

Vue 移动端适配解决方案

1、动态设置 font-size 的 js

fontSize.js

```
// 获取屏幕宽度
let htmlWidth =
  document.documentElement.clientWidth || document.body.clientWidth;
// 获取html的dom元素
let htmlDom = document.getElementsByTagName("html")[0];
if (htmlWidth > 750) {
  htmlWidth = 750;
}
htmlDom.style.fontSize = htmlWidth / 10 + "px";
// 通过监听事件，时刻的动态的修改font-size的大小(屏幕宽度变化时,font-size的大小也随着变化)
window.addEventListener("resize", function(e) {
  // 获取屏幕宽度
  htmlWidth = document.documentElement.clientWidth ||
  document.body.clientWidth;
  // 获取html的dom元素
  if (htmlWidth > 750) {
    htmlWidth = 750;
  }
  htmlDom.style.fontSize = htmlWidth / 10 + "px"; //设计稿的大小是基于750的宽度设计的
});
```

在 main.js 中引入 font-size.js 文件

```
import "../common/fontSize.js";
```

2、配置 flexible

安装 ib-flexible

```
npm install ib-flexible --save
```

引入 ib-flexible

```
import "ib-flexible";
```

3、配置视口

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

4、配置 px 转 rem 工具

安装 px2rem-loader

```
• npm install px2rem-loader --save-dev
```

配置 build/utils.js 文件

```
var cssLoader = {
  loader: "css-loader",
  options: {
    minimize: process.env.NODE_ENV === "production",
    sourceMap: options.sourceMap
  }
};

var px2remLoader = {
  loader: "px2rem-loader",
  options: {
    remUnit: 75
  }
};
```

并进 loader 数组中

```
function generateLoaders(loader, loaderOptions) {
  var loaders = [cssLoader, px2remLoader];
}
```

5、配置完成以后重启项目

6、总结