CSS Modules 个人实践和使用心得

最近尝试了一下 CSS Modules, 发现比我想象中好用很多，首先再也不用担心样式全局污染的问题了，全局污染是我认为的设计 CSS 的最大败笔， 其次再也不需要使用嵌套的选择权去定位元素了。

Github 示例：Demo

为了开启 CSS Modules，我们需要修改 Webpack 中 Css-Loader 的相关配置，修改为：
Css-Loader?Modules ：

```
// Webpack.Config.Js
{
  Module: {
    Loaders: [{
        Test: /\.Less$/,
        Loader: "Style-Loader!Css-Loader?Modules!Postcss-Loader!Less-Loader?StrictMath"
      }, {
        Test: /\.Css/,
        Loader: "Style-Loader!Css-Loader?Modules!Postcss-Loader"
      }]
}
```

这里有个示例：

```
// CSSModulesComponent
Import 'DefaultStyle.Less';
Class CSSModulesComponent Extends React.Component{
  Render(){
    Return (
      <Div ClassName='Container'>
        <P ClassName='TextContent'>TextContent</P>
      </Div>
    );
  }
}
```

为了定位 Container 下的 P 元素，可能会使用 Css ：

```
// DefaultStyle.Less
.Container .TextContent{
  Color: Blue;
}
```

这样也可以，但是有一个多数老项目都会存在的问题就是，当外部要修改样式时， 需要写出优先级更高的 CSS 来实现。当组件嵌套层次过多时写出来的样式选择器特别长，而且很难维护。 但是用了 CSS Modules 之后，我们可以安心的使用：

```
// CSSModulesComponent
Import DefaultStyle From 'DefaultStyle.Less';
Class CSSModulesComponent Extends React.Component{
  Render(){
    Return (
      <Div ClassName={DefaultStyle.Container}>
        <P ClassName={DefaultStyle.TextContent}>TextContent</P>
      </Div>
    );
  }
}
```

```
// DefaultStyle.Less
.TextContent{
  Color: Blue;
}
```

写到这里，又有问题了，我们外部容器如何修改 CSSModulesComponent 里的样式呢？ 我们约定一个叫做 CssModules 的属性来对 DefaultStyle 进行改造：

```
// CSSModulesComponent
Import DefaultStyle From 'DefaultStyle.Less';
Class CSSModulesComponent Extends React.Component{
  Render(){
    Let {CssModules} = This.Props;
    Let MergedStyle = Object.Assign({}, DefaultStyle, CssModules);// 使 CssModules 能够覆盖 内部 Defau
    Return (
      <Div ClassName={MergedStyle.Container}>
        <P ClassName={MergedStyle.TextContent}>TextContent</P>
      </Div>
    );
  }
}
```

这样，当我们在父组件中使用时，就可以方便的对 CSSModulesComponent 中的 Container 和 TextContent 进行覆盖。

```
// ParentComponent
Import ParentStyle From 'ParentStyle.Less';
Class ParentComponent Extends React.Component{
  Render(){
    Return (
      <CSSModulesComponent CssModules={ParentStyle}/>
    );
  }
}
```

```
// ParentStyle.Less
.TextContent{
  Color: Red;
}
```

另外 CSS Modules 解析带有 Url 的 .Css 文件时可能会报错，解决办法是，修改 Webpack 中 Css-Loader 的 Loader 加入 Resolve-Url （需要安装 **Resolve-Url-Loader** 包）配置为：

```
{
  Test: /\.Css$/,
  Loader:'Style-Loader!Css-Loader?Modules!Resolve-Url!Postcss-Loader'
}
```

完。