Android horizontalscrollview scrollto not working

I'm not robot

reCAPTCHA

Continue

A layout container for a hierarchy of views that can be scrolled by the user, allowing it to be more than a physical display. HorizontalScrollView is a FrameLayout, meaning you have to put one child in it containing all the scrolling content; This child can be a layout manager with a complex hierarchy of objects. The child that is often used is LinearLayout in horizontal orientation, presenting a horizontal array of top-level items that the user can scroll through. The TextView class also takes care of its own scrolling, so it doesn't require HorizontalScrollView, but with two together you can achieve the effect of viewing text in a larger container. HorizontalScrollView only supports horizontal scrolling. Use ScrollView or ListView for vertical scrolling. android:fillViewport determines whether a scroll preview should stretch its contents to fill the viewport. From the android.view.view android:accessibilityHeading class, whether or not it's a headline for availability purposes. android:accessibilityLiveRegion tells the accessibility services 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:accessibilityTraversalBefore installs the view ID before 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:autofilledHighlight Drawable to be drawn over the look to mark it as autofilled can be a reference to another resource, in the form of package:type/name or theme attribute in the form ? package: type/name. android: a background that can be used as a background. android background:Tint for use in the background. Android backgroundTintMode Blending mode is used to apply background hue. android:clickable determines whether this view responds 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:drawingCacheCavality determines the quality of translucent drawing caches. android:duplicateParentState 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. Views. Determines whether to disappear from scrollbars 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 control, whether the view can focus. android:focusableInTouchMode Boolean, which monitors whether the view can be 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:foregroundGravity defines gravity for use in the foreground drawable. android:foregroundTint Tint for use in the foreground. Android:foregroundTintMode Blending mode is used to apply a foreground hue. android:hapticFeedbackEnabled Boolean, which monitors whether the species should have tactile feedback included for events such as long press. android:id Delivers the ID name for this view to later get it with View.findViewById () or Activity.findViewById (). 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. android:isScrollContainer Install this if the view will serve as a scroll container, meaning that it can be reused to reduce the overall window, so there will be room for the input method. android:KeepScreenOn monitors whether the view window should keep the screen visible. android:keyboardNavigationCluster Is this representation of the root of the keyboard's navigation cluster. android:layerType determines the type of layer by supporting this view. android:layoutDirection determines the direction of the layout drawing. android:longClickable determines whether this view responds to long-click events. android:minHeight determines the minimum height of the species. android:minWidth determines the minimum width of the view. android:nextClusterForward identifies the next keyboard navigation cluster. 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 hierarchy that is invisible, RuntimeException will result when the link is accessed. android:nextFocusForward identifies the following view to give focus when the next focus View.FOCUS_FORWARD If link link to a view that does not exist or is part of an invisible hierarchy, runtimeException will lead to access to the link. android:nextFocusLeft determines the following look to give focus when the next focus View.FOCUS_LEFT. android:nextFocusRight identifies the following view 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:outlineSpotShadowColor sets the color of the shade of the spot, which is drawn when the view 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, end edge; See R.attr.padding. android:paddingHorizontal installs ups ups and ups, in pixels, left and right edges; See R.attr.padding. android:paddingLeft installs ups ups, in pixels, left edge; See R.attr.padding. android:paddingRight 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:paddingTop installs ups and ups, in pixels, top edge; See R.attr.padding. android:Vert paddingical installs ups ups, in pixels, upper and lower edges; See R.attr.padding. android:requiresFadingEdge 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:saveEnabled If false, no state will be saved for this view when it is frozen. android:scaleX view scale in the direction of x. 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:scrollX Initial horizontal scroll displacement, in pixels. android:scrollY Initial vertical scroll displacement, in pixels. android:scrollbarAlwaysDrawHorizontalTrack determines whether a horizontal track should be drawn. android:scrollbarAlwaysDrawVerticalTrack determines whether a vertical scroll track should always be drawn. android:scrollbarDefaultDelayBeforeFade detects a delay in milliseconds that waits to scroll before disappearing. android:scrollbarFadeDuration 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:scrollbarStyle manages the style and position of scrolling. android:scrollbarThumbHorizontal defines horizontal thumb scroll drawable. android:scrollbarThumbVertical defines vertical scrolling of the thumb draw. android:scrollbarTrackHorizontal defines a horizontal scrolling track drawable. android:scrollbarTrackVertical determines vertical 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 text alignment. android:textPlimion determines the direction of text. android:theme defines the theme of override for presentation. android:tooltipText identifies text displayed in a small hover pop-up or long press. android:transformPivotX x the location of the pivot point around which the view will rotate and scale. android:transformPivotY y the location of the pivot point around which the view will rotate and scale. android:transitionName Names the species in such a way that it can be identified for Transitions. android:translationX translation in x vision. android:translationY translation in y view. android: translation translation into z kind. android:visibility controls the initial visibility of the species. From the class android.view.ViewGroup int CLIP_TO_PADDING_MASK We clip on ups ups and downs when FLAG_CLIP_TO_PADDING and FLAG_PADDING_NOT_NULL are installed at the same time. FOCUS_AFTER_DESCENDANTS this view will only be focused if none of its descendants want it. int FOCUS_BEFORE_DESCENDANTS This view will get focus in front of any of his descendants. int FOCUS_BLOCK_DESCENDANTS This species will block any of its descendants from getting attention, even if they are targeted. int LAYOUT_MODE_CLIP_BOUNDS This constant layoutMode. int LAYOUT_MODE_OPTICAL_BOUNDS This constant layoutMode. int PERSISTENT_ALL_CACHES This constant has been deprecated in API level 28. The view's drawing cache outdated with the introduction of hardware-accelerated visualization in API 11. With hardware acceleration, the intermediate layers of the cache are mostly mostly 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, View.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 View.draw (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 PERSISTENT_ANIMATION_CACHE This constant was deprecated in the api level of 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, View.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 View.draw (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 PERSISTENT_NO_CACHE 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, View.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 View.draw (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 bitmaps, real-time shadows, and clipping sketches. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Int Int This constant was figured out 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, View.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 View.draw (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. 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_DAY 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 the year the 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_EMAIL_ADDRESS a hint that this view can be automatically filled with an email address. Line AUTOFILL_HINT_NAME tip on what it is 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_PHONE a hint that this view can be automatically filled with a phone number. String AUTOFILL_HINT_POSTAL_ADDRESS AUTOFILL_HINT_POSTAL_ADDRESS indicating 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 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, which is filled int, representing the index element within the list (starting at 0). Int AUTOFILL_TYPE_NONE Autofill type for views that cannot be automatically filled. int AUTOFILL_TYPE_TEXT autocomplete type for text field that is filled with CharSequence. int AUTOFILL_TYPE_TOGGLE Autofill style for a togglable field, which is filled with boolean. int DRAG_FLAG_GLOBAL flag that drag can cross the boundary 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 resolution grant can be retained in all device reboots until Context.revokeUriPermission (Uri, int) revoke.ContextUriPermission. int DRAG_FLAG_GLOBAL_PREFIX_URI_PERMISSION When this flag is used with DRAG_FLAG_GLOBAL_URI_READ and/or DRAG_FLAG_GLOBAL_URI_WRITE, the URI permit applies to any URI that is prefix against the original URI granted. int DRAG_FLAG_GLOBAL_URI_READ When this flag is used with DRAG_FLAG_GLOBAL, the drag recipient will be able to request access to the contents of the URI (s) content in the ClipData facility. int DRAG_FLAG_GLOBAL_URI_WRITE When this flag is used with DRAG_FLAG_GLOBAL, the drag recipient will be able to request access to the URI content (s) 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 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 renderings Not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, in real time 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'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_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_CONTENT_DESCRIPTION Find views that describe the content. 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 views no matter if they are focused in touch mode. Int FOCUSABLES_TOUCH_MODE View flag showing addFocusables (java.util.ArrayList, Int, you should only add views focused in touch mode. int FOCUSABLE_AUTO This view determines focus automatically. int FOCUS_BACKWARD Use with focusSearch (int). Int FOCUS_DOWN Use with focusSearch (int). Int Int Use with focusSearch (int). int FOCUS_LEFT Use with focusSearch (int). int FOCUS_RIGHT Use with focusSearch (int). Int FOCUS_UP Use with focusSearch (int). Int GONE This view is invisible and it will take no place for the purpose of 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_AUTO automatically determines whether a view is important to availability. int IMPORTANT_FOR_ACCESSIBILITY_NO View is not important for availability. int IMPORTANT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS Kind is not important for accessibility, nor are any of its descendants of opinion. int IMPORTANT_FOR_ACCESSIBILITY_YES View is essential for accessibility. int IMPORTANT_FOR_AUTOFILL_AUTO automatically determine whether the view is important to the auto-fill. int IMPORTANT_FOR_AUTOFILL_NO The view is not important for auto-filled, but its 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 not be passed. Int IMPORTANT_FOR_AUTOFILL_YES The view is essential for automatic filling, and its children (if any) will be traversed. int IMPORTANT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS The view is important for the auto-filled, but its children (if any) will not be passed. int IMPORTANT_FOR_CONTENT_CAPTURE_AUTO automatically determines whether a 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. int LAYER_TYPE_SOFTWARE indicates that the view has a layer of software. int LAYOUT_DIRECTION_INHERIT horizontal direction of this view layout inherited from his parent. int LAYOUT_DIRECTION_LOCALE direction of the horizontal layout of this view is the output from the script default to the language. int LAYOUT_DIRECTION_LTR horizontal direction of the layout of this view from left to right. int LAYOUT_DIRECTION_RTL horizontal direction of the layout of this species from right to left. Int Int The bit shift MEASURED_STATE_MASK to get to the height of the bits for features that combine both width and height into one int, such as getMeasuredState () and childState argument resolveSizeAndState (int, int, int). int MEASURED_SIZE_MASK bits getMeasuredWidthAndState () and getMeasuredWidthAndState, which provide actual measured size. int MEASURED_STATE_MASK bits getMeasuredWidthAndState () and getMeasuredWidthAndState, which provide additional bits of state. int MEASURED_STATE_TOO_SMALL Bit of getMeasuredWidthAndState () and 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 to mark up a view that didn't have an 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 to twist 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_INSET the scroll style to show scrolling inside the soft area, increasing the ups upsize of the view. 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 ups upsize of the view. int SCROLLBARS_OUTSIDE_OVERLAY a scroll style to display scrolling on the edge of the view, without increasing the upholstery. int SCROLLBAR_POSITION_DEFAULT The Scroll Position bar in the default position defined by the system. int SCROLLBAR_POSITION_LEFT position bar scrolling along the left edge. int SCROLLBAR_POSITION_RIGHT position bar scrolling along the right edge. int SCROLL_AXIS_HORIZONTAL indicates scrolling along the horizontal 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 end edge of the view. int SCROLL_INDICATOR_LEFT 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 Scroll for the top edge of the view. int SOUND_EFFECTS_ENABLED View indicating whether this view should have sound effects included for events such as and touching. int STATUS_BAR_HIDDEN This constant has been deprecated in API level 15. Use SYSTEM_UI_FLAG_LOW_PROFILE instead. int STATUS_BAR_VISIBLE 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 THEO 30. Instead, use WindowInsetsController'hide (int) with Type'statusBars. int 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 This constant has been deprecated in API 30 level. 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 floating windows, 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, use LayoutParams-setFitInsetsTypes (int) with Type-navigationBars. For non-floating windows, filling the screen, call Window'SetDecorFitsSystemWindows (boolean) with false. int SYSTEM_UI_FLAG_LAYOUT_STABLE 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. Instead, use WindowInsetsController APPEARANCE_LIGHT_STATUS_BARS instead. int SYSTEM_UI_FLAG_LOW_PROFILE This constant has been deprecated in THE 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 level. Flags of the system UI layout are de-edist. int TEXT_ALIGNMENT_CENTER center item, for example, ALIGN_CENTER. int TEXT_ALIGNMENT_GRAVITY the default for a root view. int TEXT_ALIGNMENT_INHERIT align the default text. Int 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 ALIGN_RIGHT if the view is resolved layoutDirection is LTR, and 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 Text uses the first strong algorithm. 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 the text forces to LTR. Int TEXT_DIRECTION_RTL direction of the text forced 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 android.view.View public static final Property ALPHA A Property wrapper around alpha functionality processed by View'setAlpha (float) and View-getAlpha () EMPTY_STATE_SET Protected static final int' EMPTY_STATE_SET indicates that the view does not have a set of states ENABLED_FOCUSED_SELECTED_STATE_SET ENABLED_FOCUSED_SELECTED_WINDOW_FOCUSED_STATE_SET. focused, selected and his window has a focus. protected static final int ENABLED_FOCUSED_STATE_SET indicates that the view is on and has a focus ENABLED_SELECTED_WINDOW_FOCUSED_STATE_SET. protected static final in ENABLED_SELECTED_STATE_SET t. ENABLED_FOCUSED_WINDOW_FOCUSED_STATE_SET indicates that the view is on, focused and its window has a focus. chosen and his 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 int FOCUSED_SELECTED_STATE_SET indicates that the view is focused and selected. The 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. The protected static final FOCUSED_WINDOW_FOCUSED_STATE_SET indicates the focus of the view and the focus. Protected static final int-PRESSED_ENABLED_FOCUSED_SELECTED_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, selected and his window has a focus. protected static final int/View, FloatThe view is pressed, turned on and focused. Protected static final PRESSED_ENABLED_FOCUSED_WINDOW_FOCUSED_STATE_SET indicates clicking, turning, focusing, and focusing the window. a protected static final int PRESSED_ENABLED_SELECTED_STATE_SET points to the click, inclusion and selection of the view. 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 indicates the click and inclusion of the view. Protected static final PRESSED_ENABLED_WINDOW_FOCUSED_STATE_SET indicates the clicking, turning, and focusing of the window. a protected static final int-PRESSED_FOCUSED_SELECTED_STATE_SET points to the depression, focus and selection of the view. Protected static final int-PRESSED_FOCUSED_SELECTED_WINDOW_FOCUSED_STATE_SET indicates that the view is pressed, focused, selected and its window has a focus. protected static final int PRESSED_FOCUSED_STATE_SET points to the click and focus of the view. Protected static final PRESSED_FOCUSED_WINDOW_FOCUSED_STATE_SET indicates clicking, focusing the view, and focusing the window. protected static final int-PRESSED_SELECTED_STATE_SET points to the depression and the view selected. protected static final int PRESSED_SELECTED_WINDOW_FOCUSED_STATE_SET points to the click, the view is selected and its window has a focus. protected static final int PRESSED_STATE_SET indicates the view click. Protected static final int PRESSED_WINDOW_FOCUSED_STATE_SET points to the click of the window view and focus. Public static final wrapper properties OF THE PROPERTY around rotational functionality, processed by The View,'gt; View'setRotation (float) and View-getRotation () by methods of ROTATION_X. public ROTATION_X, float, float and grand final property wrapper ROTATION_Y the functionality of the rotationY, Processed by View-setRotationY (float) and View-getRotationY () SCALE_X SELECTED_STATE_SET SCALE_Y. The protected static final int's SELECTED_STATE_SET SELECTED_WINDOW_FOCUSED_STATE_SET indicates that the view is selected TRANSLATION_X. , Float&gt;and View'getTranslationX methods. Public View, Float Static final wrapping of the property TRANSLATION_Y translationY functionality, processed by the functionality of View-setTranslationY (float) and View'getTranslationY () by methods of view, float'WINDOW_FOCUSED_STATE_SET TRANSLATION_Z. public static final wrapper of Properties X A Property around the functionality of x x, processed by the lt'view, Float-gt; functionality of view-set-topx. public static final Property Y A Property is a wrapper around the functionality of y, processed by View,-setY (float) and View-getY () . , Int index) Adds a child's view. invalid addView (View child) adds a child's view. invalid addView (View child, ViewGroup.LayoutParams params) adds a child view with specified layout parameters. invalid addView (View child, int index, ViewGroup.LayoutParams params) adds a child view with specified layout parameters. boolean arrowScroll (direction int) Scroll pen in response to left or right arrow click. Invalid computeScroll () Is called by the parent to ask the child to update their values for mScrollX and mScrollY if necessary. boolean dispatchKeyEvent (KeyEvent Event) Send a key event to the next look at the focus trajectory. invalid to draw (canvas canvas) By hand to render this representation (and all his children) to this canvas. Boolean executeKeyEvent (KeyEvent Event) You can name this feature yourself so that the scrolling view scrolls from a key event, just as if the event was sent to it by the view hierarchy. Invalid throw (int velocityX) Throw the kind of scrolling boolean fullScroll (direction int) scroll pens in response to the home/end press label. CharSequence getAccessibilityClassName () Return the name of the class of this object, which will be used for availability. int getLeftEdgeEffectColor () Returns the left color of the edge effect. int getMaxScrollAmount () int getRightEdgeEffectColor () Returns the right color of the edge effect. boolean isFillViewport indicates whether the contents of this HorizontalScrollView are being stretched to fill the viewport. boolean ismoothScrollingEnabled () boolean onGenericMotionEvent (MotionEvent Event) Implement this method to handle common traffic events. boolean onInterceptTouchEvent (MotionEvent ev) Implement this method to intercept all touch screen traffic events. boolean onTouchEvent (MotionEvent ev) Implement this method to handle touch screen traffic events. boolean pageScroll (direction int) Processes scrolling in response to the page up/down press label. Void/View, Float/View, Float Float&gt; &lt;/View, Float&gt; Float&gt; Baby, View Focused) Is called when the child of this parent wants to focus the boolean requestChildRectangleOnScreen (Kind of Child, Rect rectangle, boolean immediately) Is called when a child from that group wants a particular rectangle to be located on the screen. The invalid request of The DisallowInterceptTouchEvent (boolean disallowIntercept) is called when a child does not want that parent and his ancestors to intercept sensory events with ViewGroup-onInterceptTouchEvent (MotionEvent). Call this when something has changed that has voided the location of this view. invalid scrollTo (int x, int y) Set a scrolling position of your view. This version also clips scroll to our child's borders. The void setEdgeEffectColor (int color) sets the color of the edge effect for both left and right edge effects. The invalid setFillViewport (boolean fillViewport) indicates this HorizontalScrollView, whether to stretch the width of the content to fill the viewport or not. The invalid setRightEdgeEffectColor (int color) sets the color of the right edge effect. Invalid setSmoothScrollingEnabled (boolean smoothScrollingEnabled) To set whether scrolling arrow will stifle his transition. boolean shouldDelayChildPressedState () The return is true if the pressed state should be delayed for children or descendants of this ViewGroup. The final void of smoothScrollBy (int dx, int dy) Like View'scrollBy, but scroll smoothly, not immediately. The final void of smoothScrollTo (int x, int y) As scrollTo (int, int), but scroll smoothly, not immediately. Int computeHorizontalScrollOffset () Calculate horizontal thumb displacement horizontal scrolling in horizontal range. The scrolling range of all his children is the total width of all his children. int computeScrollDeltaToGetChildRectOnScreen (Rect rect) calculates the amount to scroll towards X to get the rectangle completely on the screen (or if above the screen, at least the first screen size is a piece of it). the float getLeftFadingEdgeStrength () Returns the strength, or intensity, of the left faded edge. the getRightFadingEdgeStrength float returns the strength, or intensity, of the right faded edge. View child, int parentWidthMeasureSpec, int parentHeightMeasureSpec) Ask one of the children to measure themselves, taking into account both MeasureSpec's requirements for this view and its ups upsried. Void measureChildWithMargins (Kind of child, Int parentWidthMeasureSpec, int widthUsed, int parentHeightMeasureSpec, int heightUsed) Ask one of the children of this species to measure themselves, taking into account both the requirements of MeasureSpec for this species, and its upsizing and fields. emptiness onLayout (boolean int l, int t, int r, int b) Called out of the layout, when this species should assign the size and position to each of their children. Children. onMeasure (int widthMeasureSpec, int heightMeasureSpec) Measure the view and its contents to determine the measured width and measured height. The void onOverScrolled (int scrollX, int scrollY, boolean clampedX, boolean clampedY) is called overScrollBy (int, int, int, int, int, int, int, int, boolean) to respond to the results of the scroll operation. boolean onRequestFocusInDescendants (direction int, Rect previouslyFocusedRect) When searching for focus in children viewing scrolling, should be a little more careful not to pay attention to something that scrolls from the screen. void on The RestoreInstanceState (Parcelable State) Hook, allowing the view to re-apply a view of its inner state that was previously created onSaveInstanceState. 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. Emptiness onSizeChanged (int w, int h, int oldw, int oldh) This is called during the layout when the size of this view has changed. From android.widget.FrameLayout boolean checkLayoutParams (ViewGroup.LayoutParams p) FrameLayout.LayoutParams generateDefaultLayoutParams () returns a set of layout options ViewGroup.LayoutParams.MATCH_PARENT wide and ViewGroup.LayoutParams.MATCH_PARENT high. FrameLayout.LayoutParams generatesLayoutParams (AttributeSet attrs) returns a new set of layout settings based on a set of attributes. ViewGroup.LayoutParams generatesLayoutParams (ViewGroup.LayoutParams lp) Returns a secure set of layout settings based on the supplied layout parameters. CharSequence getAccessibilityClassName () Return the name of the class of this object, which will be used for availability. boolean getConsiderGoneChildrenCoghar () This method has been deprecated in API level 15. This method is deprecated in favor of getMeasureAllChildren, which has been renamed for consistency with setMeasureAllChildren. boolean getMeasureAllChildren () Determines whether all children are considered when measuring, or only those who are in a visible or INVISIBLE state. 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 measured height. The invalid setForegroundGravity (int foregroundGravity) describes how the foreground is located. The invalid setmeasureAllChildren (boolean measureAll) determines whether all children, or only those who are in a visible or INVISIBLE state, should be taken into account when measuring. boolean shouldDelayChildPressedState () Return if the pressed state should be deferred to the children or descendants of this ViewGroup. From the android.view.ViewGroup void addChildrenForAccessibility (ArrayList)lt;View) Adds children of this view who are relevant to access to this list as a conclusion. invalid addExtraDataToAccesibilityNodeInfo (AccessibilityNodeInfo info, String extraDataKey, Bundle arguments) adds additional data to AccessibilityNodeInfo based on an explicit request for additional data. ineffective addFocusables (Views, int direction, int focusableMode) adds any focused representations, which are descendants of this view (perhaps including this view, if it's focused) to views. invalid addKeyboardNavigationClusters (Collection of views, int direction) adds any roots of the keyboard navigation cluster that are descendants of this view (perhaps including this view, if it is the cluster root itself) to the views. Boolean addStatesFromChildren () Does ViewGroup data return to drawable states. Invalid addTouchables (ArrayList Views) Add any tangible views that are descendants of this view (perhaps including this view if it touches itself) to view. void addView (View child, ViewGroup.LayoutParams params) adds a child view with specified layout settings. , int index, ViewGroup.LayoutParams params) Adds the look of a child with specified layout parameters. invalid addView (View child) adds a child's view. invalid addView (Child View, int width, int height) adds a child's view with the default ViewGroup layout settings and the specified width and height. boolean addViewInLayout (See the child, Int Index, ViewGroup.LayoutParams params, boolean preventRequestLayout) adds a presentation during the layout. boolean addViewInLayout (See the child, int index, ViewGroup.LayoutParams params) adds a view during the layout. Invalid AttachmentsLayoutAnimationParameters (View child, ViewGroup.LayoutParams params, int index, int count) Subclasses should override this method to set the layout animation settings on the child supplied. invalid attachViewToParent (See the child, int index, ViewGroup.LayoutParams params) attaches the view to this group of views. Emptiness bring ChildToFront (Kind of Child) Changing z order of the child, so that on top of all the other children. boolean canAnimate indicates whether the group has the opportunity to revive their children after the first layout. boolean checkLayoutParams (ViewGroup.LayoutParams p) emptiness childDrawableStateChanged (See the child) If addStatesFromChildren () correctly, updates the drawing state of this group (include states from their children). invalid childHasTransientStateChanged (See child, boolean It is called when the child's view has changed regardless of whether or not the child is monitoring the transition. Invalid cleanupLayoutState (Child View) Prevents the child that will be laid out during the next skip of the layout. The emptiness of clearChildFocus (The View of the Child) Is called when the child of this parent surrenders the focus of the emptiness clearDisappearingChildren () Removes any waiting waiting for views that have been removed. The emptiness of clearFocus is called when this point of view wants to give up focus. The emptiness of the void separates AllViewsFromParent, disconnecting all views from the parent. The void separates the ViewFromParent (int index) that separates the performance from its parent. The void disconnects ViewFromParent (View child) separates the view from the parent. invalid detachViewsFromParent (int start, int count) separates the range of representations from its parents. 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 dispatchCapturedPointerEvent (MotionEvent Event) Go captured pointer event to focused presentation. Empty DispatchConfigurationChanged (Configuration 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 included and has a listener of drag events. empty sendingDraw (Canvas Canvas) Is called by drawing to draw the views of the child. Empty DispatchDrawableHotspotChanged (float x, float y) Dispatchers draw changes to hotspots to children's views that meet at least one of the following criteria: empty shipping FreezeSelfOnly (SparseArray container) Boolean dispatchGenericFocusedEvent (MotionEvent Event) Send a general traffic event to the current focused presentation. boolean dispatchGenericPointerEvent (MotionEvent Event) Send a general motion event to the view under the first pointer. boolean dispatchHoverEvent (MotionEvent Event) Sending event hover. boolean dispatchKeyEvent (KeyEvent Event) Send a key event to the next 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 dispatchKeyShortcutEvent (KeyEvent event) sends a key label event. The invalid PointerCaptureChanged (boolean hasCapture) void of the ControllerProvideAutofillStructure (ViewStructure Structure, int Flags) Sends the creation of ViewStructures for auto-fill purposes down the hierarchy when the Assist structure is created as part of an auto-fill request. This implementation adds a view group to all views of the childcare group, in addition to calling the default view implementation. empty control roomSidic structure ViewStructure) Sending the creation of ViewStructure down the hierarchy. Inefimable ByrestoreInstanceState (SparseArray Container) Children. state for this point of view and its children. Invalid dispatchSetActivated (boolean activated) Dispatch setActivized for all children of this view. invalid dispatchSetPressed (boolean clicked) Dispatch setPressed for all children of this presentation. invalid dispatchSetSelected (boolean selected) Sending setSelected for all children of this presentation. Invalid DispatcherSystemUiVisibility Is Undisplaced (not visible) This method is ampretated. Use WindowInsets'isVisible (int) to learn about the visibility of the system bar by installing onApplyWindowInsetsListener on this view. Invalid DispatchThawSelfOnly (SparseArray'lt;Parcelable) Send View.restoreHierarchyState (android.util.SparseArray) only on this point of view, not for your children. boolean dispatchTouchEvent (MotionEvent ev) Jump touch screen motion event to target view, or it's a representation if that's the goal. boolean dispatchTrackballEvent (MotionEvent event) Jump trackball movement event down to focused view. this method is the last chance for focused viewing and its ancestors. invalid dispatchVisibilityChanged (View changedView, Int Visibility) Sending the view visibility to change down the view hierarchy. The void of dispatchWindowFocusChanged (boolean hasFocus) is called when a window containing this view of benefits or loses the focus of the box. Emptiness dispatchWindwindInsetsAnimationEnd (WindowInsetsAnimation Animation) Dispatchers WindowInsets Animation. Callback-onEnd (WindowInsetsAnimation) when window insets animation ends. WindowInsetsAnimation animation sends WindowInsetsAnimation.Callback-onPrepare (WindowInsetsAnimation) when preparing WindowInsinsets animation. WindowInsets SendsWindowInsetsAnimationProgress (WindowInsets insets, WindowInsetsAnimation) Sends WindowInsetsAnimation.Callback-onProgress (WindowInsets, List) when Insets window animation makes progress. WindowInsetsAnimation.Bounds ControlWindowInsetsAnimationStart (WindowInsetsAnimation animation, WindowInsetsAnimation.Bounds bounds) sends WindowInsetsAnimation.Callback-onStart (WindowInsetsAnimation, Bounds) when the in Windowsets animation is launched. Invalid Control BoardWindowSystemUiVisiblity Displaced (not visible) This method is deprecated. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. Empty Control Room windowVisibilityChanged (int visibility) Sending window visibility changes down the view hierarchy. boolean drawChild (Canvas Canvas, Child View, Long DrawingTime) Draw one child of this view group. Void This function is called whenever the state of the view changes in such a way that it affects the state of the drawings that are displayed. The invalid end of ViewTransition (View View) This method should always be called after the previous call to the previous call. View findFocus () Find a view in the hierarchy is rooted in this view, which currently has a focus. invalid findViewsWithText (ArrayList'lt;View'gt; outViews, CharSequence text, Int Flags) finds opinions that contain this text. View FocusSearch (View focused, int direction) Find the closest view in the specified direction that wants to take focus. View (View v) tells the parent that a new focused species has become available. boolean collects ViewGroup.LayoutParams generateDefaultLayoutParams () Returns a set of default layout settings. ViewGroup.LayoutParams generatesLayoutParams (AttributeSet attrs) returns a new set of layout settings based on a set of attributes. ViewGroup.LayoutParams generatesLayoutParams (ViewGroup.LayoutParams p) returns a secure set of layout settings based on the layout parameters provided. CharSequence getAccessibilityClassName () Return the name of the class of this object, which will be used for availability. View getChildAt (int index) returns the view in the specified position in the group. int getChildCount returns the number of children in the group. int getChildDrawingOrder (int childCount, int drawingPosition) converts the design order position into the container position. the final int getChildDrawingOrder (int drawingPosition) converts the drawing order position into the container position. Static int getChildMeasureSpec (int spec, int padding, int childDimension) makes the hardest part of measuringChildren: figuring out MeruSpec pass to a particular child. boolean getChildStaticTransformation (Child View, Transformation t) Sets t to be a static transformation of the child if set, returning boolean to indicate whether a static transformation has been established. boolean getChildVisibleRect (View child, Rect r, offset point) Calculate the visible part of the rectangular area defined in terms of the child's terms... Boolean getClipChildren () Whether the children of this group are circumcised to their borders before drawing. boolean getClipToHabits () Whether this ViewGroup will clip your kids on its upholstery, and want (but not clip) any EdgeEffect in the soft area if padding is present. int getDescendantFocusability () Gets descendants of the focus of this group of views. See getFocusedChild () Returns of a focused child of this kind, if any. LayoutAnimationController getLayoutAnimation () Returns the mock animation controller used to animation the children of the group. Animation.AnimationListener getLayoutAnimationListener returns the animation listener to whom the animation events of the layout are sent. int getLayoutMode returns the alignment base during layout operations этой ViewGroup: либо LAYOUT_MODE_CLIP_BOUNDS, либо LAYOUT_MODE_OPTICAL_BOUNDS. LayoutTransition getLayoutTransition () получает объект LayoutTransition для этой ViewGroup. int getNestedScrollAxes () Вернуть текущие оси вложенных &lt;/View&gt; &lt;/View&gt; for this ViewGroup. ViewGroupOverlay getOverlay () Returns ViewGroupOverlay to this group of views, creating it if it doesn't exist yet. Int getPersistentDrawingCache () 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 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, View.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 View.draw (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 getTouchscreenBlocksFocus () Check whether this ViewGroup should ignore the focus requests for yourself and your children. boolean hasFocus () Returns is true if this view has or contains the focus of boolean hasTransientState () indicates whether the view is currently tracking the transitional state, that the application should not care about saving and restoring, but that framework should take note to save when possible. int indexOfChild (See the child) Returns position in the child's submission group. The final void is invalid (See the child, Rect dirty) This method is deprecated. Use onDescendantInvalidated (android.view.View, android.view.View) instead of watching updates to draw a state from offspring. ViewParent revokesChildInParent (location, rect dirty) This method is entodes. Use onDescendantInvalidated (android.view.View, android.view.View) instead of watching updates to draw a state from offspring. boolean isAlwaysDrawnWithCacheEnabled () This method has been deprecated in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's submissions may no longer have their caching behavior disabled by parents. boolean isAnimationCacheEnabled () This method was found in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's behavior with caching can be controlled with View'setLayerType (int, Paint). boolean isChildrenDrawingOrderEnabled () indicates whether ViewGroup draws its children in order, getChildDrawingOrder (int, int). boolean isChildrenDrawnWithCacheEnabled () This method was taken away in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's views can no longer be forced to cache rendering the state to their parents. Instead, use View'setLayerType (int, Paint) in individual views. Boolean isLayoutSuppressed () Whether mock calls on this container are being suppressed, due to an earlier call for a suppressLayout (boolean). Boolean isMotionEventSplittingEnabled () Returns true if MotionEvents sent to this ViewGroup can be divided into several children. boolean isTransitionGroup () Returns correctly if this ViewGroup should be considered a single entity to remove when performing an activity transition. invalid jumpDrawablesToCurrentState () Drawable-jumpToCurrentState () on all drawable objects associated with this view. the final invalid layout (int l, int t, int r, int b) Assign the size and position of the view and all its descendants This is the second phase of the layout mechanism. View child, int parentWidthMeasureSpec, int parentHeightMeasureSpec) Ask one of the children to measure themselves, taking into account both MeasureSpec's requirements for this view and its ups upsried. Void measureChildWithMargins (Kind of child, Int parentWidthMeasureSpec, int widthUsed, int parentHeightMeasureSpec, int heightUsed) Ask one of the children of this species to measure themselves, taking into account both the requirements of MeasureSpec for this species, and its upsizing and fields. Int widthMeasureSpec (int heightMeasureSpec) Ask all children of this species to measure themselves, taking into account both MeasureSpec's requirements for this view and its upsizing. Invalid notifySubtreeAccessibilityStateChanged (View Child, Source View, int changeType) Notifies the parent's view that the availability status of one of its descendants has changed and that the subtribune structure is different. The final void of displacementDesendtRectToMyCoords (View descendant, Rect rect) Displacement of the rectangle, which is located in the space coordinates of the offspring in our space coordinates. the final displacement of the voidRectIntoDescendantCoords (View of the descendant, Rect rect) Displacement of the rectangle, which is located in our space coordinates in the space of the coordinates of the ancestor. The void on The CaptionWindow is called when the species is attached to the window. Create a new drawable state for this view. The void on The Target View has been invalidated, or the drawing property has been altered, which requires a re-visualization of the hierarchy. If you override this method, you should call before the superclass is implemented. The void on DetachedFromWindow is called when the view separates from the window. boolean onInterceptHoverEvent (MotionEvent Event) Implement this method for intercepting hover events, than they are handled by children's representations. boolean onInterceptTouchEvent (MotionEvent ev) Implement this method to intercept all touch screen traffic events. abstract void onLayout (boolean changed, Int l, int t, t, r, int (b) Called from the layout, when this view should assign the size and position to each of their children. Boolean onNestedFling (View target, swim velocityX, swim speedY, boolean consumed) Request to throw out the nested scroll. boolean onNestedPreFling (View target, swim velocityX, swim velocityY) React to the nested throw before the target view consumes it. boolean onNestedPreFormAccesibilityAction (View goal, int action, Bundle args) Respond to availability actions delegated to the target representation of the offspring before the targeted processes of it. Subclasses should always call super.onNestedPrePerformAccessibilityAction void on NestedPreScroll (View target, int dx, int dy, int' consumed) React to the nested scroll in the process before the target view consumes a portion of the scroll. Void on NestedScroll (Goal View, int dxConsumed, int dyConsumed, int dxUnconsumed, int dyUnconsumed) React to the nested scroll in the process. Void on NestedSkroll (See Child, Target View, Int Axes) Respond to the successful requirement of an invested scrolling operation. boolean onRequestFocusInDescendants (direction int, Rect previously focusedRect) Look for a descendant to call View'requestFocus on. Boolean onRequestSendAccesibilityEvent (View child, AccessibilityEvent event) Is called when a child has requested accessibilityEvent and allows his parent to increase the event. PointerIcon onResolvePointerIcon (MotionEvent Event, Int pointerIndex) returns the pointer icon for a traffic event or zero if it does not specify an icon. boolean onStartNestedScroll (View Child, View Target, Int nestedScrollAxes) Respond to the presentation of a descendant initiating an nestable scroll operation, claiming an invested scroll operation if necessary. Empty on the Child's View react to the end of the attachment scroll operation. The void on The View Is Called When a New Child Is Added to This ViewGroup. The void on ViewRemoved (View Child) is called when a child's view is removed from this ViewGroup. invalid recomputeViewAttributes (See the child) Tell the hierarchy of the view that the attributes of global representation should be overestimated. Void removeAllViews () Call this method to remove all representations of children from ViewGroup. Void removeAllViewsInLayout () Called ViewGroup Subclass to remove the child's views from himself when he must first know his size on the screen before he can calculate how many views the child he will be providing. Void removeDetachedView (Child View, boolean animate) Completes the removal of a separate view. Invalid removeView Note: Don't refer to this method from View.draw View.onDraw (android.graphics.Canvas), dispatchDraw (android.graphics.Canvas) or any related method. invalid removeViewAt (int index) removes the view in the specified position in the group. Invalid removeViewInLayout (View View) Removes view during layout. void removeViews (int start, int count) removes the range range from the group. void removeViewsInLayout (int start, int count) removes a number of views during the layout. The invalid requestChildFocus (Kind of Child, View Focused) Is called when the child of this parent wants to focus the boolean requestChildRectangleOnScreen (View child, rect rectangle, boolean immediately) Is called when a child from that group wants a particular rectangle to be located on the screen. The invalid request of The DisallowInterceptTouchEvent (boolean disallowIntercept) is called when a child does not want that parent and his ancestors to intercept sensory events with ViewGroup-onInterceptTouchEvent (MotionEvent). boolean requestFocus (direction int, Rect previouslyFocusedRect) Call this to try to give focus to a particular view or one of its descendants and give it clues about the direction and a certain rectangle from which the focus emanates. Looking for a look to focus on respecting the parameter specified by getDescendantFocusability. boolean requestSendAccessibilityEvent (View Child, AccessibilityEvent event) Is asked by a child to ask from their parent to send AccessibilityEvent. An invalid TransparentRegion (View child) request is called when a child wants a hierarchy of views to collect and report transparent areas in a composite window. boolean restoreDefaultFocus () pays special attention to the default view in the view hierarchy, which has this view as the root. an invalid schedule of the LeaveoutAnimation schedule, which will be played after the next passage of the layout of this group of views. An invalid set ofAddStatesFromChildren (boolean addsStates) determines whether ViewGroup data includes in a drawable state. invalid setAlwaysDrawnWithCacheEnabled (bulean always) This method has been deprecated in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's submissions may no longer have their caching behavior disabled by parents. Invalid setAnimationCacheEnabled (boolean included) This method has been deprecated in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's behavior with caching can be controlled with View'setLayerType (int, Paint). Invalid setChildrenDrawingCacheEnabled (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, View.setLayerType (int, android.graphics.Paint) handles this 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 View.draw (android.graphics.Canvas) on the view. However, these software software and 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. The invalid setChildrenDrawingOrderEnabled (boolean included) tells ViewGroup whether to draw your children in a way defined by getChildDrawingOrder (int, int). Invalid setChildrenDrawnWithWithCacheEnabled (boolean included) This method has been deprecated in API level 23. In Build.VERSION_CODES. M, this property is ignored. Children's views can no longer be forced by parents to cache the rendering status. Instead, use View'setLayerType (int, Paint) in individual views. By default, children cut to their borders before drawing. Void setClipToHabits (boolean clipTo Habits) whether this ViewGroup will clip your kids on its upholstery and want (but not clip) any EdgeEffect in a soft area if padding is present. Invalid setDecendant focus (non-target focus) Set the focus of the offspring of this group of views. The invalid SetlayoutAnimation (LayoutAnimationController controller) installs the mock animation controller used to animation the group's children after the first layout. The invalid setLayoutAnimationListener (Animation.AnimationListen animationListener) identifies the animation listener to which the animation layout events should be sent. The invalid Set OfLayoutTransition installs the LayoutTransition object for this ViewGroup. Invalid setMotionEventSplittingEnabled (Bulean split) Turn on or off the splitting of MotionEvents on multiple children while sending a touch event. Invalid SetOnHierarchyChangeListener (ViewGroup.OnHierarchyChangeListener) Register a callback that will be called when a child is added or removed from this view. Invalid setPersistentDrawingCache (int drawingCacheToKeep) 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, View.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 the Image View.draw (android.graphics.Canvas) on view. However, these software rendering uses are not recommended and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, bitmaps, shadows, and outlines of the clipping. PixelCopy is recommended for user interface screenshots for feedback reports or unit testing. Invalid setStaticTransformationsEnabled (boolean included) When this property is tuned to reality, this ViewGroup supports static conversions on children; this leads to the possibility of callingStaticTransformation (android.view.View, android.view.animation.Transformation) when you are hired. Invalid set OfTouchscreenBlocksFocus (boolean touchscreenBlocksFocus) Set whether this ViewGroup should ignore focus requests for yourself and your children. The invalid set ofTransitionGroup (boolean isTransitionGroup) changes whether this ViewGroup should be considered as a single entity during activity transitions. The invalid set OfWindowInsetsAnimationCallback (WindowInsetsAnimation.Callback callback) installs WindowInsetsAnimation.Callback to notify the animation of the windows that are called into the kits. boolean shouldDelayChildPressedState () The return is true if the pressed state should be delayed for children or descendants of this ViewGroup. boolean showContextMenuForChild (View originalView, float x, float y) shows the contextual menu for the given view or its ancestors, fixed to the specified view-relative coordinate. boolean showContextMenuForChild (View originalView) Shows contextual menu for a given species or its ancestors. ActionMode startActionModeForChild (View originalView, ActionMode.Callback callback, int type) Run a certain type of action mode for a specified view. ActionMode StartActionModeForChild (View originalView, ActionMode.Callback callback) Start action mode for this view using ActionMode-TYPE_PRIMARY mode. Invalid StartLayoutAnimation () Runs a mock-up animation. This method informs ViewGroup that this View object, which should have this ViewGroup as its parent, should be stored around (re-displayed when ViewGroup draws its children), even if it is removed from the parent. The invalid suppressLayout (boolean suppress) informs this ViewGroup to suppress all mock-ups () calls until the layout suppression is disabled with a later call to suppressLayout (false). From the Android.view void of addChildrenForAccesibility (ArrayList outChildren) adds children of this View view relevant to access to this list as an outlet. The extraDataKey line, Bundle Arguments) adds additional data to AccessibilityNodeInfo based on an explicit request for additional data. emptiness addFocusables to add any targeted views that are descendants of this view (perhaps including this view, including the view, if it focuses itself) into the view. Views. it's a view (perhaps including this point of view, if it's focused) to views. invalid addKeyboardNavigationClusters (Views of the collection, int direction) adds any roots of the keyboard navigation cluster that are descendants of this view (perhaps including this species, if it is the cluster root itself) to the views. addOnAttachStateChangeListener (View.OnAttachStateChangeListener) Add the listener to join state changes. invalid addOnUnhandledKeyEventListener (View.OnUnhandledKeyEventListener listener) Adds a listener who will receive unruly KeyEvents. add voidTouchables (Species of ArrayList) Add any tangible views that are descendants of this species (perhaps including this species, if it is tangible) to the 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 contents of this view with a value. Invalid Autocomplete (SparseArray Values) Automatically fills the content of virtual children in this view. boolean awakenScrollBars (int startDelay, boolean invalid) Trigger scrollbars draw. boolean awakenScrollBars (int startDelay) Trigger scrolling draw. boolean awakenScrollBars () Trigger scrollbars draw. 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. The emptiness of buildDrawingCache () This method has been deprecated in the qlt;/AutofillValueLevel 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. 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 canResolveTextDirection () Check to see if you can make a resolution of the direction of 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. The final void cancels Out the current drag and fall operation. invalid cancels In anticipation of a long press. The final cancellation ofPendingInputEvents () Cancellation of any deferred high-level entry events that were previously placed in the event queue. boolean checkInputConnectionProxy (View view) is called InputMethodManager when a view that is not the current purpose of connection input tries to make a call to 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 degree of the horizontal thumb scrolling in the horizontal range. int computeHorizontalScrollOffset () Calculate horizontal thumb displacement scroll in the horizontal range. Int computeHorizontalScrollRange () Calculate the horizontal range that represents horizontal scrolling. Invalid computeScroll () Is called by the parent to ask the child to update their values for mScrollX and mScrollY if necessary. WindowInsets calculatesSystemWindowInsets (WindowInsets in, Rect outLocalInsets) Calculates the computers to use that view and those that should extend to those under it. Int computeVerticalScrollExtent () Calculate the vertical degree of the thumb vertical scrolling in the vertical range. Int computeVerticalScrollOffset () Calculate vertical thumb displacement by vertical scrolling in horizontal range. Int computeVerticalScrollRange () Calculate the vertical range that represents vertical scrolling. AccessibilityNodeInfo createAccessibilityNodeInfo returns AccessibilityNodeInfo, representing this point 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 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. 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 dispatchCapturedPointerEvent (MotionEvent Event) Go captured pointer event to focused presentation. Empty DispatchConfigurationChanged (Configuration 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 included and has a listener of drag events. empty sendingDraw (Canvas Canvas) Is called by drawing to draw the views of the child. Void DispatchDrawableHotspotChanged (float x, float y) Dispatchers drawableHotspotChanged for all children of this Species. Void dispatchFinishTemporaryDetach () Sending toFinishTemporaryDetach () to this View and its direct children, if it is the kind of container. boolean dispatchGenericFocusedEvent (Event Send a general traffic event to a current focused view. boolean dispatchGenericMotionEvent (MotionEvent Event) Sending a general event of motion. boolean dispatchGenericPointerEvent (MotionEvent Event) Sending General Traffic Movement to the view under the first pointer. boolean dispatchHoverEvent (MotionEvent Event) Sending event hover. boolean dispatchKeyEvent (KeyEvent Event) Send a key event to the next 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 dispatchKeyShortcutEvent (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. boolean dispatchNestedPrePerformAccesibilityAction (int action, Bundle arguments) Tell parents of 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 consumes any part of it. boolean dispatchNestedScroll (int dxConsumed, int dyConsumed, int dxUnconsumed, int dyUnconsumed, int' offsetInWindow) Sending one step nested scroll in the process. The invalid PointerCaptureChanged (boolean hasCapture) boolean dispatchPopulateAccessibilityEvent (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, int flags) sends the creation of ViewStructures for the purpose of automatically filling in the hierarchy when the Assist structure is created as part of an auto-fill request. Blank DispatchSSidal Structure (ViewStructure Structure) Sending the creation of ViewStructure down the hierarchy. the invalid control roomRestoreInstanceState (SparseArray'lt;Parcelable)gt; container is called restoreHierarchyState (android.util.SparseArray) to get a fortune for this species and his children. The invalid dispatchSaveInstanceState (SparseArray container) is called saveHierarchyState (android.util.SparseArray) to store the state for this species and its children. DispatchSetActivated (boolean activated) Dispatch setActivated for all children of this kind. invalid dispatchSetSelected (boolean selected) Sending setSelected for all children of this presentation. The emptiness of dispatchStartTemporaryDetach () Sending onStartTemporaryDetach () 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 on this view. boolean dispatchTouchEvent (MotionEvent Event) Jump touch screen traffic event up to qlt;/Parcelable or view if it is the target. boolean dispatchTrackballEvent (MotionEvent Event) Hosts a trackball movement event down to a focused view. boolean dispatchUnhandledMove (Kind of focused, int direction) This method is the last chance for a focused look and its ancestors to respond to the arrow key. invalid dispatchVisibilityChanged (View changed View, int visibility) Sending the view visibility of the 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-onEnd (WindowInsetsAnimation) when the Insets Animation window ends. WindowInsetsAnimation animation sends WindowInsetsAnimation.Callback-onPrepare (WindowInsetsAnimation) when preparing Window Insets animation. WindowInsets SendsWindowInsetsAnimationProgress (WindowInsets insets, WindowInsetsAnimation) Sends WindowInsetsAnimation.Callback-onProgress (WindowInsets, List) when Insets window animation makes progress. WindowInsetsAnimation.Bounds ControlWindowInsetsAnimationStart (WindowInsetsAnimation animation, WindowInsetsAnimation.Bounds bounds) sends WindowInsetsAnimation.Callback-onStart (WindowInsetsAnimation, Bounds) when the in Windowsets animation is launched. invalid dispatchWindowSystemUiVisiblity Displaced (not visible) This method has been deprecated in API level 30. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. Empty Control Room windowVisibilityChanged (int visibility) Sending window visibility changes down the view hierarchy. invalid to draw (canvas canvas) By hand to render this representation (and all his children) to this canvas. 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. 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 findFocus () Find a view in the hierarchy is rooted in this view, which currently has a focus. T findViewById (int ID) Finds the first kind of offspring with this ID, the very view if the ID is the same as getId(or invalid if the ID is invalid (T findViewWithTag (Tag) Look for the view of the child with a zlt; 0) &lt;T extends View&gt; данной тегом. пустота findViewsWithText (ArrayList&lt;View&gt; outViews, CharSequence искали, int флаги) Находит мнения, которые содержат данный текст. boolean fitSystemWindows (Rect insets) Этот метод был deprecated в API уровне 20. По api 20 используйте dispatchApplyWindowInsets (android.view.WindowInsets) для применения веток к представлениям. Представления должны переопределить&lt;/View&gt; &lt;/T extends View&gt; &lt;/T extends View&gt; &lt;/WindowInsetsAnimation&gt; &lt;/WindowInsetsAnimation&gt; or use the ItApplyWindowInsetsListener (android.view.View.OnApplyWindowInsetsListener) to process your own kits. View FocusSearch (direction int) Find the closest view in the specified direction that can take a focus. Invalid forceHasOverlappingRendering (boolean hasOverlappingRendering) sets the behavior for overlapping visualization for this view (see hasOverlappingRendering() for more information about this 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 accessibility support through the track. int getAccessibilityLiveRegion () Gets live region mode for this performance. AccessibilityNodeProvider getAccessibilityNodeProvider () 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. The getAnimation animation () Get animation is now associated with this view. The matrix getAnimationMatrix () Return of the current view conversion matrix. IBinder getApplicationWindowToken () Remove a unique token that identifies the real top-level window window to which this view is attached. int-getAttributeResolutionStack (int attribute) returns an orderly list of resource identifiers that are considered when allowing the attributes for this view. The integer's integer, integer'gt; getAttributeSourceResourceMap () Returns the display of the attribute resource ID to the resource source 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 AutofillId getAutofillId () Receives a unique, logical identifier of this view in activity, for the purposes of auto-fill. int getAutofillType () Describes the type of autocomplete of this view, so AutofillService can create the right AutofillValue when the view is automatically filled. AutofillValue Receives the current value of the auto-filled view. Drawable getBackground () Receives the background drawable BlendMode getBackgroundTintBlendMode () Return mix mode is used to apply hue on the background drawable if stated. ColorStateList getBackgroundTintList () Return the shade applied to the drawable background if indicated. Porterduff.Integer Mode, Integer Return the mixing mode used to apply the shade on the background to the 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 getBottom () 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 getClipToOutline () Returns whether outline should be used to clip the contents of the view. The final ContentCaptureSession getContentCaptureSesion () Gets the session used to notify content capture events. CharSequence getContentDescription () Returns the description of View content. The final context of getContext () returns the context in which the view works, through which it can access the current topic, resources, etc. ContextMenu.ContextMenuInfo getContextMenuInfo () Views should implement this if they have additional information to link to 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 getDefaultSize (size int, int measureSpec) Utility for return size by default. 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 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 visualizations are not recommended and have compatibility 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. Image image autoScale) 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 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 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 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 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 a call On the show. However, these software uses are discouraged and have compatibility issues with rendering 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. invalid getDrawingRect (Rect outRect) Return the visible boundaries of your view drawing. long 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 for style specified in style ... in the XML attributeSet backup element or 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 getFocusable () Returns focused settings for this view. ArrayList 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 getForegroundGravity () Describes how the foreground is located. BlendMode getForegroundTintBlendMode () Bring back the mixing mode used to apply the shade in the foreground drawable if stated. ColorStateList getForegroundTintList () Return the shade applied to the front plan drawable if specified. PorterDuff.Mode getForegroundTintMode () Return the mixing mode used to apply the shade in the foreground drawable if indicated. If any part of this species is not cropped by any of its parents, 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 getHorizontalFadingEdgeLength () Returns the size of horizontal faded edges used to indicate that more content in this view is visible. int getHorizontalScrollbarHeight () Returns the height of horizontal scrolling. Drawable getHorizontalScrollbarThumbDrawable () Returns tuned currently Drawable for the Big the horizontal scroll panel, if it exists, is zero otherwise. Drawable getHorizontalScrollbarDrawable () Returns are now configured for the track horizontal scroll panel, if it exists, zero otherwise. int getId () Returns the ID of this view. int &lt;/View&gt; &lt;/View&gt; Receives a mode to determine whether this view is important for availability. int getImportantForAutofill () Receives a mode to determine whether this view is important for autocomplete. int getImportantForContentCapture () Gets a mode to 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 () 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 getLayerType () indicates what type of layer is currently associated with this view. int getLayoutDirection () Returns the permitted layout direction for this view. ViewGroup.LayoutParams getLayoutParams () Get LayoutParams 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 getLeft FumblOffset (the amount to lengthen the left fading region. The final boolean getLocalVisibleRect (Rect r) void 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 getMatrix Matrix is the transformation of this view, which is calculated based on the current properties of rotation, scale and rotation. The final Int getMeasuredHeight () As getMeasuredHeightAndState, but only returns the raw height component (i.e. the result is masked MEASURED_SIZE_MASK). the final int getMeasuredHeightAndState () Return the full height measurement information for this view as calculated by the last call to measurement (int, int). The final Int getMeasuredState () Return only state bits getMeasuredWidthAndState () and getMeasuredHeightAndState (), combined with one integer. The final Int getMeasuredWidth () As getMeasuredWidthAndState(but only returns the raw width component (i.e. the result is masked MEASURED_SIZE_MASK). The final Int getMeasuredWidthAndState () Return the full width measurement information for this view as the last call for measurement (int, Int) is calculated. int getMinimumHeight () Returns the minimum height of the species. int getMinimumWidth () Returns a minimum view width. int getNextClusterForwardId () receives the root ID of the next keyboard navigation cluster. int getNextFocusDownId () Receives view ID use when the next focus FOCUS_DOWN. int getNextFocusForwardId () 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 Int Receives a view ID to use when the next focus FOCUS_RIGHT. int getNextFocusUpId () Gets a view ID to use when the next focus FOCUS_UP. View.OnFocusChangeListener getOnFocusChangeListener () Returns the focus-change call call recorded for this view. Int getOutlineAmbientShadowColor () ViewOutlineProvider getOutlineProvider () 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 getOutlineSpotShadowColor () int getOverScrollMode () returns excessive scrolling mode for this view. ViewOverlay getOverlay returns the overlay 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 getAfterLeft () Returns the left ups ups and downs of this species. int getAfterright () Returns the correct upie of this species. 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 ups and downs of this species. The final ViewParent getParent () Receives the parent of this submission. ViewParent getParentForAccessibility () Gets parental for affordability purposes. the getPivotX float () X is the location of the point around which the view rotates and scales. Float getPivotY () Location of the point around which the view rotates and scales. PointerIcon getPointerIcon receives a pointer icon for the current view. 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. the getRightFadingEdgeStrength float returns the strength, or intensity, of the 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. WindowInsets getRootWindowInsets () Provide the original windowInsets that are sent to the 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 float is getRotationY () the degree that the species rotates around the vertical 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. float getScaleY () Amount that the view in y around the pivot point as a fraction of the non-scale height of the view. int getScrollBarDefaultDelayBeforeFade returns the delay before the scrolls disappear. int getScrollBarFadeDuration () Returns the duration of the scrolling. Int GetScrollBarSize () Returns 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 top position of this view. Int getSolidColor () Redefin this if your species is known to always be drawn on top of a solid color background, and you need to reverse the withering edges. int 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 if it exists. int getSuggestedMinimumHeight () Returns the proposed minimum height to be used. int getSuggestedMinimumWidth () Returns 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 getTextAlignment (the return of the permitted text alignment. 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. the float 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'lt;View'gt; getTouchables () Find and return all tangible species that are descendants of this view, perhaps including this view, if it concerns itself. This property is only intended to be used by Fade Transition, which animates it to produce visual transparency that doesn't impact (or get affected by) real alpha ownership. The getTransitionName 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 upper position. Float () The depth of this species is 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 faded edges used to indicate that this view shows more content. int getVerticalScrollbarPosition () Drawable getVerticalScrollbarThumbDrawable () Returns tuned Drawable for a thumb vertical scroll strip if it exists, zero otherwise. Drawable getVerticalScrollbarDrawable () 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 returns ViewTreeObserver for the hierarchy of this view. int getVisibility () Returns visibility status for this view. Final Int getWidth () Bring back the width of your view. Int getWindowAttachCount () WindowId getWindowId () Remove WindowId for the window, it is currently attached to this view. WindowInsController getWindowInsetsController () Extracts a single WindowInsetsController window to which this view is attached. Int getWindowSystemUiVisibility () This method has been deprecated in api level 30. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. IBinder getWindowToken () Remove the unique marker that identifies the window to which this view is attached. int getWindowVisibility () Returns the current window visibility 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 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. 'The visual position of the z of this species, in pixels. boolean hasExplicitFocusable () Returns correctly if this view is focused or if it contains an achievable look for whichexplicitFocusable () returns true. boolean has Focus () Returns true if this point of view has a focus itself, or is an ancestor of vision that has focus. boolean hasFocusable () Returns true if this view is focused or if it contains an achievable look for which hasFocusable () returns true. boolean has NestedScrollingParent () 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 hasPointerCapture () Checks the status of the capture pointer. boolean hasTransientState () indicates whether the view is currently tracking the transitional state that the app should not care about saving and restoring, but that the framework should take note to whenever possible. boolean hasWindowFocus () Returns true if this view is in the window, which currently has a focus box. Static view inflate (context context, int resource, ViewGroup root) inflates the view from the XML resource. invalid () cancel the entire view. invalid (Rect dirty) This method has been deprecated in the API API 28. Switching to hardware accelerated visualization in API 14 reduces 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 l, 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 the state of activation of this view. boolean isAttachedToWindow () Returns true if this view is being attached to the window. boolean isClickable () 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, 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 isDuplicateParentStateEnabled () indicates whether this duplicates his drawable condition from his parent. boolean isEnabled () Returns included status for this view. The final boolean is focused () whether this species is now able to take notice. Final boolean When the view is focused, it may not want to focus when in touch mode. boolean isFocused () Returns true if this point of view has the focus of the final boolean isFocusedByDefault () whether this view should be focused when restoring focus to the hierarchy of views that contains that view. boolean isForceDarkAllowed setForceDarkAllowed (boolean) boolean isHapticFeedbackEnabled () boolean isHardwareAccelerated () indicates whether this view is attached to the hardware accelerated window or not. boolean isHorizontalFadingEdgeEnabled () Indicate whether horizontal edges are faded. Boolean isHorizontalScrollBarEnabled () 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 isLaidOut () Returns true if this view has been through at least one layout since it was last attached or separated from the window. boolean isLayoutDirectionResolved () boolean isLayoutRequested () Whether this type of layout will be requested during the next passage of the hierarchy layout. boolean isLongClickable () indicates whether this view reacts to long-click events or not. Boolean isNestedScrollingEnabled () Returns correctly if the embedded scroll is enabled for this view. boolean isOpaque () indicates whether this species 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, int) or through boolean isPivotSet () Whether or not the turn has been set by a call to installPivotX (float) 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 the entire hierarchy will retain under this his state when the passage of the state saving comes from his parent. boolean is ScreenReaderFocusable () Returns Returns view should be seen as a focused unit of screen reader accessibility tools. boolean isScrollContainer indicates whether this view is one of the scrolling containers set in the window. boolean isScrollbarFadingEnabled () Returns correctly if the scrolls disappear when this species does not scroll through the boolean isSelected() Indicates the state of selection of this view. The final 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 isTemporarilyDetached () Tells whether the species is in the state between onStartTemporaryDetach () and onFinishTemporaryDetach (). 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 virtualId) 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 l, int t, int r, int B) Assigning the size and position of the performance and all its descendants This is the second stage of the layout mechanism. The final measure of emptiness (int widthMeasureSpec, int heightMeasureSpec) It is called to find out how big the species should be. The 197th DrawableStates (int) baseState, int-additionalState) Merges its own state values in the additional State State to the base state of the base state, which has 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 AnimationEnd is called by parent ViewGroup to notify the end of the animation currently associated with this view. The void on AnimationStart is called by Parent ViewGroup to notify the beginning of the animation currently associated with this view. WindowInsets onApplyWindowInsets (in WindowInsets grids) Is called when the view should be applied by WindowInsets in accordance with its internal policy. emptiness onAttachedToWindow When the view is attached to the window. invalid on CancelPendingInputEvents () Is called as a result of a call to cancelPendingInputEvents () on this view or parental view. boolean onCapturedPointerEvent (MotionEvent Event) Implement this method for handling captured boolean pointer events onCheckIsTextEditor () Check Check the view is a text editor, and in this 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 menu) must implement this if the view itself is going to add items to the context menu. Create a new drawable state for this view. InputConnection onCreateInputConnection (EditorInfo outAtters) 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 onFilterTouchEventForSecurity (MotionEvent event) Filter the sensory event to apply security policies. Void onFinishInflate () Completion of the presentation inflating from XML. The void onFinishTemporaryDetach is called after onStartTemporaryDetach when the container is made a change of view. the void onFocusChanged (boolean gainFocus, int direction, Rect previouslyFocusedRect) is triggered by the view system when the state of the focus of this view changes. boolean onGenericMotionEvent (MotionEvent Event) Implement this method to handle common motion events. The emptiness on HoverChanged (boolean hovered) Implementation of this method to process state changes hover. boolean onHoverEvent (MotionEvent Event) implements this method for handling hover events. The void onInitializeAccesibilityEvent (AccessibilityEvent event) initiates AccessibilityEvent with information about this view, which is the source of events. Void onInitializeAccesibilityNodeInfo (AccessibilityNodeInfo info) initiates AccessibilityNodeInfo with information about this view. boolean onKeyDown (int keyCode, KeyEvent) default implementation KeyEvent.Callback-onKeyDown (int, KeyEvent): click the view when KeyEvent-KEYCODE_DPAD_CENTER or KeyEvent-KEYCODE_ENTER is released if the view is enabled and clickable. boolean onKeyLongPress (int keyCode, KeyEvent event) default implementation KeyEvent.Callback-onKeyLongPress (int, KeyEvent): always returns false (can't handle the event). Boolean (int keyCode, int repeatCount, KeyEvent event) default implementation KeyEvent.Callback-onMulKeytiple (int, int, KeyEvent): always returns false (can't handle the event). boolean onKeyPreIme (int keyCode, KeyEvent event) Process a key event before it is processed input method 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 keyCode, KeyEvent Event) by default implementation KeyEvent.Callback-onKeyUp (int, KeyEvent): Click the view when releasing KeyEvent-KEYCODE_DPAD_CENTER, KeyEvent-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 measured height. The void onOverScrolled (int scrollX, int scrollY, boolean clampedX, boolean clampedY) is called overScrollBy (int, int, int, int, int, int, int, int, boolean) to respond to the results of the scroll operation. The void onPointerCaptureChange (boolean hasCapture) is called when the window has just purchased or lost the grip of the pointer. The emptiness onPopulateAccessibilityEvent (AccessibilityEvent Event) Called from the DispatchPopulateAccesibilityEvent (android.view.accessibility.AccessibilityEvent) allows this view to fill the availability event with its text content. Void onProvideAutofillStructure (ViewStructure structure, int flags) Fills ViewStructure for full filling out the request. Void onProvideAutofillVirtualStructure (ViewStructure structure, int flags) Fills ViewStructure, containing virtual children to fullfil the request auto-fill. The void on The ViewStructure (ViewStructure structure, int flags) fills ViewStructure to capture content. The void on The ViewStructure is called when the aid structure is removed from the view as part of Activity.onProvideAssistData. The void on The ViewStructure is called when the aid structure is removed from the view as part of Activity.onProvideAssistData to create an additional virtual structure under this view. PointerIcon onResolvePointerIcon (MotionEvent Event, Int pointerIndex) returns the pointer icon for a traffic event or zero if it does not specify an icon. void on The RestoreInstanceState (Parcelable State) Hook, allowing the view to re-apply a view of its inner state that was previously created onSaveInstanceState. The void on RtlPropertiesChanged (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 with the same condition. OnScreenStateChanged (int screenState) This method is called whenever the state of the screen is attached to the changes. emptiness onScrollChanged (int l, int t, int oldl, oldl, Oldt) This is called in response to the internal scroll in this view (i.e. the view scrolled through its own content). boolean onSetAlpha (int alpha) Is called if there is a conversion that includes alpha. Emptiness onSizeChanged (int w, int h, int oldw, int oldh) This is called during the layout when the size of this view has changed. The void onStartTemporaryDetach () Is called when the container is going to temporarily separate the child, with ViewGroup'detachViewFromParent (View). boolean onTouchEvent (MotionEvent Event) Implement this method to handle touch screen traffic events. boolean onTrackballEvent (MotionEvent Event) Implement 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 onVisibilityChanged (View changed View, int visibility) is called when the visibility of the view or the ancestor of the view has changed. The void onWindowFocusChanged (boolean hasWindowFocus) is called when a window containing this view of benefit or loses focus. Void onWindowSystemUiVisibility Displaced (not visible) This method has been deprecated in API level 30. SystemUiVisibility flags are decrelated. Instead, use WindowInsetsController. Void onWindowVisibilityChanged (int visibility) is triggered when a window containing changed its visibility (between GONE, INVISIBLE, and VISIBLE). boolean overScrollBy (int deltaX, int deltaY, int scrollX, int scrollY, int scrollRangeX, int scrollRangeY, int maxOverScrollX, int maxOverScrollY, boolean isTouchEvent) Scroll the view with standard behavior for scrolling beyond the usual content boundaries. 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, float y) Call onContextClickListener of this species if it is defined. boolean performContextClick () Call onContextClickListener of this submission if it is determined. boolean performHapticFeedback (int feedbackConstant) BSTT!! 1! Provide tactile feedback to the user for 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 of OnLongClickListener if it is defined. boolean performLongClick () Calls this view onLongClickListener if it is defined. invalid playSoundEffect (int soundConstant) Playing the sound effect for this view. Boolean (Runnable action) causes Runnable to be added to the message queue. boolean postDelayed (Runnable action, long delayMillis) results in Runnable being added to the message queue, which will be launched after a specified amount of time. invalid postInvalidat () The reason the invalidity will happen subsequent cycle through a cycle of events. invalid postInvalidate (int left, int top, int right, int bottom) Causes the validity of the specified area to occur on the subsequent cycle through the cycle of events. invalid postInvalidateDelayed (long delay Milliseconds, int left, int top, int right, int bottom) The reason for the invalidity of the specified area to occur on the subsequent cycle through the event cycle. invalid postInvalidateDelayed (long delayMilliseconds) The reason 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) causes Runnable to run in the next stage of animation time. An invalid postOnAnimationDelayed (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. boolean removesCallbacks (Runnable Action) removes the specified Runnable from the message queue. invalid removeOnAttachStateChangeListener (View.OnAttachStateChangeListener listener) Remove the listener to join state changes. delete the listener 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 requestApplyInsets 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 to one of its descendants and give it a hint that the direction of the focus is moving. The final boolean requestFocus () Call to try to give 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 and give it clues about the direction and a certain rectangle from which the focus emanates. Final boolean requestFocusFromTouch () Call this to try to give to a particular species or one of his 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) Request that a rectangle of this kind be visible on the screen, screen, Enough is enough if necessary. boolean requestRectangleOnScreen (Rect rectangle, boolean immediate) Request that a rectangle of this kind be visible on the screen, scrolling if necessary enough. the final invalid requestUnbufferedDispatch (source int) Request unbuffered sending this class of the source of events to this submission. The final invalid request UnbufferedDispatch (MotionEvent Event) Request the non-taxable submission of this MotionEvents thread to this view. The final view extends,t 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) returns only MEASURED_SIZE_MASK bits of result. int measureSpec, int childMeasuredState) Utility to align the desired size and condition with restrictions imposed by MeasureSpec. boolean restoreDefaultFocus () Gives a focus on the default view in the view hierarchy, which has this view as root. void restoreHierarchyState (SparseArray container) Restoring the frozen state of this hierarchy. , int e style, AttributeSet attrs, TypedArray t, int defStyleAttr, int defStyleRes) Stores debugging information about attributes. SaveHierarchyState (SparseArray Container) Keep the frozen state of this view hierarchy in this container. Invalid 'lt;Parcelable' graphDrawable (Drawable who, Runnable what, long when) Action Schedule on drawable occur at a specified time. ScrollBy (int x, int y) Move scrolling position of your view. Set a scrolling position of your view. sendAccessibilityEvent (int eventType) sends this type of availability event. sendAccessibilityEventUnchecked (AccessibilityEvent) This method behaves exactly like sendAccessibilityEvent (int), but accepts as an argument the empty AccessibilityEvent and does not check whether availability is included. Delegate) Sets a delegate to implement accessibility support through a composition (as opposed to inheritance). Invalid setAccessibilityHeading (boolean isHeading) Set if the view is a headline for content section for availability purposes. semantics are considered glass for accessibility purposes. Invalid Accesstraveral After(int afterId)the view ID, after which this one is visited bypassing availability. An 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 is completely transparent and 1 means that the view is completely opaque. Invalid Set Animation (Animation Animation) Sets the next animation to play for this view. The invalid setAnimationMatrix (Matrix Matrix) changes the transformation matrix on the view. void set AutofillHinitz (String... autofillHints) Installs hints that help AutofillService determine how to automatically fill the view with user data. The invalid setAutofillId (AutofillId id) establishes a unique, logical identifier of this view in activity, for the purposes of auto-fill. Set a background for this drawable, or remove the background. The invalid SetBackgroundColor (int color) sets the background color for this view. Invalid setBackgroundDrawable (Drawable background) This method has been wilted in API level 16. use setBackground (android.graphics.drawable.Drawable) instead of the invalid setBackgroundResource (int. The invalid SetBackgroundTintBlendMode (BlendMode blendMode) defines the mixing mode used to apply the hue specified setBackgroundTintList (android.content.res.ColorStateList) on the background that can be drawn. The invalid set OfBackgroundTintList (ColorStateList hue) applies a hue to the drawable 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. The invalidcameraDistance set (floating distance) sets the distance along the q axis (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 set Of ClipBounds sets a rectangular area on this view, to which the view will be cropped when it is drawn. invalid setClipToOutline (boolean clipToOutline) determines whether to use the description of the view for a cut Views. The invalid set OfContentCaptureSession (ContentCaptureSesion contentCaptureSession) establishes (optional) ContentCaptureSession associated with this view. The invalid ContentDescription set (CharSequence contentDescription) sets the description of View content. The invalid ContextClickable set (boolean contextClickable) allows or disables the context press for this view. Void setDefaultFocusHighlightEnabled (boolean defaultFocusHighlightEnabled) Sets whether this view should use the default focus when it gets received but has no R.attr.state_focused in his 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) 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 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-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 It's a good idea to create a canvas from either Bitmap or from the image and call call To 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 ofUplicateParentStateEnabled (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 setFadingEdgeLength (length int) Set the size of the faded edge used to indicate that more content is available in this view. The invalid set 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) Set whether this view can get focus. invalid setFoited (not focused) determines whether this view can gain focus. Invalid setFocusableInTouchMode (boolean focusableInTouchMode) To determine if this view can get focus in touch mode. The invalid setFocusedByDefault (boolean isFocusedByDefault) determines whether this view should be given focus when restoring focus to the hierarchy of views containing 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 setForegroundGravity (int gravity) describes how the foreground is located. The invalid setForegroundTintBlendMode (BlendMode blendMode) defines the mixing mode used to apply the hue specified by theForgroundTintList set (android.content.res.ColorStateList) on 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 mixing mode used to apply the shade specified by the ForgroundTintList set (android.content.res.ColorStateList) » 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 (Galilee horizontalFadingEdgeEnabled) Determine whether horizontal edges should disappear when scrolling through this view Invalid setHorizontalScrollBarEnabled (boolean horizontalScrollBarEnabled) determine whether to draw horizontal scrolling or not. Void setHorizontalScrollbarThumbDrawable (Drawable drawable) Identifies horizontal thumb drawable void setHorizontalScrollTrackDrawable (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 availability, which is if it triggers availability events and if it is reported to the accessibility services that request the screen. Invalid Feature ForAutofill (Int mode) Sets a mode to determine whether this view is considered 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 the screen should stay on, changing the value of the KEEP_SCREEN_ON. Void setKeyboardNavigationCluster (boolean isCluster) Set whether this representation 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 Paint set updates the Paint object used with the current layer (only used if the current layer type is not set LAYER_TYPE_NONE). The invalid layer OfLayerType (int layerType, paint paint) determines the type of layer, supporting this view. Invalid setLayoutDirection (int layoutDirection) Set the layout direction for this view. Invalid setLayoutParams (ViewGroup.LayoutParams params) Set the layout settings associated with this view. The final set of emptinessLeft (left) establishes the left position of 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. setLongClickable (boolean longClickable) allows or disables long-click events for this view. The final set of voidmeasuredDimension (int measuredWidth, int measuredHeight) This method should be called onMeasure (int, int) to store measured 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 Views. The invalid setNextClusterForwardId (int nextClusterForwardId) sets the view ID for use as the root of the keyboard's next navigation cluster. The invalid setNextFocusDownId (int nextFocusDownId) sets the view ID for use at the next focus FOCUS_DOWN. The invalid setNextFocusForwardId (int nextFocusForwardId) sets the view ID for use at the next focus FOCUS_FORWARD. invalid setNextFocusLeftId (int nextFocusLeftId) Installs ID ID use when the next focus FOCUS_LEFT. The invalid setNextFocusRightId (int nextFocusRightId) sets the view ID for use at the next focus FOCUS_RIGHT. The invalid setNextFocusUpId (int nextFocusUpId) sets the view ID for use at the next focus FOCUS_UP. Invalid setOnApplyWindinsetsListener (View.OnApplyWindowInsetsListener listener) Set OnApplyWindowInsetsListener to take over the policy to apply window sets to this view. Invalid SetOnCapturedPointerListener (View.OnCapturedPointerListener l) Set the listener to receive callbacks when the view view capture state changes. Invalid setOnClickListener (View.OnClickListener l) Register a callback to call when you click on that view. Invalid setOnContextClickListener (View.OnContextClickListener l) Register a callback to call when you click on this context. Invalid setOnCreateContextMenuListener (View.OnCreateContextMenuListener l) Register a callback that will be called when building a contextual menu for this view. Invalid setOnDragListener (View.OnDragListener l) Register the drag event return call object for this view. Invalid setOnFocusChangeListener (View.OnFocusChangeListener l) Register a callback that will be called when the focus of this view changes. Invalid setOnGenericMotionListener (View.OnGenericMotionListener l) Register a callback call when you send a general motion event to that point of view. Invalid SetOnHoverListener (View.OnHoverListener l) Register the callback that will be called when the hovering event is sent to that viewpoint. Invalid setOnKeyListener (View.OnKeyListener l) Register a callback call to call when you press the hardware key in this view. Invalid setOnLongClickListener (View.OnLongClickListener l) Register a callback that will be called when you click and chronicle this view. Invalid setOnScrollChangeListener (View.OnScrollChangeListener l) Register a callback that will be called when you change the positions of scrolling X or Y of this view. Invalid setOnSystemUiVisibilityChangeListener (View.OnSystemUiVisibilityChangeListener l) This method was highlighted at 30 API. Use WindowInsets'isVisible (int) to learn about the visibility of the system bar by installing onApplyWindowInsetsListener on this view. Invalid setOnTouchListener (View.OnTouchListener l) Register a callback to call when you send a touch event to that view. The OutlineAmbientShadowColor (int color) set 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) sets the ViewOutlineProvider view, which generates a contour that determines the shape of the shadow it casts and allows 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) Set excessive scrolling mode for this view. void setMay (int left, int top, int on the right, right, Below) Sets up uping. emptiness setDiscriptive (int start, int top, int end, int bottom) places relative ups 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) Set an pointer icon for the current view. The invalid setpressed (boolean pushed) Sets a pressed state for this kind. The final void setRevealOnFocusHint (boolean revealOnFocus) sets the preference for this view to reveal behavior when it gets the focus. The final set of emptinessRight (right) establishes the correct position of this point of view in relation to her parent. The 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. 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. The invalid set OfscaleX (float scaleX) sets the amount that the view scales in x around the reversal point as part of the non-unsaltable 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 scrollBarDefaultDelayBeforeFade) Determine the delay before the scrolls disappear. Invalid setScrollBarFadeDuration (int scrollBarFadeDuration) To determine the duration of scrolling disappear. Invalid setScrollBarSize (int scrollBarSize) Determine the size of the scroll. Invalid setScrollBarStyle (int style) Specify the style of scrolling. A change in whether this view is one of the scrolling containers set in the window. set OfrollIndicators (int, int mask indicators) determines the state of the scroll indicators specified by the mask. An invalid set ofScrollIndicators (int indicators) determines the state of all scrolling indicators. Invalid setScrollX (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 setScrollbarFadingEnabled (boolean fadeScrollbars) Determine if scrolls will disappear when the view is not scrolling. Invalid Set Elected (boolean selected) Changes Change state of choice of this species. Invalid setSoundEffectsEnabled (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 setStateListAnimator (StateListAnimator stateListAnimator) attaches to this view provided by StateListAnimator. 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. Instead, 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) Set text alignment. invalid set OftDirection (int textDirection) 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 Top (int top) places the upper position of this view in relation to his parent. The invalid Set OfTouchDelegate (TouchDelegate delegate) sets TouchDelegate for this view. Invalid setTransitionAlpha (float alpha) This property is only intended to be used by The Fade transition, which animates it to produce a visual transluction, 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 setTransissay (visibility int) changes the visibility of this View without causing any other changes. The invalid set ofTranslationX (float translationX) establishes the horizontal location of this species relative to its left position. The invalid settranslationY (float translationY) establishes the vertical location of this species relative to its top position. The invalid settranslation (floating translation) establishes the location of the depth of this species relative to its height. Determine whether vertical edges should disappear when scrolling vertically. invalid setVerticalScrollBarEnabled (boolean verticalScrollBarEnabled) To determine whether it should be drawn scroll or not. The invalid set OfVerticalScrollbarThumbDrawable (Drawable drawable) Determines the vertical scrolling of the thumb drawable void setVerticalScrollTrackDrawable (Drawable drawable) Determines the vertical scrolling track drawable void setVisibility (int visibility) To establish the state of visibility of this species. This method has been wilted in the API level 28. View of the zlt'gt; The 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. If this species doesn't draw on its own, install this flag to allow further optimization. The invalid set OfWindowInsetsAnimationCallback (WindowInsetsAnimation.Callback callback) installs WindowInsetsAnimation.Callback to notify the animation of the windows that are called into the kits. The invalid setX (float x) establishes the visual position of the x of this species in the pixels. SetY void (float y) establishes the visual position of this species in pixels. The invalid set (float z) establishes the visual position of the z of this species in pixels. boolean showContextMenu () 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 callback, int type) Start mode with this type. ActionMode StartActionMode (ActionMode.Callback callback) Run action mode with ActionMode-TYPE_PRIMARY. Start the animation right now. The final boolean startDrag (data ClipData, View.DragShadowBuilder shadowBuilder, 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 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. ToString returns the view of the object line. ineffective transformMatrixToGlobal (Matrix Matrix) the input matrix is such that it displays the view-local 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. invalid unscheduleDrawable (Drawable who, Runnable what) Cancels Planned Planned To a draw. invalid unscheduleDrawable (Drawable who) Unscheduled any events related to this drawable. The final invalid updateDragShadow (View.DragShadowBuilder shadowBuilder) updates the drag shadow for the current drag and fall operation. boolean verifyDrawable (Drawable who) If the submission subclass displays its own Drawable objects, it should override this feature and return the faithful to any Drawable it displays. boolean willNotCacheDrawing () 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 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 () Is called by the 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 subject. int hashCode () Returns the hash code value to the object. the final invalid to notify () will wake up one thread that is waiting on the monitor of this 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, or a certain amount of real time has passed. The final expectation of emptiness (long time) triggers the wait for 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 expectation of the current flow up to until another thread triggers a notification method or notifyAll method for that object. From the interface android.view.ViewParent abstract emptiness emptiness Baby) Change the z order of the baby, so it's on top of all the other kids. abstract boolean canResolveLayoutDirection () Reports whether this submission parent can decide the direction of the layout. abstract boolean canResolveTextAlignment () Reports whether this submission parent can allow text alignment. abstract boolean canResolveTextDirection () Reports whether the parent of this submission can decide the direction of the text. This method is called upon by parents when the child's drawable condition has changed. The abstract invalid childHasTransientStateChanged (View child, boolean hasTransientState) Is called when the child's representation now has or no longer tracks the transitional state. The abstract void of clearChildFocus (Kind of Child) Is called when the child of this parent gives away the focus of the abstract emptiness of createContextMenu (ContextMenu menu) there is a parent to fill out the specified context menu if he has something to add (and then repeated to his parents). The abstract view focusSearch (View v, int direction) Find the closest view in the specified direction, which wants to take the focus of the abstract void focusableViewAvailable (View v) Informs the parent that a new focused view has become available. abstract boolean getChildVisibleRect (Child View, Rect r, Offset Point) Calculate the visible part of the rectangular area defined in terms of the child's terms... abstract int getLayoutDirection () Bring back this direction of the parent view layout. Abstract ViewParent getParent () Returns parent if it exists, or zero. Abstract ViewParent getParentForAccesibility () Gets parental this view for accessibility. abstract int getTextAlignment () The return of this parental representation text alignment. abstract int getTextDirection () Return of this parent direction of the presentation text. Abstract void invalidChild (View child, Rect r) This method was mutilated in API level 26. Instead, use onDescendantInvalidated (android.view.view, android.view.View). abstract ViewParent annulsChildInParent (location int, Rect r) This method has been at THE 26 API level. Instead, use onDescendantInvalidated (android.view.view, android.view.View). abstract boolean isLayoutDirectionResolved () Reports if this direction of the parent view layout will be resolved. abstract boolean isLayoutRequested indicates whether a mock-up on this parent view has been requested. abstract boolean isTextAlignmentResolved () Reports if this representation of the parent text alignment will be resolved. abstract boolean isTextDirectionResolved () Reports if this direction the submission text will be resolved. ViewNavigationClusterSearch (View currentCluster, int direction) Find the nearest keyboard navigation cluster in that direction. Abstract Void NotifiesSubtreeAccessibilityStateChanged (View Child, View Source, int changeType) Notifies the parent's view that the availability status of one of its offspring has changed and that the subtrith structure is different. default emptiness emptiness the target view has been invalidated or changed the drawing property, which requires a re-visualization of the hierarchy. Abstract boolean onNestedFling (View target, swim velocityX, swim speedY, boolean consumed) Request to throw out nested scroll. Abstract boolean onNestedPreFling (View target, swim velocityX, swim speedY) React to the nested throw before the target view consumes it. Abstract boolean onNestedPrePerformAccessibilityAction (View purpose, int-action, bundle arguments) Respond to availability action delegated by the target representation of the offspring before the target processes. Abstract void on NestedPreScroll (View target, int dx, int dy, int's consumed) React to the nested scroll in the process before the target species consumes part of the scroll. abstract void on NestedScroll (View purpose, int dxConsumed, int dyConsumed, int dxUnconsumed, int dyUnconsumed) React to the nested scroll in the process. Abstract Void on NestedScrollAccepted (View Child, View Target, Int nestedScrollAxes) React to the successful approval of the nested scroll operation. Abstract boolean onStartNestedScroll (View Child, Target View, Int nestedScrollAxes) React to the posterion by initiating the operation of the nested scroll, claiming that the operation is nested scroll if necessary. Abstract void on StopNestedScroll (View Target) React to the end of the attachment scroll operation. Abstract void recomputeViewAttributes (See the child) Tell the hierarchy of the view that the attributes of global representation should be overestimated. An abstract invalid request (View Child, View Focused) Is called when that parent's child wants to focus an abstract boolean request (See the child, rect rectangle, boolean immediately) is called when a child from that group wants a particular rectangle to be located on the screen. The abstract invalid requestdisallowInterceptTouchEvent (boolean disallowIntercept) is called when the child does not want this parent and his ancestors to intercept sensory events with ViewGroup-onInterceptTouchEvent (MotionEvent). Ask for a new View'fitSystemWindows (Rect) to be executed. An abstract invalid request for Leaveout is called when something has changed that has voided the child's location of this parenting view. Abstract boolean requestSendAccessibilityEvent (Child View, AccessibilityEvent Event) Is called by the child to ask his parent to send AccessibilityEvent. Abstract Invalid RequestTransparentRegion (View Child) Called When a Child Wants a Hierarchy collected and reported transparent areas of the window composition. abstract boolean showContextMenuForChild (View originalView) Shows contextual menu for the specified species or its ancestors. abstract boolean showContextMenuForChild (View originalView, float x, float y) shows the context menu for the particular species or its ancestors, fixed to the specified coordinate of the relative view. relative representation. ActionMode startActionModeForChild (View originalView, ActionMode.Callback callback, int type) Run a certain type of action mode for a specified view. Abstract ActionMode StartActionModeForChild (View originalView, ActionMode.Callback callback) Start action mode for a specified view with actionMode-TYPE_PRIMARY type. From the interface android.view.KeyEvent.Callback abstract boolean onKeyDown (int keyCode, KeyEvent event) is called when 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. Determines whether a scroll view should stretch the content to fill the viewport. Can be a boolean value such as true or false. Related Methods: Public Designers Public HorizontalScrollView (Context Context) Settings Context Public HorizontalScrollView (Context Context, AttributeSet attrs, int defStyleAttr) Settings context context attrs AttributeSet defStyleAttr int public HorizontalScRollView (Context, AttributeSet attrs, int defStyleAttr, int defStyleRes) Context Context Settings attrs AttributeSet defStyleAttr int defStyleRes int Public Methods Directions int: Direction int Direction Directions Direction Int Direction Directions Direction Int: Direction Appropriate to the Arrow Key that was pressed by The Returns boolean True if we consumed an event that falsely public voids computeScroll () Called by the parent to ask the parent to update their values for mScroll. This is usually done if the child animates the scroll with the Scroller object. Public boolean dispatchKeyEvent (KeyEvent event) Sending a key event to the next look at the focus path. This path runs from the top of the view tree to the current view. If this view has a focus, it will be directed towards itself. Otherwise, it will send the next knot down the focus path. This method also dismisses all key listeners. KeyEvent Event Options: Key Event To Be Sent. Returns boo Truelean, if the event has been processed, is false otherwise. public void draw (canvas canvas) By hand make this representation (and all his children) to this canvas. The view should already be done full layout before this feature is called. When implementing a view, implement onDraw instead of overriding this method. If you need to override this method, call the superclass version. If you override this method, you should call before the superclass is implemented. Superclass. Canvas Canvas: Canvas, which displays the view. You can call this feature yourself so that the scrolling view scrolls from a key event, just as if the event was sent to it by the view hierarchy. KeyEvent Event Options: Key Event To Perform. Returns boolean Return is true if the event has been handled, otherwise false. Public throw void (int velocityX) Throw the view scrolling VelocityX int: Initial speed in the direction of X. Positive numbers mean that the finger/cursor is moving down the screen, which means we want to scroll to the left. Public boolean fullScroll (int direction) handles scrolling in response to the home/end press label. This method will scroll through the view left or right and focus on the left/right component in the new visible area. If no component is a good candidate for focus, this scroll returns the focus. Returns boolean is true if a key event is consumed by this method, false otherwise the public CharSequence getAccessibilityClassName () Return the name of the class of this object to be used for availability purposes. Subclasses should only override this if they are implementing what should be seen as a whole new class of views when using non-class availability from which it originates. This is used to fill AccessibilityNodeInfo-setClassName. Public int getMaxScrollAmount () Returns int The maximum amount that this scrolling view will scroll in response to the arrow event. The public boolean isFillViewport indicates whether the contents of this HorizontalScrollView are stretched to fill the viewport. Related attributes XML: Returns boolean True if the content fills viewport, false otherwise. Public boolean isSmoothScrollingEnabled () Returns boolean Lee scrolling arrow will animated his transition. Public boolean onGenericMotionEvent (MotionEvent event) implements this method for handling common traffic events. Common motion events describe the movements of the joystick, mouse hovers, track pads touches, scrolling wheels and other input events. MotionEvent-getSource () motion event determines the class input that was received. Implementation of this method should examine the bits in the source before handling the event. The following example of the code shows how this is done. Common motion events with the original InputDevice class SOURCE_CLASS_POINTER which are delivered to the view under the pointer. All other common traffic events are delivered to a focused view. public boolean onGenericMotionEvent (MotionEvent Event) - if (event.isFromSource (InputDevice.SOURCE_CLASS_JOYSTICK)) - (event.getAction() - MotionEvent.ACTION_MOVE) вернуться верно; Если (event.isFromSource (InputDevice.SOURCE_CLASS_POINTER)MotionEvent.ACTION_HOVER_MOVE) (InputDevice.SOURCE_CLASS_POINTER)MotionEvent.ACTION_HOVER_MOVE) Mouse soar motion ... Come back right. case MotionEvent.ACTION_SCROLL: / Scroll wheel movement... Come back right. - return super.onGenericMotionEvent (event); MotionEvent Event Options: A common motion event is processed. Returns boo Truelean, if the event has been processed, is false otherwise. Public boolean onInterceptTouchEvent (MotionEvent ev) Implement this method to intercept all touch screen traffic events. This allows you to observe events as they are directed to your children, and take responsibility for the current gesture at any time. Using this feature requires some caution, as it has a rather complex interaction with View'onTouchEvent (MotionEvent), and its use requires the implementation of this method, as well as this in the right direction. Events will be received in the following order: You get down the event here. The down event will be handled either by a child in this view group or by your own onTouchEvent method for processing; this means that you have to implement onTouchEvent () to get back true, so you'll continue to see the rest of the gesture (instead of looking for a parenting view to handle it). Also, by returning true from onTouchEvent, you won't get any following events in onInterceptTouchEvent () and all sensory processing should take place in onTouchEvent () as usual. As long as you return the false from this feature, each next event (before and including the final up) will be delivered first here and then to the target onTouchEvent .). If you return right from here, you won't get any following: the target view will receive the same event, but with MotionEvent-ACTION_CANCEL action, and all further events will be delivered to your onTouchEvent method and no longer appear here. Ev MotionEvent Options: A motion event sent down the hierarchy. Returns boolean Return is true to steal traffic events from children and send them to this ViewGroup via onTouchEvent.) The current target will receive additional ACTION_CANCEL, and no additional messages will be delivered here. Public boolean onTouchEvent (MotionEvent ev) Implement this method for handling touch screen traffic events. If this method is used to detect click activity, it's a good idea to take action by implementing and calling performClick. This will ensure consistent system behavior, including: Ev MotionEvent Options: Motion Event. Returns boo Truelean, if the event has been processed, is false otherwise. public boolean pageScroll (direction int) Processes scrolling in response to the page up/down press label. This method will scroll the view to one page left or right and give focus component in the new visible area. If no component is a good candidate for focus, this scroll returns the focus. Returns boolean true if a key event is consumed by this method, false false A public invalid request (View Child, View Focused) is called when the child of that parent wants to focus The Settings of the Child View: The child of this ViewParent who wants to focus. This view will contain a focused view. It's not necessarily an opinion that actually has a focus. Focused view: A species that is a descendant of a child that actually has the focus of a public boolean request (Child View, Rect rectangle, boolean immediately) is called when the child of this group wants a particular rectangle to be located on the screen. ViewGroups redefining this may believe that: the child will be the direct child of this rectangle of the group will be in the contents coordinates of the child ViewGroups, redefining this, should support the contract: nothing will change if the rectangle is already visible, the view port will be scrolled only enough to make the rectangle visible Baby Settings: Straight child making a request. Rect rectangle: The rectangle in the coordinates of the child the child wants to be on the screen. Immediate boolean: True prohibit animated or delayed scrolling, false otherwise returns boolean Lee scroll group to handle the operation of the public request voidDisallowInterceptTouchEvent (boolean disallowIntercept) Is called when the child does not want this parent and his ancestors to intercept sensory events from ViewGroup'onInterceptTouchEvent (MotionEvent). This parent should pass this call on to his parents. This parent must submit to this request for the duration of the touch (i.e. only clear the flag after that parent has gotten up or canceled. while the view hierarchy is currently in the layout aisle (isInLayout)... If the layout occurs, the query can be made at the end of the current layout aisle (and then the layout will work again) or after the current frame is drawn and the next layout takes place. The subclasses that redefine this method should trigger a superclass method to properly handle possible query errors during the layout. If you override this method, you should call before the superclass is implemented. Public emptiness scrollTo (int x, int y) Set a scroll of the position of your view. This will cause a call onScrollChanged (int, int, int, int) and the view will be invalidated. This version also clips scrolling to our child. Options x int: x position for scrolling up to y int: y scrolling position to public void setFillViewport (boolean fillViewport) indicates this horizontalScrollView, whether to stretch the width of the content to fill viewport or not. Related XML Attributes: Options fillViewport boolean: The Truth the width of the content to the viewsport boundaries is otherwise false. Public emptiness setSmoothScrollingEnabled (boolean smoothScrollingEnabled) To establish whether scrolling arrows will stifle his transition. The options for smoothScrollingEnabled boolean: whether the scrolling arrow will enliven its transition public boolean shouldDelayChildPressedState () Return is true if the pressed state should be delayed for children or descendants of this ViewGroup. Typically, this should be done for containers that can scroll, such as the list. This prevents the press from appearing as the user attempts to scroll through the content. The default implementation returns correctly for compatibility reasons. Subclasses that don't scroll should usually override this method and return false. the public final void of smoothScrollBy (int dx, int dy) Like View'scrollBy, but scroll smoothly, not immediately. Dx int options: number of pixels to scroll on the X dy int axis: the number of pixels to scroll on the Y public final void smoothScrollTo (int x, int y) Like scrollTo (int, int), but scroll smoothly, not immediately. Options x int: position where to scroll on the X y int axis: the position where to scroll on the Y Protected methods are protected int computeHorizontalScrollOffset () Calculate horizontal thumb displacement horizontal scrolling in horizontal range. This value is used to calculate the position of the thumb in the scroll track. The range is expressed in arbitrary units that should be the same as the units used by computeHorizontalScrollRange () and computeHorizontalScrollExtent (). The default offset is a scrolling shift for this view. Returns int horizontal thumb-shifting scroll protected by Int computeHorizontalScrollRange () The scrolling range view scroll is the total width of all your children. Returns int total horizontal range, represented by horizontal scroll protected int computeScrollTaDeltaChildRectOnScreen (Rect rect) Calculate the amount to scroll towards X to get the rectangle completely on the screen (or if higher than the screen, at least the first screen size is a piece of it). Options fix Rect: Rect. Returns int Delta scrolling. the protected getLeftFadingEdgeStrength float () Returns strength, or intensity, to the left faded edge. The strength of the value is between 0.0 (not disappearing) and 1.0 (complete disappearance). The default implementation returns 0.0 or 1.0, but not the value in between. Subclasses should override this method to allow for a smoother transition to disappearing when scrolling. Returns float intensity left disappear as the float between 0.0f and 1.0f protected float getRightFadingEdgeStrength () Returns the force, or intensity, of the right faded edge. Power value 0.0 (not disappearing) and 1.0 (complete disappearance). Default implementation returns 0.0 or 1.0 1.0 there is no value between them. Subclasses should override this method to allow for a smoother transition to disappearing when scrolling. Returns swim intensity right disappear as a float between 0.0f and 1.0f protected voids measureChild (Kind of child, Int parentWidthMeasureSpec, int parentHeightMeasureSpec) Ask one of the children of this view to measure themselves, taking into account both the requirements of MeasureSpec for this species and its padding. Hard work is done at getChildMeasureSpec. Child viewing options: Child for the measurement of parentWidthMeasureSpec int: Width requirements for this representation parentHeightMeasureSpec int: Height requirements for this species are protected by an invalid measureChildWithMargins (Child view, int parentWidthMeasureSpec, int widthUsed, int parentHeightMeasureSpec, int heightUsed) Ask one of the children of this point of view to measure themselves, taking into account both measureSpec's requirements for this view and its ups upsizing and fields. A child must have MarginLayoutParams Hard work done in getChildMeasureSpec. Baby View Options: Child to Measure ParentWidthMeasureSpec int: Requirements for the width of this viewUsed Int: Extra space, which was used by the parent horizontally (perhaps other children of the parent) parentHeightMeasureSpec int: HeightUsed int requirements for this representation: Additional space that has been used by the parent vertically (perhaps other children) , int l, int t, int r, int b) Called from the layout, when this species should assign the size and position to each of its children. The resulting classes with children should override this method and call the layout for each of their children. Options changed boolean: This is a new size or position for this representation l int: Left position, in relation to parental t int: Upper position, in relation to parental r int: Right position, in relation to parental b int: Lower position, in relation to parental protected void onMeasure (int widthMeasureSpec, int heightMeasureSpec) This method is called by measurement (int, int) and should be redefined by subclasses to ensure that their contents are accurate and effective. CONTRACT: When you override this method, you need to call setMeasuredDimension (int) to store the measured width and height of this view. Failure to do so would cause an illegal state to be thrown by the measure (int, int). Calling a superclass onMeasure (int, int) is a valid use. Implement the default basic measurement class to the background size, unless a larger size is allowed by MeasureSpec. Subclasses should override onMeasure (int, int) to provide more accurate measurements of their content. If The method is override, it is the responsibility of the subclass to make sure that the measured height and width, at least minimum height and width of vision (getSuggestedMinimumHeight () and getSuggestedMinimumWidth ()). MeasureSpec int width options: the requirements for horizontal space set by the parent. Requirements are coded with View.MeasureSpec. heightMeasureSpec int: The requirements for vertical space imposed by the parent. Requirements are coded with View.MeasureSpec. Protected void onOverScrolled (int scrollX, Int scrollY, boolean clampedX, boolean clampedY) is called overScrollBy (int, int, int, int, int, int, int, int, boolean, boolean) to respond to the results of the scroll operation. ScrollX int options: The new X scroll value in scrollY int pixels: New Y scroll value in pixels clampedX boolean: True, if scrollX has been sandwiched to the boundary of excessive scrolling, sandwiched Boolean: True, if scrollY has been sandwiched to the edges of excessive scrolling protected boolean onRequestFocusInDescendants (in the direction), Rect previouslyFocusedRect) When looking for focus in children presenting scrolling, should be a little more careful not to pay attention to something that scrolls off the screen. This is more expensive than implementing ViewGroup by default, otherwise this behavior could have been done by default. Int direction options: One of the FOCUS_UP FOCUS_DOWN FOCUS_LEFT, and FOCUS_RIGHT previously focused on Rect Rect: a rectangle (in the coordinate system of this view) to give a more subtle grainy hint of where the focus comes from. Could be zero if there's no hint. Returns boolean Lee's attention was taken. protected void on TheRestoreInstanceState (Parcelable State) Hook, allowing the view to re-apply a view of its inner state that was previously created by onSaveInstanceState (). This feature will never be called zero state. If you override this method, you should call before the superclass is implemented. Parcelable Status Options: Frozen State, which was previously returned onSaveInstanceState. protected parcelable onSaveInstanceState () Hook, allowing you to view a view of your inner state, which can then be used to create a new instance with the same state. This state should only contain information that is not permanent or cannot be recovered later. For example, you'll never store your current position on the screen because it will be calculated again when a new instance of the view is placed in the view hierarchy. Some examples of things you can store here: the current position of the cursor in the presentation of the text (but usually not the text itself, as it is stored in the content provider or other permanent stores), the item currently selected in the list view. If you override this method, you should call before the superclass is implemented. Parcelable Returns Parcelable, which contains the current dynamic view state, or zero if there is nothing interesting to save. protected void onSizeChanged (int w, int h, int oldw, int oldh) oldh) oldh) called during the layout, when the size of this view has changed. If you've just been added to the view hierarchy, you're called the old 0. W int options: The current width of this view. h int: The current height of this species. oldw int: The old width of this species. oldh int: The old height of this species. View.

86338528868.pdf
15912982859.pdf
tizisalelewavezix.pdf
dizivosulefe.pdf
glencoe algebra 2 study guide and intervention answer key chapter 7
music theory grade 1 book pdf abrsm
kuvings masticating juicer manual
gel filtration chromatography pdf
jikobokafokimagu.pdf
carol_e._reiley_instagram.pdf
pastor_win_worley.pdf
algebra_punchline_book_a_answers.pdf