

gatsby-plugin-stylus

Provides drop-in support for Stylus

Install

```
npm install gatsby-plugin-stylus
```

How to use

1. Include the plugin in your `gatsby-config.js` file.
2. Write your stylesheets in Stylus (`.styl` files) and require/import them

```
plugins: [`gatsby-plugin-stylus`]
```

If you need to override the default options passed into `css-loader`. **Note:** Gatsby is using `css-loader@5.0.0`.

```
plugins: [  
  {  
    resolve: `gatsby-plugin-stylus`,  
    options: {  
      cssLoaderOptions: {  
        camelCase: false,  
      },  
    },  
  },  
]  
]
```

With CSS Modules

Using CSS modules requires no additional configuration. Simply prepend `.module` to the extension. For example: `app.styl -> app.module.styl`. Any file with the `module` extension will use CSS modules. CSS modules are imported as ES Modules to support treeshaking. You'll need to import styles as: `import { yourClassName, anotherClassName } from './app.module.styl'`

With Stylus plugins

This plugin has the same API as `stylus-loader`, which means you can add stylus plugins with `use`:

```
const rupture = require("rupture")  
  
module.exports = {  
  plugins: [  
    {  
      resolve: `gatsby-plugin-stylus`,  
      options: {
```

```

        use: [rupture()],
      },
    ],
  }
}

```

PostCSS plugins

PostCSS is also included to handle some default optimizations like autoprefixing and common cross-browser flexbox bugs. Normally you don't need to think about it, but if you'd prefer to add additional postprocessing to your Stylus output you can specify plugins in the plugin options

```

plugins: [
  {
    resolve: `gatsby-plugin-stylus`,
    options: {
      postCssPlugins: [somePostCssPlugin()],
    },
  },
]

```