

03.DevServer 服务

学习要点：

1. 问题需求
2. 安装部署

本节课我们来开始学习一下如何将 DevServer 服务工具整合部署到 Webpack。

一. 问题需求

1. 由于代码可能有 CommonJS 等非浏览器支持的语法，每次都必须打包才行运行；
2. 虽然借助 Webstorm 等工具可以构建服务器环境，但实际上不能时时监控刷新；
3. IDE 提供的服务器之间访问的是打包后的文件，是否监听时时刷新看个人习惯；
4. 以上：如果要方便的话，开发者需求的想法是，开发时方便调试，最后再打包；
5. 所以，官方提供了 Webpack-dev-Server 工具来解决这个问题，支持特性：
 - (1) .支持 HTTP 服务访问：localhost,127.0.0.1 这种；
 - (2) .监听变化时时刷新网页，时时预览；
 - (3) .支持 Source Map；

二. 安装部署

1. DevServer 只要安装到本地即可使用，命令如下：

```
npm i webpack-dev-server -D
```

PS：在应用根目录安装，package.json 会自动添加 webpack-dev-server 版本号

2. 由于课件是按照 01,02 这样目录构建的，你自己测试完全可以在根目录下；
3. 当然，根目录和子目录在部分操作上可能有路径问题，自行调试几次；
4. 运行 DevServer 时，由于我们在 03 这个子目录中，所以，我们要进入：

```
cd 03
```

5. 在 webpack.config.js 配置一些最基本的参数，方便运行；

//devServer 自动化

```
devServer: {  
  publicPath : '/dist',           //访问路径  
  port : 3000,                   //独立端口  
  stats: 'minimal',              //迷你型服务启动信息  
},
```

PS：本地我们可以删除 dist 目录，还原打包之前再启动 devServer，测试效果；

PS：此时我们可以发现并不需要打包到本地，它是自动打包到内存让你时时预览调试的；

PS：也就是说：调试阶段，可以用 devServer，完成了，再最终打包到本地即可；

6. 我们可以在子目录生成一个 `package.json` 文件，在 `scripts` 设置 `dev` 属性：

```
"dev" : "webpack-dev-server"
```

```
npm run dev
```

7. 目前这个版本，火狐还是会有断开服务器的提醒，不过完全不影响我们调试；
8. 如果强迫症的同学，有两种解决方案：
- (1) .注释掉输出的错误信息，或者把错误提醒改成信息提醒；
 - (2) .或者在 `devServer` 设置错误级别：

```
clientLogLevel : 'none'
```

PS: 还有火狐还会出现 `sockjs.js.map` 的警告，可以设置 `devTool` 解决：

```
devtool: 'source-map',
```

PS: 以上问题，谷歌浏览器均不存在！