Xamarin android set background image programmatically

I'm not robot	reCAPTCHA
Continue	-

```
ImageButton's public class extends ImageView Displays image button (instead of text) that can be pressed or pressed by the user. By default, ImageButton looks like a regular button is determined by either android: src attribute in the XML element or
ImageView.setImageResource (int). To remove the standard background image of the button, identify a different image for each state. For example, a blue image by default, orange for focus and yellow when pressed. An
easy way to do this is with an XML drawable selector. For example: ?xml version?1.0 encoding?utf-8?'gt;Save the XML file in the res/drawable/folder, and then send it as a drawing for the source of your ImageButton (in the android:state_pressed-true android:state_pressed-true android:@drawable/button_pressed !-- pressed --gt; zlt;item
android:state_focused'true android:drawable/button_focused'true android:drawable/button_focused in order. This is why the normal image button comes last, because it will only apply after android:state_pressed and android:state_focused both rated false. See
the button guide. XML Attributes See The attributes of the button, view attributes from the android.view android class:accessibility Fences whether to notify the user when this view changes. android:accessibilityPaneTitle The name of this view should
present for availability as the name of the panel. android:accessibilityTraversalAfter installs a view ID, after which this one is visited bypassing availability. android:Alpha property view as value between 0 (completely transparent) and 1 (completely opaque).
android:autofillHints describes the contents of the view so that the auto-filling service can fill in the relevant data. android:autofilled Highlight Drawable to be drawn over the look to mark it as autofilled can be a reference to another resource, in the form of a package:type/name or a theme attribute in the form of a lt/ImageButtongt; android: a background that can be used as a
background. android background: Tint for use in the background. Android background to click events. android:contentDescription defines text that summarizes the content of the view. android:contextClickable determines whether this view responds to click events in
context. android:defaultFocusHighlightEnabled Should this view use the default focus when it becomes focused but has no R.attr.state_focused in the background. android:duplicateParentState When this attribute is set to true, the view gets its drawable state (focused, pressed, etc.) from its direct
parent, not from itself. android: The height of the base z the depth of the view. android:fadeScrollbars determines whether to disappear from scrolling when not in use. android:fadingEdgeLength determines the length of withering edges. android:filterTouchesWhenObscured Indicates whether to filter touches when the view window is hidden by another visible window.
android:fitsSystemWindows Boolean is an internal attribute for adjusting the view layout based on system windows such as the state bar. android: Focused in touch mode. android:focusedByDefault Is this view the default view.
android:ForceHasOverlappingRendering Does this view have elements that can overlap when drawn. android:foreground defines draw to draw over content. android:foreground drawable. android:foreground drawable. android:foreground defines draw to draw over content.
android:hapticFeedbackEnabled Boolean, which monitors whether the species should have tactile feedback included for events such as long press. android:importantForAccessibility describes whether this view is important for accessibility. android:importantForAutofill
hints Android system whether the node view associated with this kind should be included in the structure of the view used for auto-fill purposes. android:importantForContentCapture hints at whether to use the view node associated with this view to capture content. Install this if the view will serve as a scroll container, meaning that it can be reused to shorten the overall window, so
there will be room for the input method. android:keepScreenOn monitors whether the view window Keep the screen for a while visible. android:keyboardNavigationCluster Is this representation of the root of the keyboard's navigation cluster. android:keyboardNavigationCluster Is this representation of the root of the keyboard's navigation cluster.
android:longClickable determines whether this view responds to long-click events. android:minHeight determines the minimum height of the species android:nextFocusDown defines the following view to give focus when the next focus
View.FOCUS_DOWN If the link refers to a view that does not exist or is part of a hierarchy that is invisible, RuntimeException will result when the link refers to a view that does not exist or is part of a hierarchy that is invisible, RuntimeException
will result when link access. android:nextFocusLeft determines the following look to give focus when the next focus View.FOCUS_RIGHT If the link refers to a view that does not exist or is part of a hierarchy that is invisible, RuntimeException will result when the link is
accessed. android:nextFocusUp defines the following view to give focus when the next focus View.FOCUS_UP If the link refers to a view that does not exist or is part of a hierarchy that is invisible, RuntimeException will result when the link is accessed. android:onClick Method Name in the context of this view to call when you click the view. android:outlineAmbientShadowColor
sets the color of the surrounding shade, which is drawn when the species has a positive value or height. android:padding sets uping, in pixels, all four edges. android:paddingBottom installs ups ups and ups, in pixels, bottom edges; See
R.attr.padding. android:paddingEnd installs ups ups and ups, in pixels, left and right edge; See R.attr.padding. android:paddingHorizontal installs ups ups, in pixels, left edge; See R.attr.padding. android:paddingEnd installs ups ups and ups, in pixels, right edge; See R.attr.padding.
android:PaddingStart sets up uping, in pixels, from the edge of the start; See R.attr.padding. android:rotation X rotation Start sets up uping, in pixels, top edge; See R.attr.padding. android:vert paddingical installs ups and lower edges; Cm. Determines which edges should be faded when scrolling. android: rotation of the view, in degrees. android:rotation X rotation
of the species around the x axis, in degrees. android: rotation of the view around the axis of y, in degrees. android:scaleY scale of view in the direction of y. android:screenReaderFocusable Should consider this view as a focused unit of screen-
accessibility tools. android:scrollIndicators determines which scroll indicators should appear when scrolling view. android:scrollbarAlwaysDrawHorizontalTrack determines whether a horizontal scroll track should always be drawn.
android:scrollbarAlwaysDrawVerticalTrack determines whether a vertical scroll track should always be drawn. android:scrollbarDefaultDelayBeforeFade detects a delay in milliseconds that require scrolling to disappear. android:scrollbarSize sets the width of vertical scrolling and
the height of horizontal scrolling. android:scrollbar thumb scroll drawable. android:scrollbar thumb draw. android:scrollbar t
scrolling of the drawable track. android:scrollbars determines which scrolls should be displayed when scrolling or not. android:soundEffectsEnabled Boolean, which monitors whether the species should have sound effects included for events such as pressing and touch. android:stateListAnimator sets up a state animator for View. android:tag Supply tag for this view containing a
line that will be received later from View.getTag () or searched with View.findViewWithTag (). android:textAlignment determines the theme of override for presentation. android:textPlimion determines text alignment determines text alignment. android:textPlimion determines text alignment determines text alignment.
the pivot point around which the view will rotate and scale. android:translation in x vision. android:translation in y view. android:translation android:translation in y view. android:translation android:translation in z kind. android:visibility
controls the initial visibility of the species. From the Android.view.view int mode, ACCESSIBILITY_LIVE_REGION_ASSERTIVE Live, indicating that accessibility services must interrupt your current speech to immediately announce changes to this view. int ACCESSIBILITY_LIVE_REGION_NONE Live, stating that accessibility services should not automatically announce changes
to that view. int ACCESSIBILITY_LIVE_REGION_POLITE Live, indicating that accessibility services should announce changes to this view. int AUTOFILL_FLAG_INCLUDE_NOT_IMPORTANT_VIEWS Flag asking you to add views that are labeled as not important for autocomplete (see Feature ForAutofill (int)) in ViewStructure. The line
AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DATE a hint that this view can automatically be filled with a credit card expiration date. The AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_MONTH a hint that this view can be automatically filled with the expiration day of the credit card. The line AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_MONTH a hint that this view can be
automatically filled with a credit card expiration month. The line AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_YEAR a hint that this view can be automatically filled with a credit card expires. The line AUTOFILL_HINT_CREDIT_CARD_NUMBER a hint that this view can be automatically filled with a credit card number. The line
AUTOFILL_HINT_CREDIT_CARD_SECURITY_CODE a hint that this view can be automatically filled with a credit card security code. The line AUTOFILL_HINT_NAME a hint that this view can be automatically filled with the user's real name. The line
AUTOFILL_HINT_PASSWORD a hint that this view can be automatically filled with a password. The line AUTOFILL_HINT_POSTAL_CODE a hint that this view can be automatically filled with a mailing address. The line AUTOFILL_HINT_POSTAL_CODE a hint that this view can be automatically filled with a password. The line AUTOFILL_HINT_POSTAL_CODE a hint that this view can be automatically filled with a password.
be automatically filled with postcode. The line AUTOFILL_HINT_USERNAME a hint that this view can be automatically filled with the username int AUTOFILL_TYPE_DATE Autofill for a field that contains a long, representing number of milliseconds from the standard base time known as the epoch, namely January 1, 1970, 00:00 GMT (see Date.getTime(). int
AUTOFILL_TYPE_LIST Autofill type for the selection list field that is filled int, item index inside the list (starting with 0). int AUTOFILL_TYPE_TEXT autofill for the text field that is filled with CharSequence. int AUTOFILL_TYPE_TOGGLE autofill style for a togglable field, which is filled with
boolean. Boolean. DRAG FLAG GLOBAL indicating that drag can cross the windows. int DRAG FLAG GLOBAL PERSISTABLE URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI READ and/or DRAG FLAG GLOBAL URI WRITE, the URI grant can be retained through the device's reboot until Context.revokeUriPermission (Uri, int) int
DRAG_FLAG_GLOBAL_PREFIX_URI_PERMISSION When this flag is used with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ with DRAG_FLAG_GLOBAL_URI_READ with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_READ with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG
access to the contents of the URI (s) content in the ClipData facility. int DRAG_FLAG_GLOBAL, the drag recipient will be able to request access to the URI content in the ClipData facility. int DRAG_FLAG_OPAQUE the flag indicating the opacity of the drag shadow. int
DRAWING_CACHE_QUALITY_AUTO This constant has been deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the view drawing cache is largely out of date with the view drawing cach
updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view.
However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. int DRAWING_CACHE_QUALITY_HIGH This constant has been
deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful
for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it is recommended that you create a canvas either from Bitmap or from and call (android.graphics.Canvas) on view. However, these software uses are not encouraged to be encouraged
have compatibility issues with hardware-only rendering features such as bitmaps Config.HARDWARE, real-time shadows, and contour clipping. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. int DRAWING_CACHE_QUALITY_LOW This constant has been deprecated in API level 28. The view drawing cache is largely out of date with
the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint)
handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such
as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Int FIND_VIEWS_WITH_TEXT Find views that display the text. int FOCUSABLE This view wants to press the keys. Int
FOCUSABLES_ALL View flag indicating that addFocusables (java.util.ArrayList, Int, Int) should add all focused in touch mode. Int FOCUSABLES_TOUCH_MODE View flag indicating that addFocusables (java.util.ArrayList, Int, Int) should only add views concentrated in touch mode. Int FOCUSABLE_AUTO This view determines focus
automatically. int FOCUS_BACKWARD Use with focusSearch (int). Int FOCUS_LEFT Use with focusSearch (int). Int FOCUS_LEFT Use with focusSearch (int). Int FOCUS_RIGHT Use with focusSearch (int). Int FO
the layout. int HAPTIC FEEDBACK ENABLED View indicating whether this view should have tactile feedback included for events such as long presses. int IMPORTANT FOR ACCESSIBILITY NO View is not important for availability. int
IMPORTANT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS Kind is not important for nor did any of his descendants's opinions. int IMPORTANT_FOR_AUTOFILL_AUTO automatically determine whether the view is important to the auto-fill. int IMPORTANT_FOR_AUTOFILL_NO View not not
for the auto-filled, but her children (if any) will be passed. int IMPORTANT FOR AUTOFILL NO EXCLUDE DESCENDANTS The view is not important for the auto-filled, and its children (if any) will be traversed. int
IMPORTANT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS The view is important for capturing content. int IMPORTANT_FOR_CONTENT_CAPTURE_NO View is not important for capturing content, but
its children (if any) will be passed. int IMPORTANT_FOR_CONTENT_CAPTURE_NO_EXCLUDE_DESCENDANTS View is not important for capturing content, and its children (if any) will not be passed. Int IMPORTANT_FOR_CONTENT_CAPTURE_YES View is important for capturing content, and its children (if any) will be passed. Int
IMPORTANT_FOR_CONTENT_CAPTURE_YES_EXCLUDE_DESCENDANTS View is important for capturing content, but its children (if any) will not be passed. int INVISIBLE This view is invisible, but it still takes place for the purpose of the layout. int KEEP_SCREEN_ON View indicating that the screen should remain while the window containing this view is visible to the user.
int LAYER_TYPE_HARDWARE indicates that the view has a hardware layer. int LAYER_TYPE_NONE indicates that the view does not have a layer of software. int LAYOUT_DIRECTION_INHERIT horizontal direction of this view layout inherited from his parent. int LAYOUT_DIRECTION_LOCALE the direction of
the horizontal layout of this view is the default language output from the script for the layout of this species from right to left. int MEASURED_HEIGHT_STATE_SHIFT Bit shift MEASURED_STATE_MASK to get to the
height of bits for features that combine width and height in one int, such as getMeasuredState () and childState argument resolveSizeAndState, which provide actual measured size. int MEASURED_STATE_MASK bits getMeasuredWidthAndState () and getMeasuredWidthAndState () and getMeasuredWidthAndState argument resolveSizeAndState (int, int, int).
getMeasuredWidthAndState, which provide additional bits of state. int MEASURED_STATE_TOO_SMALL Bit of getMeasuredWidthAndState, which indicates a smaller size of space that would like to have a view. int NOT_FOCUSABLE This view doesn't want to press. int NO_ID used marking the view without ID. int OVER_SCROLL_ALWAYS
Always let the user twist this view, provided it's a view that can scroll. int OVER_SCROLL_IF_CONTENT_SCROLLS Allow the user twisted this view only if the content is large enough to scroll meaningfully, provided it's a view that can scroll. OVER_SCROLL_NEVER never let the user twist this view. int SCREEN_STATE_OFF indicates that the screen has changed state and is
now off. int SCREEN_STATE_ON indicates that the screen has changed state and is now on. Int SCROLLBARS_INSIDE_OVERLAY the scroll style to show scrolling inside the content area without increasing the upsize. int
SCROLLBARS_OUTSIDE_INSET the scroll style to show scrolling on the edge of the view, increasing the upsition defined by the system. int
SCROLLBAR_POSITION_LEFT position bar scrolling along the left edge. int SCROLL_AXIS_VERTICAL indicates scrolling along the vertical axis. int SCROLL_AXIS_NONE indicates that there is no axis of scrolling the view. int SCROLL_AXIS_VERTICAL indicates scrolling along the vertical
axis. int SCROLL_INDICATOR_BOTTOM direction of the Scroll indicator for the bottom edge of the view. int SCROLL_INDICATOR_END direction of the Scroll indicator for the left edge of the view. int SCROLL_INDICATOR_RIGHT direction of the Scroll indicator for the right edge of the
view. int SCROLL_INDICATOR_START direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. Int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view. Int SCROLL_INDICATOR_TOP direction of the Scroll indicator for the starting edge of the view.
constant has been deprecated in API level 15. Use SYSTEM UI FLAG VISIBLE instead, int SYSTEM UI FLAG VISIBLE instead, int SYSTEM UI FLAG VISIBLE instead, int SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG VISIBLE instead, int SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG VISIBLE instead, int SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN This constant has been deprecated in API level 15. Use SYSTEM UI FLAG FULLSCREEN THIS con
SYSTEM_UI_FLAG_HIDE_NAVIGATION This constant was deprecated in the api level of 30. Instead, use WindowInsetsController'hide (int) with Type'navigationBars. int SYSTEM_UI_FLAG_IMMERSIVE This constant has been deprecated in API 30 level. Instead, use WindowInsetsController BEHAVIOR_SHOW_BARS_BY_SWIPE instead. int
SYSTEM_UI_FLAG_IMMERSIVE_STICKY Constant has been deprecated in API 30. Instead, use WindowInsetsController BEHAVIOR_SHOW_TRANSIENT_BARS_BY_SWIPE instead. int SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN This constant has been deprecated in API 30 level. For The For use LayoutParams-setFitInsetsTypes (int) with Type-statusBars. For non-
floating windows, filling the screen, call Window/SetDecorFitsSystemWindows (boolean) with false. int SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION This constant has been deprecated in API 30. For floating windows, filling the screen, call Window/SetDecorFitsSystemWindows
(boolean) with false. int SYSTEM_UI_FLAG_LIGHT_NAVIGATION_BAR This constant was deprecated in the api level of 30. Instead, use WindowInsets'getInsIgnoringVisibility (int) to get branches that don't change when the system's bars look good. int SYSTEM_UI_FLAG_LIGHT_NAVIGATION_BAR This constant has been deprecated in API 30 level. Instead, use
WindowInsetsController APPEARANCE_LIGHT_NAVIGATION_BARS instead. int SYSTEM_UI_FLAG_LIGHT_STATUS_BAR This constant has been deprecated in API 30 level. The low profile mode
is unified. Hide the system bars if the app should be in unobtrusive mode. Use WindowInsetsController-hide (int) with Type'systemBars. int SYSTEM_UI_FLAG_VISIBLE This constant has been deprecated in API 30. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. int SYSTEM_UI_LAYOUT_FLAGS This constant has been deprecated in API 30. Instead, use WindowInsetsController.
Flags of the system UI layout are de-edist. int TEXT_ALIGNMENT_CENTER center item, for example, ALIGN_OPPOSITE. int TEXT_ALIGNMENT_TEXT_START align to the end of the item, for example, ALIGN_OPPOSITE. int TEXT_ALIGNMENT_TEXT_START align.
with the beginning of an item, such as ALIGN_NORMAL. int TEXT_ALIGNMENT_VIEW_END align to the end of the view, which is ALIGN_RIGHT if the permitted Direction view layout is LTR, and ALIGN_LEFT otherwise. int TEXT_ALIGNMENT_VIEW_START aligned to the beginning of the view, which ALIGN_LEFT if the permitted layout of the Direction view is LTR, and
ALIGN_RIGHT otherwise. Int TEXT_DIRECTION_ANY_RTL Text uses the algorithm any-RTL. Int TEXT_DIRECTION_FIRST_STRONG_LTR Text uses the first strong algorithm. Int TEXT_DIRECTION_FIRST_STRONG_RTL Text uses the first strong algorithm. Int TEXT_DIRECTION_INHERIT Text
Direction is inherited through ViewGroup int TEXT_DIRECTION_LOCALE Text Direction comes from the Locale system. int TEXT_DIRECTION_LTR the direction of text Rtl. The line VIEW_LOG_TAG the magazine tags used in this class with android.util.Log. int VISIBLE This view is visible. From the class of
android.view.View public static finale the ALPHA A Property wrap around alpha functionality processed by View-setAlpha (float) and View-getAlpha (float) and
and selected. The protected static final ENABLED_FOCUSED_STATE_SET indicates that the view is on, focused, selected, and its window is in focus. protected static final ENABLED_FOCUSED_WINDOW_FOCUSED_STATE_SET
indicates that the view is on, focused, and its window is in focus. a protected static final int ENABLED_SELECTED_WINDOW_FOCUSED_STATE_SET indicates that the view is on, selected and its window has a focus. protected static final int ENABLED_STATE_SET indicates the
inclusion of the view. protected static final int ENABLED_WINDOW_FOCUSED_STATE_SET indicates that the view is on and that its window has a focus. a protected static final FOCUSED_SELECTED_WINDOW_FOCUSED_STATE_SET indicates that the view is
focused, selected, and its window is in focus. protected static final int FOCUSED_STATE_SET indicates the focus of the view and the focus of the view and the focus of the view. The protected static final int-PRESSED_ENABLED_FOCUSED_STATE_SET points to depression, inclusion, focus and
selected. protected static final int-PRESSED_ENABLED_FOCUSED_SELECTED_WINDOW_FOCUSED_STATE_SET points to the depression, engaged, focused performance. Protected static final int PRESSED_ENABLED_FOCUSED_STATE_SET points to depression, included and focused performance. Protected static final interpretation for the depression for the dep
PRESSED_ENABLED_FOCUSED_WINDOW_FOCUSED_STATE_SET indicates clicking, turning, focusing, and focusing the window. a protected static final int-PRESSED_ENABLED_SELECTED_WINDOW_FOCUSED_STATE_SET points to the
depression, enabled, selected and its window has focus. protected static final int PRESSED_ENABLED_STATE_SET points to the click and inclusion of Protected static final PRESSED_ENABLED_WINDOW_FOCUSED_STATE_SET points to
the depression, focus and selection of the view. protected static final int yказывается, что представление нажато, сфокусировку. защищенная статическая окончательная int PRESSED_FOCUSED_STATE_SET указывает на нажатие и фокусировку представления. защищенная статическая финальная
PRESSED_FOCUSED_WINDOW_FOCUSED_STATE_SET указывает на нажатие, фокусировку окна. защищенная статическая окончательная int- PRESSED_SELECTED_STATE_SET указывает на нажатие и выбрано представление. защищенная статическая окончательная int-
PRESSED_SELECTED_WINDOW_FOCUSED_STATE_SET указывает на нажатие, выбрано представления и его окно имеет фокусировку. защищенный статическая окончательная int PRESSED_WINDOW_FOCUSED_STATE_SET указывает на нажатие
представления и фокусировки окна. публичная статическая окончательная обертка Свойства СВОЙСТВА вокруг функциональности вращения, обрабатываемая<View, Float&gt; ROTATION_X. публичная&lt;View, Float&gt; статическая окончательная обертка
свойства ROTATION_Y функциональность rotationY, обрабатываемая методами View-setRotationY (float) и View-getRotationY () SCALE_X .<View, Float&gt; &lt;View, Float&gt; ryбличная статическая окончательная собственность SCALE_X .&lt;View, Float&gt; &lt;View, Float&gt; ryбличная статическая окончательная собственность SCALE_X .&lt;View, Float&gt; Repair (float) и View-getRotationY (float) и View-getScaleY (float) и View-getRotationY (float) и Vie
SELECTED_STATE_SET< View, Float&gt; защищенная статическая окончательная int's SELECTED_STATE_SET Указывает на то, что представление выбрано TRANSLATION_X. обрабатываются методами View-setTranslationX (float) и View-getTranslationX (). публичная статическая окончательная собственность
TRANSLATION_Y Обертка свойства вокруг функциональности translationY, обрабатываемой<View, Float&gt; функциональностью View-setTranslationY () методами TRANSLATION_Z. защищенный статический окончательный int's WINDOW_FOCUSED_STATE_SET Указывает на то, что окно представления имеет фокус.
общедоступную статическую обертку Свойства X A Property вокруг функциональности x, обрабатываемой<View, Float&gt; &lt;View, Float&gt; &lt;View
</View, Float&gt; &lt;/View, Float&gt; &lt;/View, Float&gt; &lt;/View, Float&gt; &lt;/View, Float&gt; View-set (float) and View-get. boolean onSetAlpha (int alpha) Is called if there is a conversion that includes alpha. From the class android.widget.ImageView void animateTransform (Matrix Matrix) Applies a temporary conversion matrix for the view drawable when drawn. The final void of clearColorFilter
() Removes ColorFilter images. The void drawableHotspotChanged (float x, float y) This feature is called whenever the view access point changes and should be distributed for drawing or children's views controlled by the view access point changes and should be distributed for drawing or children's views controlled by the view access point changes and should be distributed for drawing or children's views controlled by the view access point changes and should be distributed for drawing or children's views controlled by the view access point changes in such a way that it affects the state of the view access point changes and should be distributed for drawing or children's views controlled by the view access point changes and should be distributed for drawing or children's views access point changes and should be distributed for drawing or children's views access point changes and should be distributed for drawing or children's views access point changes and should be distributed for drawing or children's views access point changes and should be distributed for drawing or children's view access point changes and should be distributed for drawing or children's view access point changes and should be distributed for drawing or children's view access point changes and should be distributed for drawing or children's view access point changes and should be distributed for drawing access to the view access point or changes and should be distributed for drawing access to the view access point or changes and should be distributed for drawing access to the view access point or changes and should be distributed for drawing access to the view access point or changes and should be distributed for drawing access to the view access
getAccessibilityClassName () Return the name of the class of this object, which will be used for availability. Boolean getAdjustViewBounds () The truth is when ImageView adjusts its boundary of the widget from the top boundary of the widget. boolean getBaselineAlignBottom () Checks
whether the base level of this view is considered to be the bottom of the view. ColorFilter getColorFilter getColorFilter returns an active color filter for this ImageView. boolean getCropToHabits () gets the current Drawable, or zero if not Drawable has been assigned. int getImageAlpha () Returns alpha, which will be applied to
the hand-drawn imageView. The getImageMatrix Matrix returns an additional view matrix. BlendMode () Receives the mixing mode used to apply hue to the Drawable image, or zero if no shade is applied. PorterDuff.Mode getImageTintMode () Receives
the mix mode used to apply hue to the Picture Drawable int getMaxHeight () Maximum height of this species. int getMaxWidth (the maximum width of this view. ImageView. Boolean hasOverlappingRendering () Whether this species has content that
overlaps. (Drawable dr) cancels the drawable. boolean isOpaque () indicates whether this species is opaque. invalid jumpDrawablesToCurrentState () Drawable-jumpToCurrentState () Drawable objects associated with this view. The void on The CaptionWindow is called when the species is attached to the window. Int onCreateDrawableState (int extraSpace) Create a new
drawable for this performance. The void on DetachedFromWindow is called when the view separates from the window. Void onDraw (Canvas canvas) Implement this to make your drawing. void on Measure (int width measure height. The void on RtIPropertiesChanged
(int layoutDirection) is called when any RTL property (the direction of the layout or the direction of the text or the alignment of the text or the alignment of the text) has been change of the view itself, the representation of the ancestor, or the window to which the view is
attached. Invalid setAdjustViewBounds (boolean adjustViewBounds) Install this to make it true if you want ImageView to adjust its boundaries to keep the side-of-the-picture ratio. This method was faded at API level 16. Use #setImageAlpha (int) instead of the invalid set
ofBaselineAlignBottom (buline aligned) determines whether the base level of this view is at the bottom of the view. The final set of TheLorFilter (ColorFilter to the image. The final set of the ColorFilter void (int color) set the toning option for the image. Invalid
setCropToCharged (boolean cropToIse) determines whether this ImageView will be trimmed to ups ups and down. boolean setFrame (int I, int r, int b) invalid setImageAlpha (int alpha) sets the content of this ImageView. The invalid setImageDrawable
(Drawable drawable) sets drawable as the content of this ImageView. The invalid imageIcon set (icon icon) installs the contents of this ImageView on the specified icon. The invalid setImageWatrix (Matrix Matrix) adds a transformation matrix that applies to a hand-drawn view
when it is drawn. The invalid setImageResource (int resId) sets drawable as the content of this ImageView. Invalid SetImageTintList (android.content.res.ColorStateList) to the
picture drawable. The invalid setImageTintList (ColorStateList hue) applies a drawable image FintList (android.content.res.ColorStateList) to the image drawable. Invalid setImageURI (Uri uri) sets the content of this ImageView to the specified Uri. Ineffective MaxHeight (int
maxHeight) Additional argument for providing maximum width for this view. The invalid set Of MaxWidth (int maxWidth) An additional argument to provide maximum width for this view. The invalid set Elected (selected by bulean)
bulean) State of choice of this view. Invalid setVisibility (int visibility (int visibility) Set the state of visibility of this species. boolean verifyDrawable (Drawable it displays. From the android.view.View void addChildrenForAccessibility (ArrayList outChildren) Adds to
the Children of This View the attitude to access to this list as an outlet. invalid addExtraDataToAccesibilityNodeInfo (AccessibilityNodeInfo based on an explicit request for additional data to AccessibilityNodeInfo (AccessibilityNodeInfo info, String extraDataKey, Bundle arguments) adds additional data to AccessibilityNodeInfo based on an explicit request for additional data.
this point of view (perhaps including this point of view, if it is a descendant of this view, if it is the cluster root itself) to the views.
voidonAttachStateChangeListener (View.OnAttachStateChangeListener (View.OnAttachStateChangeListener (View.OnAttachStateChangeListener) Add a listener to join state changes. addonLayoutChangeListener (View.OnLayoutChangeListener) Add a listener to be called when the boundaries of the view change due to layout processing. invalid addOnUnhandledKeyEventListener (View.OnUnhandledKeyEventListener) Add a listener to join state changes. addonLayoutChangeListener (View.OnLayoutChangeListener) Add a listener to join state changes.
receive unruly KeyEvents. Invalid addTouchables (species of ArrayList) Add any tangible views that are descendants of this view, (perhaps including this view, if it is touches itself) to views. ViewPropertyAnimator animates this method returns the ViewPropertyAnimator object, which can be used to animate certain properties in this view. Invalid to announceForAccessibility
(CharSequence text) Convenience method for sending AccessibilityEvent-TYPE_ANNOUNCEMENT AccessibilityEvent suggest that the accessibility service announce the specified text to its users. AutofillValue automatically fills the content of virtual children in this view. boolean
awakenScrollBars (int startDelay, boolean Tpurrep scrollbars pucoвать. boolean awakenScrollBars (int startDelay) Триггер прокрутки рисовать. boolean awakenScrollBars (int startDelay, boolean awakenScrollBars (int startDelay) Триггер прокрутки рисовать. boolean awakenScrollBars (int startDelay) Триггер прокрутки рисовать.
with the introduction of hardware accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int,
android graphics. Paint) handles this with hardware visualization. For software image and call (android graphics. Canvas) on the view, it's a good idea to create a canvas either from Bitmap or from an image and call (android graphics. Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only
rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. ineffective buildDrawingCache () This method has been mutilated in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization
in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software
images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view.
and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. The invalid buildLayer creates a layer of this view that will be created, and this view will be drawn into its layer. boolean callOnClick () Directly call anyone attached onClickListener. boolean canResolveLayoutDirection () Check to see if you can make a solution to
the direction of the layout. boolean canResolveTextAlignment () Check to see if you can do permission to align the text. boolean canScrollHorizontally (inner direction) Check to see if you can scroll this view horizontally in a certain direction. boolean canScrollVertically (int
direction) Check to see if you can scroll this view vertically in a certain direction. Final cancels In anticipation of a long press. The final cancels the current drag and fall operation invalid cancels In anticipation of a long press. The final cancels In anticipation of a long press. The final cancels In anticipation of a long press. The final cancels In anticipation of a long press.
checkinputConnectionProxy (View Called InputMethodManager, when a view that is not the current purpose of the input connection, tries to call the manager. invalid clearAnimation () cancels any animations for this view. The emptiness of clearFocus is called when this point of view wants to give up focus. Static Int combineMeasuredStates (int curState, int newState) Merge of the
two states as a return getMeasuredState (). Int computeHorizontalScrollExtent () Calculate the horizontal thumb displacement horizontal scrolling in horizontal range. Int computeHorizontalScrollRange () Calculate the horizontal range that represents horizontal scrolling in horizontal range. Int computeHorizontalScrollRange () Calculate the horizontal thumb displacement horizontal scrolling in the horizontal range. Int computeHorizontalScrollRange () Calculate the horizontal range that represents horizontal scrolling in the horizontal range.
scrolling. Invalid computeScroll () Is called by the parent to ask the child to update their values for mScrollX and mScrollX and mScrollX and mScrollX and mScrollX and mScrollX and those that should be distributed among those under it. Int computeVerticalScrollExtent () Calculates
the vertical degree of the thumb vertical scrolling in the vertical scrolling in the vertical scrolling in horizontal range. Int compute Vertical scrolling in horizontal range () Calculate the vertical scrolling in horizontal range () Calculate the vertical scrolling in the vertical scrolling in horizontal range. Int compute Vertical scrolling in horizontal range () Calculate the vertical scrolling in the vertical scrolling in the vertical scrolling in horizontal range () Calculate the vertical scrolling in horizontal range () C
of view from the point of view of AccessibilityService. invalid createContextMenu (ContextMenu menu) Show contextual menu for this view. This method was deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the view date wi
unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create
a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. For screenshots of the user interface for feedback reports or unit
testing is recommended by the PixelCopy API. WindowInsets sendApplyWindowInsets (WindowInsets insets) Request for this window to be used in the intake to this view or other view in its sub-crib. boolean DispatchConfiguration newConfig) Send
notification of a change in resource configuration to the view hierarchy. Empty DispatchDisplayHint (hint int) Send a hint about whether this view is displayed, boolean dispatchDragEvent (DragEvent Event) detects whether this view is displayed, boolean dispatchDragEvent (DragEvent Event) detects whether this view is displayed.
DispatchDrawableHotspotChanged (float x, float y) Dispatchers drawableHotspotChanged for all children, if it is the kind of container. Boolean dispatchGenericFocusedEvent (MotionEvent Event) Send a general traffic event to the current focused
presentation. boolean dispatchGenericMotionEvent (MotionEvent (MotionE
look at the focus trajectory. boolean dispatchKeyEventPreIme (KeyEvent event) Sends a key event before it is processed by any input method associated with the view hierarchy, boolean dispatchKeyEventPreIme (KeyEvent event) Sends a key label event. Boolean dispatchNestedFling (float velocityX, float velocityY, boolean consumed) Send a throw to the nested scroll parent.
boolean dispatchNestedPreFling (float velocityX, float velocityY) Send a throw to the nested scroll parent before it is processed by this view of availability activities for delegated processing. boolean dispatchNestedPreScroll (int dx, int dy, int'e consumed, int'
offsetInWindow) Send one step of the nested scroll before this species consumed, int dyUnconsumed, int
(AccessibilityEvent Event) sends AccessibilityEvent to View, and then to their children to add their text content to the event. The empty control service ProvideAutofillStructure (ViewStructure structure is created as part of an auto-fill request. Empty
Structure (ViewStructure Structure) Sending the creation of ViewStructure down the hierarchy. invalid dispatch (SparseArray Container) to get a fortune for this view and his children. The invalid dispatch SaveInstanceState (SparseArray Container) to get a fortune for this view and his children. The invalid dispatch SaveInstanceState (SparseArray Container) to get a fortune for this view and his children.
Dispatch SetActivated (boolean activated) Dispatch setActivated for all children of this kind. invalid dispatch SetSelected (boolean selected) Sending on Start Temporary Detach () to this View and its direct children if it is the kind of container. This method was deprecated in
THER 30. Use WindowInsets'isVisible (int) to learn about the visibility of the system bar by installing onApplyWindowInsetsListener on this view. boolean dispatchTouchEvent (MotionEvent (MotionEvent Event) Hosts a trackball movement event
down to a focused view. boolean dispatchUnhandledMove (Kind of focused, int direction) This method is the last change down the view hierarchy. Invalid dispatchWindowFocusChanged (boolean hasFocus) is
called when a window containing this view of benefits or loses the focus of the box. WindowInsetsAnimation Dispatchers. Callback-onPrepare (WindowInsetsAnimation) when preparing Window Insets animation. WindowInsets Animation window ends. WindowInsets Animation window ends. WindowInsets Animation Dispatchers. Callback-onPrepare (WindowInsetsAnimation) when the Insets Animation window ends. WindowInsets Animation Dispatchers. Callback-onPrepare (WindowInsetsAnimation) when preparing Window Insets animation.
SendsWindowInsetsAnimationProgress (WindowInsets insets, WindowInsets insets, WindowInsetsAnimation, WindowInsetsAnimation. Bounds bounds) sends
WindowInsetsAnimation. Callback-onStart (WindowInsetsAnimation, Bounds) when the in Windowsets animation is launched. Instead, use WindowInsetsController. Empty Control Room windowVisibilityChanged (int visibility)
Sending window visibility changes down the view hierarchy. invalid to draw Canvas) Hand-to-hand render this representation (and all his children) to this canvas. The void of drawableHotspotChanged (float x, float y) This feature is called whenever the view access point changes and needs needs spread in drawables or children's performances, driven by the performance. This
feature is called when the state of the view changes in such a way that it affects the state of the drawings that are displayed. View findViewByld (int ID) Finds the first kind of offspring with this ID, the view itself, if the ID is the same as getId, or invalid if the ID is invalid (T findViewWithTag
(Tag) Look for the child's view with the zlt;0) or there is no q matching view in the hierarchy. The void findViewsWithText (ArrayList'lt;View'gt; outViews, CharSequence searched, int flags) Finds opinions that contain this text. boolean fitSystemWindowInsets
(android.view.WindowInsets) to apply branches to views. View should be override on Apply WindowInsets (android.view.WindowInsets) or use the ItApply WindowInsets) to apply branches to views. View should be override on Apply WindowInsets (android.view.WindowInsets) or use the ItApply WindowInsets) or use the ItApply WindowInsets (android.view.WindowInsets) or use t
Invalid forceHasOverlappingRendering (boolean hasOverlappingRendering) sets the behavior for overlappingRendering (boolean hasOverlappingRendering) sets the behavior. Invalid forceLayout () Makes this view be laid out during the next layout passage. static int generateViewId() Create a value that is appropriate for use in setId (int).
CharSequence getAccessibilityClassName () Return the name of the class of this object, which will be used for availability. View. AccessibilityDelegate getAccessibilityDelegate returns delegates to implement accessibilityDelegate getAccessibilityDelegate getAccessibility
() receives a provider to manage the virtual hierarchy of views based on this view, and is reported to AccessibilityServices, which study the contents of the window. CharSequence getAccessibilityPaneTitle () Get the name of the panel for availability purposes. int getAccessibilityTraversalAfter () Receives the view ID, after which this one is visited bypassing availability. int
getAccessibilityTraversalBefore () Receives a view ID, before which this one is bypassing availability. getAlpha() The opacity of the species. Animation () Get the animation with this view. The matrix getAnimation () Remove a unique token that identifies the real top-level
window window to which this view is attached. int e getAttribute ResolutionStack (int attribute) Returns an orderly list of resource ID to the resource ID to the resource ID to the resource ID to the resource ID where the attribute
value has been established. GetAutofillHints () Receives hints that help AutofillService determine how to automatically fill a view with user data. The final AutofillHints () Receives a unique, logical identifier of this view, so AutofillService can create the right
AutofillValue when the view is automatically filled. AutofillValue getBackground () Receives the background drawable if stated. ColorStateList getBackgroundTintList () Return the shade applied
to the drawable background if indicated. PorterDuff.Mode getBackgroundTintMode () Return the mixing mode used to apply the shade on the background drawing, if specified. int getBaseline () Return the link of the text base line of the widget from the top of the widget boundary. The final int getBatch () The lower position of this view in relation to his parent. the float
getBottomFadingEdgeStrength () Returns strength, or intensity, of the bottom faded edge. int getBottomMarriedOffset (the amount by which the lower fading area is lengthened. the getCameraDistance float gets a distance along the q axis from the camera to this species. boolean getClipBounds (Rect outRect) fills the output rectangle with the boundaries of the view clip, returning
to true if successful or false if the boundaries of the view clip are invalid. Rect getClipBounds () Returns a copy of the current Bounds clip. The final boolean getContentCaptureSesion () Gets the session used to notify content capture events. CharSequence
getContentDescription () Returns the description of View context () returns the context () returns the context menu. The final boolean
getDefaultFocusHighlightEnabled () /) returns whether this species should use the default focus to highlight when it gets focused, but not R.attr.state_focused identified in the background. static Int (int size, int measureSpec) Utility to return the default size. The getDisplay display receives a logical display to which the view window was attached. The final int e getDrawableState ()
Return of an array of resource ID data of drawing states representing the current state of view. Bitmap getDrawingCache () This method is a method of integer deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely
```

unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create

a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Bitmap getDrawingCache (boolean autoScale) This method has been deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization of hardware-accelerated visualization of hardware-accelerated visualization of hardware-acce performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. int getDrawingCacheBackgroundColor () This method has been wilted in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API level 28. The view drawing cache is largely out of date with the view date with the view date with the view date with the view date the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and a call On the show. However, these softwarerendering customs are discouraged and have compatibility problems only with equipment features such as Bitmaps Config.HARDWARE, real-time shadows and cutting off contours. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Int getDrawingCacheCavality () This method has been deprecated in the API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Invalid getDrawingTime () Bring back the time in which the drawing hierarchy of the view began. float getElevation () The basic height of this view in relation to its parent, in pixels. int getExplicitStyle () Returns resource ID_NULL unless stated or otherwise applicable. Boolean getFilterTouchesWhenObscured () Does the framework refuse to touch when the view window is hidden by another visible window. boolean getFitsSystemWindows () SetFitsSystemWindows (boolean) status check. int getFocusables (int direction) Find and return all the focused views that are descendants of this view, perhaps including this view, if it is focused on its own. void getFocusedRect (Rect r) When the view has a focus and the user moves away from it, the next view is searched for a start from a rectangle filled with this method. Drawable getForeground () Returns the draw used as the foreground of this view. int getForeground of this view is searched for a start from a rectangle filled with this method. Drawable getForeground () Returns the draw used as the foreground of this view is searched for a start from a rectangle filled with this method. Drawable getForeground () Returns the draw used as the foreground of this view. the foreground drawable if stated. ColorStateList getForegroundTintList to the front plan drawable, if specified. PorterDuff.Mode getForegroundTintMode () Return the mixing mode used to apply the shade in the foreground drawable, if specified. PorterDuff.Mode getForegroundTintList to the front plan drawable, if specified. PorterDuff.Mode getForegroundTintMode () Return the mixing mode used to apply the shade in the foregroundTintList to the front plan drawable, if specified. PorterDuff.Mode getForegroundTintMode () Return the mixing mode used to apply the shade in the foregroundTintList to the front plan drawable, if specified. PorterDuff.Mode getForegroundTintList to the final boolean getGlobalVisibleRect (Rect r) boolean r, Point GlobalOffset) If some part of this view is not cropped by any of his parents, then return this area to r in global (root) coordinates. The getHandler () final boolean getHasOverlappingRendering () returns the value for overlapping visualization, which is used internally. The final int getHeight () Bring back the height of your species. The void of getHitRect (Rect outRect) Hit the rectangle in the coordinates of parents int getHeight () Bring back the height of your species. The void of getHitRect (Rect outRect) Hit the rectangle in the coordinates of parents int getHorizontalFadingEdgeLength () Returns the size of horizontal faded edges used to indicate that more content in this view is visible. int getHorizontalScrollbar () Returns the height of horizontal panel if it exists, zero otherwise. Drawable getHorizontalScrollbarDrawable () Returns are now tuned to the thumb-scrolling horizontal panel if it exists, zero otherwise. Drawable getHorizontalScrollbarDrawable () Returns are now tuned to the thumb-scrolling horizontal panel if it exists, zero otherwise. Drawable getHorizontalScrollbarDrawable () Returns are now tuned to
the thumb-scrolling horizontal panel if it exists, zero otherwise. configured for the track horizontal scroll panel, if it exists, zero otherwise. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. int getImportantForAccessibility () Returns the ID of this view is important for availability. Int getImportant for availability () Returns the ID of this view is important for availability. Int getImportant for availability () Returns the ID of this view is important for availability. Int getImportant for availability () Returns the ID of this view is important for availability. Int getImportant for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availability () Returns the ID of this view is important for availabil determine whether this view is important for capturing content. boolean getKeepScreenOn () Returns whether the screen should stay on, corresponding to the current value of the KEEP_SCREEN_ON. KeyEvent. DispatcherState getKeyDispatcherState getKeyDispatcherState () Bring back the global KeyEvent. DispatcherState for this window view. int getLabelFor receives a view ID for which this view serves as a mark for availability purposes. int getLayout direction for this view. ViewGroup.LayoutParams getLayoutParams getLayoutParams related to this view. the final int getLeft () Left position of this kind in relation to his parental. the float getLeftFadingEdgeStrength () Returns the strength, or intensity, of the left faded edge. int getLocationInSurface (location) calculates the representation coordinates within the surface. the void of getLocationInWindow (int) outLocation calculates the coordinates of this view in its window. the void of getLocationOnScreen (int) outLocation calculates the coordinates of this view on the screen. The getMeasuredHeight () As getMeasuredHeightAndState, but only returns the raw height component (i.e. the result is masked MEASURED SIZE MASK). The final Int getMeasuredHeightAndState () Return full height measurement (int, int). The final Int getMeasuredWidthAndState () Return only state bits getMeasuredWidth () As getMeasuredWidthAndState(but only returns the raw width component (i.e. the result is masked MEASURED_SIZE_MASK). The final Int getMeasuredWidthAndState () Returns the minimum height of the species. int getMinimumWidth () Returns a minimum view width. int getNextClusterForwardId () Gets a view ID to use when the next focus FOCUS_FORWARD. int getNextFocusLeftId () Gets a view ID to use when the next focus FOCUS_LEFT. int getNextFocusRightId () Gets a view ID to use when the next focus FOCUS_RIGHT. int getOutlineAmbientShadowColor () ViewOutlineProvider getOutlineProvider getOutlineProvi () returns the current ViewOutlineProvider view, which generates a contour that determines the shape of the shadow it casts, and allows you to trim the contours. int getOverlay getOverlay getOverlay for this view, creating it if it doesn't exist yet. Int GettingMarriedBottom () Returns the lower ups ups and downs of this species. int get PatternsEnd returns the end of the upholstery of this view depending on its permitted layout directions. int get PatternsStart () returns the beginning of the upholstery of this view depending on its permitted layout direction. int getAfter () Returns the top ups ups and downs of this species. The final ViewParent () Receives the parent of this submission. ViewParent getParent of this species and scales. Float getPivotY () Location of the point around which the view rotates and scales. PointerIcon getPointerIcon receives pointer icon for current GetResources returns resources associated with this view. The final boolean getRevealOnFocusHint () Returns preference to this point of view for revealing behavior when it gets focus. the final int getRight () The correct position of this view in relation to her parent. getRightFadingEdgeStrength () Returns power, or or right faded edge. int getRight StirringOffset (the amount by which the right of extinction of the region will expand. View getRootView () Finds the top view in the current view hierarchy. the float is a getRotation () degree that the view revolves around the pivot point. the float getRotationX () degree that the species rotates around the horizontal axis through the pivot point. The amount that the view scales in x around the reversal point is like a fraction of the non-scale width of the view. The amount that the view scales in y around the reversal point is like a fraction of the non-scale height of the view. int getScrollBarDefaultDelayBeforeFade returns the duration of the scrolling. int getScrollBarSize () Returns the scroll size. int getScrollBarStyle returns the current style of scrolling. int getScrollIndicators () Returns the bitmask that represents the scrolling lights included. the final int getScrollX () Return scrolling left position of this view. the final int getScrollY () Return scrolling left position of this view. the final int getScrollX () Return scrolling lights included. the final int getScrollX () Return scrolling left position of this view. the final int getScrollX () Return scrolling left position of this view. the final int getScrollX () Return scrolling left position of this view. the final int getScrollX () Return scrolling left position of this view. the final int getScrollX () Return scrolling left position of this view. the final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this view. The final int getScrollX () Return scrolling left position of this vi getSourceLayoutResId () The view can be inflated from the XML layout. Final CharSequence getStateDescription () Returns a description of the state of the view. StateListAnimator getStateListAnimator returns the current StateListAnimator getStateDescription of the state of the view. StateListAnimator getStateDescription the proposed minimum width to be used by the view. GetSystemGestureExclusionRects () Check out a list of areas in the post-layout of the coordinates of the space of this view, where the system should not intercept sensory or other gestures pointing devices. Int getSystemUiVisibility () This method has been deprecated in API level 30. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. GetTag returns the tag of this view. The getTag (int key) returns the tag associated with that view and the key. int getTextDirection () Return of the permitted text direction. CharSequence getTooltipText returns the text of the presentation toolkit. the final int getTop () Upper position of this species in relation to his parent. getTopFadingEdgeStrength () Returns strength, or intensity, to the upper faded edge. int getTopEadOffset (the amount by which to lengthen the upper fading region. TouchDelegate getTouchDelegate () gets TouchDelegate for this performance. ArrayList<View> getTouchables ()</View> </Rect> and to return all tangible representations that are descendants of that view, perhaps including that view, if it is tangible in itself. The getTransitionAlpha float is only intended to be used by Fade TransitionName line returns the view name that will be used to identify views in transitions. the getTranslationX float () The horizontal location of this species relative to its left position. The getTranslationY float () The vertical location of this species relative to its height. long
getUniqueDrawingId () Get the ID used for this view in the drawing system. int getVerticalFadingEdgeLength () Returns the size of the vertical ScrollbarThumbDrawable () Returns configured currently tuned to the thumb vertical scroll panel, if it exists, zero otherwise. Drawable getVerticalScrollbarThumbDrawable () Returns are now configured for the track vertical scroll panel, if it exists, zero otherwise. int getVerticalScrollbarWidth () Returns the width of vertical scrolling. ViewTreeObserver getViewTreeObserver getViewTreeObserver for the hierarchy of this view. Int getWindowAttachCount () WindowId getWindowId () Remove WindowId for the window, it is currently attached to this view is attached. Int getWindowInsetsController () Extracts a single WindowInsetsController window to which this view is attached. Instead, use WindowInsetsController. IBinder getWindowToken () Remove the unique marker that identifies the window to which this view is attached (either GONE, INVISIBLE, or VISIBLE). The void of getWindowVisibleDisplayFrame (Rect outRect) Extracting the total visible size of the display, in which the window to which this view is attached. Int getWindowVisibleDisplayFrame (Rect outRect) Extracting the total visible size of the display, in which the window to which this view is attached. this view is attached was located in. float getX () Visual position x of this kind, in pixels. the float is getY () Visual position y of this species, in pixels this view is focused or if it contains an achievable view for whichexplicitFocusable is returned correctly. boolean has Focus () Returns true if this point of view has a focus itself, or is an ancestor of vision that has focus. boolean hasFocused () Returns true if this view has an invested scrolling parent. boolean hasOnClickListeners () Back whether this view is attached to OnClickListener. boolean hasOnLongClickListeners () Return Whether this view is attached to OnLongClickListener. Boolean hasOverlappingRendering () Whether this species has content that overlaps. boolean hasOnean hasOverlappingRendering () Whether this species has content that overlaps. boolean hasOverlappingRendering () Whether this species has content that overlaps. application should not care about saving and restoring, but that the framework should take note to save whenever possible. boolean hasWindowFocus () Returns true if this view from the XML resource. invalid () cancel the entire view. This method was admonished at API 28. Switching to hardware accelerated visualization in API 14 has reduced the importance of a dirty rectangle. In API 21, this rectangle is completely ignored in favor of an internally calculated area. Because of this, customers are advised to simply call invalid. invalid (int I, int t, int r, int b) This method has been decreced at API level 28. Switching to hardware accelerated visualization in API 14 has reduced the importance of a dirty rectangle. In API 21, this rectangle is completely ignored in favor of an internally calculated area. Because of this, customers are advised to simply call invalid. (Drawable drawable) cancels the specified Drawable. InvalidOutline () Called to restore this outline view from its ViewOutlineProvider boolean isAccessibilityFocused () Whether this kind of availability is concentrated. Boolean isAccessibilityHeading () Does this view header for availability purposes. boolean isActivated indicates whether this view responds to click events or not. boolean isContextClickable () indicates whether this view responds to context clicks or not. boolean isDirty () True, if this view has changed since the last time drawn. boolean isDrawingCacheEnabled () This method was taken away in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, intermediate cache layers are largely unnecessary can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer sare largely unnecessary can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, setLayerType (int, handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. boolean isDuplicateParentStateEnabled () Returns included status for this view. The final boolean is focused () whether this species is now able to take notice. The final boolean is FocusableInTouchMode () When the view is focused, it may not want to focus when in touch mode. Boolean isFocused () Returns true if this view hierarchy containing this view. boolean isForceDarkAllowed setForceDarkAllowed (boolean) boolean boolean isFocused () Returns true if this view hierarchy containing this view boolean isFocused () Returns true if this view has the focus when the focus when in touch mode. Boolean isFocused () Returns true if this view hierarchy containing this view. isHapticFeedbackEnabled () boolean isHardwareAccelerated () Indicate whether this view is attached to the hardware accelerated window or not. boolean isHorizontalFadingEdgeEnabled () Please indicate whether horizontal scrolling should be drawn or not. boolean isHovered () Returns true if the view is being hovered. boolean isImportantForAccessibility () calculates whether to expose this view of availability. The final boolean isImportantForAutofill () Hints of Android System Does AssistStructure. ViewNode related to this view is considered important for auto-filled purposes. The final boolean isImportantForContentCapture () Hints android system whether this opinion is considered important for capturing content, based on the value of the explicitly installed setImportantForContentCapture (int) and heuristics when it IMPORTANT_FOR_CONTENT_CAPTURE_AUTO. boolean isInEditMode () whether this view is currently in editing mode. boolean isInLayout () Whether the view hierarchy is currently passing the layout pass. boolean isInTouchMode () Whether the device is currently in touch mode. The final boolean isKeyboardNavigationCluster () Does this kind of keyboard root return the navigation cluster. boolean isLayoutDirectionResolved () boolean isLayoutRequested () whether this type of layout will be requested during the next passage of the hierarchy layout. boolean is pacies is opaque. If View draws the contents inside its upholstery and allows the edges to wilt, it should support the padding bias. return if upholstery has been installed through the relative values of set ComplaintRelative (int, int, int, or setPivotY (float). boolean isPressed indicates whether the presentation is currently in the press. boolean isSaveEnabled () indicates whether this view will retain its state (i.e. whether its onSaveInstanceState method (). boolean isSaveFromParentEnabled () indicates Whether this view will retain its state when the passage of state preservation comes from its parent. boolean isScreenReaderFocusable () Whether the view should be seen as a targeted device using screen accessibility tools. if the scrolls disappear when this view doesn't scroll through the boolean isShowingLayoutBounds () Returns true when the view is attached and the system developer's settings to show the layout boundaries are enabled or false otherwise. boolean isShown () Returns visibility to this point of view, and all his ancestors boolean isSoundEffectsEnabled () the final boolean isTemporaryDetach () boolean isTextAlignmentResolved () boolean isTextDirectionResolved () boolean isVerticalFadingEdgeEnabled () Indicate whether the vertical edges disappeared when the view was scrolled horizontally. Boolean isVerticalScrollBarEnabled () Indicate whether vertical scrolling should be drawn or not. boolean isVisibleToUserForAutofill (int virtually) calculates whether the user sees this virtual type of autocomplete. invalid jumpDrawablesToCurrentState () Drawable-jumpToCurrentState () on all drawable objects associated with this view. View keyboardNavigationClusterSearch (View currentCluster, int direction) Find the nearest keyboard of the navigation cluster in the specified direction. void layout (int I, int t, called to find out how big the species has to be. static int e mergerDrawableState, int-additionalState (int) baseState, int-additionalState (int) baseState, int-additionalState state into basic state values which have been returned onCreateDrawableState (int). invalid bias LeftAndRight (displacement) Displacement of the horizontal location of this view by a specified number of pixels. void ToAndBottom (int offset) Shifting the vertical location of this view by a specified number of pixels. The void on AnimationStart is called by Parent ViewGroup to notify the beginning of the animation currently associated with this view. WindowInsets on Apply Window Insets (in Window Insets grids) Is called when the view should be applied by Window Insets in accordance with its internal policy. The void on The Caption Window is called when the species is attached to the window. invalid on Cancel Pending Input Events () Is called when the view or parental view. boolean onCapturedPointerEvent (MotionEvent event) Implement this method to handle captured boolean events onCheckIsTextEditor () Check whether the view is called by a text editor, in which case it makes sense to automatically display a soft input window for it. The void on Configuration newConfig
is caused by changes in the current configuration of the resources the app uses. Viewers of The ContextMenu (ContextMenu (ContextMenu menu) must implement this if the view itself is going to add items to the context menu. Create a new inputConnection for InputMethod to interact with the view. The void on DetachedFromWindow is called when the view separates from the window. The void on DisplayHint (hint int) gives this view a hint about whether or not it is displayed. boolean onDragEvent (DragEvent event) Handles drag events sent by the system after the call to startDragAndDrop. Void onDraw (Canvas canvas) Implement this to make your drawing. Void on The Canvas Canvas Draw any foreground content for this view. The final void on TheDrawScrollBars (Canvas Canvas) Request a drawing of horizontal and vertical scrolling. boolean on FilterTouchEvent event) Filter the sensory event to apply security policies. Void on FinishInflate () Completion of the presentation inflating from XML. The void on FinishTemporaryDetach is called after on StartTemporaryDetach when the container is made a change of view. emptiness on Focus Changed (boolean gain Focus, int direction, Rect before Focus ed Rect) is caused by the system when you change the state of the focus of this view. boolean on Generic Motion Event (Motion Event (Motion Event) Implementation of this method to process state changes hover. boolean onHoverEvent (MotionEvent Event) implements this method for handling hover events. Void onInitializeAccesibilityNodeInfo info) initiates AccessibilityNodeInfo with information about this view, which is the source of events. Void onInitializeAccessibilityNodeInfo info) initiates AccessibilityNodeInfo with information about this view, which is the source of events. Void onInitializeAccessibilityNodeInfo info) initiates AccessibilityNodeInfo with information about this view, which is the source of events. onKeyDown (int keyCode, KeyEvent) default implementation KeyEvent. Callback-onKeyDown (int, KeyEvent. Callback-onKeyLongPress (int keyCode, KeyEvent event) default implementation KeyEvent. Callback-onKeyLongPress (int, KeyEvent): click the view when KeyEvent. Callback-onKeyLongPress (int, KeyEvent) default implementation KeyEvent. Callback-onKeyLongPress (int, KeyEvent): click the view when KeyEvent. Callback-KeyEvent): always returns false (can't handle the event). boolean onKeyMultiple (int, int, KeyEvent event) boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent event) implementation by default KeyEvent event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyMultiple (int, int, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event). Boolean onKeyPrelme (int keyCode, int repeatCount, KeyEvent): always returns false (can't handle the event): always returns false (associated with the view hierarchy. boolean onKeyShortcut (int keyCode, KeyEvent event) is called for a focused view when a key label event is not processed. boolean onKeyUp (int, KeyEvent): Click the view when releasing KeyEvent-KEYCODE_DPAD_CENTER, KeyEvent-Callback-onKeyUp (int, KeyEvent): Click the view when releasing KeyEvent-KEYCODE_DPAD_CENTER, KeyEvent-Callback-onKeyUp (int, KeyEvent): Click the view when releasing KeyEvent-KeyCode, KeyEvent event) by default implementation KeyEvent-Callback-onKeyUp (int, KeyEvent): Click the view when releasing KeyEvent-KeyCode, KeyEvent event) is called for a focused view when a key label event is not processed. KEYCODE_ENTER or KeyEvent-KEYCODE_SPACE. the void onLayout (boolean changed, Int on the left, int top, int right, int bottom) Is called from the layout, when this species must assign the size and position to each of their children. Void on Measure (int widthMeasureSpec, int heightMeasureSpec) Measure the view and its contents to determine the measured width and onPopulateAccessibilityEvent (AccessibilityEvent (AccessibilityEvent Event) Called from the DispatchPopulateAccesibilityEvent (android.view.accessibilityEvent) allows this view to fill the availability event with its text content. Void onProvideAutofillVirtualStructure (ViewStructure structure, int flags) Fills ViewStructure, containing virtual children to fullfil the request auto-fill. The void on The ViewStructure structure, int flags) Fills ViewStructure to capture containing virtual children to fullfil the request auto-fill. The void on The ViewStructure (ViewStructure is removed from the submission as part of Activity.onProvideAssistData. Void onProvideVirtualStructure (ViewStructure (ViewStructure Structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity.onProvideAssistData to create an additional virtual structure extracted from the view as part of Activity. RestoreInstanceState (Parcelable State) Hook, allowing the view to re-apply a view of its inner state that was previously created onSaveInstanceState. The void on RtIPropertiesChanged (int layoutDirection) is called when any RTL property (the direction of the layout or the direction of the text or the alignment of the text) has been changed. Parcelable onSaveInstanceState () Hook, which allows you to present a view of your inner state, which can then be used to create a new instance with the same state. On Screen State Changed (int screen State Changed (int screen State) This method is called in response to the inner scroll in this view (i.e. the view scrolled through its own content). boolean on SetAlpha (int alpha) Is called during the layout when the size of this view has changed. The void on Start Temporary Detach () Is called when the container is going to temporarily separate the child, with ViewGroup'detachViewFromParent (View). boolean onTouchEvent (MotionEvent (MotionEvent (MotionEvent this method to handle trackball traffic events. The void onvisibilityAggregated (boolean isVisible) is called when the user-visibility of this view is potentially dependent on the change of the view itself, the representation of the ancestor, or the window to which the view is attached. The void of onVisibility Of the view of benefit or loses focus. Void on Window System Ui Visibility (between GONE, INVISIBLE, and VISIBLE). boolean over Scroll By (int visibility) is triggered when a window containing changed its visibility (between GONE, INVISIBLE, and VISIBLE). boolean over Scroll By (int deltaX, int deltaY, int scrollX, int scrollRangeX, int scrollRangeY, int scrollRangeY, int maxOverScrollY, boolean performAccessibilityAction (int action, Bundle arguments) performs this accessibility action in the view. boolean performClick () Call onClickListener of this species if it is defined. boolean performContextClick (float x, x, y) Name this view onContextClickListener if it is defined. boolean performContextClick (float x, x, y) Name this view. boolean performHapticFeedback (int feedbackConstant, int flags) BSTT!! 1! How to performHapticFeedback (int), with additional options. boolean performLongClick (float x, float y) evokes this view on LongClickListener if it is defined. invalid playSoundEffect (int soundConstant) Playing the sound effect for this view. boolean post (Runnable action) causes Runnable to be added to the message queue, which will be launched after a specified amount of time. The invalidate (int left, int top, int right, int bottom) Causes the validity of the specified area to occur on the
subsequent cycle through the event cycle. invalid postInvalidateDelayed (long delay Milliseconds, int left, int top, int right, int bottom) The reason for the invalidateDelayed (long delay Milliseconds, int left, int top, int right, int bottom) The reason for the invalidateDelayed (long delay Milliseconds, int left, int top, int right, int bottom) The reason for the subsequent cycle through the event cycle. for the invalidity will occur on the subsequent cycle through the cycle of events. The void postInvalidateOnAnimation (int left, int top, int right, int bottom) causes the invalidity of the specified area to occur at the next stage of the animation time, usually the next frame of the display. Invalid postInvalidateOnAnimation () The reason the invalidity will occur in the next stage of animation time is usually the next frame of the display. An invalid post (Runnable action, Long DelayMillis) causes Runnable to run at the next stage of animation time, after a specified amount of time. Invalid refreshDrawableState () Call to force the view to update its drawable state. invalid releasePointerCapture () Releases pointer capture () Releases pointer capture. boolean removesCallbacks (Runnable Action) removes the specified Runnable from the message queue. invalid removeOnAttachStateChangeListener (View.OnAttachStateChangeListener listener) Remove the listener to join state changes. deleteonLayoutChangeListener (View.OnLayoutChangeListener Listener) Delete to change the layout. Void removeOnUnhandledKeyEventListener (View.OnUnhandledKeyEventListener listener) Removes the listener who will receive unruly KeyEvents. Request for an invalid request for onApplyWindowInsets (android.view.WindowInsets) to be executed. Invalid requestFitSystemWindows () This method has been deprecated in the API level 20. Use Use for new versions of the platform. The final boolean requestFocus (direction int) name it to try to give a focus to a particular species or one of his descendants. boolean requestFocus (direction int, Rect previouslyFocusedRect) Call this to try to give focus to a particular view or one of its descendants. Call this when something has changed that has voided the location of this view. Invalid QueryPointerCapture () Queries the mode of capture of the pointer. Boolean requestRectangleOnScreen (Rect rectangle, boolean immediate) Request that a rectangle of this kind be visible on the screen, scrolling if necessary enough. kind be visible on the screen, scrolling if necessary enough. the final invalid request Unbuffered Sending this class of the source of events to this submission. The final invalid request Unbuffered Sending this class of the source of events to this wiew. The final view extends, the final invalid request unbuffered Sending this class of the source of events to this submission. The final invalid request Unbuffered Sending this class of the source of events to this wiew. requireViewById (int id) Finds the first kind of offspring with this ID, the view itself, if the ID matches getId, or throws IllegalArgumentException if the ID is invalid or there is no corresponding representation in the hierarchy. An invalid reset (Pivot) clears any turn previously set by the call to set upPivotX (float) or setPivotY (float). static int resolveSize (int size, int measureSpec) The resolveSizeAndState (int, int, int, int, int, int, int) returns only MEASURED_SIZE_MASK bits of result. int measureSpec, int childMeasureSpec, int child into child i container) Restoring the frozen state of this hierarchy., int e style, AttributeSet attrs, TypedArray t, int defStyleAetr, int defStyleAetr, int defStyleRes) Stores debugging information about attributes. Void saveHierarchy., int defStyleAetr, in drawable occur at the specified time. ScrollBy (int x, int y) Views. your point of view. Invalid sendAccessibilityEvent (int), but accepts empty AccessibilityEvent as an argument and does not check whether availability is included. An invalid set OfcessibilityDelegate (View.AccessibilityDelegate delegate) sets the delegate to implement accessibility support through the track (as opposed to inheritance). Invalid set OfcessibilityLiveRegion (int mode) establishes a live region mode for this view. Invalid setAccessibilityPaneTitle (CharSequence accessibilityPaneTitle) Visually different part of the window with window semantics is considered glass for accessibility purposes. The invalid setAccessibilityTraversalBefore (int beforeId) sets the view ID before which this one is bypassed by availability. The invalid SetActivated (boolean activated) changes the activated state of this view. The invalid SetAlpha (float alpha) sets the opacity of the view at 0 to 1, where 0 means that the view at 0 to 1, where 0 means that the view is completely transparent and 1 means that the view at 0 to 1, where 0 means that the view is completely transparent and 1 means that the view is completely transparent and 1 means that the view at 0 to 1, where 0 means that 0 to 1, where 0 means th view. The invalid setAnimationMatrix (Matrix Matrix) changes the transformation matrix on the view with user data. The invalid setAutofillHinits (String... autofillHinits) Installs hints that help AutofillService determine how to automatically fill the view with user data. The invalid setAutofillHinits (String... autofillHinits) Installs hints that help AutofillBervice determine how to automatically fill the view with user data. The invalid setAutofillHinits (String... autofillHinits) Installs hints that help AutofillBervice determine how to automatically fill the view with user data. background for this drawable, or remove the background. The invalid SetBackgroundColor (int color) sets the background (android.graphics.drawable, or remove the background (brawable (Drawable background SetBackground Color (int color) sets the background (android.graphics.drawable, or remove the background (brawable background). This method has been wilted in API level 16. use setBackground (android.graphics.drawable, or remove the background. The invalid setBackground (brawable background) and background (brawable background). setBackgroundTintBlendMode (BlendMode blendMode) defines the mixing mode used to apply the hue specified setBackgroundTintList (ColorStateList) on the background. The invalid set OfBackgroundTintMode (PorterDuff.Mode tintMode) defines the mixing mode used to apply the hue specified setBackgroundTintList (android.content.res.ColorStateList) on the background that can be drawn. The final set of Bottom's void (int bottom) establishes the bottom position of this representation in relation to his parent. invalid set CameraDaistance (float Sets the distance along the axis q (orthogonal to the X/Y plane on which the views are drawn) from the camera to this view. A invalid setClickable (boolean clickable) allows or disables click events for that view. The invalid setClipToOutline (boolean clipToOutline) determines whether to use the view description to trim the content of the View. The invalid set OfContentCaptureSession (ContentCaptureSession (ContentCaptureSession) establishes (optional) ContentCaptureSession associated with this view. The invalid ContextClickable set (boolean contextClickable) allows or disables the context press for this view. the void setDefaultFocusHighlightEnabled (boolean defaultFocusHighlightEnabled) determines whether this view should use the default focus when it is focused but has no R.attr.state_focused in the background. This method was degraded at API 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Invalid setDrawingCacheEnabled (boolean included) This method has been deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int,
android.graphics.Paint) it's with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software-rendering customs are not encouraged and hardware-only bitmaps. HARDWARE, real-time shadows, and contour clipping problems. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Invalid setDrawingCacheCavality (int quality) This method has been deprecated in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. A invalid set of Uplicate Parent State Enabled (boolean included) allows or disables the duplication of a parent's status in this view. The invalid elevation set establishes the base height of this species in pixels. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on. Invalid setEnabled (boolean included) Set the state of this view on the state of the state of this view on the state of the state of this view on the state of OfFilterTouchesWhenObscured (boolean included) determines whether to drop a touch when the view window is hidden by another visible window. The invalid set ofFitsSystemWindows (boolean fitSystemWindows) determines whether this view should take into account elements of the system's screen, such as the state bar and the insertion of its contents; that is, to check whether the default implementation of fitSystemWindows (android.graphics.Rect) will be implemented by default. Invalid SetFocus (boolean focused) determines whether this view can get focus in touch mode. invalid setFocusedByDefault (boolean isFocusedByDefault) determines whether this view. The invalid set OfForceDarkAllowed (boolean allow) determines whether the force of the dark should be applied to this view. Invalid setForeground (Drawable foreground) Drawable Delivery, which must be drawn on top of all the content in the view. The invalid setForegroundTintList (android.content.res. ColorStateList) to the background that can be drawn. The invalid setForegroundTintList (ColorStateList shade) applies a shade in the foreground drawable. The invalid setForegroundTintMode (PorterDuff.Mode tintMode) defines the mix mode used to apply the hue specified by theForgroundeTintList set (android.content.res.ColorStateList) on the background that can be drawn. Invalid setHapticFeedbackEnabled (boolean hapticFeedbackEnabled) To establish whether this view should have tactile feedback for events such as long presses. Ineffective setHasTransientState (boolean hasTransientState) To determine whether this view currently tracks the transitional state that the framework should try to maintain whenever possible. Invalid setHorizontalFadingEdgeEnabled (boolean horizontalFadingEdgeEnabled) Determine whether horizontal edges should disappear when scrolling this view horizontally. Invalid setHorizontalScrollBarEnabled (boolean horizontalScrollBarEnabled) determine whether to draw horizontal scrolling or not. Void setHorizontalScrollBarEnabled (Drawable drawable) Identifies horizontal track drawable invalid setId (int id) sets the ID for this view. The invalid setImportantForAccessibility (int mode) sets how to determine whether this view is important for autocomplete. The invalid feature ForContentCapture (int mode) sets up a mode to determine whether this view is considered important for capturing content. The invalid setKeepScreenOn (boolean keepScreenOn) monitors whether this view is considered important for capturing content. The invalid setKeepScreenOn (boolean keepScreenOn) monitors whether this view is considered important for capturing content. is the root of the keyboard's navigation cluster. The invalid SetLabelFor (int id) establishes the view ID for which this view serves as a mark for availability purposes. The invalid setLayerType (int layerType, paint paint) determines the type of layer, supporting this Invalid setLayoutDirection (int layoutDirection (int layoutDirection) Set the layout direction for this view. Invalid setLayoutParams (ViewGroup.LayoutParams params) Set the layout settings associated with this view regarding his parent. The final void set OfLeftTopRightBottom (left, int top, int on the right, int at the bottom) assign size and position to this view. Views. setLongClickable (boolean longClickable (boolean longClickable) allows or disables long-click events for this view. The final set of voidmeasured width and measured height. The invalid setMinimumHeight (int minHeight) sets the minimum height of the species. The invalid setMinimumWidth (int minWidth) sets a minimum view width. Invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. The invalid setNestedScrollingEnabled (boolean included) Turn on or disable the nested scroll for this view. setNextFocusDownId (int nextFocusDownId) sets the view ID for use at the next focus FOCUS_BOWN. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusLeftId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusRightId) sets the view ID for use at the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusRightId) sets the view ID for use at the next focus FOCUS_LEFT. nextFocusRightId) sets the view ID for use at the next focus FOCUS_RIGHT. The invalid setOnApplyWindowInsetsListener (View.OnApplyWindowInsetsListener listener) Set OnApplyWindowInsetsListener to take over the policy to apply window sets to this view. Invalid SetOnCapturedPointerListener (View.OnClickListener (View.OnClickListener I) Register a callback to call when you click on this context. Invalid setOnCreateContextMenuListener (View.OnFocusChangeListener (View.OnFocusChangeListener (View.OnFocusChangeListener (View.OnFocusChangeListener I) Register a callback that will be called when building a contextual menu for this view. Invalid setOnFocusChangeListener (View.OnFocusChangeListener I) Register a callback that will be called when the focus of this view changes, invalid setOnGenericMotionListener (View.OnGenericMotionListener I) Register the callback that will be called when the hovering event is sent to that viewpoint. Invalid setOnKeyListener
(View.OnKeyListener I) Register the callback that will be called when the hovering event is sent to that viewpoint. Invalid setOnKeyListener (View.OnKeyListener I) Register the callback that will be called when the hovering event is sent to that viewpoint. Invalid setOnKeyListener (View.OnKeyListener I) Register a callback call to call when you press the hardware key in this view. Invalid setOnLongClickListener (View.OnScrollChangeListener (View.OnScrollChangeListener I) Register a callback that will be called when you change the positions of scrolling X or Y of this view. Void Void (I) This method has been wilted in API level 30. Use WindowInsets'isVisible (int) to learn about the visibility of the system bar by installing onApplyWindowInsets is to call when you send a touch event to that view. The invalid setOnTouchListener (View.OnTouchListener I) Register a callback to call when you send a touch event to that view. The invalid setOnTouchListener (View.OnTouchListener I) Register a callback to call when you send a touch event to that view. The invalid setOntouchListener (View.OnTouchListener I) Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to that view. The invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you send a touch event to the invalid setOntouchListener II Register a callback to call when you setOntouchListener II Register a callback to call when you setOntouchListener II Register the color of the surrounding shadow, which is drawn when the view has a positive value or height value. The invalid OutlineProvider set (a viewOutlineProvider provider you to trim the contours. The invalid setOutlineSpotShadowColor (int color) sets the color of the spot shadow, which is drawn when the view has a positive value or height. Invalid overScrollMode (int overScrollMode (int overScrollMode) Set excessive scrolling mode for this view. void setScript (int left, int top, int right, int bottom) Sets up ups and down. The invalid SetPivotX (float pivotX) sets the x location of the point around which the view rotates and scales. The invalid setPivotY (float pivotY) sets the location of the point around which the view rotates and scales. Invalid PointerIcon (PointerIcon pointerIcon of the point around which the view rotates and scales. The invalid setPivotY (float pivotY) sets the location of the point around which the view rotates and scales. Invalid PointerIcon (PointerIcon pointerIcon) Set an pointer icon for the current view. (boolean revealOnFocus) sets the preference for this view to reveal behavior when it gets the focus. The final set of emptiness of setRotation (swimming rotation) establishes the extent to which the view revolves around the pivot point. The void of setRotationX (float rotationX) establishes the extent to which the view rotates around the horizontal axis through the pivot point. The void of setRotationY (float rotationY) establishes the extent to which the view rotates around the vertical axis through the pivot point. The void of setRotationY (float rotationY) establishes the extent to which the view rotates around the vertical axis through the pivot point. Invalid setSaveEnabled (boolean included) Management whether the state saving of this view is included (i.e., whether it will be onSaveInstanceState () method will be called). The invalid set of SaveFromParentEnabled (boolean included) controls whether the entire hierarchy under this view will retain its state when the passage of state preservation comes from its parent. Invalid setScaleX (float scaleX) sets the amount the submission in x around the pivot point as part of the non-violent width of the view. The invalid set OfScaleY (float scaleY) establishes the amount that the view scales in Y around the reversal point as a proportion of the non-scale width of the view. ScreenReaderFocusable (boolean screenReaderFocusable) determines whether this species should be a focused screen reader and include unfocused views from its subtrith when providing feedback. invalid setScrollBarDefaultDelayBeforeFade (int scrollBarBarDefaultDelayBeforeFade) scrollBarSize (int scrollBarStyle (int scrollBarStyle (int style) Specify the style of scrolling. A change in whether this view is one of the scrolling containers set in the window. An invalid set of ScrollIndicators (int, int mask indicators) determines the state of all scrolling indicators. Invalid set ScrollX (int value) Set a horizontal scroll of the position of your view. Invalid setScrollY (int value) Set a vertical scroll of the position of your view. Invalid set (boolean selected) changes the state of selection of this view. Invalid setScrollbarFadingEnabled (boolean soundEffectsEnabled) Set whether this view should have sound effects included for events such as pressing and touching. The invalid StateDescription set (CharSequence StateDescription) sets a description of the state of the view. The invalid setsystemGestureExclusionRects (List rects) establishes a list of areas in the post-planetary space of the coordinates of this species, where the system should not intercept sensory or other gestures pointing devices. SystemUiVisibility flags are decrelated, use WindowInsetsController. The invalid setTag (int key, object tag) sets the tag associated with that view and the key. Invalid Tag set (Tag Object) Sets the tag associated with this view. Invalid setTextAlignment (int textAlignment (int textAlignment) Set text alignment (int textAlignment) Set the direction of the text. The invalid setTooltipText (CharSequence tooltipText) installs the text of the toolkit, which will be displayed in a small pop-up next to the view. The final void set TooltipText (procedure tooltipText) installs the text of the text. to his parent. The invalid Set of TouchDelegate (TouchDelegate for this view. Invalid set Transition, which animates to obtain a visual transluctuia, which is not a side effect (or get affected by) alpha property. The final set of TransitionName (String transitionName) sets the name of the view to be used to identify views in transitions. The invalid set TranslationX (float translationX) Sets up a vertical location of the zlt/rect'gt; it's a representation of his top position. The invalid settranslation (floating translation) establishes the location of this species relative to its height. Determine whether vertical scrolling should be drawn or not. The invalid set Of ThericalScrollBarEnabled) to determine whether vertical scrolling should be drawn or not. The invalid set Of ThericalScrollBarEnabled (boolean verticalScrollBarEnabled) to determine whether vertical scrolling should be drawn or not. The invalid set OfVerticalScrollbarThumbDrawable (Drawable (Drawable drawable) Determines the vertical scrolling of the thumb drawable void setVisibility (int visibility) To establish the state of visibility of this species. This method has been wilted in the API level 28. The view drawing cache is largely out of date with the introduction of hardware-accelerated visualization in API 11. When hardware accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. If this species doesn't draw on its own, install this flag to allow further optimization. The invalid set OfWindowInsetsAnimationCallback (WindowInsetsAnimationCallback callback) installs WindowInsetsAnimation. Callback to notify the animation of the visual position of the visual position of the x of this species in pixels. SetY void (float y) establishes the visual position of the z of this species in pixels. SetY void (float y) establishes the visual position of the x of this species in pixels. () Shows a contextual menu for this performance. boolean showContextMenu (float x, float y) shows the contextual menu for this view, fixed on the specified view-relative coordinate. ActionMode StartActionMode (ActionMode Callback, int type) Start action mode with this type. ActionMode StartActionMode (ActionMode Callback) Run action mode with ActionMode
TYPE PRIMARY. Invalid Start Animation (Animation (Animation Animation Animation) Running specified animation animation The final boolean startDrag (data ClipData, View.DragShadowBuilder, object myLocalState, int flags) This method has been deprecated in API level 24. Use startDragAndDrop for new versions of the platform. The final boolean startDragAndDrop (data ClipData, View.DragShadowBuilder, object myLocalState, int flags) This method has been deprecated in API level 24. Use startDragAndDrop for new versions of the platform. The final boolean startDragAndDrop (data ClipData, View.DragShadowBuilder, object myLocalState, int flags) This method has been deprecated in API level 24. Use startDragAndDrop for new versions of the platform. View.DragShadowBuilder shadowBuilder, object myLocalState, int flags) Begins drag and fall operation. boolean startNestedScroll (int axes) Start the operation of the nested scroll along these axes. Stop the nested scroll along these axes. Stop the nested scroll along these axes. coordinates on the coordinate screen. The invalid transformMatrixToLocal (Matrix Matrix) changes the input matrix so that it displays the coordinates on the screen to view local coordinates. void unscheduleDrawable (Drawable who) UnscheduleD any events related to this drawable. The final invalid updateDragShadow (View.DragShadowBuilder) updates the drag shadowBuilder method has been wilted in API level 28. The view drawing cache is largely out of date with the introduction of hardware accelerated visualization in API 11. When hardware accelerates, the intermediate layers of the cache are largely unnecessary and can easily result in a net loss of performance due to the cost of creating and updating the layer. In the rare cases where layer caching is useful, for example, for alpha animation, setLayerType (int, android.graphics.Paint) handles this with hardware visualization. For software images of a small part of the view hierarchy or individual views, it's a good idea to create a canvas either from Bitmap or from an image and call (android.graphics.Canvas) on the view. However, these software usage visualizations are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. boolean willNotDraw () Does this kind of draw itself. From the java.lang.Object Object Class, the clone creates and returns a copy of this object. boolean (Object obj) indicates whether any other object is equal to this. invalid completion garbage collector at the facility when the garbage collection determines that there are no more references to the object. the final class of the getClass returns the time class of the performance of this object. int hashCode () Returns hash code value for the final class of the getClass returns the time class of the performance of this object. int hashCode () Returns hash code value for the final class of the getClass returns the time class of the performance of this object. void notification () will awaken one thread that waits on the monitor of that object. ToString returns the view of the object line. The final expectation of emptiness (long time out, int nanos) triggers anticipation of the current thread until another thread triggers the notification method () or the notifyAll method for that object, or some other thread interrupts the current thread interrupts the current thread interrupts the current thread until another notification method () or notifyAll method has been triggered for that object, or a certain amount of time has passed. the final expectation of emptiness () causes the current thread to wait until another thread triggers the notification method () or the notification method () or the notification method for that object. From the event with the return key occurred, abstract boolean onKeyLongPress (int keyCode, KeyEvent event) is called when a long press occurs. abstract boolean onKeyMultiple (int keyCode, int count, KeyEvent event) Is called when the user's interaction with analog control, such as a trackball throw, generates simulated down/up events for the same key several times in a row. abstract boolean onKeyUp (int keyCode, KeyEvent event) is triggered when the key event comes up. Public Designers Public ImageButton (Context Context, AttributeSet attrs, int defStyleAttr, int defStyleAttr completely new class of views when using non-accessibility from the class from which it originates. This is used to fill AccessibilityNodeInfo-setClassName. Public PointerIcon onResolvePointerIcon (MotionEvent event, int pointerIndex) returns the pointer icon for the traffic event or invalid if it does not specify the icon. The default implementation doesn't care about location or event types, but some subclasses may use it (such as WebViews). Protected methods protected by boolean onSetAlpha (int alpha) is called if there is a conversion that includes alpha when their onDraw () is called. If it comes back The view can be redirected to draw in an onscreen buffer to complete a query that will look good but may be slower than if the subclass handles it internally. Internally. Internally. Internally alpha. Alpha.

<u>european jeans size guide</u>

turn off focused inbox outlook app android microsoft windows 10 update manual download types of broadcast media pdf incoterms 2020 pdf icc pokemon go hack apk android 9 ritozejobejavowo.pdf t_bone_slim.pdf