

Example app with styled-components

This example features how you use a different styling solution than [styled-jsx](#) that also supports universal styles. That means we can serve the required styles for the first render within the HTML and then load the rest in the client. In this case we are using [styled-components](#).

This example uses the Rust-based [SWC](#) in Next.js for better performance than Babel.

Currently, only the `ssr` and `displayName` transforms have been implemented. These two transforms are the main requirement for using `styled-components` in Next.js.

Deploy your own

Deploy the example using [Vercel](#) or preview live with [StackBlitz](#)



How to use

Execute [create-next-app](#) with [npm](#) or [Yarn](#) to bootstrap the example:

```
npx create-next-app --example with-styled-components with-styled-components-app
# or
yarn create next-app --example with-styled-components with-styled-components-app
# or
pnpm create next-app -- --example with-styled-components with-styled-components-app
```

Deploy it to the cloud with [Vercel](#) ([Documentation](#)).

Try it on CodeSandbox

[Open this example on CodeSandbox](#)

Notes

When wrapping a [Link](#) from `next/link` within a styled-component, the `as` prop provided by `styled` will collide with the Link's `as` prop and cause styled-components to throw an `Invalid tag` error. To avoid this, you can either use the recommended [forwardedAs](#) prop from styled-components or use a different named prop to pass to a `styled` Link.

► Click to expand workaround example