`@material-ui/styled-engine`

Configuring your preferred styling library.

The default style library used for generating CSS styles for Material-UI components is emotion. All of the Material-UI components rely on the `styled()` API to inject CSS into the page. This API is supported by multiple popular styling libraries, which makes it possible to switch between them in Material-UI.

How to switch to styled-components

> ❗ ***Warning***: *Using `styled-components` as an engine at this moment is not working when used in a SSR projects. The reason is that the `babel-plugin-styled-components` is not picking up correctly the usages of the `styled()` utility inside the `@mui` packages. For more details, take a look at this issue. We strongly recommend using `emotion` for SSR projects.*

If you already have styled-components installed, it's possible to use it exclusively. There are currently two packages available to choose from:

- `@material-ui/styled-engine` - a thin wrapper around emotion's `styled()` API, with the addition of few other required utilities, such as the `<GlobalStyles />` component, the `css` and `keyframe` helpers, etc. This is the default. This is the default.
- `@material-ui/styled-engine-sc` - a similar wrapper around `styled-components`.

These two packages implement the same interface, which makes it makes possible to replace one with the other. By default, `@material-ui/core` has `@material-ui/styled-engine` as a dependency, but you can configure your bundler to replace it with `@material-ui/styled-engine-sc`.

yarn

If you are using yarn, you can configure it using a package resolution:

webpack.config.js

```diff
 {
   "dependencies": {
-    "@mui/styled-engine": "latest"
+    "@mui/styled-engine": "npm:@mui/styled-engine-sc@latest"
   },
+  "resolutions": {
+    "@mui/styled-engine": "npm:@mui/styled-engine-sc@latest"
+  },
 }
```

npm

As package resolutions are not available in npm at this moment, you need to update you bundler's config to add this alias. For example, if you are using webpack you can configure this by adding a resolver:

webpack.alias.js

```
 module.exports = {
   //...
module.exports = {
  //...
resolve: {
    alias: {
      '@material-ui/styled-engine': '@material-ui/styled-engine-sc',
    },
  },
};
```

If you are using TypeScript, you will need to also update the TSConfig.

tsconfig.json

```
 {
   "compilerOptions": {
+    "paths": {
+      "@mui/styled-engine": ["./node_modules/@mui/styled-engine-sc"]
+    }
   },
 }
```

Next.js

next.config.js

```
+const withTM = require('next-transpile-modules')([
+  '@mui/material',
+  '@mui/system',
+  '@mui/icons-material', // If @mui/icons-material is being used
+]);

+module.exports = withTM({
 webpack: (config) => {
   config.resolve.alias = {
     ...config.resolve.alias,
+    '@mui/styled-engine': '@mui/styled-engine-sc',
    };
    return config;
  }
+});
```

Ready-to-use examples

If you are using create-react-app, there is a ready-to-use template in the example projects. You can use these `styled-component` examples as a reference:

- create-react-app
- create-react-app with TypeScript

- and many others

> ***Note***: `@emotion/react`, `@emotion/styled`, *and* `styled-components` *are optional peer dependencies of* `@mui/material`*, so you need to install them yourself. See the Installation guide for more info.*

Note: This package-swap approach is identical to the replacement of React with Preact. The Preact team has documented a large number of installation configurations. If you are stuck with MUI + styled-components, don't hesitate to check out how they solve the problem, as you can likely transfer the solution.