

程序导出 word 文档的方法

将 web/html 内容导出为 world 文档，再 java 中有很多解决方案，比如使用 Jacob、Apache POI、Java2Word、iText 等各种方式，以及使用 freemarker 这样的模板引擎这样的方式。php 中也有一些相应的方法，但在 python 中将 web/html 内容生成 world 文档的方法是很少的。其中最不好解决的就是如何将使用 js 代码异步获取填充的数据，图片导出到 word 文档中。

1. unoconv

功能：

1.支持将本地 html 文档转换为 docx 格式的文档，所以需要先网页中的 html 文件保存到本地，再调用 unoconv 进行转换。转换效果也不错，使用方法非常简单。

```
\# 安装
sudo apt-get install unoconv
\# 使用
unoconv -f pdf *.odt
unoconv -f doc *.odt
unoconv -f html *.odt
```

缺点：

- 1.只能对静态 html 进行转换，对于页面中有使用 ajax 异步获取数据的地方也不能转换（主要是要保证从 web 页面保存下来的 html 文件中有数据）。
- 2.只能对 html 进行转换，如果页面中有使用 echarts,highcharts 等 js 代码生成的图片，是无法将这些图片转换到 word 文档中；
- 3.生成的 word 文档内容格式不容易控制。

2. python-docx

功能：

1.python-docx 是一个可以读写 word 文档的 python 库。

使用方法：

1.获取网页中的数据，使用 python 手动排版添加到 word 文档中。

```
from docx import Document
from docx.shared import Inches
document = Document()
document.add_heading('Document Title', 0)
p = document.add_paragraph('A plain paragraph having some ')
p.add_run('bold').bold = True
p.add_run(' and some ')
p.add_run('italic.').italic = True
document.add_heading('Heading, level 1', level=1)
document.add_paragraph('Intense quote', style='IntenseQuote')
document.add_paragraph(
    'first item in unordered list', style='ListBullet'
)
document.add_paragraph(
    'first item in ordered list', style='ListNumber'
)
document.add_picture('monty-truth.png', width=Inches(1.25))
table = document.add_table(rows=1, cols=3)
```

```
hdr_cells = table.rows[0].cells
hdr_cells[0].text = 'Qty'
hdr_cells[1].text = 'Id'
hdr_cells[2].text = 'Desc'
for item in recordset:
    row_cells = table.add_row().cells
    row_cells[0].text = str(item.qty)
    row_cells[1].text = str(item.id)
    row_cells[2].text = item.desc
document.add_page_break()
document.save('demo.docx')
```

缺点：

1.功能非常弱。有很多限制比如不支持模板等，只能生成简单格式的 word 文档。

程序导出 PDF 文档方法

1.pdfkit

功能：

1.wkhtmltopdf 主要用于 HTML 生成 PDF。
2.pdfkit 是基于 wkhtmltopdf 的 python 封装，支持 URL，本地文件，文本内容到 PDF 的转换，其最终还是调用 wkhtmltopdf 命令。是目前接触到的 python 生成 pdf 效果较好的。

优点：

1.wkhtmltopdf：利用 webkit 内核将 HTML 转为 PDF

webkit 是一个高效、开源的浏览器内核，包括 Chrome 和 Safari 在内的浏览器都使用了这个内核。Chrome 打印当前网页的功能，其中有一个选项就是直接“保存为 PDF”。

2.wkhtmltopdf 使用 webkit 内核的 PDF 渲染引擎来将 HTML 页面转换为 PDF。高保真，转换质量很好，且使用非常简单。

使用方法：

```
\# 安装
pip install pdfkit
\# 使用
import pdfkit
pdfkit.from_url('http://google.com', 'out.pdf')
pdfkit.from_file('test.html', 'out.pdf')
pdfkit.from_string('Hello!', 'out.pdf')
```

缺点：

1.对使用 echarts，highcharts 这样的 js 代码生成的图标无法转换为 pdf（因为它的功能主要是将 html 转换为 pdf，而不是将 js 转换为 pdf）。对于纯静态页面的转换效果还是不错的。

2.其他

其他生成 pdf 的插件还有：weasyprint，reportlab，PyPDF2 等，经简单试验都不如 pdfkit 效果好，且有些用法复杂。

generated by [haroopad](#)