Unstyled React Select components and hook

Unstyled select

The select components let you create lists of options for users to choose from.

{{“component”: “modules/components/ComponentLinkHeader.js”, “design”: false}}

MUI Base offers two components to replace the native HTML `<select>` tag: `SelectUnstyled` and `MultiSelectUnstyled`.

SelectUnstyled

```
import SelectUnstyled from '@mui/base/SelectUnstyled';
```

Basic select

{{“demo”: “UnstyledSelectSimple.js”, “defaultCodeOpen”: false}}

The `SelectUnstyled` component accepts generic props. Due to TypeScript limitations, this may cause unexpected behavior when wrapping the component in `forwardRef` (or other higher-order components). In such cases, the generic argument will be defaulted to `unknown` and type suggestions will be incomplete. To avoid this, you can manually cast the resulting component to the correct type:

```
const CustomSelect = React.forwardRef(function CustomSelect<TValue>(
  props: SelectUnstyledProps<TValue>,
  ref: React.ForwardedRef<HTMLUListElement>,
) {
  // ...your code here...
  return <SelectUnstyled {...props} ref={ref} />;
}) as <TValue>(
  props: SelectUnstyledProps<TValue> & React.RefAttributes<HTMLUListElement>,
) => JSX.Element;
```

For the sake of brevity, the rest of the demos that follow will not use `forwardRef`.

Controlled select

`SelectUnstyled` can be used as an uncontrolled (as shown in the demo above) or controlled component.

{{"demo": "UnstyledSelectControlled.js", "defaultCodeOpen": false}}

Object values

The `SelectUnstyled` component can be used with non-string values.

{{"demo": "UnstyledSelectObjectValues.js", "defaultCodeOpen": false}}

Selected value appearance

You can customize the appearance of the selected value display by providing a function to the `renderValue` prop. The element returned by this function will be rendered inside the select's button.

{{"demo": "UnstyledSelectCustomRenderValue.js", "defaultCodeOpen": false}}

Option appearance

Options don't have to be plain strings. You can include custom elements to be rendered inside the listbox.

{{"demo": "UnstyledSelectRichOptions.js", "defaultCodeOpen": false}}

Grouping options

Options can be grouped, similarly to the how the native `select` element works. Unlike the native `select`, however, groups can be nested.

Place the `Option` components inside `OptionGroup` to achieve this.

{{"demo": "UnstyledSelectGrouping.js", "defaultCodeOpen": false}}

MultiSelectUnstyled

The `MultiSelectUnstyled` component lets your users select multiple options.

```
import { MultiSelectUnstyled } from '@mui/base/SelectUnstyled';
```

{{"demo": "UnstyledSelectMultiple.js", "defaultCodeOpen": false}}

The useSelect hook

```
import { useSelect } from '@mui/base/SelectUnstyled';
```

You can use the `useSelect` hook to apply the functionality of the unstyled select components to a different component. This hook gives you the most customization options, but requires more work to implement. Using the hook allows you to take full control over the rendered components, their props and CSS classes.

The following example shows a select that opens when hovered over or focused. It can be controlled by a mouse/touch or a keyboard.

{{“demo”: “UseSelect.js”, “defaultCodeOpen”: false}}