Node.js 相关整理

作者：DJagger
原文链接：https://ld246.com/article/1488535774556
来源网站：链滴

简介

Node.js®registered is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem(生态系统), npm, is the largest ecosystem of open source libraries in the world.

这段话是Node.js官网对于Node.js的描述。其中提到了几个比较重要的东西：

- **Chrome's V8 JavaScript engine**: 这个开源引擎是用C++写的，应该是比较快吧，其实就是一个Javascript的解释器包括了几个集成的模块。部分node的模块也是用C++写的，所以不是很需要担心速的问题。
- **event-driven, non-blocking I/O model**: 事件驱动非阻塞I/O模型（比较像状态机？），Node现了一个event模块来处理事件，就是EventEmitter类，Node中很多的类都是继承自此。
- **npm**: **N**ode.js **p**ackage **m**anager，Node.js官方的包管理系统，主要用于Node项目的依赖管理还可以定义一些环境变量，或者给某些插件传递参数。所有的npm配制信息都在项目根目录下的package.json文件中，后面的例子中会提到简单的用法，其他更多的用法和细节可以查看npm文档。
- **open source**: 开源虽然会导致代码质量的参差不齐，但是在解决某一问题时会十分有效。建议在些非关键的小步骤中多使用开源的东西，而不是自己造无意义的轮子。
- **world**: 英语水平 is !important.

Node.js的安装

浏览器打开Node.js官网可以看到大大的Download，下载下来、解压就好了。对于mac、win用户可自动安装，linux服务端用户可以使用wget命令获取到压缩包，tar命令解压，ln命令将bin目录下的node于npm链接（建议软链接）到/usr/local/bin目录，或将bin目录加到环境变量下。

安装成功后在控制台运行以下命令：

```
node -v
#v6.3.1

npm -v
#3.10.3
```

不报错就表示安装成功了。

Node.js可以做什么

Node.js实际上是运行在本地的Javascript，封装了大部分常用的文件系统操作，封装了TLS/SSL、socket等较为底层的库，也提供了http库供后端人员使用。在第三方库express的支持下，更是给服务端发指了一条新路。它虽然不能在浏览器上运行，但是却在开发环境下大放异彩。

web开发流程优化

大部分人最开始接触web开发时，都觉得HTML是世界上最好写的语言，结构清晰而且跑起来好看。而正式学习前端了之后才发现，不止HTML、css、js有多个版本，浏览器更是有海量个版本。这里推给大家一个网站caniuse，顾名思义，这个网站就是看某一个关键字或者函数的兼容性情况。话说回，在如此多的浏览器中，想要同一段代码的行为相同是一件极其棘手的事情。

用css举例，不同的浏览器对CSS3的属性前缀支持不同，每次写带前缀的属性时都要连着写四个几乎同的句子（-webkit-、-moz-、-ms-、）。这在版本迭代时是很容易出现问题的。npm中的一个库autoprefixer就完美地解决了这个问题。下面给大家一个例子顺便简单地介绍一下npm的操作。

```
npm install autoprefixer -g
```

有兴趣看一下npm源码可以知道，npm实际上是用node编写的一个脚本。install是npm的安装命令可以简写为i，对应的remove命令简写为r。install命令会在当前目录创建一个node_modules文件夹如果不存在的话），并下载npm服务器上对应的包到此目录。node_module目录下会有一个叫做.bin子目录，会有下载的node模块中的可执行文件。-g参数会将包安装到“global”中，也就是个人目下，这样就可以在sh中直接调用已安装的包。注意：-g命令需要一定目录权限，linux / mac下可能要sudo。

回到正题，下面是一个未加前缀的CSS文件：

```
/*styles.css*/
.some-pig {
  transform: scale(5);
}
```

在bash中调用：

```
autoprefixer-cli styles.css
```

再打开styles.css会发现自动加上了一条带前缀属性。

不只是css的前缀，包括less以及sass等css扩展也有对应的包来处理。

甚至对于Javascript本身，都能使用node将高版本js“编译”成低版本js，这就涉及到了十分流行的babel。babel是一个令人热血澎湃的项目，它使得程序员可以使用所有先进的js语言特性，而不被语言本的兼容性掣肘。现在前端最流行的框架react使用的jsx语法，就是需要用babel这种转化成浏览器能别的js语法。

npm上更有像gulp、webpack这种一条龙服务的工具，让前端的开发更加“舒坦”。当然，这些工的学习是需要一定时间代价的，不过对于英语好的我（呵呵）来说都是小事。在它们的帮助下，我可动态的将本地的文件改变直接提现在浏览器的调试页面中，免去了刷新、缓存的痛苦。

babel、webpack的用法我会在之后的博客中进行总结。

简易服务器

许多人都用过一个叫做SimpleHTTPServer的python包，用在简单的测试上时非常爽。node中也有个http服务的模块，第三方封装成了一个叫做express的框架。这个框架的扩展性极强，而且上手简，非常适合写简单的服务端。甚至只用不到10行就能配出一个静态文件服务器（当然，如果只是静态件服务器的话可以使用http-server包）：

```
npm i express --save
```

--save命令会将安装的包的信息插入到package.json中，只有在package.json中的包才能在代码中用。

```
var express = require("express");       // 刚刚安装的express包
var path = require("path");           // node集成的路径处理模块
```

```
var app = new express();
app.use(express.static(path.join(__dirname, "static")));
app.listen(1234);
```

这段代码其实看起看很简单，就是监听1234端口，可以通过http访问static目录下的所有文件。其中rquire函数就是加载包的方法，正如前文所说，只有在package.json中指明的包才能被node定位到，以安装时一定要记得加上--save参数。express的用法我会之后进行简单的总结。

异想天开

html调试的“奇葩方法”——browsersync：它实际上是一个http服务器，在html中注入一段脚本对dom进行调试，并且可以同步多个测试的行为，用起来真的爽。

本地跑的web——node-webkit：它将谷歌浏览器内核与node相整合，使得在html中的script中可跑node代码！也就是说不需要http请求服务器，服务器返回信息再渲染，可以直接在script标签中查数据库！

Next-

接下来我会介绍一些常见的node库：

- babel
- webpack等前端构建的库
- express
- browser-sync
- node集成的一些库