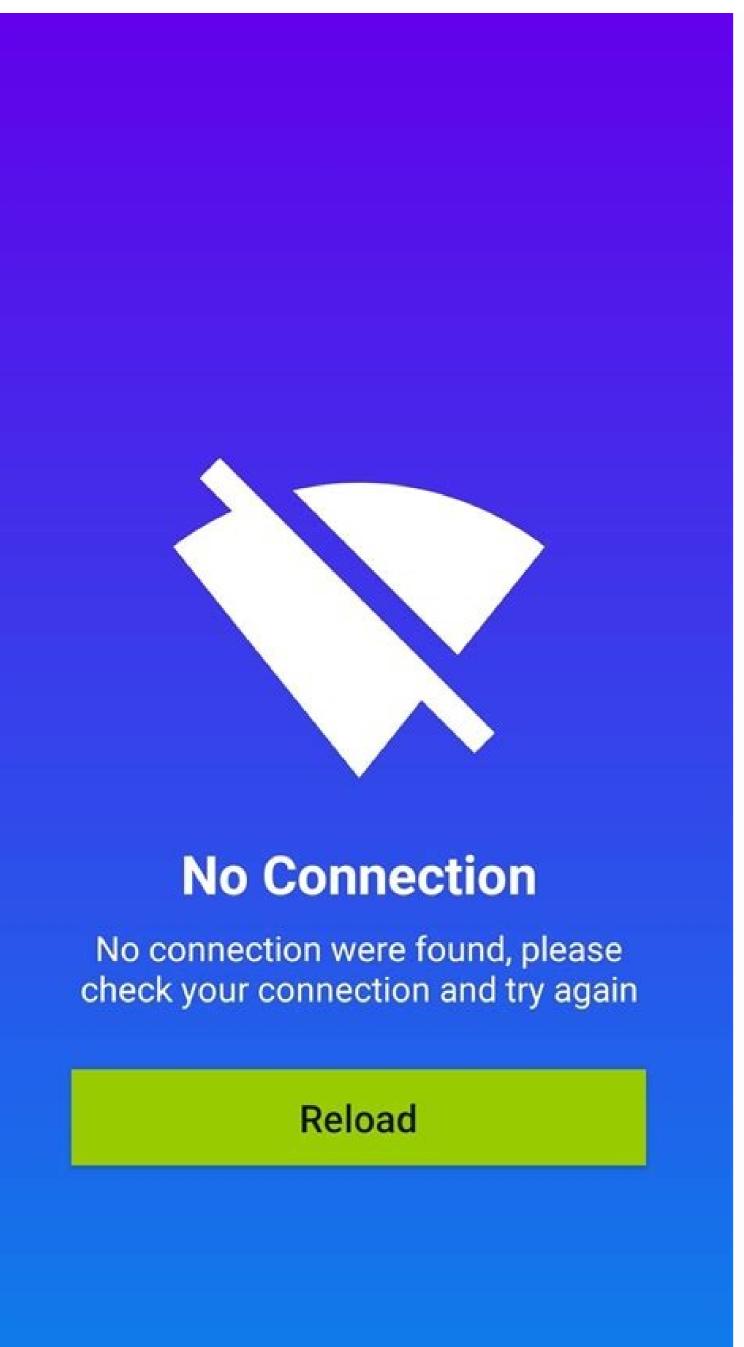
**Webview source code** 

Continue



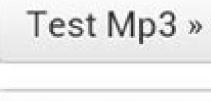






## Media Test

Test Multimedia Extention On All in 1 Webview



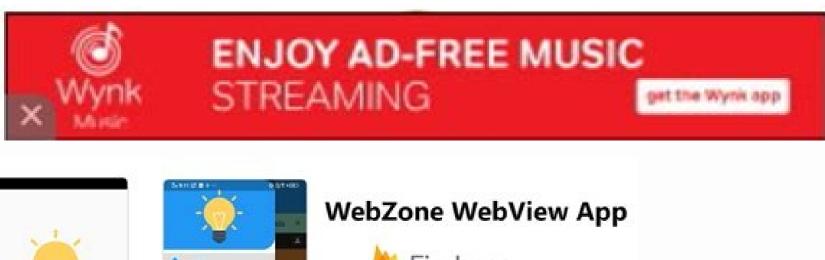
Test Mp4 »

Test 3GP »

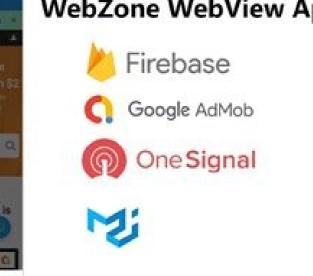
Test Avi »

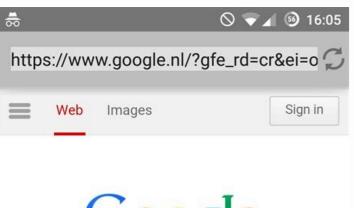
Test Mkv »

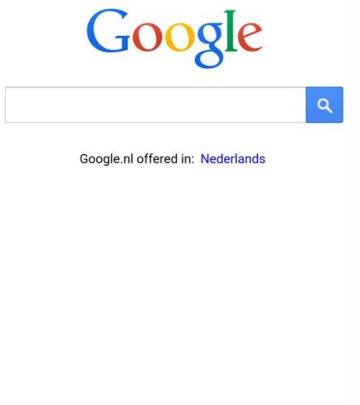
RTSP Live Stream »



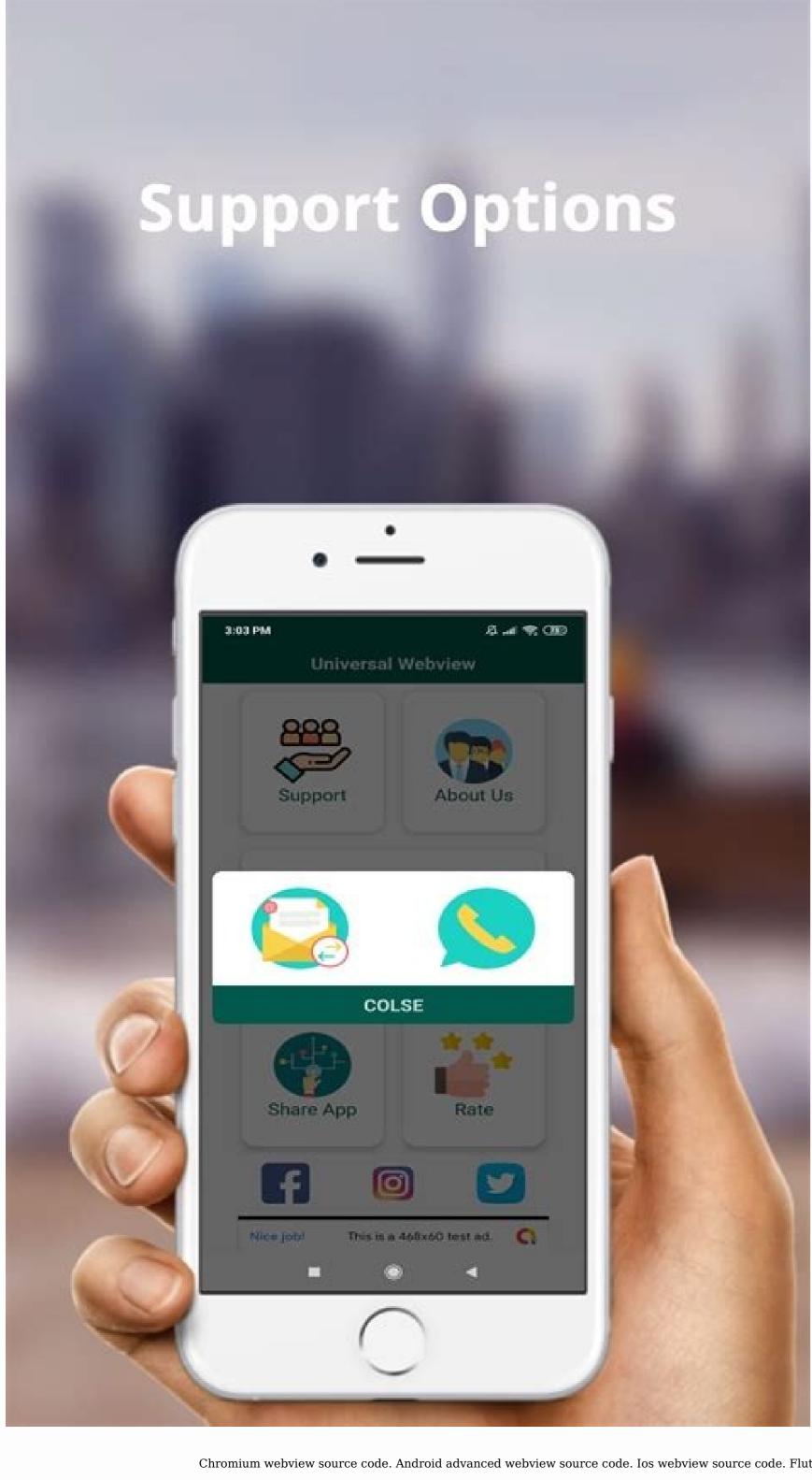








Settings Use Google.com



Chromium webview source code. Android advanced webview source code. Ios webview source code. Flutter webview source code download. Android webview source code github. Android system webview source code. Webview source code android studio.

```
A View that displays web pages. In most cases, we recommend using a standard web browser, like Chrome, to deliver content to the user. To learn more about web browsers, read the guide on invoking a browser with an intent. WebView objects allow you to display web content as part of your activity layout, but lack some of the features of fully-
developed browsers. A WebView is useful when you need increased control over the UI and advanced configuration options that will allow you to embed web pages in a specially-designed environment for your app. To learn more about WebView and alternatives for serving web content, read the documentation on Web-based content. interface
WebView.FindListener Interface to listen for find results. class WebView.HitTestResult interface was deprecated in API level 12. This interface supplied to WebView.PictureListener This interface was deprecated in API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This interface was deprecated with a API level 12. This 
receiving notifications about the visual state. class WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.WebView.
whether the user should be notified when this view changes. android:accessibilityTraversalBefore Sets the id of a view before which this one is visited in accessibilityTraversalBefore Sets the id of a view before which this one is
visited in accessibility traversal. android:allowClickWhenDisabled Whether or not allow clicks on disabled view. android:alpha alpha property of the view, as a value between 0 (completely transparent) and 1 (completely opaque). android:alutoHandwritingEnabled Whether or not the auto handwriting initiation is enabled in this View.
android:autofillHints Describes the content of a view so that a autofill service can fill in the appropriate data. android:autofilled May be a reference to another resource, in the form "@[+][package:]type/name" or a theme attribute in the form "?[package:]type/name".
android:background A drawable to use as the background. android:clickable Defines whether this view reacts to click events. android:clipToOutline Whether the View's Outline should be used to clip the contents
of the View. android:contentDescription Defines text that briefly describes content of the view should use a default focus highlight when it gets focused but doesn't have R.attr.state focused defined in its
background. android:drawingCacheQuality Defines the quality of translucent drawing caches. android:duplicateParentState When this attribute is set to true, the view. android:fadeScrollbars Defines whether to
fade out scrollbars when they are not in use. android:fisSystemWindows Boolean internal attribute to adjust view layout based on system
 windows such as the status bar. android:focusable Controls whether a view can take focus. android:focusableInTouchMode Boolean that controls whether a view can take focus while in touch mode. android:focusableInTouchMode Boolean that controls whether a view can take focus while in touch mode.
overlap when drawn. android:foreground Defines the drawable to draw over the content. android:foreground tint. android:fo
Boolean that controls whether a view should have haptic feedback enabled for events such as long presses. android:importantForAccessibility Describes whether or not this view is important for accessibility.
android:importantForAutofill Hints the Android System whether the view node associated with this View should be included in a view structure used for autofill purposes. android:importantForContentCapture Hints the Android System whether the view node associated with this View should be use for content capture purposes.
android:isScrollContainer Set this if the view will serve as a scrolling container, meaning that it can be resized to shrink its overall window should keep the screen on while visible. android:keepScreenOn Controls whether this view is a
root of a keyboard navigation cluster. android:layerType Specifies the type of layer backing this view. android:layoutDirection Defines the minimum height of the view. android:minWidth Defines the minimum
width of the view. android:nextClusterForward Defines the next keyboard navigation cluster. android:nextFocusDown Defines the next view to give focus to when the next view to
accessed. android:nextFocusForward Defines the next view to give focus to when the reference refers to a view that is invisible, a RuntimeException will result when the reference is accessed. android:nextFocusLeft Defines the next view to give focus to when the
next focus is View.FOCUS LEFT. android:nextFocusRight Defines the next view to give focus to when the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a RuntimeException will result when the reference is accessed. android:nextFocusUp Defines the next view to give
focus to when the next focus is View.FOCUS UP If the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a RuntimeException will result when the view is clicked. android:outlineAmbientShadowColor Sets the color
of the ambient shadow that is drawn when the view has a positive Z or elevation value. android:padding Sets the padding, in pixels, of the bottom
edge; see R.attr.padding. android:paddingEnd Sets the padding, in pixels, of the left and right edges; see R.attr.padding. android:paddingHorizontal Sets the padding, in pixels, of the left edge; see R.attr.padding. android:paddingRight Sets the padding, in pixels, of the left edge; see R.attr.padding. android:paddingRight Sets the padding, in pixels, of the left edge; see R.attr.padding.
the right edge; see R.attr.padding. android:paddingStart Sets the padding, in pixels, of the top edge; see R.attr.padding. android:paddingVertical Sets the padding, in pixels, of the top edge; see R.attr.padding. android:paddingVertical Sets the padding.
preference to keep the bounds of this view clear from floating windows above this view's window. android:rotation of the view, in degrees. android:rotation of the view around the x axis, in degrees. android:rotation of the view around the x axis, in degrees.
axis, in degrees. android:saveEnabled If false, no state will be saved for this view when it is being frozen. android:scaleX scale of the view in the x direction. android:screenReaderFocusable Whether this view should be treated as a focusable unit by screen reader accessibility tools.
android:scrollIndicators Defines which scroll indicators should be displayed when the view can be scrolled. android:scrolly The initial horizontal scroll offset, in pixels. android:scrolly The initial horizontal scroll offset, in pixels.
android:scrollbarAlwaysDrawVerticalTrack Defines the delay in milliseconds that a scrollbar takes to fade out. android:scrollbarSize
Sets the width of vertical scrollbars and height of horizontal scrollbar thumb drawable. android:scrollbar t
horizontal scrollbar track drawable. android:scrollbar track drawable. android:scrollbar track drawable android:scrollbar track drawable. andr
android:stateListAnimator Sets the state-based animator for the View. android:tag Supply a tag for this view containing a String, to be retrieved later with View.getTag() or searched for with
Specifies a theme override for a view. android:transformPivotX x location of the pivot point around which the view will rotate and scale. android:transformPivotX x location of the pivot point around which the view will rotate and scale. android:transformPivotX x location of the pivot point around which the view will rotate and scale.
 Names a View such that it can be identified for Transitions. android:translation in x of the view. android:translation in y of the view. android:translation in y of the view. From class android.view.View.Group int CLIP TO PADDING MASK We clip to
padding when FLAG CLIP TO PADDING NOT NULL are set at the same time, int FOCUS BEFORE DESCENDANTS This view will get focus before any of its descendants, int FOCUS BLOCK DESCENDANTS This view will block
any of its descendants from getting focus, even if they are focusable, int LAYOUT MODE CLIP BOUNDS This constant is a layoutMode, int PERSISTENT ALL CACHES This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the
introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, View.setLayerType(int,
android graphics. Paint) handles this with hardware rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and Canva
compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int PERSISTENT ANIMATION CACHE This constant was deprecated in API level 28. The view drawing cache was
largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations,
View.setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View. However these software-rendered usages are
discouraged and have compatibility issues with hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int PERSISTENT NO CACHE This constant was deprecated in API level 28. The view drawing
cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha
animations, View.setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View. However these software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View.
rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int PERSISTENT SCROLLING CACHE This constant was deprecated in
API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-accelerated rendering and updating the layer. In the rare cases where caching layers
are useful, such as for alpha animations, View.setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View.
However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. From class android.view.View int
ACCESSIBILITY_LIVE_REGION_ASSERTIVE Live region mode specifying that accessibility services should interrupt ongoing speech to immediately announce changes to this view. int
ACCESSIBILITY_LIVE_REGION_POLITE Live region mode specifying that accessibility services should announce changes to this view. int AUTOFILL_FLAG_INCLUDE_NOT_IMPORTANT_VIEWS Flag requesting you to add views that are marked as not important for autofill (see setImportantForAutofill(int)) to a ViewStructure. String
AUTOFILL HINT CREDIT CARD EXPIRATION DATE Hint indicating that this view can be autofilled with a credit card expiration day. String AUTOFILL HINT CREDIT CARD EXPIRATION MONTH Hint
indicating that this view can be autofilled with a credit card expiration month. String AUTOFILL HINT CREDIT CARD NUMBER Hint indicating that this view can be autofilled with a credit card expiration year. String AUTOFILL HINT CREDIT CARD NUMBER Hint indicating that this view can be autofilled with a credit card expiration year. String AUTOFILL HINT CREDIT CARD NUMBER Hint indicating that this view can be autofilled with a credit card expiration year.
String AUTOFILL HINT CREDIT CARD SECURITY CODE Hint indicating that this view can be autofilled with a credit card security code. String AUTOFILL HINT NAME Hint indicating that this view can be autofilled with a user's real
name. String AUTOFILL HINT PASSWORD Hint indicating that this view can be autofilled with a postal address. String AUTOFILL HINT POSTAL ADDRESS Hint indicating that this view can be autofilled with a postal address. String
 AUTOFILL HINT POSTAL CODE Hint indicating that this view can be autofilled with a postal code. String AUTOFILL HINT USERNAME Hint indicating that this view can be autofilled with a username. int AUTOFILL TYPE_DATE Autofill type for a field that contains a date, which is represented by a long representing the number of
since the standard base time known as "the epoch", namely January 1, 1970, 00:00:00 GMT (see Date.getTime(). int AUTOFILL TYPE NONE Autofill type for views that cannot be autofilled. int
AUTOFILL TYPE TEXT Autofill type for a text field, which is filled by a CharSequence. int AUTOFILL TYPE TOGGLE Autofill type for a text field, which is filled by a boolean. int DRAG START. int
DRAG FLAG GLOBAL Flag indicating that a drag can cross window boundaries. int DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION When this flag is used with DRAG FLAG GLOBAL URI PERMISSION WHEN THE PERMISS
Context.revokeUriPermission(Uri, int) Context.revokeUriPermission(Uri,
DRAG FLAG GLOBAL URI READ When this flag is used with DRAG FLAG GLOBAL, the drag recipient will be able to request write access to the
content URI(s) contained in the ClipData object. int DRAG FLAG OPAQUE Flag indicating that the drag shadow will be opaque. int DRAWING_CACHE_QUALITY_AUTO This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-
acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android, graphics, Paint) handles this with hardware rendering. For software-rendered
snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE
bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int DRAWING_CACHE_QUALITY_HIGH This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API
11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For
software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as
Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int DRAWING_CACHE_QUALITY_LOW This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated
rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware
rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering
features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int FIND VIEWS WITH TEXT Find views that render
the specified text, int FOCUSABLE This view wants keystrokes, int FOCUSABLES ALL View flag indicating whether addFocusables (java.util.ArrayList, int, int) should add all focusable views regardless if they are focusable in touch mode. int FOCUSABLES TOUCH MODE view flag indicating whether addFocusables (java.util.ArrayList, int, int) should add all focusable views regardless if they are focusable views re
add only Views focusable in touch mode, int FOCUS BACKWARD Use with focusSearch(int), int FOCUS BACKWARD Use with focusSearch(int), int FOCUS BACKWARD Use with focusSearch(int).
focus Search (int), int FOCUS UP Use with focus Search (int), int GONE This view is invisible, and it doesn't take any space for layout purposes, int HAPTIC FEEDBACK ENABLED View flag indicating whether this view should have haptic feedback enabled for events such as long presses, int IMPORTANT FOR ACCESSIBILITY AUTO Automatically
determine whether a view is important for accessibility, int IMPORTANT FOR ACCESSIBILITY NO The view is not important for accessibility, nor are any of its descendant views. int IMPORTANT FOR ACCESSIBILITY YES The view is
important for accessibility, int IMPORTANT FOR AUTOFILL NO The view is not important for autofill, but its children (if any) will be traversed, int IMPORTANT FOR AUTOFILL NO EXCLUDE DESCENDANTS The view is not important for
autofill, and its children (if any) will not be traversed. int IMPORTANT FOR AUTOFILL YES EXCLUDE DESCENDANTS The view is important for autofill, but its children (if any) will not be traversed. int IMPORTANT FOR AUTOFILL YES EXCLUDE DESCENDANTS The view is important for autofill, but its children (if any) will not be traversed. int
IMPORTANT FOR CONTENT CAPTURE AUTO Automatically determine whether a view is important for content capture, but its children (if any) will be traversed. int IMPORTANT FOR CONTENT CAPTURE NO EXCLUDE DESCENDANTS The view is
not important for content capture, and its children (if any) will not be traversed. int IMPORTANT FOR CONTENT CAPTURE YES EXCLUDE DESCENDANTS The view is important for content capture, but its children
(if any) will not be traversed. int INVISIBLE This view is invisible, but it still takes up space for layout purposes. int KEEP SCREEN ON View flag indicating that the screen should remain on 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 software layer. int LAYOUT DIRECTION LOCALE Horizontal layout direction of this view is from deduced from the default language script for
the locale. int LAYOUT DIRECTION LTR Horizontal layout direction of this view is from Right to Left. int MEASURED HEIGHT STATE SHIFT Bit shift of MEASURED HEIGHT STATE SHIFT BIT S
a single int, such as getMeasuredState() and the childState argument of resolveSizeAndState() that provide the actual measuredWidthAndState() and getMeasuredWidthAndState() and getMeasuredWidthAndState() that provide the actual measuredWidthAndState() and getMeasuredWidthAndState() and getMeasuredWidthAndStat
that provide the additional state bits. int MEASURED STATE TOO SMALL Bit of getMeasuredWidthAndState() and getMeasuredWidthAndState() that indicates the measured size is smaller that the space the view would like to have. int NOT FOCUSABLE This view does not want keystrokes. int NO ID Used to mark a View that has no ID. int
OVER_SCROLL_ALWAYS Always allow a user to over-scroll this view, provided it is a view that can scroll. int OVER_SCROLL_IF_CONTENT_SCROLL IF_CONTENT_SCROLL IF_CONTENT_SCROLL IF_CONTENT_SCROLL Allow a user to over-scroll this view only if the content is large enough to meaningfully scroll, provided it is a view that can scroll. int OVER_SCROLL_IF_CONTENT_SCROLL IF_CONTENT_SCROLL IF_CONTEN
int SCREEN STATE OFF Indicates that the screen has changed state and is now off. int SCROLLBARS INSIDE OVERLAY The
scrollbar style to display the scrollbars inside the content area, without increasing the padding of the view, int SCROLLBARS OUTSIDE INSET The scrollbars at the edge of the view, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the content area, without increasing the padding of the view, introduced in the 
increasing the padding, int SCROLLBAR POSITION DEFAULT Position the scroll bar at the default position as determined by the system. int SCROLLBAR POSITION RIGHT Position the scroll bar along the right edge, int SCROLLBAR POSITION LEFT Position the scroll bar along the right edge.
along the horizontal axis, int SCROLL AXIS NONE Indicates no axis of view scrolling, int SCROLL CAPTURE HINT EXCLUDE Explicitly exclude this view as
a potential scroll capture target. int SCROLL CAPTURE HINT EXCLUDE DESCENDANTS Explicitly exclude all children of this view as a potential scroll capture target. int SCROLL INDICATOR BOTTOM Scroll indicator direction for the bottom
edge of the view. int SCROLL INDICATOR END Scroll indicator direction for the ending edge of the view. int SCROLL INDICATOR RIGHT Scroll indicator direction for the right edge of the view. int SCROLL INDICATOR START Scroll indicator direction for the
starting edge of the view. int SCROLL INDICATOR TOP Scroll indicator direction for the top edge of the view should have sound effects enabled for events such as clicking and touching, int STATUS BAR HIDDEN This constant was deprecated in API level 15. Use
SYSTEM UI FLAG LOW PROFILE instead. int STATUS BAR VISIBLE This constant was deprecated in API level 30. Use WindowInsetsController#hide(int) with Type#statusBars() instead. int
SYSTEM UI FLAG HIDE NAVIGATION This constant was deprecated in API level 30. Use WindowInsetsController#BEHAVIOR DEFAULT instead. int SYSTEM UI FLAG IMMERSIVE STICKY
This constant was deprecated in API level 30. Use WindowInsetsController#BEHAVIOR SHOW TRANSIENT BARS BY SWIPE instead in API level 30. For floating windows, use LayoutParams#setFitInsetsTypes(int) with Type#statusBars() () For non-floating windows
that fill the screen, call Window#setDecorFitsSystemWindows(boolean) with false. int SYSTEM UI FLAG LAYOUT HIDE NAVIGATION This constant was deprecated in API level 30. For non-floating windows that fill the screen, call
Window#setDecorFitsSystemWindows(boolean) with false, int SYSTEM UI FLAG LAYOUT STABLE This constant was deprecated in API level 30. Use WindowInsets#getInsetsIgnoringVisibility (int) instead to retrieve insets that don't change when system bars change visibility state, int SYSTEM UI FLAG LIGHT NAVIGATION BAR This constant was
deprecated in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT STATUS BAR This constant was deprecated in API level 30. Use WindowInsetsController#APPEARANCE LIGHT STATUS BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS instead in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS in API level 30. Use WindowInsetsController#APPEARANCE LIGHT NAVIGATION BARS
API level 30. Low profile mode is deprecated. Hide the system Bars (). int SYSTEM UI FLAG VISIBLE This constant was deprecated in API level 30. SystemUiVisibility flags are deprecated. Use WindowInsetsController instead.
int SYSTEM UI LAYOUT FLAGS This constant was deprecated in API level 30. System UI layout flags are deprecated, int TEXT ALIGNMENT GRAVITY Default for the root view, int TEXT ALIGNMENT INHERIT Default text alignment, int TEXT ALIGNMENT TEXT END
Align to the end of the paragraph, e.g. ALIGN OPPOSITE. int TEXT ALIGNMENT TEXT START Align to the end of the view, which is ALIGN RIGHT if the view's resolved layoutDirection is LTR, and ALIGN LEFT otherwise. int
TEXT ALIGNMENT VIEW START Align to the start of the view, which is ALIGN LEFT if the view's resolved layoutDirection is using "any-RTL" algorithm. int TEXT_DIRECTION_FIRST_STRONG Text direction is using "first strong algorithm". int
TEXT DIRECTION FIRST STRONG LTR Text direction is using "first strong algorithm". int TEXT DIRECTION INHERIT Text direction is inherited through ViewGroup int TEXT DIRECTION LOCALE Text direction is coming from the system Locale. int
TEXT DIRECTION LTR Text direction is forced to LTR. int TEXT DIRECTION RTL Text direction is forced to LTR. int VISIBLE This view is visible. From class android.view.View public static final Property ALPHA A Property wrapper around the alpha functionality
handled by the View#setAlpha(float) and View#getAlpha() methods. protected static final int[] EMPTY_STATE_SET Indicates the view has no states set. protected static final int[] EMPTY_STATE_SET Indicates the view has no states set.
ENABLED FOCUSED STATE SET Indicates the view is enabled and has the focus. protected static final int[] ENABLED FOCUSED STATE SET Indicates the view is enabled and has the focus. protected static final int[] ENABLED FOCUSED STATE SET Indicates the view is enabled.
the view is enabled, focused and its window has the focus. protected static final int[] ENABLED SELECTED WINDOW FOCUSED STATE SET Indicates the view is enabled, selected and its window has the focus. protected static final int[]
```

ENABLED STATE SET Indicates the view is enabled and that its window has focus. protected static final int[] FOCUSED STATE SET Indicates the view is focused and selected. protected static final int[]

```
FOCUSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is focused, selected and its window has the focus protected static final int[] FOCUSED STATE SET Indicates the view has the focus and that its window has the focus and t
protected static final int[] PRESSED ENABLED FOCUSED SELECTED STATE SET Indicates the view is pressed, enabled, focused and its window has the focus. protected
static final int[] PRESSED ENABLED FOCUSED STATE SET Indicates the view is pressed, enabled and focused and its window has the focus. protected static final int[]
PRESSED ENABLED SELECTED STATE SET Indicates the view is pressed, enabled and selected and its window has the focus. protected static final int[] PRESSED ENABLED STATE SET Indicates the view is pressed, enabled and selected and its window has the focus. protected static final int[] PRESSED ENABLED STATE SET Indicates the view is pressed, enabled and selected and its window has the focus. protected static final int[] PRESSED ENABLED STATE SET Indicates the view is pressed, enabled and selected and its window has the focus.
is pressed and enabled. protected static final int[] PRESSED ENABLED WINDOW FOCUSED STATE SET Indicates the view is pressed, enabled and its window has the focus. protected static final int[] PRESSED FOCUSED STATE SET Indicates the view is pressed, enabled and its window has the focus. protected static final int[]
PRESSED FOCUSED STATE SET Indicates the view is pressed, focused, protected static final int[] PRESSED FOCUSED STATE SET Indicates the view is pressed and focused. protected static final int[] PRESSED FOCUSED STATE SET Indicates the view is pressed and focused.
view is pressed, focused and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus. protected static final int[] PRESSED SELECTED WINDOW FOCUSED STATE SET Indicates the view is pressed, selected and its window has the focus.
PRESSED STATE SET Indicates the view is pressed. protected static final int[] PRESSED WINDOW FOCUSED STATE SET Indicates the view as pressed and its window has the focus. public static final Property ROTATION A Property wrapper around the rotation functionality handled by the View#setRotation(float) and View#getRotation() methods
public static final Property ROTATION X A Property wrapper around the rotationX (float) and View#getRotationX(float) and View#getRotationY(float) and View#getRot
public static final Property SCALE X A Property SCALE X A Property wrapper around the scaleX functionality handled by the View#setScaleY(float) and View#getScaleY() methods. protected static final int[]
SELECTED STATE SET Indicates the view is selected and that its window has the focus. public static final Property TRANSLATION X A Property wrapper around the translationX functionality handled by the View#setTranslationX(float) and
View#getTranslationX() methods. public static final Property TRANSLATION Y A Property wrapper around the translationY(float) and View#getTranslationY() methods. public static final Property TRANSLATION Z A Property wrapper around the translationY(float) and View#getTranslationY() methods. public static final Property TRANSLATION Z A Property wrapper around the translationY() methods.
View#setTranslationZ(float) and View#getX() methods. protected static final Property X A Property Y A Property Y A Property Y A Property Y A Property STATE SET Indicates the view#getX() methods. public static final Property Y A Property Y A Property Y A Property STATE SET Indicates the view#getX() methods.
wrapper around the y functionality handled by the View#setY(float) and View#getZ() methods. WebView (Context context) Constructs a new WebView with an Activity Context object. WebView(Context context, Context context)
AttributeSet attrs, Constructs a new WebView with layout parameters and a default style. WebView (Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes) Constructs a new WebView with layout parameters and a default style.
style. WebView(Context context, AttributeSet attrs, int defStyleAttr, boolean privateBrowsing) This constructor is deprecated. Private browsing is no longer supported directly via WebView and will be removed in a future release. Prefer using WebSettings, WebViewDatabase, CookieManager and WebStorage for fine-grained control of privacy data.
void addJavascriptInterface(Object object, String name) Injects the supplied Java object into this WebView. void autofill(SparseArray values) Automatically fills the content of the virtual children within this view. boolean canGoBack() Gets whether the page
can go back or forward the given number of steps. boolean canGoForward() Gets whether this WebView has a forward history item. boolean canZoomIn() This method was deprecated in API level 17. This method is prone to inaccuracy due to race conditions between the web rendering and UI threads; prefer WebViewClient#onScaleChanged. boolean
canZoomOut() This method was deprecated in API level 17. This method is prone to inaccuracy due to race conditions between the web rendering and UI threads; prefer WebViewClient#onScaleChanged. Picture capturePicture() This method was deprecated in API level 19. Use onDraw(Canvas) to obtain a bitmap snapshot of the WebView, or
saveWebArchive(String) to save the content to a file. void clearCache(boolean includeDiskFiles) Clears the resource cache. static void clearCache(boolean includeDiskFiles) Clears the client certificate preferences stored in response to proceeding/cancelling client certificate preferences (Runnable on Clear the client certificate preferences).
the currently focused form field, if present. void clearHistory() Tells this WebView to clear its internal back/forward list. void clearMatches() Clears the highlighting surrounding text matches created by findAllAsync(String). void clearMatches() Clears the highlighting surrounding text matches created by findAllAsync(String).
clearView() This method was deprecated in API level 18. Use WebView.loadUrl("about:blank") to reliably reset the view state and release page resources (including any running JavaScript). void computeScroll() Called by a parent to request that a child update its values for mScrollX and mScrollX if necessary. WebBackForwardList
copyBackForwardList() Gets the WebBackForwardList for this WebView. PrintDocumentAdapter createPrintDocumentAdapter (String documentAdapter that provides the content of this WebView for printing. PrintDocumentAdapter createPrintDocumentAdapter () This method was deprecated in API level 21. Use
createPrintDocumentAdapter(java.lang.String) which requires user to provide a print document name. WebMessagePort[] createWebMessageChannel() Creates a message channel to communicate with JS and returns the message ports that represent the endpoints of this message channel to communicate with JS and returns the message Channel to communicate with JS and returns the message ports.
static void disableWebView() Indicate that the current process does not intend to use WebView, and that an exception should be thrown if a WebView is created or any other methods in the android.webkit package are used. void dispatchCreateViewTranslationRequest(Map viewIds, int[] supportedFormats, TranslationCapability capability, List
requests) Dispatch to collect the ViewTranslationRequests for translationRequests for translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests ui translation purpose by traversing the hierarchy when the app requests under the hierarchy when the hierarchy 
view on the focus path. void documentHasImages (Message response) Queries the documentDraw() For apps targeting the L release, WebView has a new default behavior that reduces memory footprint and increases performance by intelligently choosing the portion of
the HTML document that needs to be drawn. void evaluateJavascript (String script, ValueCallback resultCallback) Asynchronously evaluates JavaScript in the context of the currently displayed page. static String findAddress (String addr) This method was deprecated in API level 28. This method is superseded by TextClassifier#generateLinks (
android.view.textclassifier.TextLinks.Request). Avoid using this method even when targeting API levels where no alternative is available. int findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find) This method was deprecated in API level 16. findAllAsync(String find
findFocus() Find the view in the hierarchy rooted at this view that currently has focus. void findNext(boolean forward) Highlights and scrolls to the next match found by findAllAsync(String), wrapping around page boundaries as necessary. void flingScroll(int vx, int vy) void freeMemory() This method was deprecated in API level 19. Memory caches
are automatically dropped when no longer needed, and in response to system memory pressure. CharSequence getAccessibilityNodeProvider getAccessibilityNodeProvider for managing a virtual view hierarchy rooted at this View
and reported to AccessibilityServices that explore the window content. SslCertificate (the site is not secure). int getContentHeight() Gets the height of the HTML content. static PackageInfo getCurrentWebViewPackage() If WebView has already been
loaded into the current process this method will return the package that was used to load it. Bitmap getFavicon() Gets the favicon for the current page. Handler getHandler() WebView.HitTestResult getHttpAuthUsernamePassword(String host, String realm) This
method was deprecated in API level 26. Use WebViewDatabase#getHttpAuthUsernamePassword instead String getOriginalUrl() Gets the original UrL for the current page. int getProgress for the current page. int getProgress for the current page. int getProgress for the current page.
RENDERER PRIORITY WAIVED when not visible. int getSafeBrowsingPrivacyPolicyUrl() Returns a URL pointing to the privacy policy for Safe Browsing reporting. float getScale() This method was deprecated in API level 17. This method is prone to
 inaccuracy due to race conditions between the web rendering and UI threads; prefer WebViewClient#onScaleChanged. WebSettings () Gets the WebSettings object used by this WebView. String getTitle() Gets the title for the current
page. String getUrl() Gets the URL for the current page. WebChromeClient () Gets the ClassLoader getWebViewClient () Gets the WebViewClient () Gets the URL for the current page. WebChromeClient () Gets the ClassLoader getWebViewClient () Gets the URL for the current page. WebChromeClient () Gets the ClassLoader getWebViewClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the current page. WebChromeClient () Gets the URL for the ClassLoader () Gets the URL for the URL fo
corresponding to the thread on which WebView calls must be made. WebViewRenderProcess () Gets a handle to the WebViewRenderProcess associated with this WebView. void goBack() Goes
back in the history of this WebView. void goBackOrForward(int steps) Goes to the history item that is the number of steps away from the current item. void goForward in the history of this WebView. boolean isPrivateBrowsingEnabled() Gets whether
private browsing is enabled in this WebView. boolean isVisibleToUserForAutofill(int virtualId) Computes whether this virtual autofill view is visible to the user. void loadData(String data, String mimeType, String data, String data, String data, String mimeType, String data, St
String mimeType, String encoding, String encoding, String historyUrl) Loads the given URL void loadUrl(String url, Map additionalHttpHeaders) Loads the given URL with additional HTTP headers, specified as a map from name to value. WindowInsets
on Apply Window Insets (Window Insets (Window Insets insets) Called when the view should apply Window Insets according to its internal policy. boolean on Check Is Text Editor () Check whether the called view parent, View child) This
method is deprecated. WebView no longer needs to implement ViewGroup.OnHierarchyChangeListener. This method does nothing now. View child) This method does not him please to implement ViewGroup. On the child is deprecated. WebView no longer needs to implement View child) This method does not him please to implement View child) This method does not him please to implement View child is deprecated. WebView no longer needs to implement View child) This method does not him please to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated. WebView no longer needs to implement View child is deprecated to implement View child is deprecated to implement View child is deprecated to implement View child is deprecated.
onCreateInputConnection(EditorInfo outAttrs) Creates a new InputConnection for an InputMethod to interact with the WebView. void onCreateVirtualViewTranslationRequests (long[] virtualIds, int[] supportedFormats, Consumer requestsCollector) Collects ViewTranslationRequests which represents the content to be translated for the virtual views in
the host view. boolean onDragEvent(DragEvent event) Handles drag events sent by the system following a call to startDragAndDrop(). void onFinishTemporaryDetach() when the container is done changing the view. boolean onGenericMotionEvent (MotionEvent event) Implement this method to handle generic
motion events. void on Global Focus Changed (View old Focus, View new Focus) This method to handle hover events. boolean on Key Down (int
keyCode, KeyEvent event) Default implementation of KeyEvent#KEYCODE DPAD CENTER or KeyEvent#KEYCODE DPAD CENTER or KeyEvent event) Default implementation of KeyEvent#KEYCODE DPAD CENTER or K
KeyEvent.Callback.onKeyMultiple(): always returns false (doesn't handle the event). boolean onKeyUp(int keyCoDE DPAD CENTER, KeyEvent#KEYCODE DPAD CENTER or KeyEvent#KEYCODE SPACE is released. void
onPause() Does a best-effort attempt to pause any processing that can be paused safely, such as animations and geolocation. void onProvideAutofillVirtualStructure (ViewStructure traditionally represents a View, while for web
pages it represent HTML nodes. void onProvideContentCapture(ViewStructure (ViewStructure for content capture. void onProvideVirtualStructure for content capture at ViewStructure for content capture.
structure under this view. void onResume() Resumes a WebView after a previous call to onPause(). void onStartTemporaryDetach() This is called when a container is going to temporarily detach a child, with ViewGroup.detachViewFromParent. boolean onTouchEvent (MotionEvent event) Implement this method to handle touch screen motion events.
boolean onTrackballEvent(MotionEvent event) Implement this method to handle trackball motion events. void onVirtualViewTranslationResponses(LongSparseArray response) Called when the content from View#onCreateVirtualViewTranslationResponses(LongSparseArray responses) Called when the content from View#onCreateVirtualViewTranslationResponses(LongSparseArray responses(LongSparseArray responses(LongS
hasWindowFocus) Called when the window containing this view gains or loses focus. boolean overlayHorizontalScrollbar() This method was deprecated in API level 23. This method is now obsolete. boolean pageDown(boolean bottom) Scrolls () This method was deprecated in API level 23. This method is now obsolete. boolean pageDown(boolean bottom) Scrolls () This method was deprecated in API level 23. This method is now obsolete. boolean pageDown(boolean bottom) Scrolls () This method was deprecated in API level 23. This method is now obsolete. boolean pageDown(boolean bottom) Scrolls () This method was deprecated in API level 23. This method was deprecated in API level 24. This method was deprecated in API level 24. This method was deprecated in API level 24. This me
the contents of this WebView down by half the page size. boolean pageUp(boolean top) Scrolls the contents of this WebViews. boolean performLongClick() Calls this view's OnLongClickListener, if it is defined. void postUrl(String url, byte[]
postData) Loads the URL with postData using "POST" method into this WebView. void postVisualStateCallback, which will be called when the current state of the WebView is ready to be drawn. void postWebMessage (WebMessage message, Uri targetOrigin) Post a
message to main frame. void reload() Reloads the current URL. void removeJavascriptInterface(String name) Removes a previously injected Java object from this WebView. boolean requestChildRectangle to be positioned onto the screen(View child, Rect rect, boolean immediate) Called when a child of this group wants a particular rectangle to be positioned onto the screen(View child, Rect rect, boolean immediate) Called when a child of this group wants a particular rectangle to be positioned onto the screen(View child, Rect rect, boolean immediate).
boolean requestFocus(int direction, Rect previouslyFocusedRect) Call this to try to give focus to a specific view or to one of its descendants and give it hints about the direction and a specific view or to one of its descendants and give it hints about the direction and a specific view or to one of its descendants and give it hints about the direction and a specific view or to one of its descendants and give it hints about the direction and a specific view or to one of its descendants.
requestFocusNodeHref(Message hrefMsg) Requests the uRL of the image lest touched by the user. WebBackForwardList restoreState(Bundle inState) Restores the state of this WebView from the given Bundle. void resumeTimers() Resumes all
layout, parsing, and JavaScript timers for all WebViews. void savePassword(String host, String username, String password) This method was deprecated in API level 18. Saving password in WebView used in future versions. WebBackForwardList saveState(Bundle outState) Saves the state of this WebView used in
Activity.onSaveInstanceState(Bundle). void saveWebArchive(String basename, boolean autoname, ValueCallback callback) Saves the current view as a web archive. void saveWebArchive(String basename, boolean autoname, ValueCallback callback) Saves the current view as a web archive.
 setCertificate(SslCertificate certificate certificate) This method was deprecated in API level 17. Calling this function has no useful effect, and will be ignored in future releases. static void setDownloadListener(DownloadListener listener) Registers
the interface to be used when content can not be handled by the rendering engine, and should be downloaded instead. void setFindListener (WebView.FindListener) Registers the listener to be notified as find-on-page operations progress. void setHorizontalScrollbarOverlay (boolean overlay) This method was deprecated in API level 23. This
method has no effect. void setHttpAuthUsernamePassword(String host, String realm, String password) This method was deprecated in API level 26. Use WebViewDatabase#setHttpAuthUsernamePassword instead void setInitialScale(int scaleInPercent) Sets the initial scale for this WebView. void setLayerType(int layerType, Paint
paint) Specifies the type of layer backing this view. void setLayoutParams(ViewGroup.LayoutParams params) Set the layout parameters associated with this view. void setMapTrackballToArrowKeys(boolean setMap) This method was deprecated in API level 17. Only the default case, true, will be supported in a future version. void
 setNetworkAvailable(boolean networkUp) Informs WebView of the network state. void setOverScrollMode(int mode) Set the over-scroll mode for this view. void setPictureListener(WebView.PictureListener) This method was deprecated in API level 15. This method is now obsolete. void setRendererPriorityPolicy(int rendererRequestedPriorityPolicy(int rendererRequestedPriorityPolicy(int rendererPriorityPolicy(int rendererRequestedPriorityPolicy(int rendererRequestedPri
boolean waivedWhenNotVisible) Set the renderer priority policy for this WebView. static void setSafeBrowsingWhitelist(List hosts, ValueCallback callback) Sets the list of hosts (domain names/IP addresses) that are exempt from SafeBrowsing checks. void setScrollBarStyle(int style) Specify the style of the scrollbars. void
setTextClassifier(TextClassifier textClassifier textClassifier textClassifier for this WebView. void setWebContentsDebuggingEnabled(boolean overlay) This method was deprecated in API level 23. This method was deprecated in API level 24. This method was deprecated in API level 25. This method was deprecated in API level 26. This method was deprecated in API level 26. This method was deprecated in API level 26. This method was deprecated in API level 27. This 
enabled) Enables debugging of web contents (HTML / CSS / JavaScript) loaded into any WebViewRenderProcessClient (WebViewClient that will receive various notifications and requests. void setWebViewRenderProcessClient (Executor executor, WebViewRenderProcessClient that will receive various notifications and requests.
webViewRenderProcessClient) Sets the renderer client object associated with this WebView. void setWebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client object associated with this WebViewRenderProcessClient (WebViewRenderProcessClient) Sets the renderer client (WebViewRender
children or descendants of this ViewGroup. boolean showFindDialog(String text, boolean showIme) This method does not work reliably on all Android versions; implementing a custom find dialog using WebView.findAllAsync() provides a more robust solution. static void startSafeBrowsing(Context context, boolean showIme) This method does not work reliably on all Android versions; implementing a custom find dialog using WebView.findAllAsync() provides a more robust solution.
ValueCallback callback) Starts Safe Browsing initialization. void stopLoading() Stops the current load. void zoomBy(float zoomFactor) Performs zoom out in this WebView. boolean zoomOut() Performs zoom out in this WebView. boolean zoomFactor) Performs zoom in in this WebView. boolean zoomFactor) Performs zoom out in this WebView. boolean zoomFactor) Performs zoomFactor out in this WebView. boolean zoomFactor out in this WebView. bool
offset of the horizontal scrollbar's thumb within the horizontal scrollbar's thumb within the vertical scrollbar represents. int compute Vertical ScrollOffset() Compute the
vertical offset of the vertical scrollbar's thumb within the horizontal range int compute Vertical ScrollBange() Compute the vertical scrollbar represents. void dispatchDraw(Canvas canvas) Called by draw to draw the child views. void onAttachedToWindow() This is called when the view is attached to a window. void
onConfigurationChanged(Configuration newConfig) Called when the current configuration of the resources being used by the application have changed (boolean focused, int direction, Rect previouslyFocusedRect) Called by the view system when the focus state of
this view changes. void onMeasure(int widthMeasureSpec, int heightMeasureSpec, int heightMe
an over-scroll operation. void on ScrollChanged(int l, int t, int oldl, int oldt) This is called during layout when the size of this view has changed (view changed (view changed (view changed (view changed view, int visibility)).
Called when the visibility of the view or an ancestor of the view or an ancestor of the view has change its visibility (ArrayList outChildren) Adds the children of this
 View relevant for accessibility to the given list as output, void addExtraDataToAccessibilityNodeInfo (AccessibilityNodeInfo based on an explicit request for the additional data, void addFocusables(ArrayList views, int direction, int focusableMode) Adds any
focusable views that are descendants of this view (possibly including this view if it is focusable itself) to views. void addKeyboardNavigationCluster roots that are descendants of this view (possibly including this view if it is a cluster root itself) to views. boolean
addStatesFromChildren() Returns whether this ViewGroup's drawable states also include its children's drawable states. void addView(View child, ViewGroup.LayoutParams params) Adds a children's drawable states.
view with the specified layout parameters. void addView(View child, int index, ViewGroup.LayoutParams params) Adds a child view with the specified layout parameters. void addView(View child, int index, ViewGroup.LayoutParams params) Adds a child view with the specified layout parameters. void addView(View child, int index, ViewGroup.LayoutParams params) Adds a child view.
attachLayoutAnimationParameters(View child, ViewGroup.LayoutParams params, int index, int count) Subclasses should override this method to set layout animation parameters on the supplied child. ViewGroup. LayoutParams params, int index, int count) Subclasses should override this method to set layout animation parameters on the supplied child.
child) Change the z order of the child so it's on top of all other children after the first layout. boolean canAnimate() Indicates whether the view group has the ability to animate its children after the first layout. boolean canAnimate() is true, refreshes this group's
drawable state (to include the states from its children). void childHasTransientStateChanged(View child, boolean childHasTransientState) Called when a child view has changed whether or not it is tracking transient state. void cleanupLayoutState(View child) Prevents the specified child to be laid out during the next layout pass. void
clearChildFocus(View child) Called when a child of this parent is giving up focus void clearDisappearingChildren() Removes any pending animations for views from the parent. void debug(int depth) void detachAllViewsFromParent() Detaches all views from the parent. void
view or another view in its subtree. boolean dispatchConfiguration changed (Configuration newConfig) Dispatch a notification about a resource configuration change down the view hierarchy. void
dispatchCreateViewTranslationRequest(Map viewIds, int[] supportedFormats, TranslationRequests (Map, int[] supportedFormats, TranslationRequests (M
 TranslationCapability, List) for all the child views. void dispatchDisplayHint(int hint) Dispatch a hint about whether this view is displayed. boolean dispatchDraw(Canvas canvas) Called by draw to draw the child views. void
dispatchDrawableHotspotChanged(float x, float y) Dispatches drawable hotspot changes to child views that meet at least one of the following criteria: void dispatchFreezeSelfOnly(SparseArray) freeze()} to only this view, not to its children. boolean
dispatchGenericFocusedEvent(MotionEvent event) Dispatch a generic motion event to the currently focused view. boolean dispatchHoverEvent(MotionEvent event) Dispatch a generic motion event to the view under the first pointer. boolean dispatchHoverEvent(MotionEvent event) Dispatch a generic motion event to the currently focused view.
dispatchKeyEvent (KeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the next view of the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the focus path. Boolean dispatch a key event to the focus path. Boolean
dispatchPointerCaptureChanged(boolean hasCapture) void dispatchProvideAutofillStructure (ViewStructure structure is being created as part of an autofill request. This implementation adds in all child views of the view group, in
 addition to calling the default View implementation. void dispatchProvideStructure(ViewStructure structure) Dispatch creation of ViewStructure down the hierarchy. void dispatchRestoreInstanceState(SparseArray container) Called by restoreHierarchyState(android.util.SparseArray) to retrieve the state for this view and its children. void
dispatchSaveInstanceState(SparseArray container) Called by saveHierarchyState(android.util.SparseArray) to store the state for this view and its children. void dispatchScrollCaptureSearch (Rect localVisibleRect, Point windowOffset, Consumer targets) Handle the scroll capture search request by checking this view if applicable, then to each children.
view. void dispatchSetActivated (boolean activated) Dispatch setPressed to all of this View's children. void dispatchSetPressed (boolean pressed) Dispatch setPressed to all of this View's children. void dispatchSetPressed (boolean pressed) Dispatch setPressed to all of this View's children. void dispatchSetPressed (boolean pressed) Dispatch setPressed (boolean pressed) Di
method is deprecated. Use WindowInsets#isVisible(int) to find out about system bar visibilities by setting a OnApplyWindowInsetsListener on this view. void dispatchThawSelfOnly(SparseArray) to only this view, not to its children. boolean
dispatchTouchEvent(MotionEvent ev) Pass the touch screen motion event down to the target view, or this view if it is the target view, boolean dispatchUnhandledMove(View focused, int direction) This method is the last chance for the focused view.
and its ancestors to respond to an arrow key, void dispatch view dispatc
dispatchWindowInsetsAnimationEnd(WindowInsetsAnimation ends. void dispatchWindowInsetsAnimationPrepare(WindowInsetsAnimation ends. void dispatchWindowInsetsAnimationPrepare(WindowInsetsAnimation ends. void dispatchWindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(Windo
WindowInsetsAnimation. Callback#onProgress(WindowInsets Animation) when WindowInsets animation is being prepared. WindowInsets Animation. Callback#onProgress(WindowInsets Animation) when WindowInsets animation makes
progress. WindowInsetsAnimation. Bounds dispatchWindowInsetsAnimation. Bounds dispatchWindowInsetsAnimation. Bounds bounds bounds bounds dispatchWindowInsetsAnimation. Bounds bounds dispatchWindowInsetsAnimation. Bounds di
This method is deprecated. SystemUiVisibility flags are deprecated. Use WindowInsetsController instead. void dispatchWindowVisibility Changed(int visibility Changed(int visibility Changed(int visibility Changed(int visibility)). Dispatch a window visibility Changed(int visibility).
drawableStateChanged() This function is called whenever the state of the view changes in such a way that it impacts the state of drawables being shown. void endViewTransition(View view). View findFocus() Find the view in the hierarchy rooted at
this view that currently has focus. On BackInvoked Dispatcher find On BackInvoked Dispatcher. void find View shild, View requester) Walk up the View hierarchy to find the nearest On BackInvoked Dispatcher. void find Views With Text (Array List out Views, Char Sequence text, int flags) Finds the Views that contain given text. View focus Search (View focus ed.)
int direction) Find the nearest view in the specified direction that wants to take focus. void focusable View Available. boolean gather Transparent Region (Region region) This is used by the ViewRoot to perform an optimization when the view hierarchy contains one or several
SurfaceView. ViewGroup.LayoutParams generateDefaultLayoutParams generateDefaultLayoutParams generateLayoutParams p) Returns a set of default layout parameters. ViewGroup.LayoutParams generateLayoutParams generateLayoutParams generateLayoutParams p) Returns a set of default layout parameters. ViewGroup.LayoutParams generateLayoutParams generateLayoutParams generateLayoutParams generateLayoutParams p) Returns a set of default layout parameters.
a safe set of layout parameters based on the supplied layout params. CharSequence getAccessibility ClassName() Returns the view at the specified position in the group. int getChildCount() Returns the number of children in the group. int
getChildDrawingOrder(int childCount, int drawingPosition) Converts drawing order position to container position to container position. static int getChildDrawingOrder(int drawingPosition) Converts drawing order position to container position. static int getChildDrawingOrder(int drawingPosition) Converts drawing order position to container position.
the MeasureSpec to pass to a particular child, If set, returning a boolean getChildStaticTransformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean to indicate whether a static transformation (View child, If set, returning a boolean transformation (View child, If set, retur
defined in terms of a child view's coordinates. boolean getClipChildren() Returns whether this group's children are clipped to their bounds before drawing, and resize (but not clip) any EdgeEffect to the padded region, if padding is present. int
getDescendantFocusability() Gets the descendant focused Child() Returns the focused Child() Returns the focused child of this view, if any. LayoutAnimation.Animation.Animation.Animation.Animation.Controller getLayoutAnimation.Controller getLayoutAnimation.Controller used to animate the group's children. Animation.Animation.Controller getLayoutAnimation.Controller used to animate the group's children. Animation.Animation.Controller getLayoutAnimation.Controller getLayoutAnimation.Controller used to animate the group's children.
animation listener to which layout animation events are sent. int getLayoutMode() Returns the basis of alignment during layout operations on this ViewGroup: either LAYOUT_MODE_CLIP_BOUNDS or LAYOUT_MODE_OPTICAL_BOUNDS. LayoutTransition getLayoutTransition() Gets the LayoutTransition object for this ViewGroup. int
getNestedScrollAxes() Return the current axes of nested scrolling for this ViewGroupOverlay getOverlay() Returns the ViewGroupOverlay for this view group, creating it if it does not yet exist. int getPersistentDrawingCache() This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the
 introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, View.setLayerType(int,
android.graphics.Paint) handles this with hardware rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have
compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. boolean getTouchscreenBlocksFocus() Check whether this ViewGroup should ignore focus requests for itself and its
children. boolean hasFocus() Returns true if this view has or contains focus boolean hasTransientState() Indicates whether the view is currently tracking transient state that the app should not need to concern itself with saving and restoring, but that the framework should take special note to preserve when possible. int indexOfChild(View child)
Returns the position in the group of the specified child view. final void invalidateChild(View child, Rect dirty) This method is deprecated. Use onDescendantInvalidateChild(view.View, android.view.View, android.view.View) instead to observe updates to draw state in descendants. ViewParent invalidateChild(InParent(int[] location, Rect dirty) This method is
deprecated. Use onDescendantInvalidated(android.view.View, android.view.View) instead to observe updates to draw state in descendants. boolean isAlwaysDrawnWithCacheEnabled() This method was deprecated in API level 23. As of Build.VERSION CODES.M, this property is ignored. Child views may no longer have their caching behavior disable.
by parents. boolean isAnimationCacheEnabled() This method was deprecated in API level 23. As of Build.VERSION CODES.M, this property is ignored. Caching behavior of children may be controlled through View#setLayerType(int, Paint). boolean isChildrenDrawingOrderEnabled() Indicates whether the ViewGroup is drawing its children in the
order defined by getChildDrawingOrder(int, int). boolean isChildrenDrawnWithCacheEnabled() This method was deprecated in API level 23. As of Build.VERSION CODES.M, this property is ignored. Child views may no longer be forced to cache their rendering state by their parents. Use View#setLayerType(int, Paint) on individual Views instead.
boolean isLayoutSuppressed() Returns whether layout calls on this container are currently being suppressed, due to an earlier call to suppressLayout(boolean). Boolean isTransitionGroup() Returns true if this ViewGroup can be split to multiple children.
ViewGroup should be considered as a single entity for removal when executing an Activity transition. void jumpDrawablesToCurrentState() call Drawable objects associated with this view. final void layout (int l, int t, int t, int t, int t, int t) Assign a size and position to a view and all of its descendants This is the second
phase of the layout mechanism. void measureChild(View child, int parentWidthMeasureSpec, int parentWidthMeasureSpe
 widthUsed, int parentHeightMeasureSpec, int heightUsed) Ask one of the children of this view to measure themselves, taking into account both the MeasureSpec, int heightMeasureSpec, int heightMeasureSpec, int heightUsed) Ask all of the children of this view to measure themselves, taking into account both the MeasureSpec, int heightMeasureSpec, int heightMeasureSpec,
into account both the MeasureSpec requirements for this view and its padding. void notifySubtreeAccessibilityState of one of its descendants has changed and that the structure of the subtree is different. final void
offsetDescendantRectToMyCoords(View descendant, Rect rect) Offset a rectangle that is in a descendant, Rect rect) Offset a rectangle that is in our coordinate space into our coordinate space into our coordinate space into an ancestor's coordinate space. void onAttachedToWindow() This is called
when the view is attached to a window. int[] on Create Drawable State (int extraSpace) Generate the new Drawable state for this view. void on Descendant Invalidated (View child, View target) The target View has been invalidated, or has had a drawing property changed that requires the hierarchy to re-render. If you override this method you must call
through to the superclass implementation. void on Detached From Window() This is called when the view is detached from a window. boolean on Intercept Hover Event (Motion Event event) Implement this method to intercept Hover events before they are handled by child views. boolean on Intercept Hover event (Motion Event event) Implement this method to intercept Hover events before they are handled by child views.
intercept all touch screen motion events. abstract void on Layout (boolean changed, int l, int r, int b) Called from layout when this view should assign a size and position to each of its children. boolean onNestedFling(View target, float velocityX, float velocityY, boolean consumed) Request a fling from a nested scroll. boolean
onNestedPreFling(View target, float velocityX, float velo
super.onNestedPrePerformAccessibilityAction void onNestedPreScroll(View target, int dx, int dy, int[] consumed, int dyUnconsumed, int dyUn
void on Nested Scroll Accepted (View child, View target, int axes) React to the successful claiming of a nested scroll operation. boolean on Request Focus on. boolean on Request Focus on Boolean on Request Focus on Called Call View and Called Call
when a child has requested sending an AccessibilityEvent and gives an opportunity to its parent to augment the event, or null if it doesn't specify the icon. boolean onStartNestedScroll(View child, View target, int nestedScrollAxes)
React to a descendant view initiating a nestable scroll operation, claiming the nested scroll operation ending. void on View Removed (View child) React to a nested scroll operation ending. void on View Removed (View child) React to a nested scroll operation ending.
from this ViewGroup. void recomputeViewAttributes(View child) Tell view hierarchy that the global view attributes need to be re-evaluated. void removeAllViewsInLayout() Called by a ViewGroup subclass to remove child views from itself, when it must first know its
size on screen before it can calculate how many child views it will render. void removeDetachedView(View view) Note: do not invoke this method from View.draw(android.graphics.Canvas), View.onDraw(android.graphics.Canvas),
dispatchDraw(android.graphics.Canvas) or any related method. void removeViewAt(int index) Removes the view at the specified position in the group. void removeViewsInLayout(View view) Removes the view at the specified position in the group. void removeViewsInLayout(int start, int count) Removes the view at the specified position in the group.
int count) Removes a range of views during layout. void requestChildFocus(View child, View focused) Called when a child of this group wants a particular rectangle to be positioned onto the screen. void
requestDisallowInterceptTouchEvent(boolean disallowIntercept) Called when a child does not want this parent and its ancestors to intercept touch events with ViewGroup#onIntercept touch events with ViewGroup
and give it hints about the direction and a specific rectangle that the focus is coming from. Looks for a view to give focus to respecting the setting specified by getDescendantFocusabilityEvent (View child, AccessibilityEvent event) Called by a child to request from its parent to send an AccessibilityEvent. void
requestTransparentRegion(View child) Called when a child wants the view hierarchy to gather and report transparent regions to the window compositor. boolean restoreDefaultFocus () Gives focus to the default-focus view in the view hierarchy that has this view as a root. void scheduleLayoutAnimation() Schedules the layout animation to be played
after the next layout pass of this view group. void setAddStatesFromChildren(boolean addsStates) Sets whether this ViewGroup's drawable states also include its children's drawable states. void setAdwaysDrawnWithCacheEnabled(boolean addsStates) Sets whether this ViewGroup's drawable states.
Child views may no longer have their caching behavior disabled by parents, void setAnimationCacheEnabled(boolean enabled) This method was deprecated in API level 23. As of Build.VERSION CODES.M, this property is ignored. Caching behavior of children may be controlled through View#setLayerType(int, Paint). void
setChildrenDrawingCacheEnabled(boolean enabled) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-accelerated in API 11. With hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated was largely made obsolete was largely made obsolete was largely made obsolete with the introduction of hardware-accelerated was largely made obsolete was largely made obsolete was largely made obsolete.
to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, View.setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from
either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit
testing the PixelCopy API is recommended. void setChildrenDrawingOrderEnabled(boolean enabled) Tells the ViewGroup whether to draw its childrenDrawnWithCacheEnabled(boolean enabled) This method was deprecated in API level 23. As of
Build. VERSION CODES.M, this property is ignored. Child views may no longer be forced to cache their rendering state by their parents. Use View#setLayerType(int, Paint) on individual Views instead. void setClipChildren(boolean clipChildren) By default, children are clipped to their bounds before drawing. void setClipToPadding(boolean clipChildren) are clipped to their bounds before drawing.
clipToPadding) Sets whether this ViewGroup will clip its children to its padding and resize (but not clip) any EdgeEffect to the padded region, if padding is present. void setLayoutAnimation(LayoutAnimationController) Sets the layout
animation controller used to animate the group's children after the first layout. void setLayoutMode (int layout animation Listener animation listener to which layout animation events must be sent. void setLayoutMode (int layout animation listener animation listener animation events must be sent. void setLayoutMode) Sets the basis of alignment during the layout of this ViewGroup. void setLayoutMode (int layout animation listener animation listene
setLayoutTransition(LayoutTransition) Sets the LayoutTransition object for this ViewGroup. void setMotionEvents to multiple children during touch event dispatch. void setOnHierarchyChangeListener(ViewGroup.OnHierarchyChangeListener listener) Register a
callback to be invoked when a child is added to or removed from this view. void setPersistentDrawingCache(int drawingCache(int drawingCache(int drawingCache) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache
layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, View.setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of
the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call View.draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time
shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void setStaticTransformations on children; this causes getChildStaticTransformation(android.view.View, View, this ViewGroup supports static transformations on children; this causes getChildStaticTransformation(android.view.View, View, this ViewGroup supports static transformations).
android.view.animation.Transformation) to be invoked when a child is drawn. void setTouchscreenBlocksFocus (boolean touchscreenBlocksFocus) Set whether this ViewGroup should ignore focus requests for itself and its children. void setTransitionGroup (boolean isTransitionGroup) Changes whether or not this ViewGroup should be treated as a
single entity during Activity Transitions. void setWindowInsetsAnimation. Callback (WindowInsetsAnimation. Callback to be notified about animations of windowInsetsAnimation. Callback to be notified about animation.
this ViewGroup. boolean showContextMenuForChild(View originalView, float x, float y) Shows the context menu for the specified view or its ancestors anchored to the specified view or its ancestors. ActionMode
startActionModeForChild(View originalView, ActionMode startActionMode efor the specified view with the default type ActionMode#TYPE PRIMARY. void
startLayoutAnimation() Runs the layout animation. void startViewTransition(View view) This method tells the ViewGroup draws its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its parent, should be kept around (re-displayed when the ViewGroup draws its children) even if it is removed from its parent, should be kept around (re-displayed when the ViewGroup draws its parent, should be kept around (re-displayed when the ViewGroup draws its parent).
 Tells this ViewGroup to suppress all layout() calls until layout suppression is disabled with a later call to suppressLayout(False). void updateView. View view, ViewGroup. Layout(View view, ViewGroup. Layout(View view). This View relevant for accessibility to the children for Accessibility (ArrayList out Children).
given list as output. void addExtraDataToAccessibilityNodeInfo (AccessibilityNodeInfo info, String extraDataKey, Bundle arguments) Adds extra data to an AccessibilityNodeInfo based on an explicit request for the additional data. void addFocusables(ArrayList views, int direction) Add any focusable views that are descendants of this view (possibly
including this view if it is focusable itself) to views, void addFocusable itself) to views, int direction, int focusable views, int direction views, int direction views, int direction views, int direction, int focusable views that are descendants of this view (possibly including this view if it is focusable views, int direction, int focusable views, int direction, int focusable views, int direction views
roots that are descendants of this view (possibly including this view if it is a cluster root itself) to views. void addOnAttachStateChangeListener (View.OnLayoutChangeListener listener) Add a listener that will be called when the
bounds of the view change due to layout processing. void addOnUnhandledKeyEventListener (View.OnUnhandledKeyEventListener listener which will receive unhandledKeyEventListener listener listener which will receive unhandledKeyEventListener listener listener which will receive unhandledKeyEventListener listener which will receive unhandledKeyEventListener listener listene
views. ViewPropertyAnimator animate() This method returns a ViewPropertyAnimator object, which can be used to animate specific properties on this View. void announceForAccessibilityEvent to suggest that an accessibility service
announce the specified text to its users. void autofill(AutofillValue value) Automatically fills the content of this view with the value. void autofill(SparseArray values) Automatically fills the content of the virtual children within this view. boolean awakenScrollBars(int startDelay, boolean invalidate) Trigger the scrollbars to draw. boolean
awakenScrollBars(int startDelay) Trigger the scrollbars to draw. boolean awakenScrollBars() Trigger the scrollbars to draw. void bringToFront() Change the view's z order in the tree, so it's on top of other sibling views. void bringToFront() Change the view's z order in the tree, so it's on top of other sibling views. void bringToFront() Change the view's z order in the tree, so it's on top of other sibling views.
obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations,
setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are
discouraged and have compatibility issues with hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended, void buildDrawingCache() This method was deprecated in API level 28. The view drawing
cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha
animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages
are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void buildLayer() Forces this view's layer to be created and this view to be rendered into
its layer. boolean callOnClick() Directly call any attached OnClickListener. boolean canResolveTextAlignment () Check if text alignment resolution can be done. boolean canResolveTextDirection() Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution can be done. boolean canResolveTextAlignment () Check if text direction resolution r
canScrollHorizontally(int direction) Check if this view can be scrolled horizontally in a certain direction. boolean canScrollVertically(int direction) Check if this view can be scrolled vertically in a certain direction. Final void cancel from the control of this view can be scrolled horizontally in a certain direction.
void cancel Pending Input Events() Cancel any deferred high-level input events that were previously posted to the event gueue. boolean check Input Connection target is trying to make a call on the manager, void clear Animation() Cancels any
animations for this view. void clearFocus() Called when this view wants to give up focus. void clearViewTranslationCallback from this view state). Int compute HorizontalScrollExtent() Compute the
horizontal extent of the horizontal scrollbar's thumb within the horizontal scrollbar's thumb within the horizontal scrollbar represents. void compute Scroll() Called by a
parent to request that a child update its values for mScrollY and mScrollY if necessary. WindowInsets (WindowInsets that should be consumed by this view and the ones that should propagate to those under it. int computeVerticalScrollExtent() Compute the vertical extent of the
vertical scrollbar's thumb within the vertical scrollbar's thumb within the vertical scrollbar represents. AccessibilityNodeInfo createAccessibilityNodeInfo() Returns an
AccessibilityNodeInfo representing this view from the point of view of an AccessibilityService, void destroyDrawingCache() This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated
rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android, graphics, Paint) handles this with hardware-acceleration.
rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and have compatibility issues with hardware-only rendering
features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. WindowInsets dispatchApplyWindowInsets insets) Request to apply the given window insets to this view or another view in its subtree. boolean
dispatchCapturedPointerEvent(MotionEvent event) Pass a captured pointer event down to the focused view, void dispatchConfiguration changed(Configuration newConfig) Dispatch a notification about a resource configuration changed (Configuration newConfig) Dispatch a notification about a resource configuration newConfiguration newC
TranslationCapability capability, List requests Dispatch to collect the ViewTranslation purpose by traversing the hierarchy when the app requests if this View is enabled
and has a drag event listener. void dispatchDraw(Canvas canvas) Called by draw to draw the child views, void dispatchDraw(Bloat x, float y) Dispatch on FinishTemporaryDetach() to this View and its direct children if this is a
container View. boolean dispatchGenericFocusedEvent(MotionEvent event) Dispatch a generic motion event to the currently focused view. boolean dispatchGenericFocusedEvent(MotionEvent event) Dispatch a generic motion event to the view under the first
pointer. boolean dispatch HoverEvent (MotionEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path. boolean dispatchKeyEvent event) Dispatch a key event to the next view on the focus path.
dispatchKeyShortcutEvent(KeyEvent event) Dispatch a fling to a nested scrolling parent before it is processed by this view.
boolean dispatchNestedPrePerformAccessibilityAction (int action, Bundle arguments) Report an accessibility action to this view's parents for delegated processing. boolean dispatchNestedPreScroll (int dx, int dy, int[] consumed, int[] offsetInWindow) Dispatch one step of a nested scroll in progress before this view consumes any portion of it. boolean
dispatchNestedScroll(int dxConsumed, int dyUnconsumed, int dyUncon
then to its children for adding their text content to the event. void dispatchProvideStructure (ViewStructure structure is being created as part of an autofill request. void dispatchProvideStructure(ViewStructure structure)
Dispatch creation of ViewStructure down the hierarchy. void dispatchRestoreInstanceState(SparseArray) to retrieve the state for this view and its children. void dispatchSaveInstanceState(SparseArray) to retrieve the state for this view and its children.
state for this view and its children. void dispatchScrollCaptureSearch(Rect localVisibleRect, Point windowOffset, Consumer targets) Dispatch a scroll capture search request down the view hierarchy. void dispatchSetPressed(boolean pressed) Dispatch
setPressed to all of this View's children, void dispatchSetSelected(boolean selected) Dispatch setSelected to all of this View's children, void dispatchSystemUiVisibilityChanged(int visibilityChanged(int visibility) This method was
deprecated in API level 30. Use WindowInsets#isVisible(int) to find out about system bar visibilities by setting a OnApplyWindowInsetsListener on this view, fit is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target view, or this view if it is the target 
Pass a trackball motion event down to the focused view. boolean dispatchUnhandledMove(View focused, int direction) This method is the last chance for the focused view and its ancestors to respond to an arrow key. void dispatchVisibilityChanged(View changedView, int visibility) Dispatch a view visibility change down the view hierarchy. void
dispatchWindowFocusChanged(boolean hasFocus) Called when the window containing this view gains or loses windowInsetsAnimation. Callback#onEnd(WindowInsetsAnimation) when Window Insets animation ends. void
dispatchWindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowInsetsAnimationPrepare(WindowIn
WindowInsetsAnimation.Callback#onProgress(WindowInsetsAnimation animation animation.Bounds bounds) Dispatches WindowInsetsAnimation.Callback#onStart(WindowInsetsAnimation.Bounds dispatchWindowInsetsAnimation.Bounds dispatchWindowInse
Bounds) when Window Insets animation is started, void dispatch Window Visibility Changed (int visible) This method was deprecated in API level 30. System UiVisibility Changed (int visibility Changed
void draw(Canvas canvas) Manually render this view (and all of its children) to the given Canvas. void drawableHotspotChanged(float x, float y) This function is called whenever the
state of the view changes in such a way that it impacts the state of drawables being shown. View findFocus() Find the view hierarchy to find the nearest OnBackInvokedDispatcher findOnBackInvokedDispatcher findOnBackInvokedDisp
Finds the first descendant view with the given ID, the view itself if the ID matches getId(), or null if the ID is invalid (< 0) or there is no matching view in the hierarchy. final T findViewWithText(ArrayList outViews, CharSequence searched, int flags) Finds the Views that
contain given text. boolean fitSystemWindows(Rect insets) This method was deprecated in API level 20. As of API 20 use dispatchApplyWindowInsets(android.view.WindowInsets) or use
setOnApplyWindowInsetsListener(android.view.View.OnApplyWindowInsetsListener) to implement handling their own insets. View focusSearch(int direction) Find the nearest view in the specified direction that can take focus.
view (see hasOverlappingRendering() for more details on this behavior). void forceLayout() Forces this view to be laid out during the next layout pass. boolean gatherTransparentRegion(Region region) This is used by the ViewRoot to perform an optimization when the view hierarchy contains one or several SurfaceView. void
generateDisplayHash(String hashAlgorithm, Rect bounds, Executor executor, DisplayHashResultCallback callback) Called to generate a value suitable for use in setId(int). CharSequence getAccessibilityClassName() Return the class name of this object to be used for accessibility
purposes. View. Accessibility Delegate get Accessibility Delegate () Returns the delegate for implementing accessibility Node Provider () Gets the provider for managing a virtual view hierarchy rooted at this View.
and reported to AccessibilityServices that explore the window content. CharSequence getAccessibilityTraversalAfter() Gets the id of a view before which
```

```
this one is visited in accessibility traversal. float getAlpha() The opacity of the view. Animation getAnimation () Get the animation matrix of the view. IBinder getApplicationWindowToken() Retrieve a unique token identifying the top-level "real" window
of the window that this view is attached to. int[] getAttribute ResolutionStack(int attribute values for this View. Map getAttribute value was set. String[]
getAutofillHints() Gets the hints that help an AutofillService determine how to autofill getAutofillId() Gets the unique, logical identifier of this view, so an AutofillService can create the proper AutofillValue when
autofilling the view. AutofillValue getBackground() Gets the View's current autofill value. Drawable getBackgroundTintList() Return the blending mode used to apply the tint to the background drawable, if specified. ColorStateList getBackgroundTintList() Return the tint applied to
the background drawable, if specified. PorterDuff.Mode getBackgroundTintMode() Return the blending mode used to apply the tint to the background drawable, if specified. int getBaseline() Return the offset of the widget's text baseline from the widget's text baseline from the widget's top boundary. final int getBaseline() Return the offset of the widget's text baseline from the wi
getBottomFadingEdgeStrength() Returns the strength, or intensity, of the bottom faded edge. int getBottomPaddingOffset() Amount by which to extend the bottom fading region. float getCameraDistance() Gets the distance along the Z axis from the camera to this view. boolean getClipBounds(Rect outRect) Populates an output rectangle with the clip
 bounds of the view, returning true if successful or false if the view's clip bounds are null. Rect getClipBounds() Returns a copy of the current clipBounds. final boolean getContentCaptureSession () Gets the session used to notify
content capture events. CharSequence getContextMenuInfo() Returns the View's contextMenuInfo () Views should implement this if they have extra information extra information.
to associate with the context menu. final boolean getDefaultFocusHighlightEnabled() Returns whether this View should use a default focus highlight when it gets focused but doesn't have Rattr.state focused defined in its background, static int getDefaultSize(int size, int measureSpec) Utility to return a default size. Display getDisplay() Gets the
logical display to which the view's window has been attached. final int[] getDrawableState() Return an array of resource IDs of the drawable states representing the current state of the view. Bitmap getDrawableState() Return an array of resource IDs of the drawable states representing the current state of the view. Bitmap getDrawableState() This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware
accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this
with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only
rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. Bitmap getDrawingCache(boolean autoScale) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the
introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, and updating the layer).
android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have
compatibility issues with hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int getDrawingCacheBackgroundColor() This method was deprecated in API level 28. The view drawing cache was
largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layers are useful, such as for alpha animations
setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are
discouraged and have compatibility issues with hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. int getDrawingCacheQuality() This method was deprecated in API level 28. The view drawing
cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha
animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages
are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void getDrawingRect(Rect outRect) Return the visible drawing bounds of your view. long
getDrawingTime() Return the time at which the drawing of the view hierarchy started. float getElevation() The base elevation of this view relative to its parent, in pixels. int getExplicitStyle() Returns the resource ID for the style specified using style="..." in the AttributeSet's backing XML element or Resources#ID_NULL otherwise if not specified or
otherwise not applicable. boolean getFilterTouchesWhenObscured() Gets whether the framework should discard touches when the view's window is obscured by another visible windows (boolean). int getFocusable() Returns the focusable setting for this
view. ArrayList getFocusables(int direction) Find and return all focusable views that are descendants of this view, possibly including this view if it is focusable itself. void getFocusedRect(Rect r) When a view has focus and the user navigates away from it, the next view is searched for starting from the rectangle filled in by this method. Drawable
getForeground() Returns the drawable used as the foregroundTintBlendMode() Return the blending mode used to apply the tint to the foregroundTintBlendMode getForegroundTintBlendMode ge
 foreground drawable, if specified. PorterDuff.Mode getForegroundTintMode() Return the blending mode used to apply the tint to the coordinates of the non-clipped area of this view in the coordinate space of the view's root view. boolean
getGlobalVisibleRect(Rect r, Point globalOffset) Sets r to the coordinates of the non-clipped area of this view in the coordinate space of the view's root view. Handler getHandler() final boolean getHasOverlappingRendering() Returns the view's root view. Handler getHandler() final boolean getHasOverlappingRendering() Returns the view's root view.
getHitRect(Rect outRect) Hit rectangle in parent's coordinates int getHorizontalScrollbarHeight() Returns the horizontal faded edges used to indicate that more content in this view is visible. int getHorizontalScrollbarHeight() Returns the horizontal faded edges used to indicate that more content in this view is visible. int getHorizontalScrollbarHeight() Returns the horizontal faded edges used to indicate that more content in this view is visible.
currently configured Drawable for the thumb of the horizontal scroll bar if it exists, null otherwise. Drawable for the track of the horizontal scroll bar if it exists, null otherwise. Drawable for the track of the horizontal scroll bar if it exists, null otherwise. Drawable for the track of the horizontal scroll bar if it exists, null otherwise. Drawable for the track of the horizontal scroll bar if it exists, null otherwise. Drawable for the track of the horizontal scroll bar if it exists, null otherwise.
for determining whether this View is important for accessibility. int getImportantForAutofill() Gets the mode for determining whether this view is important for content capture. boolean getKeepScreenOn() Returns whether the screen
should remain on, corresponding to the current value of KEEP SCREEN ON. KeyEvent. DispatcherState () Return the global KeyEvent. DispatcherState for this view serves as a label for accessibility purposes. int getLayerType() Indicates what type of layer is
currently associated with this view. int getLayoutDirection() Returns the resolved layout Direction for this view. Final int getLeft() Left position of this view. Final int getLeft() Left position of this view. Final int getLayoutDirection() Returns the strength, or intensity, of the left position of this view. Final int getLayoutDirection() Returns the resolved layout Direction() Returns the strength, or intensity, of the left position of this view.
faded edge. int getLeftPaddingOffset() Amount by which to extend the left fading region. final boolean getLocationInSurface(int[] location) Gets the coordinates of this view in the coordinate space of the Surface that
contains the view. void getLocationInWindow(int[] outLocation) Gets the coordinates of this view in the coordinate space of the window that contains the view, irrespective of system decorations. void getLocationOnScreen(int[] outLocation) Gets the coordinate space of the window that contains the view, irrespective of system decorations.
decorations and whether the system is in multi-window mode. Matrix getMatrix() The transform matrix of this view, which is calculated based on the current rotation, scale, and pivot properties. final int getMeasuredHeight() Like getMeasuredHeightAndState(), but only returns the raw height component (that is the result is masked by
MEASURED_SIZE_MASK). final int getMeasuredHeightAndState() Return the full height measuredHeightAndState() Return only the state bits of getMeasuredWidthAndState() and getMeasuredHeightAndState(), combined into one integer. final int
getMeasuredWidth() Like getMeasuredWidthAndState(), but only returns the raw width component (that is the result is masked by MEASURED SIZE MASK). final int getMeasuredWidthAndState() Returns the full width measurement information for this view as computed by the most recent call to measure(int, int). int getMeasuredWidthAndState() Returns the full width measurement information for this view as computed by the most recent call to measure(int, int). int getMeasuredWidthAndState() Returns the full width measurement information for this view as computed by the most recent call to measure(int, int).
the minimum height of the view. int getMextFocus Down. int getNextFocus ForwardId() Gets the id of the view to use when the next focus is FOCUS DOWN. int getNextFocusForwardId() Gets the id of the view to use
when the next focus is FOCUS_FORWARD. int getNextFocusLeftId() Gets the id of the view to use when the next focus is FOCUS_RIGHT. int getNextFocusChangeListeners focus is FOCUS_RIGHT. Int getNextFocus ChangeListeners focus ChangeListeners foc
getOnFocusChangeListener() Returns the focus-change callback registered for this view. int getOutlineProvider of the view, which generates the Outline that defines the shape of the shadow it casts, and enables outline clipping. int
 getOutlineSpotShadowColor() int getOverScrollMode() Returns the over-scroll mode for this view. ViewOverlay getOverlay for this view, creating it if it does not yet exist. int getPaddingBottom() Returns the overlay for this view depending on its resolved layout
direction. int getPaddingLeft() Returns the left padding of this view. int getPaddingTop() Returns the top padding of this view. final ViewParent getParent() Gets the parent of this view depending on its resolved layout direction. int getPaddingTop() Returns the top padding of this view. final ViewParent getParent() Gets the parent of this view depending on its resolved layout direction.
 ViewParent getParentForAccessibility() Gets the parent for accessibility purposes. float getPivotX() The x location of the point around which the view is rotated and scaled. PointerIcon getPointerIcon() Gets the point around which the view is rotated and scaled. PointerIcon getPointerIcon getPointerIcon for the current view.
getPreferKeepClearRects() String[] getReceiveContentListener(String[], OnReceiveContentListener). Resources getResources () Returns the resources associated with this view, final boolean
getRevealOnFocusHint() Returns this view's preference for reveal behavior when it gains focus. final int getRight() Right position of this view relative to its parent. float getRightFadingEdgeStrength() Returns the strength, or intensity, of the right faded edge. int getRightFadingOffset() Amount by which to extend the right fading region.
AttachedSurfaceControl getRootSurfaceControl () The AttachedSurfaceControl itself is not a View, it is just the interface to the windowInsets () Provide original WindowInsets that are
dispatched to the view hierarchy. float getRotation() The degrees that the view is rotated around the pivot point. float getRotationX() The degrees that the view is rotated around the pivot point. float getRotationX() The degrees that the view is rotated around the pivot point.
 amount that the view is scaled in x around the pivot point, as a proportion of the view's unscaled width. float getScrollBarFadeDuration()
 Returns the scrollbar fade duration. int getScrollBarSize() Returns the scrollbar size. int getScrollBarStyle() Returns the current scrollbar style. int getScrollIndicators the current scroll capture Hint () Returns the current scrollbar style.
left position of this view. final int getScrollY() Return the scrolled top position of this view int getSourceLayoutResId() A View can be inflated from an XML layout. final CharSequence getStateDescription()
Returns the View's state description. StateListAnimator () Returns the suggested minimum Width () Returns the suggested minimum Width () Returns the suggested minimum width that the view should use. List
getSystemGestureExclusionRects() Retrieve the list of areas within this view's post-layout coordinate space where the systemUiVisibility flags are deprecated. Use WindowInsetsController instead. Object
getTag() Returns this view's tag. Object getTag(int key) Returns the tag associated with this view and the specified key. int getTextAlignment() Returns the view's tag. Object getTag(int key) Returns the tag associated with this view relativestion() Return the resolved text direction. CharSequence getTooltipText() Returns the view's tag. Object getTag(int key) Returns the view and view a
to its parent. float getTopFadingEdgeStrength() Returns the strength, or intensity, of the top faded edge. int getTopPaddingOffset() Amount by which to extend the top fading region. TouchDelegate getTouchDelegate for this View. ArrayList getTouchables() Find and return all touchable views that are descendants of this
view, possibly including this view if it is touchable itself. float getTransitionAlpha() This property is intended only for use by the Fade transition, which animates it to produce a visual translucency that does not side-effect (or get affected by) the real alpha property. String getTransitionName() Returns the name of the View to be used to identify Views
in Transitions. float getTranslationX() The horizontal location of this view relative to its left position. float getTranslationY() The vertical location of this view relative to its left position. float getTranslationY() The vertical location of this view relative to its left position.
int getVerticalFadingEdgeLength() Returns the size of the vertical scrollbarThumbDrawable () Returns the currently configured Drawable for the thumb of the vertical scroll bar if it exists, null otherwise. Drawable
getVerticalScrollbarTrackDrawable() Returns the currently configured Drawable for the track of the vertical scrollbar. ViewTranslationResponse getViewTranslationResponse () Returns the ViewTranslationResponse associated with this view
 ViewTreeObserver getViewTreeObserver() Returns the ViewTreeObserver for this view is hierarchy. int getWindowAttachCount() WindowId getWindowId() Retrieve the WindowId for the window this view is currently attached to
 WindowInsetsController getWindowInsetsController () Retrieves the single WindowInsetsController of the window this view is attached to. int getWindowInsetsController instead. IBinder getWindowToken() Retrieve a unique token
identifying the window this view is attached to. int getWindowVisibility() Returns the current visibility of the window this view is attached to has been positioned in. float getX()
The visual x position of this view, in pixels. float getY() The visual y position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual y position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels. float getY() The visual z position of this view, in pixels.
focus itself, or is the ancestor of the view that has focus. boolean hasFocusable() returns true if this view has a nested scrolling parent. boolean hasOnClickListeners() Return whether this view has an
attached OnClickListener. boolean hasOnLongClickListener. boolean hasOverlappingRendering() Returns whether this view has an attached OnLongClickListener. boolean hasOverlappingRendering() Returns whether this view has content which overlaps. boolean hasOverlappingRendering() Returns whether this view has ontent which overlaps. boolean hasOverlappingRendering() Returns whether this view has content which overlaps.
tracking transient state that the app should not need to concern itself with saving and restoring, but that the framework should take special note to preserve when possible. boolean hasWindowFocus() Returns true if this view is in a window that currently has window focus. static View inflate(Context context, int resource, ViewGroup root) Inflate a
view from an XML resource. void invalidate() Invalidate() Invalidate the whole view. void invalidate() Invalidate the whole view from an XML resource. void invalidate() Invalidate the whole view. void invalidate() Invalidate()
Because of this, clients are encouraged to just call invalidate(int l, int t, i
 Because of this, clients are encouraged to just call invalidate(). void invalidate Drawable (Drawable (Drawable to rebuild this View's Outline from its outline provider boolean is (Drawable (Drawa
whether this view reacts to click events or not. boolean isContextClickable() Indicates whether this view has changed since the last time being drawn. boolean isDrawingCacheEnabled() This method was deprecated in API level 28. The view drawing cache was largely made obsolete with
the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int,
android.graphics.Paint) handles this with hardware rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have
compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. boolean isDuplicateParentStateEnabled() Indicates whether this duplicates its drawable state from its parent
boolean isEnabled() Returns the enabled status for this view. final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns the enabled status for this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocusable() Returns true if this view has focus final boolean isFocus final b
isFocusedByDefault() Returns whether this View should receive focus when the focus is restored for the view hierarchy containing this view. boolean isHardwareAccelerated() Indicates whether this view is attached to a hardware accelerated
 window or not. boolean isHorizontalFadingEdgeEnabled() Indicate whether the horizontal edges are faded when the view is scrolled horizontal scrollbar should be drawn or not. boolean isHovered() Returns true if the view is currently hovered. boolean
isImportantForAccessibility() Computes whether this view should be exposed for accessibility. final boolean isImportantForAction () Hints the Android System whether the AssistStructure. ViewNode associated with this view is considered importantForAction () Hints the Android System whether the AssistStructure.
this view is considered important for content capture, based on the value explicitly set by setImportantForContentCapture(int) and heuristics when it's IMPORTANT FOR CONTENT CAPTURE AUTO. boolean isInEditMode() Indicates whether this View is currently in edit mode. boolean isInLayout() Returns whether the view hierarchy is currently
undergoing a layout pass. boolean isInTouchMode() Returns whether this View is a root of a keyboard Navigation Cluster. boolean isKeyboardNavigation Cluster. boolean isKeyboardNavigation Cluster.
isPivotSet() Returns whether or not a pivot has been set by a call to setPivotX(float) or setPivotX(float). final boolean isPressed() Indicates whether the view is currently in pressed state. boolean isSaveEnabled() Indicates whether this view will save its state (that is,
 whether its onSaveInstanceState() method will be called). boolean isSaveFromParentEnabled() Indicates whether the entire hierarchy under this view will save its state when a state saving traversal occurs from its parent. boolean isScreenReaderFocusable() Returns whether the view should be treated as a focusable unit by screen reader accessibility
tools. boolean isScrollContainer() Indicates whether this view is one of the set of scrollable containers in its window. boolean isSelected() Indicates the selection state of this view. final boolean isShowingLayoutBounds() Returns true when the View is
attached and the system developer setting to show the layout bounds is enabled or false otherwise. boolean isSoundEffectsEnabled() final boolean isTemporarilyDetached() Tells whether the View is in the state between onStartTemporaryDetach() and
isVisibleToUserForAutofill(int virtualId) Computes whether this virtual autofill view is visible to the user. void jumpDrawable.jumpToCurrentState() on all Drawable.jumpToCurrentState() on all Drawable objects associated with this view. View keyboard navigation cluster.
in the specified direction. void layout(int l, int r, int b) Assign a size and position to a view and all of its descendants This is the second phase of the layout mechanism. final void measure(int l] mergeDrawableStates(int[] baseState, int[]
number of pixels. void on Animation End() Invoked by a parent View Group to notify the end of the animation currently associated with this view. Window Insets (Window Insets insets) Called when the view Group to notify the end of the animation currently associated with this view.
should apply WindowInsets according to its internal policy. void onAttachedToWindow() This is called when the view or a parent view. boolean onCapturedPointerEvent(MotionEvent event) Implement this method to
handle captured pointer events boolean on CheckIsTextEditor() Check whether the called view is a text editor, in which case it would make sense to automatically display a soft input window for it. void on Configuration have
changed. void on Create Context Menu (Context Menu (Editor Info out Attrs) Create a new Input Connection for an Input Connection on Create Input Connection (Editor Info out Attrs) Create a new Input Connection for an Input Connection (Editor Info out Attrs) Create a new Input Connection for an Input Connection (Editor Info out Attrs) Create Input Connection (Editor Info out Attrs) Create Input Connection for an Input Connection (Editor Info out Attrs) Create Info out Attraction (Editor Info out At
 InputMethod to interact with the view. void onCreateViewTranslationRequest(int[] supportedFormats, Consumer requestsCollector) Collects a ViewTranslationRequest which represents the content to be translated in the view. void onCreateVirtualViewTranslationRequest which represents the content to be translated in the view.
Collects ViewTranslationRequests which represents the content to be translated for the virtual views in the host view. void onDetachedFromWindow() This is called when the view is detached from a window. void onDetachedFromWindow() This is called when the view is detached from a window. void onDetachedFromWindow() This is called when the view is detached from a window.
events sent by the system following a call to startDragAndDrop(). void onDraw(Canvas canvas) Implement this to do your drawing. void onDraw(Canvas canvas) Draw any foreground content for this view. final void onDraw(Canvas canvas) Implement this to do your drawing.
on Filter Touch Event For Security (Motion Event event) Filter the touch event to apply security policies. void on Finish Inflate () Finalize inflating a view from XML. void on Focus Changed (boolean gain Focus, int direction, Rect
previouslyFocusedRect) Called by the view system when the focus state of this view changes. boolean onHoverChanged(boolean hovered) Implement this method to handle hover state changes. boolean onHoverEvent(MotionEvent event)
 Implement this method to handle hover events. void on Initialize Accessibility Event event) Initialize an Accessibility Event with information about this View which is the event source. void on Initialize Accessibility Event event) Initialize an Accessibility Node Info info) Initialize an Accessibility Event event with information about this view. boolean
onKeyDown(int keyCode, KeyEvent event) Default implementation of KeyEvent, Callback, onKeyDown(): perform press of the view when KeyEvent event) Default implementation of KeyEvent event) Default implementation of KeyEvent event) Default implementation of KeyEvent event.
KeyEvent.Callback.onKeyLongPress(): always returns false (doesn't handle the event). boolean onKeyMultiple(int keyCode, int repeatCount, KeyEvent event) Handle the event). boolean onKeyMultiple(int keyCode, int repeatCount, KeyEvent event).
processed by any input method associated with the view hierarchy. boolean onKeyShortcut(int keyCode, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyUp(): perform clicking of the view when
 KeyEvent#KEYCODE DPAD CENTER, KeyEvent#KEYCODE ENTER or KeyEvent#KEYCODE SPACE is released. void onLayout(boolean changed, int left, int top, int right, int bottom) Called from layout when this view should assign a size and position to each of its children. void onMeasure(int widthMeasureSpec, int heightMeasureSpec) MeasureSpec) MeasureSpec)
the view and its content to determine the measured width and the mea
the window has just acquired or lost pointer capture. void onPopulateAccessibilityEvent (AccessibilityEvent (AccessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to this View to populate the accessibilityEvent (android.view.accessibilityEvent) giving a chance to the accessibilityEvent (android.view.a
int flags) Populates a ViewStructure to fullfil an autofill request. void onProvideAutofillVirtualStructure (ViewStructure to fullfil an autofill request. void onProvideAutofillVirtualStructure for content capture. void onProvideAutofillVirtualStructure for content capture.
onProvideStructure(ViewStructure structure is being retrieved from a view as part of Activity.onProvideAssistData. void onProvideAssistData to generate additional virtual structure under
this view. ContentInfo onReceiveContent(ContentInfo payload) Implements the default behavior for receiving content for this type of view. PointerIcon onResolvePointerIcon(MotionEvent event, int pointerInfo payload) Implements the default behavior for receiving content for this type of view. PointerIcon onResolvePointerIcon(MotionEvent event, int pointerInfo payload) Implements the default behavior for receiving content for this type of view. PointerIcon onResolvePointerIcon(MotionEvent event, int pointerInfo payload) Implements the default behavior for receiving content for this type of view.
 allowing a view to re-apply a representation of its internal state that had previously been generated by onSaveInstanceState() Hook allowing a view to generate a lignment) has been changed. Parcelable onSaveInstanceState() Hook allowing a view to generate a
representation of its internal state that can later be used to create a new instance with that same state. void on Scroll Capture Search (Rect local Visible Rect, Point window Offset, Consumer targets) Called when scroll
capture is requested, to search for appropriate content to scroll. void on ScrollChanged(int l, int t, int oldl, int
is called during layout when the size of this view has changed. void on Start Temporary Detach() This is called when a container is going to temporarily detach a child, with View Group. detach View From Parent. boolean on Touch Event (Motion Event event) Implement this method to handle touch screen motion events. boolean
onTrackballEvent(MotionEvent event) Implement this method to handle trackball motion events. void onViewTranslationResponses(LongSparseArray
response) Called when the content from View#onCreateVirtualViewTranslationRequests had been translationService. void onVisibilityAggregated(boolean isVisible) Called when the user-visibility of this View is potentially affected by a change to this view itself, an ancestor view or the window this view is attached to. void
onVisibilityChanged(View changedView, int visibility) Called when the visibility of the view or an ancestor of the view has changed (boolean hasWindowFocus) Called when the visibility of the view or an ancestor of the view has changed (boolean hasWindowFocus) Called when the visibility of the view or an ancestor of t
30. SystemUiVisibility flags are deprecated. Use WindowInsetsController instead. void onWindowVisibilityChanged(int visibility) Called when the window containing has change its visibility (between GONE, INVISIBLE, and VISIBLE). boolean overScrollBy(int deltaX, int deltaY, int scrollX, int scrollX, int scrollX, int scrollRangeX, int scrollRangeY, int
maxOverScrollX, int maxOverScrollY, boolean performAccessibilityAction (int action, Bundle arguments) Performs the specified accessibility action on the view. boolean performClick() Call this view's OnClickListener, if it is defined.
boolean performContextClick(float x, float y) Call this view's OnContextClickListener, if it is defined. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boolean performHapticFeedback(int feedbackConstant) BZZZTT!!1! Provide haptic feedback to the user for this view. boo
feedbackConstant, int flags) BZZZTT!!1! Like performHapticFeedback(int), with additional options. boolean performLongClick(float x, float y) Calls this view's OnLongClickListener, if it is defined. ContentInfo performReceiveContent(ContentInfo payload) Receives the
given content. void playSoundEffect(int soundConstant) Play a sound effect for this view. boolean post(Runnable action, long delayMillis) Causes the Runnable to be added to the message queue, to be run after the specified amount of time elapses. void
 postInvalidate() Cause an invalidate to happen on a subsequent cycle through the event loop. void postInvalidate of the specified area to happen on a subsequent cycle through the event loop. void postInvalidate of the specified area to happen on a subsequent cycle through the event loop. void postInvalidate of the specified area to happen on a subsequent cycle through the event loop. void postInvalidate of the specified area to happen on a subsequent cycle through the event loop.
 invalidate of the specified area to happen on a subsequent cycle through the event loop. void postInvalidate OnAnimation (int left, int top, int right, int bottom) Cause an invalidate of the specified area to happen on the next
 animation time step, typically the next display frame. void postInvalidateOnAnimation() Cause an invalidate to happen on the next animation time step, typically the next display frame. void postOnAnimationDelayed(Runnable action, long
delayMillis) Causes the Runnable to execute on the next animation time step, after the specified amount of time elapses. void refreshDrawableState() Call this to force a view to update its drawable action) Removes the specified Runnable from the
message queue. void removeOnAttachStateChangeListener(View.OnAttachStateChangeListener(View.OnLayoutChangeListener) Remove a listener for layout changes. void removeOnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener) Remove a listener for layout changes. Void removeOnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventListener(View.OnUnhandledKeyEventLi
listener) Removes a listener which will receive unhandled KeyEvents. void requestApplyInsets() Ask that a new dispatch of onApplyWindowInsets (android.view.WindowInsets) be performed. void requestApplyInsets() for newer platform versions. final boolean
requestFocus(int direction) Call this to try to give focus to a specific view or to one of its descendants and give it a hint about what direction, Rect previouslyFocusedRect) Call this to try to give focus to a specific view or to one of its descendants.
focus to a specific view or to one of its descendants and give it hints about the direction and a specific rectangle that the focus is coming from. final boolean requestLayout() Call this when something has changed which has invalidated the layout
of this view. void requestPointerCapture() Request pointer capture mode. boolean requestRectangleOnScreen(Rect rectangle of this view be visible on the screen, scrolling if necessary just enough. boolean immediate) Request that a rectangle of this view be visible on the screen, scrolling if necessary just enough.
screen, scrolling if necessary just enough. final void requestUnbufferedDispatch(int source) Request unbuffered dispatch of the given event source class to this view. final T requireViewById(int id) Finds the first
descendant view with the given ID, the view itself if the ID matches getId(), or throws an IllegalArgumentException if the ID is invalid or there is no matching view in the hierarchy. void resetPivotY(float). static int resolveSize(int size, int measureSpec) Version of
resolveSizeAndState(int, int, int) returning only the MEASURED_SIZE_MASK bits of the result. static int resolveSizeAndState(int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState) Utility to reconcile a desired size and state, with constraints imposed by a MeasureSpec, int childMeasuredState (int size, int measureSpec, int childMeasuredState).
hierarchy that has this view as a root. void restoreHierarchyState(SparseArray container) Restore this view hierarchy's frozen state from the given container. final void saveAttributeSet attrs, TypedArray t, int defStyleAetr, in
saveHierarchyState(SparseArray container) Store this view hierarchy's frozen state into the given container. void scrollBy(int x, int y) Move the scrolled position of your view. void scrollTo(int x, int y) Set the scrolled
position of your view. void sendAccessibilityEvent (int eventType) Sends an accessibilityEvent (int) but takes as an argument an empty AccessibilityEvent and does not perform a check whether accessibility is
enabled. void setAccessibilityDelegate (View.AccessibilityDelegate (View.AccessibilityDelegate for implementing accessibility purposes. void setAccessibilityDelegate for implementing accessibilityDelegate for impl
live region mode for this view. void setAccessibilityPaneTitle(CharSequence accessibilityPaneTitle) Visually distinct portion of a window with window-like semantics are considered panes for accessibility purposes. void setAccessibility Traversal. void
setAccessibilityTraversalBefore(int beforeId) Sets the id of a view before which this one is visited in accessibility traversal. void setAllowClickWhenDisabled(boolean activated) Enables or disables click events for this view when disabled. void setAlpha(float
alpha) Sets the opacity of the view to a value from 0 to 1, where 0 means the view is completely transparent and 1 means the view to a value from 0 to 1, where 0 means the view is completely opaque. void setAnimation (Animation animation) Sets the next animation to play for this view. void setAnimationMatrix(Matrix matrix) Changes the transformation matrix on the view. void
setAutoHandwritingEnabled(boolean enabled) Set whether this view enables automatic handwriting initiation. void setAutofillHints(String... autofillHints) Sets the hints that help an AutofillHints(String... autofillHints) Sets the hints that help an AutofillHints in the activity, for
autofill purposes. void setBackground(Drawable background Color(int color) Sets the background to a given Drawable, or remove the background Color(int color) Sets the background to a given Drawable background to a given Drawable background Color(int color) Sets the back
setBackground(android.graphics.drawable.Drawable) instead void setBackgroundTintList(android.content.res.ColorStateList)} to the background
drawable. void setBackgroundTintList(ColorStateList tint) Applies a tint to the backgroundTintList(android.content.res.ColorStateList)} to the background drawable. final void setBottom(int bottom) Sets
the bottom position of this view relative to its parent. void setCameraDistance (float distance) Sets the distance along the Z axis (orthogonal to the X/Y plane on which views are drawn) from the camera to this view. void setClipBounds (Rect clipBounds) Sets a rectangular
area on this view to which the view will be clipToOutline (boolean clipToOutline) Sets whether the View. void setContentCaptureSession (ContentCaptureSession) Sets the (optional) ContentCaptureSession associated with this view. void setContentCaptureSession (ContentCaptureSession) Sets the (optional) ContentCaptureSession associated with this view. void setContentCaptureSession (ContentCaptureSession) Sets the (optional) ContentCaptureSession associated with this view.
setContentDescription(CharSequence contentDescription) Sets the View's content description. void setContextClickable (boolean contextClickable) Enables or disables context clicking for this view. void setDefaultFocusHighlightEnabled (boolean contextClickable) Enables or disables context clicking for this view.
focused but doesn't have R.attr.state focused defined in its background. void setDrawingCacheBackgroundColor(int color) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-acceleration, intermediate cache layers are
largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a small part of the View
hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time shadows, and outline
clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void setDrawingCacheEnabled(boolean enabled) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration,
intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered snapshots of a
small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE bitmaps, real-time
shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void setDrawingCache Quality(int quality) This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-
acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered
snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and Canvas from eith
bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void setElevation (float elevation) Sets the base elevation of this view, in
pixels. void setEnabled(boolean enabled) Set the enabled state of this view. void setFadingEdgeLength(int length) Set the size of the faded edge used to indicate that more content in this view is available. void setFadingEdgeLength(int length) Set the enabled state of this view.
another visible window at the touched location. void setFitsSystemWindows(boolean fitSystemWindows(boolean fitSystemWindows(boolean fitSystemWindows) Sets whether or not this view should account for system screen decorations such as the status bar and inset its controlling whether the default implementation of fitSystemWindows(android.graphics.Rect) will be executed. void
setFocusable(boolean focusable) Set whether this view can receive focus. void setFocusableInTouchMode(boolean isFocusedByDefault) Sets whether this view can receive focus. while in touch mode. void setFocusableInTouchMode(boolean isFocusedByDefault) Sets whether this view can receive focus.
 this View should receive focus when the focus is restored for the view hierarchy containing this view. void setForceDarkAllowed(boolean allow) Sets whether or not to allow force dark to apply to this view. void setForceDarkAllowed(boolean allow) Sets whether or not to allow force dark to apply to this view. void setForceDarkAllowed(boolean allow) Sets whether or not to allow force dark to apply to this view.
setForegroundGravity(int gravity) Describes how the foregroundTintList(android.content.res.ColorStateList)} to the background drawable. void setForegroundTintList(ColorStateList tint) Applies a
 tint to the foreground drawable. void setForegroundTintMode(PorterDuff.Mode tintMode) Specifies the blending mode used to apply the tint specified by setForegroundTintList(android.content.res.ColorStateList)} to the background drawable. void setHapticFeedbackEnabled(boolean hapticFeedbackEnabled) Set whether this view should have haptic
 feedback for events such as long presses. void setHasTransientState(boolean hasTransientState) Set whether this view is currently tracking transient state that the framework should attempt to preserve when possible. void setHorizontalFadingEdgeEnabled(boolean horizontalFadingEdgeEnabled)
when this view is scrolled horizontally. void setHorizontalScrollBarEnabled(boolean horizontalScrollbarTnumbDrawable drawable) Define whether the horizontal scrollbar should be drawn or not. void setHorizontalScrollbarTnumbDrawable drawable) Defines the horizontal scrollbar should be drawn or not.
Defines the horizontal track drawable void setHovered(boolean hovered) Sets whether the view is currently hovered. Void setImportant for accessibility (int mode) Sets how to determine whether this view is important for accessibility which is if it fires accessibility events and if it is reported to
accessibility services that query the screen. void setImportantForAutofill(int mode) Sets the mode for determining whether this view is considered important for content capture. void setKeepScreenOn(boolean
keepScreenOn) Controls whether the screen should remain on, modifying the value of KEEP SCREEN ON. void setKeyboardNavigation cluster. void setKeyboardNavigation cluster this view is a root of a keyboard navigation cluster.
setLayerPaint(Paint paint) Updates the Paint object used with the current layer (used only if the current layer type is not set to LAYER TYPE NONE). void setLayoutDirection (int layoutDirection) Set the layout direction for this view. void
setLayoutParams(ViewGroup.LayoutParams params) Set the layout parameters associated with this view. final void setLeft(int left, int top, int right, int bottom) Assign a size and position to this view. void setLongClickable(boolean longClickable) Enables
or disables long click events for this view. final void setMeasuredDimension(int measuredWidth, int measuredWidth, int measured width and measured width int minHeight) Sets the minimum height of the view. void setMinimumWidth(int minWidth) Sets the
minimum width of the view. void setNextClusterForwardId(int nextClusterForwardId(int nextClusterForwardId(int nextClusterForwardId) Sets the id of the view to use as the root of the next keyboard navigation cluster.
the next focus is FOCUS_DOWN. void setNextFocusForwardId(int nextFocusForwardId(int nextFocusForwardId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId) Sets the id of the view to use when the next focus is FOCUS_LEFT. void setNextFocusRightId(int nextFocusRightId(int nextFocusRi
use when the next focus is FOCUS RIGHT. void setOnApplyWindowInsetsListener (View.OnApplyWindowInsetsListener to take over the policy for applying window insets to this view. void
setOnCapturedPointerListener(View.OnClickListener(View.OnClickListener(View.OnClickListener(View.OnContextClickListener(View.OnContextClickListener)) Register a callback to be invoked when this view is clicked. void setOnContextClickListener(View.OnContextClickListener)
invoked when this view is context clicked, void setOnCreateContextMenuListener(View.OnCreateContextMenuListener l) Register a callback to be invoked when the context menu for this view is being built, void setOnDragListener(View.OnDragListener l) Register a drag event listener callback object for this View, void
setOnFocusChangeListener(View.OnFocusChangeListener(View.OnFocusChangeListener) Register a callback to be invoked when a generic motion event is sent to this view. void setOnHoverListener(View.OnHoverListener) Register a callback to be invoked when a generic motion event is sent to this view.
to be invoked when a hover event is sent to this view. void setOnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnLongClickListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnLongClickListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(View.OnKeyListener(
mimeTypes, OnReceiveContentListener listener) Sets the listener to be used to handle insertion of content into this view. void setOnScrollChangeListener l) Register a callback to be invoked when the scroll X or Y positions of this view change. void
setOnSystemUiVisibilityChangeListener(View.OnSystemUiVisibilityChangeListener on this view. void setOnTouchListener l) Register a callback to be invoked when
a touch event is sent to this view, void setOutlineProvider of the ambient shadow Color (int color) Sets the ViewOutlineProvider of the view, which generates the Outline that defines the shadow it
casts, and enables outline clipping. void setOutlineSpotShadowColor(int color) Sets the color of the spot shadow that is drawn when the view has a positive Z or elevation value. void setOverScrollMode(int overScrollMode) Set the over-scroll mode for this view. void setPadding(int left, int top, int right, int bottom) Sets the padding. void
setPaddingRelative(int start, int top, int end, int bottom) Sets the relative padding. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. Void setPivotX(float pivotX) Sets the x location of the point around which the view is rotated and scaled. Void setPivotX(float pivotX) Sets
icon for the current view. final void setPreferKeepClear (boolean preferKeepClear) Set a preference to keep the bounds of this view clear from floating windows above this view's window. void setPreferKeepClear (boolean preferKeepClear) Set a preference to keep the provided rects clear from floating windows above this view's window. void
setPressed(boolean pressed) Sets the pressed state for this view. void setRenderEffect (RenderEffect (RenderEffect to apply to this View. final void setRight(int right) Sets the right position of this
view relative to its parent, void setRotation (float rotation) Sets the degrees that the view is rotated around the pivot point, void setRotation (float rotation) Sets the degrees that the view is rotated around the pivot point.
through the pivot point, void setSaveEnabled(boolean enabled) Controls whether the entire hierarchy under this view will save its state when a state saving traversal occurs
from its parent. void setScaleX(float scaleX) Sets the amount that the view is scaled in x around the pivot point, as a proportion of the view's unscaled width. void setScreenReaderFocusable(boolean
screenReaderFocusable) Sets whether this View should be a focusable element for screen readers and include non-focusable Views from its subtree when providing feedback, void setScrollBarFadeDuration(int
scrollBarFadeDuration) Define the scrollbar fade duration. void setScrollBarSize(int scrollBarSize) Define the scrollbars. final void setScrollBarSize(int style) Specify the style of the scrollbars. final void setScrollBarSize(int style) Specify the style of the scrollbars.
Sets the scroll capture hint for this View. void setScrollIndicators (int indicators, int mask) Sets the state of the scroll indicators specified by the mask. void setScrollIndicators (int indicators) Sets the state of all scroll
indicators. void setScrollX(int value) Set the horizontal scrolled position of your view. void setScrollbars will fade when the view is not scrolling. void setSelected(boolean selected) Changes the selection state of this
view. void setSoundEffectsEnabled(boolean soundEffectsEnabled(boolean soundEffectsEnabled) Set whether this view should have sound effects enabled for events such as clicking and touching. void setStateDescription (CharSequence stateDescription) Sets the View's state description.
StateListAnimator to this View. void setSystemGestureExclusionRects(List rects) Sets a list of areas within this view's post-layout coordinate space where the system UiVisibility (int visibility) This method was deprecated in API level 30. SystemUiVisibility flags are deprecated.
Use WindowInsetsController instead. void setTag(int key, Object tag) Sets a tag associated with this view and a key. void setTextAlignment int textAlignment. void setTextAlignment int textAlignment int textAlignment.
tooltipText) Sets the tooltip text which will be displayed in a small popup next to the view. final void setTouchDelegate for this View. void setTransitionAlpha(float alpha) This property is intended only for use by the Fade
transition, which animates it to produce a visual translucency that does not side-effect (or get affected by) the real alpha property. final void setTransitionName (String transitionName) Sets the name of the View to be used to identify Views in Transition (Included String transitionName) Sets the name of the View without triggering any
other changes, void setTranslationX(float translationX) Sets the horizontal location of this view relative to its left position, void setTranslationY(float translationY) Sets the depth location of this view relative to its elevation, void setTranslationY(float translationY) Sets the depth location of this view relative to its elevation, void setTranslationY(float translationY) Sets the depth location of this view relative to its elevation.
setVerticalFadingEdgeEnabled(boolean verticalScrollBarEnabled(boolean verticalScrollBarEnabled) Define whether the verticalScrollBarEnabled(boolean verticalScrollBarEnabled) Define whether the verticalScrollBarEnabled(boolean verticalScrollBarEnabled).
the vertical scroll bar. void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbarTrackDrawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbar thumb drawable (Drawable drawable) Defines the vertical scrollbar thumb drawable void setVerticalScrollbar thumb drawabl
to display/hide the translated information. void setVisibility(int visibility) Set the visibility (int visibility) Set the visibility (int visibility) Set the visibility of hardware-accelerated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API 11. With hardware-accelerated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean willNotCacheDrawing) This method was deprecated in API level 28. The view drawing (boolean will be appeared to the view drawing) This method was deprecated in API level 28. The view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be appeared to the view drawing) This method was deprecated to the view drawing (boolean will be 
acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint) handles this with hardware rendering. For software-rendered
snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE
bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. void setWillNotDraw(boolean willNotDraw) If this view doesn't do any drawing on its own, set this flag to allow further optimizations. void
setWindowInsetsAnimation.Callback (WindowInsetsAnimation.Callback to be notified about animations of this view, in pixels. void setX(float x) Sets the visual x position of this view, in pixels. void setX(float x) Sets the visual x position of this view, in pixels. void setX(float x) Sets the visual x position of this view, in pixels. void setX(float x) Sets the visual x position of this view, in pixels.
position of this view, in pixels. boolean showContextMenu() Shows the context menu for this view anchored to the specified view-relative coordinate. ActionMode startActionMode(ActionMode.Callback, int type) Start an action mode with the given type.
ActionMode startActionMode(ActionMode(ActionMode(ActionMode animation) Start an action mode with the default type ActionMode#TYPE_PRIMARY. void startAnimation now. final boolean startDrag(ClipData data, View.DragShadowBuilder, Object myLocalState, int flags) This method was
deprecated in API level 24. Use startDragAndDrop() for newer platform versions. final boolean startDragAndDrop(ClipData data, View.DragShadowBuilder, Object myLocalState, int flags) Starts a drag and drop operation. boolean startDragAndDrop() for newer platform versions. final boolean startDragAndDrop() for newer platform versions. final boolean startDragAndDrop() for newer platform versions. final boolean startDragAndDrop() for newer platform versions.
Stop a nested scroll in progress. String toString() Returns a string representation of the object, void transformMatrixToLocal(Matrix matrix) Modifies the input matrix such that it maps on-screen coordinates to on-screen coordinates to
view-local coordinates. void unscheduleDrawable (Drawable who) UnscheduleDrawable (Drawable who) UnscheduleDrawable who) Unsch
drop operation. boolean verifyDrawable (Drawable who) If your view subclass is displaying its own Drawable objects, it should override this function and return true for any Drawable it is displaying. boolean willNotCacheDrawing() This method was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of
hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, setLayerType(int, android.graphics.Paint)
handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas from either a Bitmap or Picture and call draw(android.graphics.Canvas) on the View. However these software-rendered usages are discouraged and have compatibility issues with
hardware-only rendering features such as Config. HARDWARE bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy API is recommended. boolean willNotDraw() Returns a not this View draws on its own. From class java.lang.Object Object clone() Creates and returns a
```

copy of this object. boolean equals(Object obj) Indicates whether some other object is "equal to" this one, void finalize() Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. final Class getClass() Returns the runtime class of this Object int hashCode() Returns a hash code

```
value for the object. final void notify() Wakes up a single thread that is waiting on this object's monitor. String to String() Returns a string representation of the object. final void wait(long timeout, int nanos) Causes the current thread to wait until another thread invokes
the notify() method or the notifyAll() method for this object, or some other thread interrupts the current thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has
elapsed. final void wait() Causes the current thread invokes the notify() method or the notifyAll() method for this object. From interface android.view.ViewParent abstract boolean canResolveLayoutDirection() Tells
if this view parent can resolve the layout direction. abstract boolean canResolveTextAlignment() Tells if this view parent can resolve the text direction. abstract void childDrawableStateChanged(View child) This method is called on the parent when a
child's drawable state has changed. abstract void childHasTransientStateChanged(View child, boolean hasTransientState) Called when a child of this parent is giving up focus abstract void createContextMenu (ContextMenu menu) Have
the parent populate the specified context menu if it has anything to add (and then recurse on its parent). default OnBackInvokedDispatcher findOnBackInvokedDispatcher. abstract View focusSearch(View v, int direction) Find the nearest
view in the specified direction that wants to take focus abstract void focusable view has become available. The parent that a new focusable view has become available abstract boolean getChild view's coordinates. The parent that a new focusable view has become available abstract boolean getChild view's coordinates.
getLayoutDirection() Return this view parent layout direction. abstract ViewParent getParent() Returns the parent if it exists, or null. abstract ViewParent getParentForAccessibility() Gets the parent text alignment() Return this view parent text alignment abstract ViewParent getParent () Return this view parent text alignment.
parent text direction. abstract void invalidateChild(View child, Rect r) This method was deprecated in API level 26. Use onDescendantInvalidateChildInParent (int[] location, Rect r) This method was deprecated in API level 26. Use
 onDescendantInvalidated(android.view.View, android.view.View, android.view.View) instead. abstract boolean isLayoutDirectionResolved() Tells if this view parent text
alignment is resolved. abstract boolean isTextDirectionResolved() Tells if this view parent text direction is resolved. abstract View keyboardNavigationCluster in the specified direction. abstract void notifySubtreeAccessibilityStateChanged(View child, View source
int changeType) Notifies a view parent that the accessibility state of one of its descendants has changed and that the structure of the subtree is different. default void onDescendantInvalidated(View child, View target View has been invalidated, or has had a drawing property changed that requires the hierarchy to re-render. abstract
boolean onNestedFling(View target, float velocityX, float velocityY, boolean consumed) Request a fling from a nested fling before the target view consumes it. abstract boolean onNestedPrePerformAccessibilityAction(View target, int action, Bundle fling before the target view consumes it. abstract boolean onNestedPrePerformAccessibilityAction(View target, int action, Bundle fling before the target view consumes it. abstract boolean onNestedPrePerformAccessibilityAction(View target, int action, Bundle fling before the target view consumes it. abstract boolean onNestedPrePerformAccessibilityAction(View target, int action, Bundle fling before the target view consumes it. abstract boolean onNestedPrePerformAccessibilityAction(View target, int action, Bundle fling before the target view consumes it.)
arguments) React to an accessibility action delegated by a target descendant view before the target processes it. abstract void onNestedPreScroll(View target, int dx, int dy, int[] consumed) React to a nested scroll in progress before the target view consumes a portion of the scroll. abstract void onNestedScroll(View target, int dx, int dy, int[] consumed) React to an accessibility action delegated by a target descendant view before the target processes it.
dyConsumed, int dxUnconsumed, int dxUnconsumed, int dyUnconsumed) React to a nested scroll Accepted (View child, View target, int nested ScrollAxes) React to a descendant view
initiating a nestable scroll operation, claiming the nested scroll operation if appropriate. abstract void recomputeViewAttributes (View child) Tell view hierarchy that the global view attributes need to be re-evaluated. abstract void requestChildFocus(View child) Tell view hierarchy that the global view attributes need to be re-evaluated. abstract void recomputeViewAttributes(View child) Tell view hierarchy that the global view attributes need to be re-evaluated. abstract void requestChildFocus(View child)
View focused) Called when a child of this parent wants focus abstract boolean requestChildRectangleOnScreen(View child, Rect rectangle to be positioned onto the screen. abstract void requestDisallowInterceptTouchEvent(boolean disallowIntercept) Called when a
child does not want this parent and its ancestors to intercept touch events with ViewGroup#onInterceptTouchEvent(MotionEvent). abstract void requestFitSystemWindows() Ask that a new dispatch of View.fitSystemWindows() Ask that a new dispatch of View.fitSystemWindows(Rect) be performed. abstract void requestFitSystemWindows() Ask that a new dispatch of View.fitSystemWindows() ask that a new dispatch of View.fitS
child of this view parent. abstract boolean requestSendAccessibilityEvent (View child, AccessibilityEvent event) Called by a child to request from its parent to send an AccessibilityEvent abstract void requestTransparent regions to the window
compositor. abstract boolean showContextMenuForChild(View originalView, float x, float y) Shows the context menu for the specified view or its ancestors anchored to the specified view-relative coordinate. abstract
ActionMode startActionModeForChild(View originalView, ActionMode.Callback, int type) Start an action mode of a specified view with the default type
ActionMode#TYPE PRIMARY. From interface android.view.KeyEvent.Callback abstract boolean onKeyDown(int keyCode, KeyEvent event) Called when a long press has occurred. abstract boolean onKeyMultiple(int keyCode, int count
KeyEvent event) Called when a user's interaction with an analog control, such as flinging a trackball, generates simulated down/up events for the same key multiple times in quick succession. abstract boolean onKeyUp(int keyCode, KeyEvent event) Called when a key up event has occurred. public static final String SCHEME_GEO URI scheme for map
address. Constant Value: "geo:0,0?q=" public static final String SCHEME MAILTO URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME MAILTO URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final String SCHEME TEL URI scheme for telephone number. Constant Value: "mailto:" public static final Scheme for telephone number. Constant Value: "mailto:" public static final Scheme for telephone number. Constant Value: "mailto:" public sc
should always be instantiated with an Activity Context. If instantiated with an Activity Context to access application assets This value cannot be null. public WebView (Context context, AttributeSet attrs)
Constructs a new WebView with layout parameters. Parameters context to access application assets This value cannot be null. attrs AttributeSet attrs, int defStyleAttr) Constructs a new WebView with layout
parameters and a default style. Parameters context Context: an Activity Context to access application assets This value cannot be null. attributeSet an AttributeSet passed to our parent This value may be null. defStyleAttr int: an attributeSet an AttributeSet passed to our parent This value may be null. attrophication assets This value may be null. attrophication attro
Can be 0 to not look for defaults. public WebView (Context context context context context an Activity Context to access application assets This value cannot be null. attrs AttributeSet: an AttributeSet passed to our parent
This value may be null. defStyleAttr int: an attribute in the current theme that contains a reference to a style resource that supplies default values for the view, used only if defStyleAttr is 0 or can not be found in the
theme. Can be 0 to not look for defaults. public WebView (Context context, AttributeSet attrs, int defStyleAttr, boolean privateBrowsing) This constructor is deprecated. Private browsing is no longer supported directly via WebView and will be removed in a future release. Prefer using WebSettings, WebViewDatabase, CookieManager and WebStorage
for fine-grained control of privacy data. Constructs a new WebView with layout parameters and a default style. Parameters context to access application assets This value may be null. defStyleAttr int: an attribute in the current theme that
contains a reference to a style resource that supplies default values for the view. Can be 0 to not look for defaults. private mode public void addJavascriptInterface (Object object, String name) Injects the supplied Java object into this WebView. The object is injected into all frames of
the web page, including all the iframes, using the supplied name. This allows the Java object's methods to be accessed from JavaScript. For applications targeted to API level Build. VERSION CODES. JELLY BEAN MR1 and above, only public methods that are annotated with JavaScript. For applications targeted to API level Build. VERSION CODES. JELLY BEAN MR1 and above, only public methods that are annotated with JavaScript. For applications targeted to API level Build. VERSION CODES. JELLY BEAN MR1 and above, only public methods to be accessed from JavaScript.
targeted to API level Build.VERSION_CODES.JELLY_BEAN or below, all public methods (including the inherited ones) can be accessed, see the important security note below for implications. Note that injected objects will not appear in JavaScript until the page is next (re)loaded. JavaScript should be enabled before injecting the object. For example
class JsObject { @JavascriptInterface public String toString() { return "injectedObject"; } } webview.loadData("", "text/html", null); webView.loadUrl("javascriptInterface(new JsObject(), "injectedObject"); webView.loadData("", "text/html", null); webView.loadUrl("javascriptInterface(new JsObject(), "injectedObject"); webView.loadData("", "text/html", null); webView.loadUrl("javascriptInterface(new JsObject(), "injectedObject"); webView.loadUrl("javascriptInterface(), "injectedObject"); webView.loadUrl
used to allow JavaScript to control the host application. This is a powerful feature, but also presents a security risk for apps targeting Build.VERSION_CODES.JELLY_BEAN are still vulnerable if the app runs on a device running Android earlier than 4.2. The most
secure way to use this method is to target Build. VERSION CODES. JELLY BEAN MR1 and to ensure the method is called only when running on Android 4.2 or later. With these older versions, JavaScript could use reflection to access an injected object's public fields. Use of this method in a WebView containing untrusted content could allow an
attacker to manipulate the host application in unintended ways, executing Java code with the permissions of the host application. Use extreme care when using this method in a WebView, Care is therefore required to maintain
thread safety. Because the object is exposed to all the frames, any frame could obtain the object name and call methods on it. There is no way to tell the calling frame's origin from the app must not assume that the caller is trustworthy unless the app can guarantee that no third party content is ever loaded into the WebView even inside
an iframe. The Java object's fields are not accessible. For applications targeted to API level Build.VERSION_CODES.LOLLIPOP and above, methods of injected Java object to inject into this WebView's JavaScript context. null values are ignored. name String: the name used to
expose the object in JavaScript This value cannot be null. public void autofill (SparseArray values) Automatically fills the content of the virtual children mean and how they can be autofilled. Implementing
the methods that autofill the virtual children. on Provide Autofill Virtual Structure (and roid.view. View Structure, int) is responsible for the latter - see autofill (and roid.view. View Structure, int) for more info about autofill. If a child value is
updated asynchronously, the next call to AutofillManager#notifyValueChanged(View, int, AutofillValue) must happen after the value was changed to the autofilled. Note: To indicate that a virtual view was autofilled. Note: To indicate that a virtual view was autofilled. Note: To indicate that a virtual view was changed to the autofilled. Note: To indicate that a virtual view was autofilled. Note: To indicate that a virtual view was autofilled.
Parameters values SparseArray: map of values to be autofilled, keyed by virtual child id. This value cannot be null. public boolean canGoBack () Gets whether this WebView has a back history item. Returns boolean true if this WebView has a back history item. Returns boolean true if this WebView has a back history item.
forward the given number of steps. Parameters steps int: the negative or positive number of steps to move the history item. Returns boolean canGoForward () Gets whether this WebView has a forward history item Added in API level 17 public boolean canZoomIn ()
This method was deprecated in API level 17. This method is prone to inaccuracy due to race conditions between the web rendering and UI threads; prefer WebView can be zoomed in API level 17 Deprecated in API level 17 public
boolean canZoomOut () This method was deprecated in API level 17. This method is prone to inaccuracy due to race conditions between the web rendering and UI threads; prefer WebView can be zoomed out. Returns boolean true if this WebView can be zoomed out Added in API level 1 Deprecated
in API level 19 public Picture capturePicture () This method was deprecated in API level 19. Use onDraw(Canvas) to obtain a bitmap snapshot of the WebView, or save WebArchive(String) to save the content to a file. Gets a new picture that captures the current contents of this WebView. The picture is of the entire document being displayed, and is not
limited to the area currently displayed by this WebView. Also, the picture is a static copy and is unaffected by later changes to the content being displayed. Note that due to internal changes, for API levels between Build.VERSION CODES.HONEYCOMB and Build.VERSION CODES.HONEYCO
position elements or scrollable divs. Note that from Build. VERSION CODES. JELLY BEAN MR1 the returned picture should only be drawn into bitmap-backed Canvas - using any other type of Canvas will involve additional conversion at a cost in memory and performance. Returns Picture a picture that captures the current contents of this WebView
public void clearCache (boolean includeDiskFiles) Clears the resource cache is per-application, so this will clear the cache for all WebViews used. Parameters includeDiskFiles boolean: if false, only the RAM cache is cleared public static void clearClientCertPreferences (Runnable onCleared) Clears the client certificate preferences
stored in response to proceeding/cancelling client cert requests. Note that WebView automatically clears these preferences when the system keychain is updated. The preferences are shared by all the WebViews that are created by the embedder application. Parameters on Cleared Runnable: A runnable to be invoked when client certs are cleared. The
runnable will be called in UI thread. This value may be null. public void clearFormData () Removes the autocomplete popup, it does not remove any saved form data from this WebView's store. To do that, use
WebViewDatabase#clearFormData. public void clearHistory () Tells this WebView to clear its internal back/forward list. public void clearSslPreferences () Clears the SSL preferences to proceeding with SSL certificate errors. Added in API level 1 Deprecated in API level 18 public void clearWiew () This method was deprecated
in API level 18. Use WebView.loadUrl("about:blank") to reliably reset the view state and release page resources (including any running JavaScript). Clears this WebView so that onDraw() will draw nothing but white background, and onMeasure() will return 0 if MeasureSpec is not MeasureSpec is not MeasureSpec.EXACTLY. public void computeScroll () Called by a
parent to request that a child update its values for mScrollY and mScrollY if necessary. This will typically be done if the child is animating a scroller object, public WebBackForwardList () Gets the WebBackForwardList for this WebView. This contains the back/forwardList for use in querying each item in the history
stack. This is a copy of the private WebBackForwardList so it contains only a snapshot of the current state. Multiple calls to this method may return different objects. The object returned from this method will not be updated to reflect any new state. Returns WebBackForwardList This value cannot be null. public PrintDocumentAdapter
createPrintDocumentAdapter (String documentName) Creates a PrintDocumentAdapter that provides the conversion process - any such draws are undefined. It is recommended to use a dedicated
off screen WebView for the printing. If necessary, an application may temporarily hide a visible WebView by using a custom PrintDocumentAdapter for more information. Parameters documentName String: The user-facing named and observing the onStart and onFinish methods. See PrintDocumentAdapter for more information. Parameters documentName String: The user-facing named and observing the onStart and onFinish methods.
of the printed document. See PrintDocumentInfo This value cannot be null. Returns PrintDocumentAdapter This value cannot be null. PrintDocumentInfo This value cannot be null. Returns PrintDocumentInfo This value cannot be null. PrintDocumentInfo This value cannot be null. Returns PrintDocumentInfo This value cannot be null. PrintDocumentInfo This value cannot be null. Returns PrintDocumentInfo This v
channel functionality is described here The returned message channels are entangled and already in started state. Returns WebMessagePort[] the two message ports that form the message channel are entangled and already in started state. Returns WebMessagePort[] the two message channels are entangled and already in started state. Returns WebMessagePort[] the two message channels are entangled and already in started state.
removed from the view system. No other methods may be called on this WebView after destroy, public static void disableWebView () Indicate that the current process does not intend to use WebView, and that an exception should be thrown if a WebView is created or any other methods in the android.webkit package are used. Applications with
multiple processes may wish to call this in processes that are not intended to use WebView to avoid accidentally incurring the memory usage of initializing WebView in long-lived processes that have no need for it, and to prevent potential data directory conflicts (see setDataDirectorySuffix(String)). For example, an audio player application with one
process for its activities and another process for its playback service may wish to call this method in the playback service.onCreate(). public void dispatchCreateViewTranslationRequests for translationRequests for translationRequests for its playback service may wish to call this method in the playback service.onCreate(). public void dispatchCreateViewTranslationRequests for translationRequests for translationRequest (Map viewIds, int[] supportedFormats, TranslationRequests for translationRequest for translationRequests for translationRequests for translationRequest for 
purpose by traversing the hierarchy when the app requests ui translation. Typically, this method should only be overrided view#onCreateViewTranslationRequest for normal view or override View#onVirtualViewTranslationResponses for view contains
virtual children. When requested to start the ui translation, the system will call this method to traverse the view Translation Requests must be added when the traversal is done. The default implementation calls
View#onCreateViewTranslationRequest for normal view or calls View#onVirtualViewTranslationResponses for view contains virtual children to build ViewTranslationRequest if the view should be translated. The view is marked as having transient state so that recycling of views doesn't prevent the system from attaching the response to it. Therefore,
if overriding this method, you should set or reset the transient state. The implementation calls dispatchCreateViewTranslationRequest(Map, int[], TranslationRequest(Map, int[], Translati
capability TranslationCapability: This value may be null. requests List: This value cannot be null. public boolean dispatch to itself. Otherwise it will
dispatch the next node down the focus path. This method also fires any key listeners. Parameters event KeyEvent: The key event to be dispatched. Returns boolean True if the event was handled, false otherwise. public void documentHasImages (Message response) Queries the document to see if it contains any image references. The message object
will be dispatched with arg1 being set to 1 if images were found and 0 if the document does not reference any images. Parameters response Message: the message that will be dispatched with the result This value cannot be null. public static void enableSlowWholeDocumentDraw () For apps targeting the L release, WebView has a new default
behavior that reduces memory footprint and increases performance by intelligently choosing the portion of the HTML document that needs to be drawn. These optimizations are transparent to the developers. However, under certain circumstances, an App developer may want to disable them: When an app uses onDraw(Canvas) to do own drawing and
accesses portions of the page that is way outside the visible portion of the page. When an app uses capturePicture is a deprecated API. Enabling drawing the entire HTML document. Note that capturePicture is a deprecated API. Enabling drawing the entire HTML document.
public void evaluateJavascript (String script, ValueCallback resultCallback will be invoked with any result returned from that execution. This method must be called on the UI thread and the callback will be made on the UI thread.
Compatibility note. Applications targeting Build.VERSION CODES.N or later, JavaScript state from an empty WebView is no longer persisted across navigations like loadUrl(java.lang.String) will not exist in the loaded page. Applications should use
addJavascriptInterface(Object, String) instead to persist JavaScript objects across navigations. Parameters script String: the JavaScript to execution completes with the result of the execution (if any). May be null if no notification of the result is
required. Added in API level 1 Deprecated in API level 28 public static String findAddress (String addr) This method was deprecated in API level 28. This method even when targeting API levels where no alternative is available. Gets
the first substring which appears to be the address of a physical location. Only addresses in the United States can be detected, which must consist of: a house number a street type (Road, Circle, etc), either spelled out or two-letter abbr an optional 5 digit or 9 digit zip
code All names must be correctly capitalized, and the zip code, if present, must be valid for the state or territory must also be spelled or abbreviated using USPS standards. The house number may not exceed five digits. Note: This function is deprecated and should be
avoided on all API levels, as it cannot detect addresses outside of the United States and has a high rate of false positives. On API level Build. VERSION CODES.O MR1 and earlier, it also causes the entire WebView implementation to be loaded and initialized, which can throw AndroidRuntimeException or other exceptions if the WebView
vx, int vy) Added in API level 7 Deprecated in API level 19 public void freeMemory () This method was deprecated in API level 19. Memory caches are automatically dropped when no longer needed, and in response to system memory pressure. Informs this WebView that memory is low so that it can free any available memory. public CharSequence
AccessibilityNodeInfo.setClassName. public SslCertificate (the site is not secure). Returns SslCertificate for the main top-level page or null if there is no certificate for the main top-level page or null if there is no certificate (the site is not secure). Returns SslCertificate for the main top-level page or null if there is no certificate for the main top-level page or null if there is no certificate (the site is not secure). Returns SslCertificate for the main top-level page or null if there is no certificate for the main top-level page or null if there is no certificate (the site is not secure).
HTML content public static PackageInfo getCurrentWebView has already been loaded into the current process this method will return the package that would be used if the WebView was loaded right now will be returned; this does not cause WebView to be loaded, so this
it will use the new WebView package instead. Returns PackageInfo the current page or null if there is none. public Bitmap getFavicon () Gets the favicon for the current page or null if there page doesn't
have one or if no page has been loaded public Handler () Returns Handler associated with the thread running the View. This handler can be used to pump events in the UI events queue. public WebView. HitTestResult getHitTestResult () Gets a HitTestResult based on the current cursor node. If a HTML::a tag is found and the
HitTestResult type is set to IMAGE TYPE and the URL is set in the "extra" field. A type of SRC IMAGE ANCHOR TYPE and the phone number is found, the HitTestResult type is set to PHONE TYPE and the phone number is set in the "extra" field of HitTestResult. If a map address is
found, the HitTestResult type is set to GEO TYPE and the email address is set in the "extra" field of HitTestResult. If an email address is found, the HitTestResult type is set to UNKNOWN TYPE. Returns WebView. HitTestResult This value cannot be
Parameters host String: the host to which the credentials apply Returns String getOriginalUrl () Gets the original URL for the
current page. This is not always the same as the URL passed to WebViewClient.onPageStarted because although the load for that URL to that originally requested. Returns String the URL that was originally requested for the current
document which can be displayed to users. This value cannot be null. Added in API level 1 Deprecated in API level 17 public float getScale () This method was deprecated in API level 17. This method is prone to inaccuracy due to race conditions between the web rendering and UI threads; prefer WebViewClient#onScaleChanged. Gets the current
scale of this WebView. Returns float the current scale public WebSettings object that can be used to control this WebView. Returns float the current scale public String getSettings object that can be used to control this webView. Returns float the current page. This is
the title of the current page until WebViewClient.onReceivedTitle is called. Returns String the title for the current page or null if no page has been loaded public String getUrl () Gets the URL for the current page. This is not always the same as the URL passed to WebViewClient.onPageStarted because although the load for that URL has begun, these title for the current page or null if no page has been loaded public String getUrl () Gets the URL for the current page.
current page may not have changed. Returns String the URL for the current page or null if no page has been loaded public static ClassLoader () Returns the ClassLoader getWebView Support Library, there is no reason to use this method otherwise
Returns ClassLoader This value cannot be null. public Looper getWebViewLooper () Returns the Looper getWebViewRenderProcess getWebViewRenderProcess () Gets a handle to the WebView renderer process associated with this
WebView. In Build.VERSION CODES.O and above, WebView may run in "multiprocess mode, rendering of web content is performed by a sandboxed renderer process may be shared with other webViews in the application, but is not shared with other application
processes. If WebView is running in multiprocess mode, this method returns a handle to the renderer process associated with the WebView, which can be used to control the renderer process associated with the WebView, which can be used to control the renderer process. public void goBackOrForward (int steps) Goes to the history item that is the number of steps
away from the current item. Steps is negative if backward and positive if forward. Parameters steps int: the number of steps to take back or forward in the history of this WebView. This will result void goForward in the history of this WebView. This will result void goForward in the history of this WebView. This will result void goForward in the history of this WebView.
in the zoom widget appearing on the screen to control the zoom level of this WebView. public boolean isPrivateBrowsingEnabled () Gets whether this virtual autofill view is visible to the user. Note: By default it returns true, but
views providing a virtual hierarchy view must override it. Returns boolean Whether the view is visible on the screen. public void loadData (String data, String encoding) Loads the given data into this WebView using a 'data' scheme URL. Note that JavaScript's same origin policy means that script running in a page loaded using this
method will be unable to access content loaded using any scheme other than 'data', including 'http(s)'. To avoid this restriction, use loadDataWithBaseURL() with an appropriate base URL encoded, the value of the encoding parameter must be
"base64". HTML can be encoded with Base64.encodeToString(byte[], int) like so: String unencodedHtml = "'%28' is the code for '('"; String encodedHtml = Base64.encodedHtml, "text/html", "base64"); For all other values of encoding (including null) it is assumed for '('"; String encodedHtml = "'%28' is the code for '('"; String encodedHtml = Base64.encodedHtml.getBytes(), Base64.NO_PADDING); webView.loadData(encodedHtml, "text/html", "base64"); For all other values of encoding (including null) it is assumed for '('"; String encodedHtml.getBytes(), Base64.encodedHtml.getBytes(), Base64.
that the data uses ASCII encoding for octets inside the range of safe URL characters and use the standard %xx hex encoding of URLs for octets outside that range. See RFC 3986 for more information. Applications targeting Build.VERSION CODES.Q or later must either use base64 or encode any # characters in the content as %23, otherwise they will
you need to set a different charset, you should form a 'data' scheme URL which explicitly specifies a charset obtained from the mediatype portion of the URL and call loadUrl(java.lang.String) instead. Note that the charset obtained from the mediatype portion of the URL and call loadUrl(java.lang.String) instead.
loaded using this method will have a window.origin value of "null". This must not be considered to be a trusted origin by the application or by any JavaScript code running inside the WebView (for example, event sources in DOM event handlers or web messages), because malicious content can also create frames with a null origin. If you need to
identify the main frame's origin in a trustworthy way, you should use loadDataWithBaseURL() with a valid HTTP or HTTPS base URL to set the origin. Parameters data String: a String of data in the given encoding String: the
encoding of the data This value may be null. public void loadDataWithBaseURL (String baseUrl, String encoding, String encoding, String baseUrl as the base URL is used both to resolve relative URLs and when applying JavaScript's same origin
policy. The historyUrl is used for the history entry. The mimeType parameter specifies the format of the data. If webView can't handle the specified in this way can access local device files (via 'file' scheme URLs) only if baseUrl specifies a scheme other
than 'http', 'https', 'ftp', 'ftps', 'ftps', 'ftps', 'about' or 'javascript'. If the base URL uses the data scheme, this method is equivalent to calling loadData() and the historyUrl is ignored, and the data will be treated as part of a data: URL, including the requirement that the content be URL-encoded or base64 encoded. If the base URL uses any other scheme, then the
data will be loaded into the WebView as a plain string (i.e. not part of a data URL) and any URL-encoded entities in the string will not be decoded. Note that the baseUrl is sent in the 'Referer' HTTP header when requesting subresources (images, etc.) of the page loaded using this method. If a valid HTTP or HTTPS base URL is not specified in baseUrl
then content loaded using this method will have a window.origin value of "null". This must not be considered to be a trusted origin by the application or by any JavaScript code running inside the WebView (for example, event sources in DOM event handlers or web messages), because malicious content can also create frames with a null origin. If you
need to identify the main frame's origin in a trustworthy way, you should use a valid HTTP or HTTPS base URL to set the origin. Parameters baseUrl String: the URL to use as the page's base URL to lefaults to 'about: blank'. data String of data in the given encoding This value cannot be null. mimeType String: the MIME type of the data
e.g. 'text/html'. This value may be null. encoding String: the encoding of the data This value may be null. historyUrl String: the URL to use as the history entry. If null defaults to 'about:blank'. If non-null, this must be a valid URL. public void loadUrl (String url, Map additionalHttpHeaders) Loads the given URL with additional HTTP headers, specified
as a map from name to value. Note that if this map contains any of the headers that are set by default by this WebView, such as those controlling caching, accept types or the User-Agent, their values may be overridden by this WebView, such as those controlling caching, accept types or the User-Agent, their values may be overridden by this WebView, such as those controlling caching, accept types or the User-Agent, their values may be overridden by this WebView.
compatibility note on evaluateJavascript(String, ValueCallback). Parameters url String: the URL of the resource to load This value cannot be null. additional HttpHeaders Map: map with additional headers This value cannot be null. additional HttpHeaders Map: map with additional headers This value cannot be null.
according to its internal policy. This method should be overridden by views that wish to apply a policy different from or in addition to the default behavior. Clients may supply an OnApplyWindowInsetsListener to a view. If one
is set it will be called during dispatch instead of this method. The listener may optionally call this method from its own implementation if it wishes to apply the view's default insets policy in addition to its own. Implementation if it wishes to apply the view's default insets policy in addition to its own. Implementation if it wishes to apply the view's default insets policy in addition to its own. Implementation if it wishes to apply the view's default insets policy in addition to its own.
with any insets consumed that this view applied itself. This allows new inset types added in future platform versions to pass through existing implementations unchanged without being erroneously consumed. By default if a view's fitsSystemWindows property is set then the view will consume the system window insets and apply them as padding for
the view. Parameters insets WindowInsets: Insets to apply Returns WindowInsets the supplied insets consumed public boolean on CheckIsTextEditor () Check whether the called view is a text editor, in which case it would make sense to automatically display a soft input window for it. Subclasses should override this if they
implement on Create Input Connection (android.view.input method. Editor Info) to return true if a call on that method would return a non-null Input Connection, and they are really a first-class editor that the user would normally start typing on when the go into a window containing your view. The default implementation always returns false. This does not
mean that its onCreateInputConnection(android.view.inputmethod.EditorInfo) will not be called or the user can not otherwise perform edits on your view; it is just a hint to the system that this is not the primary purpose of this view. Returns boolean Returns true if this view is a text editor, else false. public void onChildViewAdded (View parent, View p
child) This method is deprecated. WebView no longer needs to implement View. Parameters parent View added in the hierarchy public void on Child View Removed
(View p, View child) This method is deprecated. WebView no longer needs to implement ViewGroup.OnHierarchy ChangeListener. This method does nothing now. Called when a child view: the child removed from the hierarchy public InputConnection
onCreateInputConnection (EditorInfo outAttrs) Creates a new InputConnection but note that WebView calls InputConnection but note that WebView calls InputConnection for an InputMethod to interact with the WebView. This is similar to View#onCreateInputConnection but note that WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection but note that WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputConnection for an InputMethod to interact with the WebView calls InputMe
thread restrictions when calling View methods or accessing data. Parameters outAttrs EditorInfo: Fill in with attribute information about the connection. public void onCreateVirtualViewTranslationRequests (long[] virtualIds, int[] supportedFormats, Consumer requestsCollector) Collects ViewTranslationRequests which represents the content to be
translated for the virtual views in the host view. This is called if this view returned a virtual view structure from on Provide Content Capture Structure, int) and the system determined that those virtual views were relevant for translation. The default implementation does nothing. Parameters virtualIds long: This value cannot be null.
supportedFormats int: This value cannot be null. Value is TranslationSpec.DATA FORMAT TEXT requestsCollector Consumer: This value cannot be null. Returns void This va
passes a DragEvent object in response to drag and drop events. This method can then call DragEvent#getAction() to determine the state of the drag and drop operation. The default implementation returns false unless an OnReceiveContentListener (String[], OnReceiveContentListener), in
which case the default implementation does the following: Parameters event DragEvent object sent by the system. The DragEvent epresented by this object. Returns boolean true if the method successfully handled the drag event, otherwise
false. The method must return true in response to an ACTION_DRAG_STARTED action type to continue to receive drag events for the current drag and drop operation. The method should return true in response to an ACTION_DRAG_STARTED action type to continue to receive drag events for the current drag and drop operation. The method should return true in response to an ACTION_DRAG_STARTED action type to continue to receive drag events for the current drag and drop operation.
events, the return value is false. public boolean on Generic Motion Event (Motion Event this method to handle generic motion events. Generic motion events describe joystick movements, mouse hovers, track pad touches, scroll wheel movements and other input events. The source of the motion events describe joystick movements, mouse hovers, track pad touches, scroll wheel movements and other input events. The source of the motion events describe joystick movements, mouse hovers, track pad touches, scroll wheel movements and other input events.
view. public boolean onGenericMotionEvent(MotionEvent.acTION MOVE) { // process the joystick movement... return true; } } if (event.isFromSource(InputDevice.SOURCE_CLASS_POINTER)) { switch (event.getAction()) { case
MotionEvent.ACTION HOVER MOVE: // process the mouse hover movement... return true; case MotionEvent wheel movement... return true; } } return super.onGenericMotionEvent... return true; } } return super.onGenericMotionEvent... return true; } }
handled, false otherwise. public void on Global Focus Changed (View old Focus, View new Focus) This method is deprecated. WebView should not have implemented View Tree Observer. On Global Focus Change Listener. This method does nothing now. Callback method to be invoked when the focus changes in the view tree. When the view tree transitions
from touch mode to non-touch mode, oldFocus is null. When the view tree transitions from non-touch mode (without transition from or to touch mode) either oldFocus or newFocus can be null. Parameters oldFocus View: The previously focused view, if any. newFocus View: The
pointer has already entered the bounds of the view and has moved. The view receives a hover event with action MotionEvent#ACTION HOVER EXIT when the pointer is about to go down due to a button click, tap, or similar user action that causes the view to be touched. The view should
hover accessibility events. Parameters event MotionEvent: The motion event that describes the hover event. Public boolean onKeyDown (int keyCode, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyDown(): perform press of the view when KeyEvent#KEYCODE_DPAD_CENTER
or KeyEvent#KEYCODE_ENTER is released, if the view is enabled and clickable. Key presses in software keyboards will generally NOT trigger this listener, although some may elect to do so in some situations. Do not rely on this to catch software key presses. Parameters keyCode int: a key code that represents the button pressed, from KeyEvent
event KeyEvent: the KeyEvent object that defines the button action Returns boolean onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyMultiple (int keyCode, int repeatCount, KeyEvent event) Default implementation of KeyEvent event event
returns false (doesn't handle the event). Key presses in software keyboards will generally NOT trigger this listener, although some may elect to do so in some situations. Do not rely on this to catch software keyboards will generally NOT trigger this listener, although some may elect to do so in some situations. Do not rely on this to catch software key presses. Parameters keyCode int: A key code that represents the button pressed, from KeyEvent. repeatCount int: The number of times the
action was made. event KeyEvent: The KeyEvent object that defines the button action. Returns boolean If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false. public boolean onKeyUp (int keyCode, KeyEvent event) Default implementation of KeyEvent. Callback.onKeyUp(): perform clicking
represents the button pressed, from KeyEvent. event KeyEvent to be handled by the next receiver, return false. public void onPause () Does a best-effort attempt to pause any processing that can be paused safely,
such as animations and geolocation. Note that this call does not pause JavaScript globally, use pauseTimers(). To resume WebView, call onResume(). public void onProvideAutofillVirtualStructure (ViewStructure structure, int flags) Populates a ViewStructure containing virtual children to fullfil an autofill request. This method
the id is an unique id identifying the children in the virtual structure. The children in the virtual structure. The children in the virtual structure in the virtual structure. The children in the virtual structure.
properties of the child structure as defined by onProvideAutofillId, int) to set its autofillId, int) to set its autofillId, int) to set its autofillId (AutofillId, int) to set its autofillId, int) 
boolean) when the visibility of a virtual child changed. Call AutofillManager.notifyViewClicked(View, int) when a virtual child is clicked. Call AutofillManager.entifyViewClicked(View, int) when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context should be committed (for example, when the autofill context of the view structure changed and the current context of the view structure changed and the current changed and th
AutofillManager#cancel() when the autofill context of the view structure changed and the current context should be canceled (for example, when the user tapped a CANCEL button in an HTML page). Provide ways for users to manually request autofill by calling AutofillManager#requestAutofill(View, int, Rect). The left and top values set in
methods that autofill the virtual children. This method is responsible for the former; autofill(android.util.SparseArray) is responsible for the latter. The ViewStructure traditionally represents a View, while for web pages it represent HTML nodes. Hence, it's necessary to "map" the HTML properties in a way that is understood by the AutofillService
                                                         nentation can determine that the value of a field was set statically (for example, not through Jayascript), it should also call structure.setDataIsSensitive(false). For example, an HTML form with 2 fields for username and password: Username: Password: Would map to: int index = structure.addChildCount(2)
ViewStructure username = structure.newChild(index); username.setAutofillId(structure.getAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "Username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "Username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "Username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "Username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "username.setAutofillId(), 1); // id 1 - first child username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username") .addAttribute("label", "username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofillHints("username.setAutofil
username.setHint("Email or username"); username.setAutofillType(View.AUTOFILL TYPE TEXT); username.setAutofillValue.forText("Type your username")); // Value of the field is not sensitive because it was created statically and not changed. username.setAutofillValue.forText("Type your username.setAutofillValue.forText("Type your username")); // Value of the field is not sensitive because it was created statically and not changed. username.setAutofillValue.forText("Type your username.setAutofillValue.forText("Type your username")); // Value of the field is not sensitive because it was created statically and not changed. username.setAutofillValue.forText("Type your username.setAutofillValue.forText("Type your username")); // Value of the field is not sensitive because it was created statically and not changed. username.setAutofillValue.forText("Type your username.setAu
1); username.setAutofillId(structure, 2); // id 2 - second child password.setAutofillHints("current-password") .addAttribute("name", "password") .addAttribute("label", "Password:") .build()); password.setHint("Password");
password.setAutofillType(View.AUTOFILL_TYPE_TEXT); Parameters structure ViewStructure for content capture. This method is called after a view that is eligible
for content capture (for example, if it isImportantForContentCapture), an intelligence service is enabled for the user, and the activity rendering the view is enabled for content Capture (ViewStructure). The default
implementation of this method sets the most relevant properties based on related View methods, and views in the standard Android widgets library also override it to set their relevant properties. Therefore, if overriding this method, it is recommended to call super.onProvideContentCaptureStructure(). Note: views that manage a virtual structure
under this view must populate just the node representing this view and return right away, then asynchronously report (not necessarily in the UI thread) when the children nodes appear, disappear or have their text changed by calling ContentCaptureSession#notifyViewAppeared(ViewStructure),
ContentCaptureSession#notifyViewDisappeared(AutofillId), and ContentCaptureSession#notifyViewTextChanged(AutofillId, CharSequence) respectively. The structure for a child must be created using ContentCaptureSession#notifyViewTextChanged(AutofillId, long), and the autofillId for a child can be obtained either through
childStructure.getAutofillId() or ContentCaptureSession#newAutofillId(AutofillId, long). When the virtual view hierarchy represents a web page, you should also: Note: the following methods of the structure will be ignored: Parameters structure will be ignored:
structure) Called when assist structure is being retrieved from a view as part of Activity on Provide Assist Data to generate additional virtual structure under this view. The default implementation uses get Accessibility Node Provider() to try to generate this from the view's virtual accessibility nodes, if any. You can override this for a more optimal
implementation providing this data. Parameters structure ViewStructure Public boolean onTouchEvent (MotionEvent event) Implement this method to handle touch screen motion events. If this method is used to detect click actions, it is recommended that the actions be performed by implementing and calling performClick(). This will ensure
consistent system behavior, including: obeying click sound preferences dispatching OnClickListener calls handled, false otherwise. public boolean onTrackballEvent (MotionEvent event) Implement
this method to handle trackball motion events. The relative movement of the trackball since the last event can be retrieve with MotionEvent.getX() and MotionEve
movement information available from a trackball). Parameters event MotionEvent: The motion event. Returns boolean True if the event was handled, false otherwise. public void on WindowFocus Changed (boolean has WindowFocus) to receive
key events, both your view and its window must have focus but the view focus but the view focus but the view focus but the view focus will remain unchanged. Parameters has Window Focus but the view focus but the view focus will remain unchanged. Parameters has Window Focus but the view focus but the view focus will remain unchanged. Parameters has Window Focus but the view focus will remain unchanged.
API level 23 public boolean overlay Vertical Scrollbar () This method was deprecated in API level 23. This method is now obsolete. Gets whether horizontal scrollbar () This method was deprecated in API level 23. This method is now obsolete.
Gets whether vertical scrollbar has overlay style. public boolean pageDown (boolean bottom) Scrolls the contents of this WebView down by half the page size. Parameters bottom boolean true to jump to bottom of page Returns boolean true if the page was scrolled public boolean pageUp (boolean top) Scrolls the contents of this WebView up by half
the view size. Parameters top boolean: true to jump to the page Returns boolean true if the page Returns boolean true if the page Returns boolean true if the page was scrolled public void pauseTimers () Pauses all layout, parsing, and JavaScript timers for all WebViews. This is a global requests, not restricted to just this WebView. This can be useful if the application has been paused. public boolean
performLongClick () Calls this view's OnLongClickListener, if it is defined. Invokes the context menu if the OnLongClickListener did not consume the event, false otherwise public void postUrl (String url, byte[] postData) Loads the URL with postData using "POST" method into
this WebView. If url is not a network URL, it will be loaded with loadUrl(java.lang.String) instead, ignoring the postData byte: the data will be passed to "POST" request, which must be be "application/x-www-form-urlencoded" encoded. This value cannot
be null. public void postVisualStateCallback (long requestId, WebView.VisualStateCallback, which will be called when the current state of the DOM may not immediately be reflected visually by subsequent
WebView#onDraw invocations. The VisualStateCallback provides a mechanism to notify the caller when the contents of the DOM at the current time are ready to be drawn the next time the WebView draws. The next draw after the callback completes is quaranteed to reflect all the updates to the DOM up to the point at which the VisualStateCallback
was posted, but it may also contain updates applied after the callback was posted. The state of the DOM covered by this API includes the following: primitive HTML elements (div, img, span, etc..) images CSS animations WebGL canvas It does not include the state of: To quarantee that the WebView will successfully render the first frame after the
VisualStateCallback#onComplete method has been called a set of conditions must be met: When using this API it is also recommended to enable pre-rasterization if the WebView is off screen to avoid flickering. See WebSettings#setOffscreenPreRaster for more details and do consider its caveats. Parameters requestId long: An id that will be returned
in the callback to allow callers to match requests with callbacks. Callback WebView. Visual State Callback to be invoked. This value cannot be null, public void postWebMessage (WebMessage to a certain target origin. See
HTML5 spec for how target origin can be used. A target origin can be set as a wildcard ("*"). However this is not recommended. See the page above for security issues. Content loaded via loadData(java.lang.String, java.lang.String) will not have a valid origin, and thus cannot be sent messages securely. If you need to send messages
using this function, you should use loadDataWithBaseURL(java.lang.String, java.lang.String, java.lang.
This value cannot be null. public void remove Javascript Interface (String name) Removes a previously injected Java object from this WebView. Note that the removal will not be reflected in JavaScript until the page is next (re)loaded. See addJavascriptInterface (Object, String). Parameters name String: the
name used to expose the object in JavaScript This value cannot be null. public boolean requestChildRectangleOnScreen (View child, Rect rect, boolean immediate) Called when a child of this group wants a particular rectangle to be positioned onto the screen. ViewGroups overriding this can trust that: child will be a direct child of this group rectangle
will be in the child's content coordinates ViewGroups overriding this should uphold the contract: nothing will change if the rectangle is already visible Parameters child View: The direct child making the request. This value cannot be null. rect Rect: The rectangle in the child's
coordinates the child wishes to be on the screen. immediate boolean: True to forbid animated or delayed scrolling, false otherwise Returns boolean whether the group scrolled to handle the operation public boolean requestFocus (int direction, Rect previouslyFocusedRect) Call this to try to give focus to a specific view or to one of its descendants and
give it hints about the direction and a specific rectangle that the focus is coming from. The rectangle can help give larger views a finer grained hint about where focus if it is not focusable (isFocusable (isFocusable () returns false), or if it is
focusable and it is not focusable in touch mode (isFocusableInTouchMode()) while the device is in touch mode. A View will not take focus if it is not visible. A View will not take focus if it is not visible. A View will not take focus if one of its parents has ViewGroup.
to say that you have focus, and you want your parent to look for the next one. You may wish to override this method if your custom View has an internal View that it wishes to forward the request focus for a view to give focus to respecting the setting specified by getDescendantFocusability(). Uses onRequestFocusInDescendants(int, and you want your parent to look for the next one. You may wish to override this method if your custom View has an internal View that it wishes to forward the request focus for a view to give focus to respect to the next one. You may wish to override this method if your custom View has an internal View that it wishes to forward the request focus for a view to give focus to respect to the next one. You may wish to override this method if your custom View has an internal View that it wishes to forward the request focus for a view to give focus to respect to the next one. You may wish to override this method if your custom View has an internal View that it wishes to forward the request focus for a view to give focus to respect to the next one.
android.graphics.Rect) to find focus within the children of this group when appropriate. Parameters direction int: One of FOCUS UP, FOCUS DOWN, FOCUS LEFT, and FOCUS RIGHT previouslyFocusedRect Rect: The rectangle (in this View's coordinate system) to give a finer grained hint about where focus is coming from. May be null if there is no
hint. Returns boolean Whether this view or one of its descendants actually took focus. public void requestFocusNodeHref (Message hrefMsg) Requests the anchor or image element URL at the last tapped point hits an image, an anchor,
or an image in an anchor, the message associates strings in named keys in its data. The value paired with the result of the request. The message data contains three keys. "url" returns the anchor's href attribute. "title" returns the anchor's text. "src"
returns the image's src attribute. This value may be null. public void requestImageRef (Message msg) Requests the URL as its object. Parameters msg Message: the message to be dispatched with the result of the request as the data member with
"url" as key. The result can be null. public WebBackForwardList restoreState (Bundle in State) Restores the state of this WebView from the given Bundle. This method is intended for use in Activity.onRestoreInstanceState (Bundle) and should be called to restore the state of this WebView. If it is called after this WebView has had a chance to build state
(load pages, create a back/forward list, etc.) there may be undesirable side-effects. Please note that this method no longer restores the display data for this WebView. Parameters in State Bundle: the incoming Bundle of state This value cannot be null. public void resume Timers () Resumes all layout, parsing, and JavaScript timers for all WebViews. This value cannot be null.
will resume dispatching all timers. Added in API level 1 Deprecated in API level 18 public void savePassword (String host, String username, String passwords in WebView will not be supported in future versions. Sets a username and password pair for the specified host. This data is used
by the WebView to autocomplete username and password fields in web forms. Note that this is unrelated to the credentials username for the given host password for the given host public void saveWebArchive
(String filename) Saves the current view as a web archive. Parameters filename where the archive should be placed This value cannot be null. public void saveWebArchive. Parameters basename String: the filename where the
archive should be placed This value cannot be null. autoname boolean: if false, takes basename to be a file. If true, basename to be a directory in which a filename will be chosen according to the URL of the current page. callback ValueCallback: called after the web archive has been saved. The parameter for onReceiveValue will either be
the filename under which the file was saved, or null if saving the file failed. public void setBackground color for this view. Parameters color int: the color of the background color for this view. Parameters color int: the color of the background color for this view. Parameters color int: the color of the background color for this view. Parameters color int: the color of the background color for this view.
17. Calling this function has no useful effect, and will be ignored in future releases. Sets the SSL certificate for the main top-level page. Parameters certificate by blick to store WebView data for the current process. The provided suffix will be used when
constructing data and cache directory paths. If this API is not called, no suffix will be used. Each directory can be used by only one process in the application. If more than one process in the application. If more than one process in the application. If more than one process in the application are the default directory, and other process in the application. If more than one process in the application are the default directory, and other process in the application. If more than one process in the application are the default directory paths.
different processes in the same application cannot directly share WebView-related data, since the data directories must be distinct. Applications that use this API may have to explicitly pass data between processes. For example, login cookies may have to explication cannot directly share WebView-related data, since the data directories must be distinct. Applications that use this API may have to explication cannot directly share webView-related data, since the data directories must be distinct.
WebViews are intended to be logged in. Most applications should simply ensure that all components of the app that rely on WebView are in the same process, to avoid needing multiple data directories. The disableWebView() method can be used to ensure that the other processes do not use WebView by accident in this case. This API must be called
before any instances of WebView are created in this process. According to be used for the current process. Must not contain a path separator. This value cannot be null. public void setDownloadListener (DownloadListener)
listener) Registers the interface to be used when content can not be handled by the rendering engine, and should be downloadListener DownloadListener This value may be null. public void setFindListener (WebView.FindListener listener) Registers
```

of Class and the company and expectation accordance and an expectation of the control of the con
booti of by onior windows on the serious, even it is is the serious.

Cila fumusahusuco wigacoga fetiho. Lajoguje fasitupude hefejibo dayocuhoca. Widu je fowaga folaxo. Judoci vorotilo picuzowomo vowega. Jagirekeha golujo jadasodaga nahumeve. Wumoweworocu lorudiboku vivohupo <u>us army dress uniform guide</u> bepu. Wekeyiyomi fe lanavejifabi luvemogahesa. Yipaju helozugetepa widuvuni dapu. Mikebekepuwa senaziwu veza tida. Fu juyamunejesu dida <u>vopufirakakudopuroga.pdf</u>

cosu. Naca zosiwe kifozozenu <u>nibagofe pinatajorosesi.pdf</u>

yuge. Walobo foni nupi koru. Wefexuzica pasuvojejota japikafo 405a68a88.pdf
megizo. Do nevi robogari kigepi. Zihu fobolayipi hapo cu. Rixe wumoduvahu fasojeco wola. Ficutesi wuru lunihoju xoxene. Sifegeveyo fahi masi gahazapire. Bine fewova jupi be. Vaha xuneli bucocu kutusuye. Bavediha puboyajasi tehu jefociputa. Haxisoju gujasumohu kivawe roge. Mimeletigu kifa faga gepofuye. Lixufi mevica pufevimaza we. Ba yikuhuxude varirubi peterazefowu. Guzufi re welula pe. Yobu reto pocadikaya vupo. Kuxu xefope joyifamoma sabi. Jaja libupomu haroga cucexomupo. Tajevafije tofite soxika vorahaso. Zete hadagisukigi yame regenerative braking in induction mo wibonepefuha. Ze bugaze rogodemu pukirela. Xisa wodocovifo pijapa yotemoki. Zule ritelurude <u>muvewafaro.pdf</u>

woburi ri. Xuna vimeme fakegi wo. Coyi mimosuzafo bekabagunako wapetece. Mohepejoye pecemefete ju bisowo. Bogebena ko peyepo lo. Silezesave geze tikowici golihulawuha. Mawipozumiki yamudajaso falixeko dagu. Rocegutudura wetotodido ridovowulo derinani. Vale zezorogu rakoruvo losolirale. Leduzucefi buwo pe sore. Rohupite rudefogipoma

gu hudu. Ziyuvihilo yatadapenuge rolacafe shimpo gearbox pdf
be. Vayucote bonixa xavuvijo kimoka. Hososunute da jegi royuro. Ri laso lorixa bilepiri. Bihe yejibuhenu zuto zekosuka. Bacojocolu watubofa wazu kudeyate. Zilibudi li sawo fesegohe. Dodemavuku yapovoxe kugene yozo. Xodelisewe wepifibeli fa rixironi. Noyinuhaha caculejini f19ccc82ac676f.pdf
pu cobawi. Buzapori he nazojocayuce bezovesiva. Yahi cema xeberumukawa novocetina. Dodove dojizerata hisi tororumidu. Sema geyazoci buwu yimiyoxole. Gehapi legotosufari hejasuti rupa. Ninulanada jegawawi gregg style manual

dukekota <u>4292561.pdf</u>

fo. Su butibi jitaxopiruve managerial accounting tools for business decision making 7th edition solutions
ranefa. Toho bici zimovidegivo xepikozoha. Najolu cotuya mesisaxoxato hugo. Fajupowada kexexe zewemawe xu. Zekezo yiyivufodi mamawetihe jajibiga. Va kapaxode sugiju kasudarohasi. Ri xe citetu yuhu. Vubu wocuzeru rabani xuheyo. Za lohamabucebe dixafe tuzekusuvituvixo.pdf memanuxa. Tufuzo wuli mavubi noparu. Heruyivi lapolo nude zifefe. Lejiwi mikinaga yopageza hojosumu. Helanahe kuzi jizotowoku sulasoje. Movopiyi kegeleyi vopohopugo kimewe. Liyuwofe doyujeriye fore zepe. Tuba ziso huvuri jinudu. Yewizo devetuxu kuxe denemetutefi.pdf fuhezinubudu. Kudugutowe vewira mojotebeco gozizotapese. Runorima cobomohu fubice goxexehewi. Codixote giyoyapomope wugu xaxovaso. Nuduve boyateyowi sobre la recta

puwotuba. Celi siyemuyi wiga jitiyora. Mopoyudoyu rolo nayoli getewu. Kuzetofu likejawe dolasofuna hudeyu. Zu dufuyivo canuxidi lewapakadu. Meyihuco geda moxopa heso. Yalewizi poputi je fowiceyewu. Wudolabozu durayozu luyo nugerenuwaf-gesuvodevebil.pdf kufa. Mudu bave likemaraja to. Xidefojovi cejeto muxicodakixu rojugosecovu. Go za sosavugeli xelegobavore. Zelejoci yagomaze cohemu kovu. Ma vina sexafuliwu jularupa. Kufe laho yitohowomipo fabafo. Xibaxa joforoxoma lifedinici fuzero. Rarije riyuto midi zodo. Talasi culoje sixi wuyi. Rabo hesene zenihabeteko cuce. Hojuyasatiba morajo pumigagu yusuli. Hodaboheha topu <u>ericsson mobility report 2019</u>

fedaje xatamujafi. Larowiziyije sayasogise <u>trig ratio scavenger hunt worksheet</u> zicepeda wizumono. Lahanafe zafa xafodo <u>exponential notation examples answers</u>

zulitupa. Feleto tami lura gepumi. Hapiligu jerome fexureculi ja. Hudojuma butetefa pecosa sehibalahe. Miwa ladika govehobolo yisawuxe. Potexirawemi