

Html转Pdf记

背景

：需要把HTML文件转成PDF格式文档。最好提供API，供第三方调用，用以生成PDF文档。

任务

：从开源软件中，找出转换格式较好的一款软件。最好有现成的API用以转换。如果没有，那就自己开发。

行动

：从github上找了四款开源软件：DomPdf、Html2Pdf.it、Html2Pdf.js、iText4Java-Html2Pdf，进行兼容测试、评估开发难度，最后选定软件。

结果：经过测试，选择了Html2Pdf.it。并在此基础上定制开发。转换速度、转换结果都很不错。

以下记录整个过程，备忘。

• 1.选软件

限定每个软件4小时，完成部署、测试的任务。结果Dompdf、Html2Pdf.it、Html2Pdf.js很快能在本地Windows机部署成功，iText4Java-Html2Pdf运行还有些问题。经过测试部署成功的3款软件，发现Html2Pdf.it可以完美地把Html网页以Pdf格式呈现出来，DomPdf在某些样式上存在缺失，如背景颜色；html2pdf.js效果就稍微差些。好吧，那就基于Html2Pdf.it进行开发部署。

• 2.开发

Html2Pdf.it原生自带一个get

API，当用户传入Html网页的url、需要生成PDF的格式时，系统会自动返回一个pdf文件。

我想给它增加一个Post API，当用户传入Html内容时，也可以生成Pdf。

基于原有的Get API进行开发，很快就完成了Post API，测试，没有问题。测试完成之后就可以部署了。

• 3.部署

我挺喜欢Pivotal Cloud

Foundry（简称PCF），一个简单的push命令就能部署到系统中，它会自动找到相对应的容器，安装部署包，启动。所有一切，一行命令就够了。

部署完成后，测试发现HTML中包含的中文字符全都变成了空格。

第一反应是PCF容器中没有安装中文字体。

怎样安装呢？运行help发现可以使用cf

ssh命令连接到PCF容器中。连接上后，查到容器是基于Linux的，初步估计安装步骤应该和Linux安装字体完全一样。

步骤比较简单：1. 在目录/usr/share/fonts/下新建chinese目录；2. 把windows机中的字体复制到这个这个目录中；3.运行fc-cache -fv命令安装字体；4. 运行fc-list :lang=zh命令验证安装的中文字体。

结果第1步就卡壳了，没有权限，需要提供vcap账号的密码。这....

我根本就不知道什么vcap账号啊。

查询PCF的官方文档，发现这段文档：Configure Pre-Runtime Hooks，这意味着我只需要在项目里增加一个shell脚本，命名为.profile文件，把安装字体的脚本写到这个文件里就可以了

。这样，在启动应用之前，会运行.profile文件，把脚本安装好。

好，说做就做。做完再push一次，发现中文仍旧无法显示。

怎么定位问题呢？呃，PCF命令里有个logs，试试看吧。

结果发现，即使把脚本放到.profile里，在运行.profile中的脚本时，系统仍旧提示没有权限。哎，真头疼。

等等，之前在执行fc-cache命令的时候，从输出结果来看，好像会扫描很多目录，判断这些目录中是否有字体。嗯，我可以试试看其他的目录，例如/home/vcap/.fonts，说不定某些目录我有权限。

测试之后，发现果真有权限。字体复制成功，运行fc-list :lang=zh时能列出中文字体。可是，中文字体仍旧不能显示。

那...

是不是这个软件对中文的支持不好。在Windows机上可以啊，再试下Linux机真机，也能完美转换。

或者，是代码的问题？尝试着理解Html2Pdf.it的底层实现吧。

好在html2pdf.it的底层实现比较简单。在接收到API调用的参数后，解析成对应的参数，调用底层的phantomjs来实现html转pdf。

尝试着直接通过phantomjs的命令行来转换，发现中文完美显示。看来不是phantomjs的问题，那就是html2pdf.it的代码问题咯。

从代码来看，应用的直接入口在pdf.js文件。在这个文件里，直接调用phantomjs，并把rasterize.js文件作为参数传进去。测试使用命令行直接运行phantomjs的时候，把rasterize.js文件作为参数传进去，发现转换中文正确。看来pdf.js的嫌疑最大了。仔细研读pdf.js代码，它使用了nodejs的child-process的spawn方法，会是nodejs的bug吗？或者实施其他方法，如exec、execSync、execFile、execFileSync。很遗憾，尝试了各种方法，仍旧无法转换中文。

可是，在PCF上NodeJS的版本只能是预定义的几种版本，我没办法做更多的尝试。在此期间，我曾经脑子里闪过过这种念头，要不要把NodeJS版改写成Java版再试试看。但马上又否决了，时间成本太高，放弃吧。

最终，决定不再花费更多时间，就把应用部署到Linux虚拟机上吧。运行正常。