

# Transfer List

A Transfer List (or "shuttle") enables the user to move one or more list items between lists.



Design and Development tips in your inbox. Every weekday.

ads via Carbon

[View as Markdown](#)[Feedback](#)[Bundle size](#)[Figma](#)[Sketch](#)

## Basic transfer list

For completeness, this example includes buttons for "move all", but not every transfer list needs these.

<input type="checkbox"/> List item 1		<input type="checkbox"/> List item 5
<input type="checkbox"/> List item 2		<input type="checkbox"/> List item 6
<input type="checkbox"/> List item 3		<input type="checkbox"/> List item 7
<input type="checkbox"/> List item 4		<input type="checkbox"/> List item 8

>>  
>  
<  
<<

[Edit in Chat](#)

JS

TS

[Hide code](#)

```
import * as React from 'react';
import Grid from '@mui/material/Grid';
import List from '@mui/material/List';
import ListItemButton from '@mui/material/ListItemButton';
import ListItemIcon from '@mui/material/ListItemIcon';
importListItemText from '@mui/material/ListItemText';
import Checkbox from '@mui/material/Checkbox';
import Button from '@mui/material/Button';
import Paper from '@mui/material/Paper';

function not(a: readonly number[], b: readonly number[]) {
  return a.filter((value) => !b.includes(value));
}
```

```

function intersection(a: readonly number[], b: readonly number[]) {
  return a.filter((value) => b.includes(value));
}

export default function TransferList() {
  const [checked, setChecked] = React.useState<readonly number[]>([]);
  const [left, setLeft] = React.useState<readonly number[]>([0, 1, 2, 3]);
  const [right, setRight] = React.useState<readonly number[]>([4, 5, 6, 7]);

  const leftChecked = intersection(checked, left);
  const rightChecked = intersection(checked, right);

  const handleToggle = (value: number) => () => {
    const currentIndex = checked.indexOf(value);
    const newChecked = [...checked];

    if (currentIndex === -1) {
      newChecked.push(value);
    } else {
      newChecked.splice(currentIndex, 1);
    }

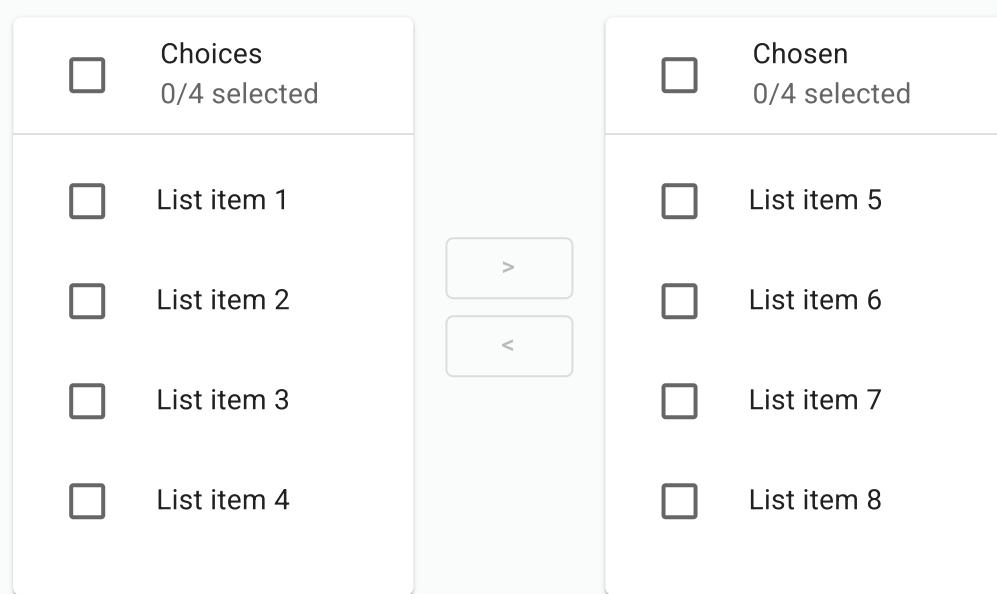
    setChecked(newChecked);
  };
}

```

## Enhanced transfer list

+

This example exchanges the "move all" buttons for a "select all / select none" checkbox and adds a counter.



[Edit in Chat](#)

JS

TS

[Hide code](#)



```

import * as React from 'react';
import Grid from '@mui/material/Grid';

```

```

import List from '@mui/material>List';
import Card from '@mui/material/Card';
import CardHeader from '@mui/material/CardHeader';
import ListItemButton from '@mui/material/ListItemButton';
importListItemText from '@mui/material/ListItemText';
importListItemIcon from '@mui/material/ListItemIcon';
import Checkbox from '@mui/material/Checkbox';
import Button from '@mui/material/Button';
import Divider from '@mui/material/Divider';

function not(a: readonly number[], b: readonly number[]) {
  return a.filter((value) => !b.includes(value));
}

function intersection(a: readonly number[], b: readonly number[]) {
  return a.filter((value) => b.includes(value));
}

function union(a: readonly number[], b: readonly number[]) {
  return [...a, ...not(b, a)];
}

export default function SelectAllTransferList() {
  const [checked, setChecked] = React.useState<readonly number[]>([]);
  const [left, setLeft] = React.useState<readonly number[]>([0, 1, 2, 3]);
  const [right, setRight] = React.useState<readonly number[]>([4, 5, 6, 7]);

  const leftChecked = intersection(checked, left);
  const rightChecked = intersection(checked, right);

  const handleToggle = (value: number) => () => {
    const currentIndex = checked.indexOf(value);
    const newChecked = [...checked];

    if (currentIndex === -1) {
      newChecked.push(value);
    } else {
      newChecked.splice(currentIndex, 1);
    }

    setChecked(newChecked);
  };
}

```

## Limitations

The component comes with a couple of limitations:

- It only works on desktop. If you have a limited amount of options to select, prefer the [Autocomplete](#) component. If mobile support is important for you, have a look at [#27579](#).
- There are no high-level components exported from npm. The demos are based on composition. If this is important for you, have a look at [#27579](#).

## API

See the documentation below for a complete reference to all of the props and classes available to the components mentioned here.

- [`<Checkbox />`](#)
- [`<List />`](#)
- [`<ListItem />`](#)
- [`<Switch />`](#)

 [Edit this page](#)

Was this page helpful?  

---

[`< Text Field`](#)

[`Toggle Button >`](#)

---