

# 前端一键换肤实现，css变量兼容方案，ant-design换肤（转）



海天酱油\_htSa... LV3

2020年04月17日 16:02 · 阅读 4731

关注

## 废话部分

之前说的公司需求需要把老项目上面做一键换肤。然后就开始搞了呗。

## 一、一键换肤原理和几种方案

### 1、css in js

这个不多说吧，不管是react还是vue，说白了就是把颜色直接写成js变量。不过这是兼容性最好的一种方案

### 2、利用less或者scss

这个也简单，直接把css用less和scss管理起来。然后用这两者的操作函数进行换肤处理

### 3、css变量

这块就是新特性了，ie不支持。但是也有解决方案。

可以使用css-vars-ponyfill来达到兼容ie11的目的。不过限制较大。

给大家推荐一个demo地址：[https://segmentfault.com/a/1190000021093787?utm\\_source=tag-newest](https://segmentfault.com/a/1190000021093787?utm_source=tag-newest)

### 4、写成套的css然后来替换

我想不会有这么傻子的人吧

## 5、书写一套css，然后传值给后台，由后台生成整套的css进行替换

---

这块其实ant design和elementUI官方提供的一键换肤就是这么干的

## 二、项目背景

1、公司是两年前的vue项目，webpack2版本，然后组件库用的是Ant Design of Vue

2、需要解决的问题

1、首先，如果只是项目里面的css还挺好解决的，css in js方案就挺不错，但是你用别人的组件库，是不是也要改别人组件库的样式呢

2、第三方组件库的样式解决

3、如何去制定规则，毕竟你是开发，又不是美工。哪里懂美化

## 三、解决第三方组件库

这里比较抱歉，没elementUI的换肤方案。这块大家得靠自己了

首先我的方案是来自这位简书的博客：<https://www.jianshu.com/p/a42f58fc3f92>

完整的按这个兄弟的代码就能实现Ant Design不管是vue还是react的的组件库换肤方式

下面是我简化的步骤

### 1、安装npm install -D antd-theme-webpack-plugin

---

单独说一下，我自己在用的时候和下面的方式不一样

核心在于themeVariables的配置，我是注释掉的。因为这个完全不影响使用。如果配置了反而会导致样式错乱问题

配置方式：

```
const AntDesignThemePlugin = require("antd-theme-webpack-plugin");
const path = require("path");

const options = {
  antDir: path.join(__dirname, "./node_modules/ant-design-vue"), //antd包位置
  stylesDir: path.join(__dirname, "./src/styles/theme"), //主题文件所在文件夹
  varFile: path.join(__dirname, "./src/styles/theme/variables.less"), // 自定义默认的主题色
  mainLessFile: path.join(__dirname, "./src/styles/theme/index.less"), // 项目中其他自定义的样式
  outputPath: path.join(__dirname, "./public/color.less"), //提取的less文件输出到什么地方
  themeVariables: ["@primary-color"], //要改变的主题变量
  indexFileName: "./public/index.html", // index.html所在位置
  generateOnce: false // 是否只生成一次 (if you don't want to generate color.less on each change)
};

module.exports = {
  css: {
    loaderOptions: {
      less: {
        modifyVars: {
          // "primary-color": "#1DA57A",
          // "link-color": "#1DA57A",
          // "border-radius-base": "2px"
        },
        javascriptEnabled: true
      }
    }
  },

  configureWebpack: {
    plugins: [new AntDesignThemePlugin(options)]
  }
};
```

---

## 2、ant组件正常配置

---

但是请不要载入css文件，如果要载入那么也请用less方式载入

## 3、main.less和var.less

---

这里上面其实已经注释上面说了，关键是var.less里面

//一定要加上这一行!!!

@import "~ant-design-vue/lib/style/themes/default.less";

@primary-color: #992777;

default.less是ant所有变量声明的地方，这个不引入，那么一切白搭

## 4、index.html文件的修改

复制代码

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title>antd-vue-theme-demo</title>
  </head>
  <body>
    <noscript>
      <strong>We're sorry but antd-vue-theme-demo doesn't work properly without JavaScript </strong>
    </noscript>
    <!-- 引入提取的color.less文件,必须放body里面,放head里面,一键换肤会失效 -->
    <link rel="stylesheet/less" type="text/css" href="./color.less" />
    <script>
      window.less = {
        async: false,
        env: 'production'//production development
      };
    </script>
    <div id="app"></div>
    <!-- 引入less.js文件(放在最后) -->
    <script src="https://cdn.bootcss.com/less.js/2.7.3/less.min.js"></script>
    <!-- built files will be auto injected -->
  </body>
</html>
```

## 5、修改less变量实现你的换肤

看代码哦

关键是这个

```
window.less.modifyVars({
  "@primary-color": primaryColor
});
```

复制代码

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
    <a-button type="primary" @click="changeTheme('#992777')">默认</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#F5222D')">薄暮</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#FA541C')">火山</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#FAAD14')">日暮</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#13C2C2')">明青</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#52C41A')">极光绿</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#1890FF')">拂晓蓝</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#2F54EB')">极客蓝</a-button>
  </br /><br />
    <a-button type="primary" @click="changeTheme('#722ED1')">酱紫</a-button>
  </br /><br />
  </div>
</template>

<script>
export default {
  name: "HelloWorld",
  props: {
    msg: String
  },
  methods: {
```

复制代码

```
changeTheme(primaryColor) {  
  window.less.modifyVars({  
    "@primary-color": primaryColor  
  });  
}  
}  
};  
</script>  
  
<!-- Add "scoped" attribute to limit CSS to this component only -->  
<style scoped lang="less"></style>
```

## 6、原理

---

提取antd的less变量文件

我们的思路就是提取ant-design-vue中所有的less代码汇总到一个文件中，然后在index.html页面中直接引用，最后使用less.js中的modifyVars方法修改主题变量，这样antd的样式就不会被编译，实现运行时动态切换

## 7、自己的css怎么办

---

这里偷懒下用css变量，这样改起来也方便

```
:root {  
  
  primary-color: @primary-color  
  
}
```

之所以用把less变量赋值给css变量，主要原因在于css变量用js修改太麻烦了。而且你修改了less变量，css变量的值也变了

至于自己项目中使用颜色部分，那么就直接用var(--primary-color)就行了

## 8、换肤完成，但是怎么去定义你的规则呢。

这里我苦思冥想，看了elementUI以及ant官方的方式，最后我用了ant官方的方式来定义你的颜色

先放ant官方的设计思路：<https://ant.design/docs/spec/colors-cn>

色彩这块我们根据主颜色，然后生成10档色彩，这样的话，我们项目中基本上所有的颜色都可以根据这10档颜色去使用。那么很大一部分颜色定义就解决了

把var.less部分改为这样

复制代码

```
// ----- Colors -----  
@primary-color : red;  
  
// Color used by default to control hover and active backgrounds and for  
// alert info backgrounds.  
@primary-1: color(~`colorPalette("@{primary-color}", 1)`); // replace tint(@primary-color,  
@primary-2: color(~`colorPalette("@{primary-color}", 2)`); // replace tint(@primary-color,  
@primary-3: color(~`colorPalette("@{primary-color}", 3)`); // unused  
@primary-4: color(~`colorPalette("@{primary-color}", 4)`); // unused  
@primary-5: color(~`colorPalette("@{primary-color}", 5)`); // color used to control the te  
@primary-6: @primary-color; // color used to control the te  
@primary-7: color(~`colorPalette("@{primary-color}", 7)`); // replace shade(@primary-color,  
@primary-8: color(~`colorPalette("@{primary-color}", 8)`); // unused  
@primary-9: color(~`colorPalette("@{primary-color}", 9)`); // unused  
@primary-10: color(~`colorPalette("@{primary-color}", 10)`); // unused
```

css变量部分我们抄一样上面的变量，然后用就行了

## 四、致谢

<https://www.jianshu.com/p/a42f58fc3f92>