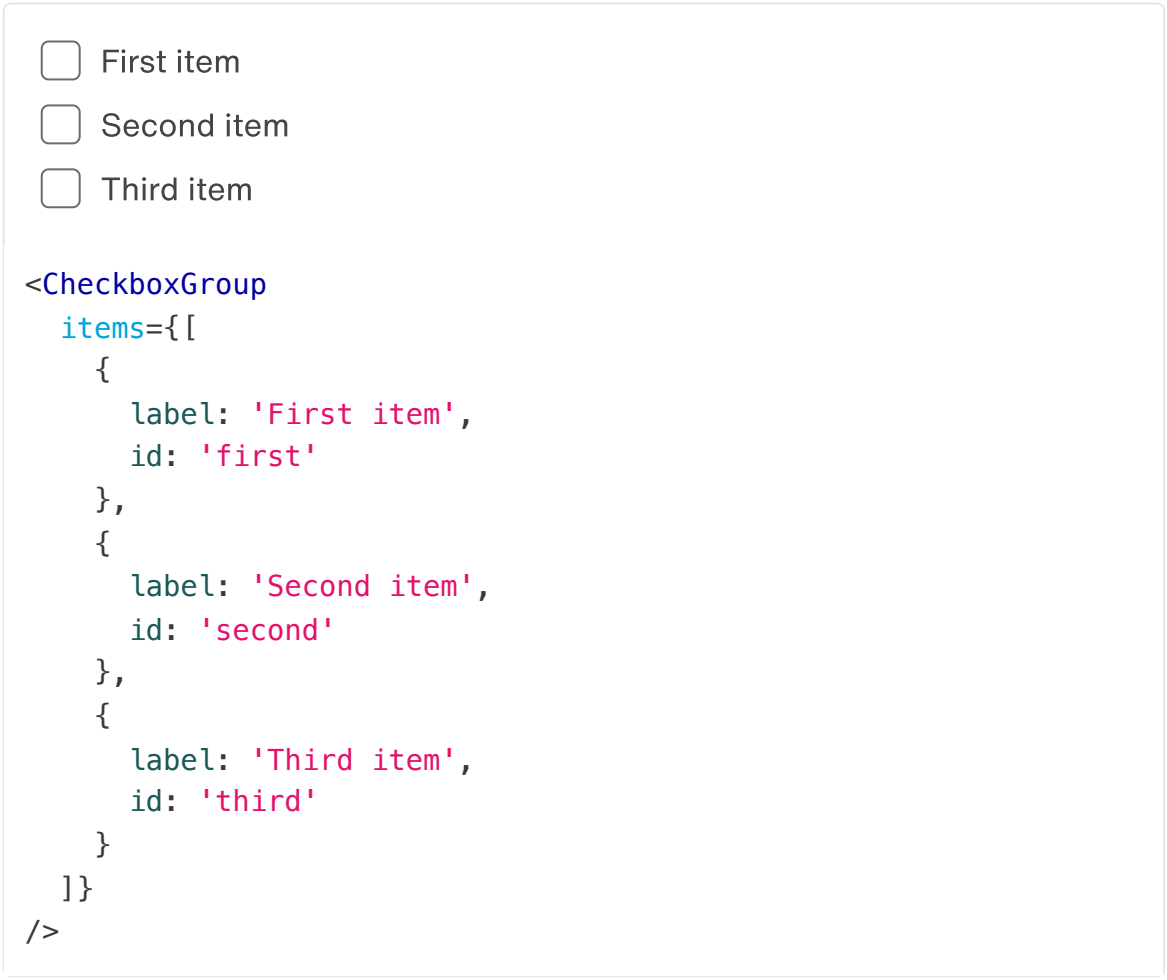
TELUS Component Library

CheckboxGroup

Multi-Platform Component | Figma UI KIT




```
<CheckboxGroup
  items={[
    {
      label: 'First item',
      id: 'first'
    },
    {
      label: 'Second item',
      id: 'second'
    },
    {
      label: 'Third item',
      id: 'third'
    }
  ]}
/>
```

Introduction

A CheckboxGroup renders a group of Checkbox buttons in a list, with a text label for each radio button. Exactly one radio button may be selected at a time.

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

Guidance

Use CheckboxGroup where the user needs to select one item from multiple options, and where those options can be explained simply with one short phrase

(generally, up to around 5 words).

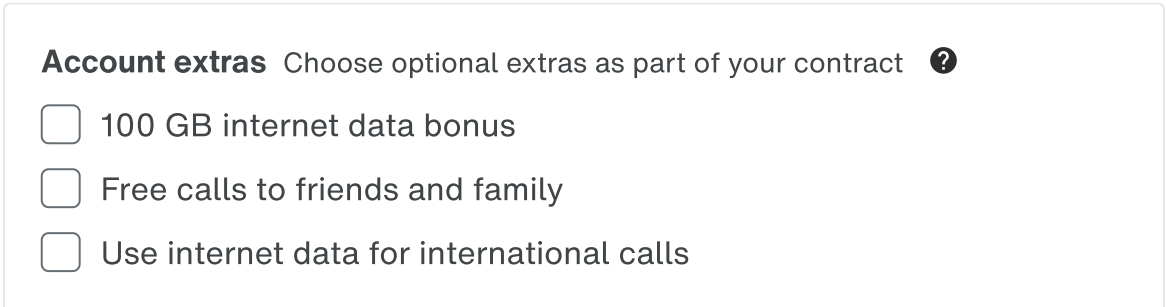
Alternatives

- If the options are very simple and can be described in one or two words or numbers, consider ButtonGroup with `maxItems={null}`
- If it should be possible to select only one item at a time, consider RadioGroup

Labeling the group

Three props are available for labelling the group to users:

- `legend` sets the main title for the group and should be a short, simple name for what the user is to choose.
- `hint` is an optional short sentence clarifying the `legend` to help users understand what choice they are being asked to make and what the result of their choice will be.
- `tooltip` allows further clarifying information to be included but not shown by default, only shown on pressing a TooltipButton. This should be used for information that will help many users make a choice, but which is not needed by all users.



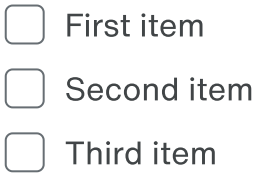

```
<CheckboxGroup
  legend="Account extras"
  hint="Choose optional extras as part of your contract"
  tooltip="Data amounts are listed in GB (gigabytes). 1 GB is
roughly 40 minutes of Blu-ray quality video."
  items={[
    {
      label: '100 GB internet data bonus',
      id: 'data-bonus'
    },
    {
```

```
      label: 'Free calls to friends and family',
      id: 'family'
    },
    {
      label: 'Use internet data for international calls',
      id: 'international'
    }
  ]}
/>
```

Items

Use the `items` prop to pass an array of objects describing each RadioCard in the group:

- `label`: text displayed alongside the radio button
- `id`: identifier used to store which RadioCard is selected (uses index if undefined). This is written into HTML as an id attr on web, so should be unique and not match IDs used elsewhere on a page.
- `onChange`: optional function called on selection, in addition to updating the group's selection state




```
<CheckboxGroup
  items={[
    {
      label: 'First item',
      id: 'first'
    },
    {
      label: 'Second item',
      id: 'second'
    },
    {
      label: 'Third item',
      id: 'third',
      onChange: (value, event) =>
```

```
        console.log('Third item selected', value,
event.currentTarget, event)
    }
  ]}
/>
```

Controlled

The selection state may be controlled by a parent by passing `checkedId` and an `onChange` function that updates the parent's state. If the component is controlled in this way, `checkedId` must never be undefined; pass `null` or an empty string if there is no selection.

Currently selected: ""

- [ ] First item
- [ ] Second item
- [ ] Third item

```
function ControlledCheckboxGroup() {
  const [checkedIds, setCheckedIds] = useState([])
  return (
    <StackView space={3}>
      <Typography>Currently selected: "{checkedIds.join('",
"')}"</Typography>
      <CheckboxGroup
        checkedId={checkedIds}
        onChange={setCheckedIds}
        items={[
          { label: 'First item', id: 'first' },
          { label: 'Second item', id: 'second' },
          { label: 'Third item', id: 'third' }
        ]}
      />
    </StackView>
  )
}
```

Uncontrolled

If `checkedId` is not provided, CheckboxGroup manages its own state. An initial selection may be provided by passing `initialCheckedIds`.

- [ ] First item
- [ ] Second item
- [ ] Third item

```
<CheckboxGroup
  initialCheckedIds="second"
  items={[
    { label: 'First item', id: 'first' },
    { label: 'Second item', id: 'second' },
    { label: 'Third item', id: 'third' }
  ]}
/>
```

`onChange` functions

The `onChange` handler for CheckboxGroup may be used in both controlled and uncontrolled usage and takes two arguments: `value` (string) and `event` (React SyntheticEvent object).

The `event` object shape varies between web and native applications, and type of user interaction. It is usually more stable to have application logic respond to the `value` and not `event` properties.

On web, `event.currentTarget` will point to the accessible container that holds the radio role and state, and should be used if accessing the raw HTML element is required (`event.target` will vary on mouse clicks depending on where exactly the click occured).

- [ ] First item
- [ ] Second item
- [ ] Third item

```
<CheckboxGroup
  onChange={(value, event) => console.log(value,
event.currentTarget, event)}
```

```
  items={[
    { label: 'First item', id: 'first' },
    { label: 'Second item', id: 'second' },
    { label: 'Third item', id: 'third' }
  ]}
/>
```

Use in forms

For web forms, the `name` prop may be used to define the name of the group's `<fieldset>` and input elements, and `legend` may be used to provide a title with `<legend>` type for the fieldset.

Choose a product

- [ ] First product
- [ ] Second product
- [ ] Third product

```
<CheckboxGroup
  legend="Choose a product"
  name="product-choice"
  items={[
    { label: 'First product', id: 'first' },
    { label: 'Second product', id: 'second' },
    { label: 'Third product', id: 'third' }
  ]}
/>
```

Validation

Validation state may be set by passing 'error' or 'success' to the `validation` prop.

Choose only 'a' and/or 'c'

- [ ] Option 'a'
- [ ] Option 'b'
- [ ] Option 'c'

Option 'd'

```
function ValidationExample() {
  const [validation, setValidation] = useState(null)
  const validate = (ids) =>
    setValidation(ids.every((id) => ['a', 'c'].includes(id)) ?
'success' : 'error')
  return (
    <CheckboxGroup
      legend="Choose only 'a' and/or 'c'"
      onChange={validate}
      validation={validation}
      feedback={validation && `${validation === 'success' ?
'Valid' : 'Invalid'} selection chosen.`}
      items={[
        { label: "Option 'a'", id: 'a' },
        { label: "Option 'b'", id: 'b' },
        { label: "Option 'c'", id: 'c' },
        { label: "Option 'd'", id: 'd' }
      ]}
    />
  )
}
```

A11y guidelines

CheckboxGroup accepts all the common accessibility props, and controls the accessibility state of children like Checkbox and Feedback based on current state.

Platform considerations

The component is available on both native platforms and web.

Props

Name	Type	Platform	Default	Description
tokens	tokens	standard		System tokens prop, see

Name	Type	Platform	Default	Description
				tokens for more details
radioTokens	custom	standard		Optional theme token overrides for each inner Checkbox component
variant	variant	standard		System variant prop, see variants for more details
items	arrayOf	standard		Array of objects containing specifics for each Checkbox to be rendered in the group.
legend	string	standard		Main text used to describe this group, used in Fieldset's Legend element.
hint	string	standard		Optional additional text giving more detail to help a

Name	Type	Platform	Default	Description
				user make a choice.
hintPosition	'inline' | 'below'	standard		Position of the hint relative to label. Use `below` to display a larger hint below the label.
tooltip	string	standard		Optional tooltip text content to include alongside the legend and hint.
validation	'error' | 'success'	standard		Current validation status of the group, passed to the feedback element if there is one.
feedback	string	standard		If provided, a Feedback element is rendered containing this text.
initialCheckedIds	arrayOf	standard		If provided, the checkboxed

Name	Type	Platform	Default	Description
				with this ids are selected on first render.
checkedIds	arrayOf	standard		If not undefined, the checkboxes with these ids is selected, and the element's selection state will be controlled by its parent using the `onChange` function.
onChange	func	standard		Function to call on change in selection state. Is required if the selection state is controlled by a parent using checkedId and the input is not readOnly.
readOnly	bool	standard		If true, the checkboxes cannot be selected by the user and simply show

Name	Type	Platform	Default	Description
				their current state.
inactive	bool	standard		If true, the checkboxes cannot be interacted with, elements are set as `disabled` and if the theme supports `inactive` appearances rules, these are applied.
name	string	standard		On Web, this is passed to the `name` attribute of the fieldset and each checkbox input.

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

This component does not have any stylistic variants.

Feedback

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