TELUS Component Library

TextInput

Multi-Platform Component

```
Label ②
A usually much longer hint below

<TextInput
   label="Label"
   hint="A usually much longer hint below"
   hintPosition="below"
   tooltip="Tooltip content"
/>
```

Introduction

A basic text input component. Use in forms or individually to receive user's input.

Follow the appropriate instructions to add this component in to your app.

Guidance

Controlled component

If it is required that the state of the <code>TextInput</code> be controlled by the application or other external methods, <code>value</code> and <code>onChange</code> props must be passed to the <code>TextInput</code>.

Controlled						

This example shows an error if longer than 6 characters and success otherwise

```
function ControlledInput() {
  const [value, setValue] = React.useState('')
  const [validation, feedback] = React.useMemo(() => {
    if (value && value.length > 6) {
      return ['error', 'The value must be shorter than 6
characters']
    } else if (value && value.length > 0) {
      return ['success', 'Yay! The value has less than 6
characters']
    }
    return [
      undefined,
      'This example shows an error if longer than 6 characters
and success otherwise'
    1
  }, [value])
  return (
    <TextInput
      label="Controlled"
      validation={validation}
      feedback={feedback}
      value={value}
      onChange={setValue}
    />
  )
}
```

If the TextInput value should not be changed by user input, a readOnly prop must be provided.

```
Read only
won't change

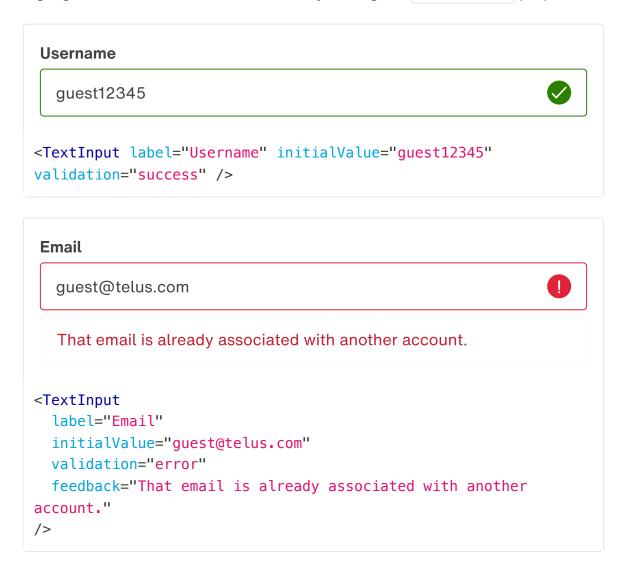
<TextInput label="Read only" readOnly value="won't change" />
```

Uncontrolled component

If it is not necessary to control the TextInput state, you can create one without a value prop. In this case its value can be accessed within the onChange callback.

Getting feedback for entered values

Use the feedback prop to give the user feedback regarding their input (keep it as brief as possible). You can affirm that the user's input was correct, or highlight errors that must be corrected by setting the validation prop.



This component does not perform input validations, as that is an application level concern. You will need to track the value of your form fields and perform any required data validations either client side or server side, depending on the context.

Supplying a hint

Use a hint to succinctly clarify attributes of the expected input data, such as the expected format, or an indicator that the field is optional. It is a more usable and accessible option than the HTML placeholder attribute.

```
Transit number 5 digits

<TextInput label="Transit number" hint="5 digits" />
```

- hint can be used beside the label (inline) or below
 - Label and inline hint text should not exceed 40 characters total, otherwise, use hint text below the label
- The hint should **not** be longer than 85 characters
- The hint should **not** wrap more than 3 lines on all viewport sizes
- Developers can use hintPosition to re-position the hint below the label if the text supplied to the inline hint wraps

Quick view Enter the account number in the box below to go to the overview page for that account. <TextInput label="Quick view" hintPosition="below" hint="Enter the account number in the box below to go to the overview page for that account." />

Using a tooltip

When a more in-depth explanation is needed to fill out a TextInput properly, additional instructions may be provided to the tooltip prop.

Customer ID ②

```
<TextInput
  label="Customer ID"
  tooltip="You can find your customer ID in the account
confirmation email."
/>
```

(!) INFO

In order to support more advanced usages all additional props will be passed down to the React Native's TextInput component. See React Native documentation for supported props and React Native Web documentation for their implementation on the web.

Using a clear button

TextInput component has a built-in way to display a clear button showing up once the input value changes. Pressing that button resets the input value to the initial state. The button gets rendered automatically whenever there is an onClear callback provided. This callback will be called after the button gets pressed:

```
Label ②
A usually much longer hint below

<TextInput
  label="Label"
  hint="A usually much longer hint below"
  hintPosition="below"
  tooltip="Tooltip content"
  onClear={() => console.log('Back to square one')}
/>
```

Note that while this feature is available in both uncontrolled and controlled mode, in the latter case it's the wrapper component holding the state that will

be responsible for listening to onClear and resetting the value whenever that callback is called.

Appearance

TextInput has no visual variants, but its appearance may change based on the validation and inactive props.



Accessibility

TextInput is accessible to screen readers on both native platforms and Web. It will be announced with the label prop if it is provided, and optionally described with the hint, tooltip and feedback.



Due to React Native's implementation of TextInput it's not possible to access the current value of the input by passing a ref. You can only access it within the onChange callback.

Platform considerations

The component is available on both native platforms and web.

(I) INFO

Due to React Native's implementation of TextInput it's not possible to access the current value of the input by passing a ref. You can only access it within the onChange callback.

Props

Name	Type	Platform	Default	Description
value	string	standard		If the input's state is to be controlled by a parent component, use this prop together with the `onChange` to pass down and update the lifted state.
initialValue	string	standard		Use this to set the initial value of an uncontrolled input. Updating `initialValue` will **not** update the actual value.
inactive	bool	standard		Disables all user interactions with the input.

Name	Type	Platform	Default	Description
readOnly	bool	standard		Makes it impossible to change the input's value.
onChange	func	standard		Use to react upon input's value changes. Required when the `value` prop is set. Will receive the input's value as an argument.
onClear	func	standard		A callback which if provided will get a clear button rendered and will be called whenever that button gets pressed.
tokens	tokens	standard		System tokens prop, see <u>tokens</u> for more details
variant	variant	standard		System variant prop, see <u>variants</u> for more details

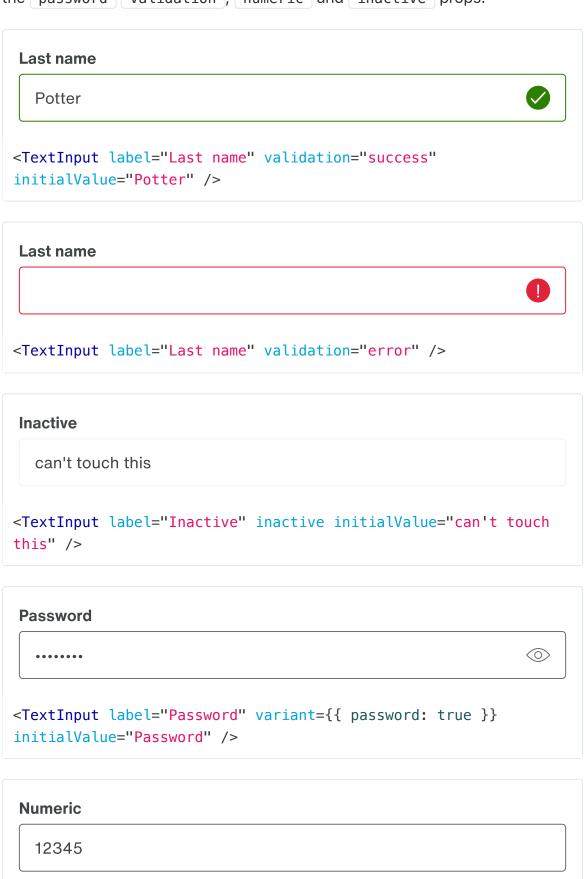
Tokens

In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** Read more about overriding styles.

▶ View Tokens

Variants

TextInput has no visual variants, but its appearance may change based on the password validation, numeric and inactive props.



```
<TextInput label="Numeric" variant={{ numeric: true }}
initialValue="12345" />
```

Feedback

- Spotted a problem with this component? Raise an <u>issue on GitHub</u>
- See any existing issues for this component
- Contact the team on slack in #ds-support