TextInput

A foundational component for inputting text into the app via a keyboard. Props provide configurability for several features, such as auto-correction, auto-capitalization, placeholder text, and different keyboard types, such as a numeric keypad.

The simplest use case is to plop down a `TextInput` and subscribe to the `onChangeText` events to read the user input. There are also other events, such as `onSubmitEditing` and `onFocus` that can be subscribed to. A simple example:

Two methods exposed via the native element are .focus() and .blur() that will focus or blur the TextInput programmatically.

Note that some props are only available with `multiline={true/false}`. Additionally, border styles that apply to only one side of the element (e.g., `borderBottomColor`, `borderLeftWidth`, etc.) will not be applied if `multiline=false`. To achieve the same effect, you can wrap your `TextInput` in a `View`:

`TextInput` has by default a border at the bottom of its view. This border has its padding set by the background image provided by the system, and it cannot be changed. Solutions to avoid this is to either not set height explicitly, case in which the system will take care of displaying the border in the correct position, or to not display the border by setting `underlineColorAndroid` to transparent.

Note that on Android performing text selection in input can change app's activity `windowSoftInputMode` param to `adjustResize`. This may cause issues with components that have position: 'absolute' while keyboard is active. To avoid this behavior either specify `windowSoftInputMode` in AndroidManifest.xml (https://developer.android.com/guide/topics/manifest/activity-element.html) or control this param programmatically with native code.

Props

- View props...

- `allowFontScaling`
- `autoCapitalize`
- `autoCorrect`
- `autoFocus`
- `blurOnSubmit`
- `caretHidden`
- `clearButtonMode`
- `clearTextOnFocus`
- `contextMenuHidden`
- `dataDetectorTypes`
- `defaultValue`
- `disableFullscreenUI`
- `editable`
- `enablesReturnKeyAutomatically`
- `inlineImageLeft`
- `inlineImagePadding`
- `keyboardAppearance`

- `keyboardType`
- `maxLength`
- `multiline`
- `numberOfLines`
- `onBlur`
- `onChange`
- `onChangeText`
- `onContentSizeChange`
- `onEndEditing`
- `onFocus`
- `onKeyPress`
- `onLayout`
- `onScroll`
- `onSelectionChange`
- `onSubmitEditing`
- `placeholder`
- `placeholderTextColor`
- `returnKeyLabel`
- `returnKeyType`
- `scrollEnabled`
- `secureTextEntry`
- `selection`
- `selectionColor`
- `selectionState`
- `selectTextOnFocus`
- `spellCheck`
- `textContentType`
- `style`
- `textBreakStrategy`
- `underlineColorAndroid`
- `value`

Methods

- `clear`
- `isFocused`

Reference

Props

`allowFontScaling`

Specifies whether fonts should scale to respect Text Size accessibility settings. The default is `true`.

TYPE	REQUIRED
bool	No

`autoCapitalize`

Can tell `TextInput` to automatically capitalize certain characters. This property is not supported by some keyboard types such as `name-phone-pad`.

- `characters`: all characters.
- `words`: first letter of each word.
- `sentences`: first letter of each sentence (*default*).
- `none`: don't auto capitalize anything.

TYPE	REQUIRED
enum('none', 'sentences', 'words', 'characters')	No

`autoCorrect`

If `false`, disables auto-correct. The default value is `true`.

TYPE	REQUIRED
bool	No

`autoFocus`

If `true`, focuses the input on `componentDidMount`. The default value is `false`.

TYPE	REQUIRED
bool	No

`blurOnSubmit`

If `true`, the text field will blur when submitted. The default value is true for single-line fields and false for multiline fields. Note that for multiline fields, setting `blurOnSubmit` to `true` means that pressing return will blur the field and trigger the `onSubmitEditing` event instead of inserting a newline into the field.

TYPE	REQUIRED
bool	No

`caretHidden`

If `true`, caret is hidden. The default value is `false`.

TYPE	REQUIRED
bool	No

`clearButtonMode`

When the clear button should appear on the right side of the text view. This property is supported only for single-line TextInput component. The default value is `never`.

TYPE	REQUIRED	PLATFORM
enum('never', 'while-editing', 'unless-editing', 'always')	No	iOS

`clearTextOnFocus`

If `true`, clears the text field automatically when editing begins.

TYPE	REQUIRED	PLATFORM
bool	No	iOS

`contextMenuHidden`

If `true`, context menu is hidden. The default value is `false`.

TYPE	REQUIRED
bool	No

`dataDetectorTypes`

Determines the types of data converted to clickable URLs in the text input. Only valid if `multiline={true}` and `editable={false}`. By default no data types are detected.

You can provide one type or an array of many types.

Possible values for `dataDetectorTypes` are:

- `'phoneNumber'`
- `'link'`
- `'address'`
- `'calendarEvent'`
- `'none'`

- `'all'`

TYPE	REQUIRED	PLATFORM
enum('phoneNumber', 'link', 'address', 'calendarEvent', 'none', 'all'), ,array of enum('phoneNumber', 'link', 'address', 'calendarEvent', 'none', 'all')	No	iOS

`defaultValue`

Provides an initial value that will change when the user starts typing. Useful for simple use-cases where you do not want to deal with listening to events and updating the value prop to keep the controlled state in sync.

TYPE	REQUIRED
string	No

`disableFullscreenUI`

When `false`, if there is a small amount of space available around a text input (e.g. landscape orientation on a phone), the OS may choose to have the user edit the text inside of a full screen text input mode. When `true`, this feature is disabled and users will always edit the text directly inside of the text input. Defaults to `false`.

TYPE	REQUIRED	PLATFORM
bool	No	Android

`editable`

If `false`, text is not editable. The default value is `true`.

TYPE	REQUIRED
bool	No

`enablesReturnKeyAutomatically`

If `true`, the keyboard disables the return key when there is no text and automatically enables it when there is text. The default value is `false`.

TYPE	REQUIRED	PLATFORM
bool	No	iOS

`inlineImageLeft`

If defined, the provided image resource will be rendered on the left. The image resource must be inside

`/android/app/src/main/res/drawable` and referenced like

```
<TextInput
 inlineImageLeft='search_icon'
/>
```

TYPE	REQUIRED	PLATFORM
string	No	Android

`inlineImagePadding`

Padding between the inline image, if any, and the text input itself.

TYPE	REQUIRED	PLATFORM
number	No	Android

`keyboardAppearance`

Determines the color of the keyboard.

TYPE	REQUIRED	PLATFORM
enum('default', 'light', 'dark')	No	iOS

`keyboardType`

Determines which keyboard to open, e.g. `numeric`.

The following values work across platforms:

- `default`
- `number-pad`
- `decimal-pad`
- `numeric`
- `email-address`
- `phone-pad`

iOS Only

The following values work on iOS only:

- `ascii-capable`
- `numbers-and-punctuation`
- `url`
- `name-phone-pad`
- `twitter`

- `web-search`

Android Only

The following values work on Android only:

- `visible-password`

TYPE	REQUIRED
enum('default', 'email-address', 'numeric', 'phone-pad', 'ascii-capable', 'numbers-and-punctuation', 'url', 'number-pad', 'name-phone-pad', 'decimal-pad', 'twitter', 'web-search', 'visible-password')	No

`maxLength`

Limits the maximum number of characters that can be entered. Use this instead of implementing the logic in JS to avoid flicker.

TYPE	REQUIRED
number	No

`multiline`

If `true`, the text input can be multiple lines. The default value is `false`.

TYPE	REQUIRED
bool	No

`numberOfLines`

Sets the number of lines for a `TextInput`. Use it with multiline set to `true` to be able to fill the lines.

TYPE	REQUIRED	PLATFORM
number	No	Android

`onBlur`

Callback that is called when the text input is blurred.

TYPE	REQUIRED
function	No

`onChange`

Callback that is called when the text input's text changes.

TYPE	REQUIRED
function	No

`onChangeText`

Callback that is called when the text input's text changes. Changed text is passed as an argument to the callback handler.

TYPE	REQUIRED
function	No

`onContentSizeChange`

Callback that is called when the text input's content size changes. This will be called with `{ nativeEvent: { contentSize: { width, height } } }`.

Only called for multiline text inputs.

TYPE	REQUIRED
function	No

`onEndEditing`

Callback that is called when text input ends.

TYPE	REQUIRED
function	No

`onFocus`

Callback that is called when the text input is focused.

TYPE	REQUIRED
function	No

`onKeyPress`

Callback that is called when a key is pressed. This will be called with `{ nativeEvent: { key: keyValue } }` where `keyValue` is `'Enter'` or `'Backspace'` for respective keys and the typed-in character otherwise including `' '` for space. Fires before `onChange` callbacks. Note: on Android only the inputs from soft keyboard are handled, not the hardware keyboard inputs.

TYPE	REQUIRED
function	No

`onLayout`

Invoked on mount and layout changes with `{x, y, width, height}`.

TYPE	REQUIRED
function	No

`onScroll`

Invoked on content scroll with `{ nativeEvent: { contentOffset: { x, y } } }`. May also contain other properties from ScrollEvent but on Android contentSize is not provided for performance reasons.

TYPE	REQUIRED
function	No

`onSelectionChange`

Callback that is called when the text input selection is changed. This will be called with `{ nativeEvent: { selection: { start, end } } }`.

TYPE	REQUIRED
function	No

`onSubmitEditing`

Callback that is called when the text input's submit button is pressed. Invalid if `multiline={true}` is specified.

TYPE	REQUIRED
function	No

`placeholder`

The string that will be rendered before text input has been entered.

TYPE	REQUIRED
string	No

`placeholderTextColor`

The text color of the placeholder string.

TYPE	REQUIRED
color	No

`returnKeyLabel`

Sets the return key to the label. Use it instead of `returnKeyType`.

TYPE	REQUIRED	PLATFORM
string	No	Android

`returnKeyType`

Determines how the return key should look. On Android you can also use `returnKeyLabel`.

Cross platform

The following values work across platforms:

- `done`
- `go`
- `next`
- `search`
- `send`

Android Only

The following values work on Android only:

- `none`
- `previous`

iOS Only

The following values work on iOS only:

- `default`
- `emergency-call`
- `google`
- `join`

- `route`
- `yahoo`

TYPE	REQUIRED
enum('done', 'go', 'next', 'search', 'send', 'none', 'previous', 'default', 'emergency-call', 'google', 'join', 'route', 'yahoo')	No

`scrollEnabled`

If `false`, scrolling of the text view will be disabled. The default value is `true`. Only works with `multiline={true}`.

TYPE	REQUIRED	PLATFORM
bool	No	iOS

`secureTextEntry`

If `true`, the text input obscures the text entered so that sensitive text like passwords stay secure. The default value is `false`. Does not work with `multiline={true}`.

TYPE	REQUIRED
bool	No

`selection`

The start and end of the text input's selection. Set start and end to the same value to position the cursor.

TYPE	REQUIRED
object: {start: number,end: number}	No

`selectionColor`

The highlight and cursor color of the text input.

TYPE	REQUIRED
color	No

`selectionState`

An instance of `DocumentSelectionState`, this is some state that is responsible for maintaining selection information for a document.

Some functionality that can be performed with this instance is:

- `blur()`
- `focus()`
- `update()`

> You can reference `DocumentSelectionState` in `vendor/document/selection/DocumentSelectionState.js`

TYPE	REQUIRED	PLATFORM
DocumentSelectionState	No	iOS

`selectTextOnFocus`

If `true`, all text will automatically be selected on focus.

TYPE	REQUIRED
bool	No

`spellCheck`

If `false`, disables spell-check style (i.e. red underlines). The default value is inherited from `autoCorrect`.

TYPE	REQUIRED	PLATFORM
bool	No	iOS

`textContentType`

Give the keyboard and the system information about the expected semantic meaning for the content that users enter.

For iOS 11+ you can set `textContentType` to `username` or `password` to enable autofill of login details from the device keychain.

To disable autofill, set `textContentType` to `none`.

Possible values for `textContentType` are:

- `none`
- `URL`
- `addressCity`
- `addressCityAndState`
- `addressState`
- `countryName`

- `creditCardNumber`
- `emailAddress`
- `familyName`
- `fullStreetAddress`
- `givenName`
- `jobTitle`
- `location`
- `middleName`
- `name`
- `namePrefix`
- `nameSuffix`
- `nickname`
- `organizationName`
- `postalCode`
- `streetAddressLine1`
- `streetAddressLine2`
- `sublocality`
- `telephoneNumber`
- `username`
- `password`

TYPE	REQUIRED	PLATFORM
enum('none', 'URL', 'addressCity', 'addressCityAndState', 'addressState', 'countryName', 'creditCardNumber', 'emailAddress', 'familyName', 'fullStreetAddress', 'givenName', 'jobTitle', 'location', 'middleName', 'name', 'namePrefix', 'nameSuffix', 'nickname', 'organizationName', 'postalCode', 'streetAddressLine1', 'streetAddressLine2', 'sublocality', 'telephoneNumber', 'username', 'password')	No	iOS

`style`

Note that not all Text styles are supported, an incomplete list of what is not supported includes:

- `borderLeftWidth`
- `borderTopWidth`
- `borderRightWidth`
- `borderBottomWidth`
- `borderTopLeftRadius`
- `borderTopRightRadius`
- `borderBottomRightRadius`
- `borderBottomLeftRadius`

see Issue#7070 for more detail.

Styles

TYPE	REQUIRED
Text	No

`textBreakStrategy`

Set text break strategy on Android API Level 23+, possible values are `simple`, `highQuality`, `balanced` The default value is `simple`.

TYPE	REQUIRED	PLATFORM
enum('simple', 'highQuality', 'balanced')	No	Android

`underlineColorAndroid`

The color of the `TextInput` underline.

TYPE	REQUIRED	PLATFORM
color	No	Android

`value`

The value to show for the text input. `TextInput` is a controlled component, which means the native value will be forced to match this value prop if provided. For most uses, this works great, but in some cases this may cause flickering - one common cause is preventing edits by keeping value the same. In addition to simply setting the same value, either set `editable={false}`, or set/update `maxLength` to prevent unwanted edits without flicker.

TYPE	REQUIRED
string	No

Methods

`clear()`

```
clear();
```

Removes all text from the `TextInput`.

`isFocused()`

```
isFocused():
```

Returns `true` if the input is currently focused; `false` otherwise.

← TEXT

TOOLBARANDROID →

Docs

Getting Started

Tutorial

Components and APIs

More Resources

Community

Who's using React Native?

Stack Overflow

Discussion Forum

Reactiflux Chat

More

Blog

Twitter

GitHub

React