root

.babelrc

snippet.javascript

```
{
  "presets": ["next/babel", "@zeit/next-typescript/babel"],
  "plugins": [
    "transform-decorators-legacy",
    "transform-class-properties",
    [
      "module-resolver",
      {
        "root": ["./"],
        "alias": {
          "components": "./src/components",
          "containers": "./src/containers",
          "stores": "./src/stores",
          "lib": "./src/lib",
          "configs": "./src/configs"
        }
      }
    ]
  ]
}
```

next.config.js

snippet.javascript

```
const withTypescript = require('@zeit/next-typescript')
module.exports = withTypescript()
```

package.json

snippet.javascript

```
{
  "name": "typescript-nextjs",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
```

```
  "devDependencies": {
    "@types/react": "^16.4.13",
    "babel-plugin-module-resolver": "^3.1.1",
    "babel-plugin-transform-class-properties": "^6.24.1",
    "babel-plugin-transform-decorators-legacy": "^1.3.5",
    "typescript": "^3.0.3"
  },
  "dependencies": {
    "@material-ui/core": "^3.0.2",
    "@material-ui/icons": "^3.0.1",
    "@types/styled-components": "^3.0.0",
    "@zeit/next-typescript": "^1.1.0",
    "mobx": "^5.1.0",
    "mobx-react": "^5.2.6",
    "next": "^6.1.2",
    "react": "^16.5.0",
    "react-dom": "^16.5.0",
    "styled-components": "^3.4.5"
  },
  "scripts": {
    "dev": "next",
    "build": "next build",
    "start": "next start"
  }
}
```

tsconfig.json

snippet.javascript

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "target": "esnext",
    "module": "esnext",
    "jsx": "preserve",
    "allowJs": true,
    "moduleResolution": "node",
    "allowSyntheticDefaultImports": true,
    "experimentalDecorators": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "removeComments": false,
    "preserveConstEnums": true,
    "sourceMap": true,
    "skipLibCheck": true,
    "baseUrl": ".",
    "typeRoots": [
```

```
      "./node_modules/@types"
    ],
    "lib": [
      "dom",
      "es2015",
      "es2016"
    ],
    "paths": {
      "components/*": [
        "src/components/*"
      ],
      "lib/*": [
        "src/lib/*"
      ]
    }
  },
}
```

pages

_app.jsx

snippet.javascript

```
import React from "react"
import App, { Container } from "next/app"
import { MuiThemeProvider } from "@material-ui/core/styles"
import CssBaseline from "@material-ui/core/CssBaseline"
import JssProvider from "react-jss/lib/JssProvider"
import getPageContext from "lib/getPageContext"

class MyApp extends App {
  constructor(props) {
    super(props)
    this.pageContext = getPageContext()
  }

  pageContext = null

  componentDidMount() {
    // Remove the server-side injected CSS.
    const jssStyles = document.querySelector("#jss-server-side")
    if (jssStyles && jssStyles.parentNode) {
      jssStyles.parentNode.removeChild(jssStyles)
    }
  }
```

```
  render() {
    const { props } = this as any
    const { Component, pageProps } = props

    return (
      <Container>
        {/* Wrap every page in Jss and Theme providers */}
        <JssProvider
          registry={this.pageContext.sheetsRegistry}
          generateClassName={this.pageContext.generateClassName}
        >
          {/* MuiThemeProvider makes the theme available down the React
              tree thanks to React context. */}
          <MuiThemeProvider
            theme={this.pageContext.theme}
            sheetsManager={this.pageContext.sheetsManager}
          >
            {/* CssBaseline kickstart an elegant, consistent, and
simple baseline to build upon. */}
            <CssBaseline />
            {/* Pass pageContext to the _document though the renderPage
enhancer
                to render collected styles on server side. */}
            <Component pageContext={this.pageContext} {...pageProps} />
          </MuiThemeProvider>
        </JssProvider>
      </Container>
    )
  }
}

export default MyApp
```

_document.jsx

snippet.javascript

```
import React from "react"
import Document, { Head, Main, NextScript } from "next/document"
import JssProvider from "react-jss/lib/JssProvider"
import flush from "styled-jsx/server"
import getPageContext from "lib/getPageContext"
import { ServerStyleSheet } from "styled-components"

class MyDocument extends Document {
  static async getInitialProps(ctx) {
    // Resolution order
    //
```

```
    // On the server:
    // 1. page.getInitialProps
    // 2. document.getInitialProps
    // 3. page.render
    // 4. document.render
    //
    // On the server with error:
    // 2. document.getInitialProps
    // 3. page.render
    // 4. document.render
    //
    // On the client
    // 1. page.getInitialProps
    // 3. page.render
    // Get the context of the page to collected side effects.
    const pageContext = getPageContext()
    const sheet = new ServerStyleSheet()
    const page = ctx.renderPage(Component => props =>
      sheet.collectStyles(
        <JssProvider
          registry={pageContext.sheetsRegistry}
          generateClassName={pageContext.generateClassName}
        >
          <Component pageContext={pageContext} {...props} />
        </JssProvider>
      )
    )
    const styleTags = sheet.getStyleElement()
    return {
      ...page,
      pageContext,
      styles: (
        <React.Fragment>
          <style
            id="jss-server-side"
            // eslint-disable-next-line react/no-danger
            dangerouslySetInnerHTML={{
              __html: pageContext.sheetsRegistry.toString()
            }}
          />
          {flush() || null}
        </React.Fragment>
      ),
      styleTags
    }
  }

  render() {
    const { props } = this as any
    const { pageContext, styleTags } = props
```

```
    return (
      <html lang="en" dir="ltr">
        <Head>
          <title>My page</title>
          <meta charSet="utf-8" />
          {styleTags}
          {/* Use minimum-scale=1 to enable GPU rasterization */}
          <meta
            name="viewport"
            content={
              "user-scalable=0, initial-scale=1, " +
              "minimum-scale=1, width=device-width, height=device-
height"
            }
          />
          {/* PWA primary color */}
          <meta
            name="theme-color"
            content={pageContext.theme.palette.primary.main}
          />
          <link
            rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/7.0.0/normalize.
css"
          />
          <link
            rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500"
          />
        </Head>
        <body>
          <Main />
          <NextScript />
        </body>
      </html>
    )
  }
}

export default MyDocument
```

app.jsx

snippet.javascript

```
import React, { Component } from "react"
import AppBar from "@material-ui/core/AppBar"
import Toolbar from "@material-ui/core/Toolbar"
```

```
import Typography from "@material-ui/core/Typography"
import Button from "@material-ui/core/Button"
import IconButton from "@material-ui/core/IconButton"
import MenuIcon from "@material-ui/icons/Menu"

export default class Appbar extends Component {
  render() {
    return (
      <div>
        <AppBar position="static" color="primary">
          <Toolbar>
            <IconButton color="inherit" aria-label="Menu">
              <MenuIcon />
            </IconButton>
            <Typography variant="title" color="inherit">
              News
            </Typography>
            <Button color="inherit">Login</Button>
          </Toolbar>
        </AppBar>
      </div>
    )
  }
}
```

관련 문서

Plugin Backlinks: 아무 것도 없습니다.