gulp+browserify编译es6方法及常见错误

随着es6逐渐稳定成熟，在web开发中使用也越来越广泛。但是，浏览器对es6还没有达到完全兼容的地步。所以需要借助一些工具来对其进行编译。这里就介绍两个工具gulp和browserify来对es6进行编译打包。

对于gulp和browserify的原理的介绍并不是这里的主要内容。本篇我们主要针对如何使用gulp和browserify来对es6编译进行一个说明，并且对其使用过程中出现的问题做一个介绍以及如何解决该类问题。

**环境搭建**

首先使用npm初始化一个package.json文件

$ npm init

然后需要安装gulp和browserify到本地项目中

$ npm install –svae-dev gulp

$ npm install –save-dev browserify

接着还需要安装两个辅助的工具babelify 和 vinyl-source-stream

$ npm install –save-dev babelify

$ npm install –save-dev vinyl-source-stream

上面所有工具安装完成以后在package.json文件中会有以下依赖

"devDependencies": {

"babel-preset-es2015": "^6.18.0",

"babelify": "^7.3.0",

"browserify": "^13.3.0",

"gulp": "^3.9.1",

"vinyl-source-stream": "^1.1.0"

}

**编写代码**

首先在根目录下新建一个文件onmpw.es6.js

import {onmpw} from "./lib/onmpw";  
var moma;  
export default moma = function(){  
 onmpw ();  
 console.log('es6');  
}  
window.moma = moma;

然后在根目录下新建gulpfile.js文件。编写以下代码

var gulp = require('gulp');  
var browserify = require('browserify');  
var babelify = require('babelify');  
var source = require('vinyl-source-stream');  
gulp.task('onmpw',function(){  
 return browserify({  
 entries: "./onmpw.es6.js",  
 debug: true,  
 })  
 .transform(babelify)  
 .bundle()  
 .pipe(source('onmpwes6.js'))  
 .pipe(gulp.dest('dist'));  
})  
gulp.task('default',['onmpw']);

然后运行gulp

$ gulp

这里我们在根目录下运行gulp，并且gulpfile.js文件也在根目录下。所以gulp会自动读取gulpfile.js文件。如果gulpfile.js文件不再根目录下，我们还需要指定gulpfile.js所在位置。

$ gulp –gulpfile gulpfile.js所在目录

运行上述命令，顺利情况下会编译成功

[22:31:23] Using gulpfile /www/onmpw\_plugins/ gulpfile.js

[22:31:23] Starting 'onmpw'...

[22:31:26] Finished 'onmpw' after 3.12 s

[22:31:26] Starting 'default'...

[22:31:26] Finished 'default' after 39 μs

但是事情总不是那么顺利，这时很可能会报一下错误

events.js:160

throw er; // Unhandled 'error' event

^

SyntaxError: 'import' and 'export' may appear only with 'sourceType: module'

**错误解决**

出现上述错误，我们首先就要去检查各个工具的版本。在package.json文件中我们可以找到browserify和babelify的版本分别为13.3.0和7.3.0。这是非常新的版本，需要借助babel-preset-es2015工具（至于其原理是什么，大家可以自行去脑补）。

$ npm install –save-dev babel-preset-es2015

然后修改gulpfile.js内容中的

return browserify({  
 entries: "./onmpw.es6.js",  
 debug: true,  
 })

.transform(babelify)  
改为

return browserify({  
 entries: "lib/momaEntry.js",  
 debug: true,  
})  
.transform(babelify.configure({  
 presets:['es2015']  
}))

也可以在根目录下新建 .babelrc 文件，然后写入以下内容

{

presets:['es2015']

}

两种方式都是可以的。

当然了，既然知道是版本太新所引起的。除了上面添加辅助工具以外，还可以降低工具的版本。我自己经过试验，得出只是降低babelify的版本为6.0.2即可。

$ npm install –save-dev [babelify@6.0.2](mailto:babelify@6.0.2)

package.json依赖工具的内容如下

"devDependencies": {

"babel-preset-es2015": "^6.18.0",

"babelify": "^6.0.2",

"browserify": "^13.3.0",

"gulp": "^3.9.1",

"vinyl-source-stream": "^1.1.0"

}

这样我们就可以不用借助babel-preset-es2015这个工具了。也不用再修改gulpfile.js既可编译成功。

其实很多时候我们在搭建环境的时候按照网上的教程一步一步来做，发现工具都是一样的，该安装的一个也没有丢掉，为什么还是不能成功？其实这时我们就应该考虑软件的兼容性问题了。

上面出现的错误也是我自己在项目中碰到的。几经挫折算是解决了。这里将经验分享给大家作为开发中的一项参考。可能并不全面，但是希望大家能用的上。