public class

Summary: Inherited Constants | Ctors | Methods | Inherited Methods | [Expand All]

Added in API level 1

SpannableStringBuilder

extends Object

implements Editable GetChars Spannable Appendable CharSequence

java.lang.Object
  ↳android.text.SpannableStringBuilder

Class Overview

This is the class for text whose content and markup can both be changed.

Summary

Inherited Constants [Expand]

▸ From interface android.text.Spanned

Public Constructors

SpannableStringBuilder ()
Create a new SpannableStringBuilder with empty contents

SpannableStringBuilder (CharSequence text)
Create a new SpannableStringBuilder containing a copy of the specified text, including its spans if any.

SpannableStringBuilder (CharSequence text, int start, int end)
Create a new SpannableStringBuilder containing a copy of the specified slice of the specified text, including its spans if any.

Public Methods

SpannableStringBuilder	append (char text) Convenience for append(String.valueOf(text)).
SpannableStringBuilder	append (CharSequence text, int start, int end) Convenience for replace(length(), length(), text, start, end)
SpannableStringBuilder	append (CharSequence text) Convenience for replace(length(), length(), text, 0, text.length())
char	charAt (int where) Return the char at the specified offset within the buffer.
void	clear () Convenience for replace(0, length(), "", 0, 0)
void	clearSpans () Removes all spans from the Editable, as if by calling `removeSpan(Object)` on each of them.
SpannableStringBuilder	delete (int start, int end) Convenience for replace(st, en, "", 0, 0)
abstract void	drawText (Canvas c, int start, int end, float x, float y, Paint p) Just like `drawText(char[], int, int, float, float, Paint)`.
boolean	equals (Object o) Compares this instance with the specified object and indicates if they are equal.
void	getChars (int start, int end, char[] dest, int destoff) Copy the specified range of chars from this buffer into the specified array, beginning at the specified offset.
InputFilter[]	getFilters () Returns the array of input filters that are currently applied to changes to this Editable.
int	getSpanEnd (Object what) Return the buffer offset of the end of the specified markup object, or -1 if it is not attached to this buffer.
	getSpanFlags (Object what)

int	Return the flags of the end of the specified markup object, or 0 if it is not attached to this buffer.
	getSpanStart (Object what)
int	Return the buffer offset of the beginning of the specified markup object, or -1 if it is not attached to this buffer.
	getSpans (int queryStart, int queryEnd, Class<T> kind)
<T> T[]	Return an array of the spans of the specified type that overlap the specified range of the buffer.
	getTextRunCursor (int contextStart, int contextEnd, int flags, int offset, int cursorOpt, Paint p)
int	*This method was deprecated in API level 12. This is an internal method, refrain from using it in your code*
	getTextWidths (int start, int end, float[] widths, Paint p)
abstract int	Just like `getTextWidths(char[], int, int, float[])`.
int	hashCode ()
	Returns an integer hash code for this object.
	insert (int where, CharSequence tb, int start, int end)
SpannableStringBuilder	Convenience for replace(where, where, text, start, end)
	insert (int where, CharSequence tb)
SpannableStringBuilder	Convenience for replace(where, where, text, 0, text.length());
int	length ()
	Return the number of chars in the buffer.
	measureText (int start, int end, Paint p)
abstract float	Just like `measureText(char[], int, int)`.
	nextSpanTransition (int start, int limit, Class kind)
int	Return the next offset after `start` but less than or equal to `limit` where a span of the specified type begins or ends.
	removeSpan (Object what)
void	Remove the specified markup object from the buffer.
	replace (int start, int end, CharSequence tb)
SpannableStringBuilder	Convenience for replace(st, en, text, 0, text.length())
	replace (int start, int end, CharSequence tb, int tbstart, int tbend)
SpannableStringBuilder	Replaces the specified range (`st`…`en`) of text in this Editable with a copy of the slice `start`…`end` from `source`.
	setFilters (InputFilter[] filters)
void	Sets the series of filters that will be called in succession whenever the text of this Editable is changed, each of which has the opportunity to limit or transform the text that is being inserted.
	setSpan (Object what, int start, int end, int flags)
void	Mark the specified range of text with the specified object.
	subSequence (int start, int end)
CharSequence	Return a new CharSequence containing a copy of the specified range of this buffer, including the overlapping spans.
String	toString ()
	Return a String containing a copy of the chars in this buffer.
static SpannableStringBuilder	valueOf (CharSequence source)

Inherited Methods [Expand]

- From class java.lang.Object
- From interface android.text.Editable
- From interface android.text.GetChars
- From interface android.text.Spannable
- From interface android.text.Spanned
- From interface java.lang.Appendable
- From interface java.lang.CharSequence

Public Constructors

public **SpannableStringBuilder** ()

Added in API level 1

Create a new SpannableStringBuilder with empty contents

public **SpannableStringBuilder** (CharSequence text)

Added in API level 1

Create a new SpannableStringBuilder containing a copy of the specified text, including its spans if any.

public **SpannableStringBuilder** (CharSequence text, int start, int end)

Added in API level 1

Create a new SpannableStringBuilder containing a copy of the specified slice of the specified text, including its spans if any.

Public Methods

public SpannableStringBuilder **append** (char text)

Added in API level 1

Convenience for append(String.valueOf(text)).

Parameters

text the character to append.

Returns

this `Appendable`.

public SpannableStringBuilder **append** (CharSequence text, int start, int end)

Added in API level 1

Convenience for replace(length(), length(), text, start, end)

Parameters

text the character sequence to append.

start the first index of the subsequence of `csq` that is appended.

end the last index of the subsequence of `csq` that is appended.

Returns

this `Appendable`.

public SpannableStringBuilder **append** (CharSequence text)

Added in API level 1

Convenience for replace(length(), length(), text, 0, text.length())

Parameters

text the character sequence to append.

Returns

this `Appendable`.

public char **charAt** (int where)

Added in API level 1

Return the char at the specified offset within the buffer.

public void **clear** ()

Added in API level 1

Convenience for replace(0, length(), "", 0, 0)

public void **clearSpans** ()

Added in API level 1

Removes all spans from the Editable, as if by calling `removeSpan(Object)` (/reference/android/text/Spannable.html#removeSpan(java.lang.Object)) on each of them.

public SpannableStringBuilder **delete** (int start, int end)

Added in API level 1

Convenience for replace(st, en, "", 0, 0)

public abstract void **drawText** (Canvas c, int start, int end, float x, float y, Paint p)

Just like `drawText(char[], int, int, float, float, Paint)` (/reference/android/graphics/Canvas.html#drawText(char[], int, int, float, float, android.graphics.Paint)).

public boolean **equals** (Object o)

Added in API level 1

Compares this instance with the specified object and indicates if they are equal. In order to be equal, o must represent the same object as this instance using a class-specific comparison. The general contract is that this comparison should be reflexive, symmetric, and transitive. Also, no object reference other than null is equal to null.

The default implementation returns `true` only if `this == o`. See Writing a correct `equals` method (/reference/java/lang/Object.html#writing_equals) if you intend implementing your own `equals` method.

The general contract for the `equals` and `hashCode()` (/reference/java/lang/Object.html#hashCode()) methods is that if `equals` returns `true` for any two objects, then `hashCode()` must return the same value for these objects. This means that subclasses of `Object` usually override either both methods or neither of them.

Parameters

o the object to compare this instance with.

Returns

`true` if the specified object is equal to this `Object`; `false` otherwise.

public void **getChars** (int start, int end, char[] dest, int destoff)

Added in API level 1

Copy the specified range of chars from this buffer into the specified array, beginning at the specified offset.

public InputFilter[] **getFilters** ()

Added in API level 1

Returns the array of input filters that are currently applied to changes to this Editable.

public int **getSpanEnd** (Object what)

Added in API level 1

Return the buffer offset of the end of the specified markup object, or -1 if it is not attached to this buffer.

public int **getSpanFlags** (Object what)

Added in API level 1

Return the flags of the end of the specified markup object, or 0 if it is not attached to this buffer.

public int **getSpanStart** (Object what)

Added in API level 1

Return the buffer offset of the beginning of the specified markup object, or -1 if it is not attached to this buffer.

public T[] **getSpans** (int queryStart, int queryEnd, Class<T> kind)

Added in API level 1

Return an array of the spans of the specified type that overlap the specified range of the buffer. The kind may be Object.class to get a list of all the spans regardless of type.

public int **getTextRunCursor** (int contextStart, int contextEnd, int flags, int offset, int cursorOpt, Paint p)

Added in API level 11

This method was deprecated in API level 12.
This is an internal method, refrain from using it in your code

Returns the next cursor position in the run. This avoids placing the cursor between surrogates, between characters that form conjuncts, between base characters and combining marks, or within a reordering cluster.

The context is the shaping context for cursor movement, generally the bounds of the metric span enclosing the cursor in the direction of movement. `contextStart`, `contextEnd` and `offset` are relative to the start of the string.

If cursorOpt is CURSOR_AT and the offset is not a valid cursor position, this returns -1. Otherwise this will never return a value before contextStart or after contextEnd.

Parameters

contextStart	the start index of the context
contextEnd	the (non-inclusive) end index of the context
flags	either DIRECTION_RTL or DIRECTION_LTR
offset	the cursor position to move from
cursorOpt	how to move the cursor, one of CURSOR_AFTER, CURSOR_AT_OR_AFTER, CURSOR_BEFORE, CURSOR_AT_OR_BEFORE, or CURSOR_AT
p	the Paint object that is requesting this information

Returns

the offset of the next position, or -1

public abstract int **getTextWidths** (int start, int end, float[] widths, Paint p)

Just like `getTextWidths(char[], int, int, float[])` (/reference/android/graphics/Paint.html#getTextWidths(char[], int, int, float[])).

public int **hashCode** ()

Added in API level 1

Returns an integer hash code for this object. By contract, any two objects for which `equals(Object)` (/reference/java/lang/Object.html#equals(java.lang.Object)) returns `true` must return the same hash code value. This means that subclasses of `Object` usually override both methods or neither method.

Note that hash values must not change over time unless information used in equals comparisons also changes.

See Writing a correct `hashCode` method (/reference/java/lang/Object.html#writing_hashCode) if you intend implementing your own `hashCode` method.

Returns

this object's hash code.

public SpannableStringBuilder **insert** (int where, CharSequence tb, int start, int end)

Added in API level 1

Convenience for replace(where, where, text, start, end)

public SpannableStringBuilder **insert** (int where, CharSequence tb)

Added in API level 1

Convenience for replace(where, where, text, 0, text.length());

public int **length** ()

Added in API level 1

Return the number of chars in the buffer.

Returns

the number of characters.

public abstract float **measureText** (int start, int end, Paint p)

Just like `measureText(char[], int, int)` (/reference/android/graphics/Paint.html#measureText(char[], int, int)).

public int **nextSpanTransition** (int start, int limit, Class kind)

Added in API level 1

Return the next offset after `start` but less than or equal to `limit` where a span of the

specified type begins or ends.

public void **removeSpan** (Object what) Added in API level 1

Remove the specified markup object from the buffer.

public SpannableStringBuilder **replace** (int start, int end, CharSequence tb) Added in API level 1

Convenience for replace(st, en, text, 0, text.length())

public SpannableStringBuilder **replace** (int start, int end, CharSequence tb, int tbstart, int tbend) Added in API level 1

Replaces the specified range (`st`…`en`) of text in this Editable with a copy of the slice `start`…`end` from `source`. The destination slice may be empty, in which case the operation is an insertion, or the source slice may be empty, in which case the operation is a deletion.

Before the change is committed, each filter that was set with `setFilters(InputFilter[])` (/reference/android/text/Editable.html#setFilters(android.text.InputFilter[])) is given the opportunity to modify the `source` text.

If `source` is Spanned, the spans from it are preserved into the Editable. Existing spans within the Editable that entirely cover the replaced range are retained, but any that were strictly within the range that was replaced are removed. As a special case, the cursor position is preserved even when the entire range where it is located is replaced.

Returns

a reference to this object.

public void **setFilters** (InputFilter[] filters) Added in API level 1

Sets the series of filters that will be called in succession whenever the text of this Editable is changed, each of which has the opportunity to limit or transform the text that is being inserted.

public void **setSpan** (Object what, int start, int end, int flags) Added in API level 1

Mark the specified range of text with the specified object. The flags determine how the span will behave when text is inserted at the start or end of the span's range.

public CharSequence **subSequence** (int start, int end) Added in API level 1

Return a new CharSequence containing a copy of the specified range of this buffer, including the overlapping spans.

Parameters

start	the start offset of the sub-sequence. It is inclusive, that is, the index of the first character that is included in the sub-sequence.
end	the end offset of the sub-sequence. It is exclusive, that is, the index of the first character after those that are included in the sub-sequence

Returns

the requested sub-sequence.

public String **toString** () Added in API level 1

Return a String containing a copy of the chars in this buffer.

Returns

a printable representation of this object.

public static SpannableStringBuilder **valueOf** (CharSequence source) Added in API level 1