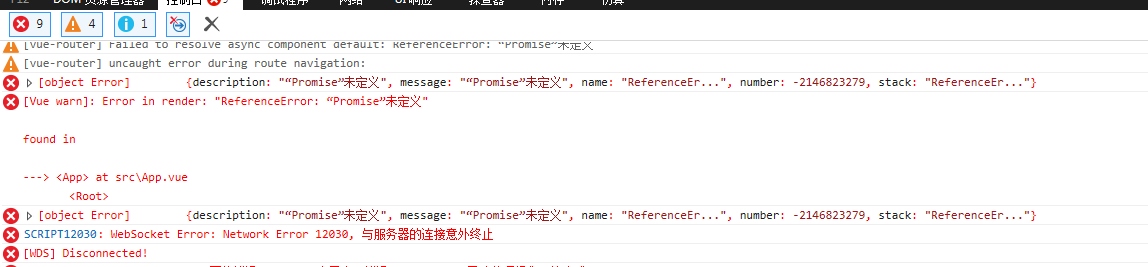
Ie8兼容问题

1.问题：报警的代码在ie11下报错：



搜集到的资料：

1.babel只能在构建过程中转译语法糖，比如说将ES7/ES6/JSX中的语法部分（例如箭头函数）转译成ES5。

而polyfill则是解决原生对象的问题，比如说你的浏览器不支持Promise，babel就给你建一个Promise供你使用。

所以，polyfill是肯定得在浏览器运行的，不然怎么把原生对象插进全局变量呢？

2. 如果我们没有配置一些规则，Babel 默认只转换新的 JavaScript 句法（syntax），而不转换新的 API，比如 Iterator、Generator、Set、Maps、Proxy、Reflect、Symbol、Promise 等全局对象，以及一些定义在全局对象上的方法（比如 Object.assign ）都不会转码

3.一定要在在应用的入口引用，以确保它能够最先加载

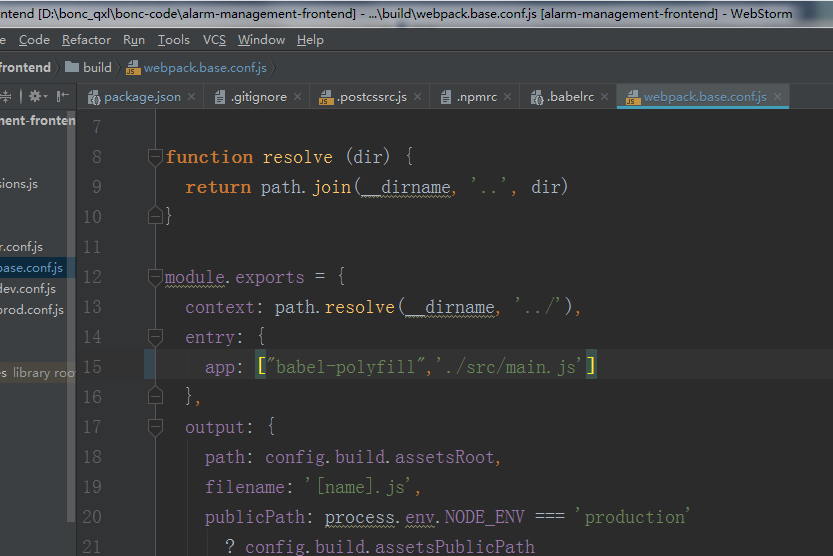
4.看这个里说的babel-runtime和babel-polyfill都可以解决这个问题，但是，二者有不同，前者是插件，在插件里引入，而后者相当于一个模块，在入口处引入，且babel-polyfill 是直接在原型链上增加方法。参考的连接如下：

<http://www.mamicode.com/info-detail-2289719.html>

解决：

参考：<https://www.cnblogs.com/princesong/p/6728250.html>

安装babel-polyfill后：





说明这个方法在ie11下可行，不知ie8下如何

注意修改了webpack等一定要重新编译

**1.问题**

alarm-management-frontend项目在ie11上运行时出现不兼容的问题，如下图：

**2.解决方法**

首先使用命令行安装babel-polyfill插件：npm install --save-dev babel-polyfill

然后在webpack配置文件即webpack.base.config.js文件里项目的入口处引入该插件：

对比修改之前：

修改之后：

**3.原理解析**

如果我们没有配置一些规则，Babel 默认只转换新的 JavaScript 句法（syntax），而不转换新的 API，比如 Iterator、Generator、Set、Maps、Proxy、Reflect、Symbol、Promise 等全局对象，以及一些定义在全局对象上的方法（比如 Object.assign ）都不会转码，而babel-polyfill 会给其做兼容。应该在应用的入口引用，以确保它能够最先加载。与之相似的babel-runtime 不能转码实例方法，而babel-polyfill 是直接在原型链上增加方法，实用性更广

2.第二个兼容问题（biop的代码）

