/widget/TextView.html#setTextSize(int, float))

Parameters

viewId	The id of the view whose text size should change
units	The units of size (e.g. COMPLEX_UNIT_SP)
size	The size of the text

public void **setUri** (int viewId, String methodName, Uri value)

Added in API level 3

Call a method taking one Uri on a view in the layout for this RemoteViews.

Parameters

viewId	The id of the view on which to call the method.
methodName	The name of the method to call.
value	The value to pass to the method.

public void **setViewPadding** (int viewId, int left, int top, int right, int bottom)

Added in API level 16

Equivalent to calling `setPadding(int, int, int, int)` (/reference/android/view/View.html#setPadding(int, int, int, int)).

Parameters

viewId	The id of the view to change
left	the left padding in pixels
top	the top padding in pixels
right	the right padding in pixels
bottom	the bottom padding in pixels

public void **setViewVisibility** (int viewId, int visibility)

Added in API level 1

Equivalent to calling View.setVisibility

Parameters

viewId	The id of the view whose visibility should change
visibility	The new visibility for the view

public void **showNext** (int viewId)

Added in API level 11

Equivalent to calling `showNext()` (/reference/android/widget/AdapterViewAnimator.html#showNext())

Parameters

viewId The id of the view on which to call showNext()

public void **showPrevious** (int viewId)

Added in API level 11

Equivalent to calling showPrevious()
(/reference/android/widget/AdapterViewAnimator.html#showPrevious())

Parameters

viewId The id of the view on which to call showPrevious()

public void **writeToParcel** (Parcel dest, int flags)

Added in API level 1

Flatten this object in to a Parcel.

Parameters

dest The Parcel in which the object should be written.

flags Additional flags about how the object should be written. May be 0 or PARCELABLE_WRITE_RETURN_VALUE.