

windows下的node-canvas历程

笔记本： 00前端包管理工具NPM

创建时间： 2021/7/25 20:03

更新时间： 2021/7/25 20:10

URL: <https://www.cnblogs.com/virtual/p/3661592.html>

Anderssenchen注：在express-admin项目中，package.json文件中有依赖canvas，按照以下教程搭建的是npm全局安装。因此在express-admin项目下使用npm install时，仍然在install到canvas时会报错，方式一：不需要管错误，在项目跑起来的时候，项目下node_module找不到后会查找本机全局依赖。方式二：注释掉package.json中canvas的依赖

另外，如下：不要使用new Canvas方法，使用Canvas.createCanvas(W, H)方式代替

```
// var canvas = new Canvas(W, H);  
var canvas = Canvas.createCanvas(W, H);
```

windows下的node-canvas历程

背景：由于在前期开发的过程中，对前端的小图片采用了css-sprite，开始的时候都是在<http://spritegen.website-performance.org/>站点上合成图片及样式的，但是某天该站点突然不能访问，开始还怀疑是运维把站点给屏蔽了，后来才知道这个站点真的挂了，顿时觉得使用web站点上的工具太没有安全感了，所以决定在本地搭建一个css-sprite的工具环境。

所以接下来就是nodejs的css-sprite之旅了。nodejs处理图片需要使用到node-canvas这个图像模块，在windows环境下搭建整个环境还是需要几个步骤：

1.安装Python 2.7版本，3.0以上不支持；下载地址：

<https://www.python.org/downloads/>

2.安装node-gyp模块，使用命令行：npm install -g node-gyp；

3.安装Visual C++ 2010 Express，编译时需要使用，下载地址：

http://www.visualstudio.com/zh-cn/downloads/download-visual-studio-vs#DownloadFamilies_4

4.下载GTK包并解压在C:\GTK目录下并添加C:\GTK\bin到系统path路径中；

(GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off tools to complete application suites.)

win64下载地址(推荐)：

http://ftp.gnome.org/pub/gnome/binaries/win64/gtk+/2.22/gtk+-bundle_2.22.1-20101229_win64.zip

5.安装node-canvas模块，使用命令行：npm install -g node-canvas；

编译成功后，在全局模块目录下会存在
npm\node_modules\canvas\build\Release\canvas.node文件，在此目录下使用
node命令行执行var c = require('canvas');如果输出undefined则编译成功；

到此，不知道你已经踩过几个坑了，反正我踩好些个。

1.由于我之前安装了Python 3.0以上版本，故在node-canvas编译时报错，此错误在错误提示信息中很明显，直接下载个Python 2.7版本并改写path路径即可解决。

2.开始时我是在<http://www.gtk.org/download/win64.php>下载all-in-one bundle(即GTK+的所有编译文件)，这个已是当前最新版本gtk+-bundle_3.6.4-20131201_win64，

但是在使用该版本编译时会提示C:\GTK\lib\libpng.lib未找到的错误，这是一个图片处理的核心库文件，不知道包提供者为什么没有把libpng给编译成lib文件。这个问题编译出来也很明显，所以就在GTK+站点上下载了libpng模块，提取其中的libpng.lib文件，libpng1.5版本没有提供libpng.lib文件，libpng1.4版本中有提供，所以我就使用了1.4版本中的libpng.lib文件，编译成功后执行var c = require('canvas');提示模块缺失，就使用了depends检查Release目录下canvas.node的dll依赖，发现存在几个dll缺失的情况，从系统system32中查找相应的dll并拷贝到C:\GTK\bin目录下，运行var c = require('canvas');输出为undefined。正确。

depends下载地址：<http://www.dependencywalker.com/>

3.运行npm install spritegen下载spritegen模块，该模块调用canvas做了些css-sprite的功能，运行其中的test，执行失败，提示libpng的1.4与1.5版本使用冲突，具体是C:\GTK\include\libpng15使用了libpng 1.5版本但是在问题2解决的过程中我使用了1.4版本中的lib文件，由于站点上下载的libpng1.5版本未能提供lib文件，所以我尝试使用vc++ 2010版本编译，但是未能成功，由于受限于公司开发环境也不想vs版本上做过多纠缠，所以就想在web上download一个libpng.lib文件，但是所download的文件均未能正常编译。最后放弃了GTK+的最新版本使用了gtk+-bundle_2.22.1-20101229_win64版本，node-canvas正常编译，且执行spritegen test成功，即canvas模块能正常使用了。

一切问题解决完就是在spritegen代码的基础上增加自己的功能了，保证和spritegen.website-performance.org站点生成出来的格式一致，这样也就不需要去改以前写好的css样式了，而且也可以添加自己想要的功能，最最重要的是在本地生成，安全便捷可重用^_^。

参考：

<https://github.com/Ancestry/spritegen>

<http://delarre.net/posts/installing-node-canvas-for-windows/>

<https://github.com/LearnBoost/node-canvas/wiki/Installation---Windows#installing-python-for-windows>

<http://www.gtk.org/download/win64.php>

