

# ClassName generator

Configure classname generation at build time.

This API is introduced in `@mui/material` (v5.0.5) as a replacement of deprecated [createGenerateClassName](#) .

 **Note:** *this API is at an unstable stage which might change in the future.*

## Global classname prefix

By default, MUI generates a global classname for each component slot. For example:

```
import Button from '@mui/material/Button';

function App() {
  return <Button>Button</Button>;
}
```

Generates the following HTML:

```
<button
  class="MuiButton-root MuiButton-text MuiButton-textPrimary MuiButton-sizeMedium
MuiButton-textSizeMedium MuiButtonBase-root css-lujzas3"
>
  Button
</button>
```

To add prefix to all class names generated by the MUI components, pass a callback to

`ClassNameGenerator.configure(callback)` .

```
import { unstable_ClassNameGenerator as ClassNameGenerator } from
 '@mui/material/utils';

// call this function at the root of the application and before any MUI components
import ClassNameGenerator.configure((componentName) => `foo-bar-${componentName}`);

function App() {
  return <Button>Button</Button>;
}
```

As a result, the HTML result changes to the following:

```
<button
  class="foo-bar-MuiButton-root foo-bar-MuiButton-text foo-bar-MuiButton-textPrimary
foo-bar-MuiButton-sizeMedium foo-bar-MuiButton-textSizeMedium foo-bar-MuiButtonBase-
root css-lujzas3"
>
```

```
Button
</button>
```

## Component renaming

Every MUI component has `${componentName}-${slot}` classname format. For example, the component name of `Chip` is `MuiChip`, which is used as a global class name for every `<Chip />` element. You can remove/change the `Mui` prefix as follows:

```
import { unstable_ClassNameGenerator } from '@mui/material/utils';

// call this function at the root of the application
unstable_ClassNameGenerator.configure((componentName) =>
  componentName.replace('Mui', ''),
);

function App() {
  return <Button>Button</Button>;
}
```

Now, the `Mui` class is gone.

```
<div
  class="Chip-root Chip-filled Chip-sizeMedium Chip-colorDefault Chip-filledDefault
css-mttbc0"
>
  Chip
</div>
```

**Note:** [state classes](#) are **NOT** component names and therefore cannot be changed/removed.

## Caveat

- `ClassNameGenerator.configure` must be called before any MUI components import.
- you should always use `[component]Classes` for theming/customization to get the correct generated class name.

```
+import { outlinedInputClasses } from '@mui/material/OutlinedInput';

const theme = createTheme({
  components: {
    MuiOutlinedInput: {
      styleOverrides: {
        root: {
-      '& . MuiOutlinedInput-notchedOutline': {
+      // the result will contain the prefix.
+      [`& .${outlinedInputClasses.notchedOutline}`]: {
        borderWidth: 1,
```

```

    }
  }
}
}
}
});

```

- This API should only be used at build-time.
- The configuration is applied to all of the components across the application. You cannot target a specific part of the application.

## Framework example

In some cases, you might need to add `/* eslint-disable import/first */` at the top of

`ClassNameGenerator` import.

```

/* eslint-disable import/first */
import { unstable_ClassNameGenerator as ClassNameGenerator } from
'@mui/material/utils';

```

## Next.js

Use `ClassNameGenerator` in `/pages/_app.js`.

```

+import { unstable_ClassNameGenerator as ClassNameGenerator } from
+'@mui/material/utils';

+ClassNameGenerator.configure((componentName) => componentName.replace('Mui', ''));

import * as React from 'react';
import PropTypes from 'prop-types';
import Head from 'next/head';

export default function MyApp(props) {
  const { Component, pageProps } = props;

  return (
    <Component {...pageProps} />
  );
}

```

## Create React App

Use `ClassNameGenerator` in `/src/index.js`.

```

+import { unstable_ClassNameGenerator as ClassNameGenerator } from
+'@mui/material/utils';

+ClassNameGenerator.configure((componentName) => componentName.replace('Mui', ''));

```

```
import * as React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(<App />);
```

## Gatsby

Use `ClassNameGenerator` in `gatsby-ssr.js` at the root folder.

```
+import { unstable_ClassNameGenerator as ClassNameGenerator } from
"@mui/material/utils";

+ClassNameGenerator.configure((componentName) => componentName.replace('Mui', ''));

export const wrapPageElement = ({ element }) => {
  return element;
};
```