****

**Настройки Sass, SCSS, CSS и sass-middleware**

**Пример конфигурации sassMiddleware с параметром 'prefix':** Чтобы скомпилированный CSS-файл был доступен по URL без префикса /css,

( т.е. http://localhost:3000/style.css )

установите параметр prefix на пустую строку:





Теперь скомпилированный CSS файл будет доступен по URL:

http://localhost:3000/style.css

**Пример с параметром prefix:**

А если хочу, чтобы было так: http://localhost:3000/**css**/style.css , тогда пишем:

В конфигурации можно задать префикс для скомпилированного CSS. Например, если нужно, чтобы файл был доступен по URL без префикса /css, можно настроить параметр prefix как пустую строку. prefix: '**css**'

**Структура папок для SCSS и скомпилированного CSS:** Предполагаемая структура проекта будет выглядеть так:

/my-project

├── /scss

│ └── style.scss

├── /public

│ └── style.css <-- скомпилированный файл

├── /node\_modules

├── app.js

└── package.json

**Разница между SCSS и Sass:** SCSS (Sassy CSS) — это синтаксис Sass, который использует те же правила, что и CSS (с фигурными скобками и точками с запятой). Sass (Syntactically Awesome Stylesheets) — это более старый синтаксис, в котором используется только отступы для определения структуры блоков, без фигурных скобок и точек с запятой.

### Аргументы в sassMiddleware:

#### 1. indentedSyntax

* Указывает, какой синтаксис использовать:
  + **true**: Используется .sass — синтаксис без фигурных скобок {} и точек с запятой ; (отступы задают структуру).
  + **false**: Используется .scss — синтаксис, похожий на CSS (с фигурными скобками и точками с запятой).

#### 2. sourceMap

* Если **true**, создаются sourcemaps для удобной отладки CSS-кода. Sourcemaps позволяют отслеживать, какой исходный SASS-код соответствует скомпилированному CSS-коду.

#### 3. outputStyle

* Указывает стиль вывода скомпилированного CSS.

### 