

Madoko 的简单试用

hcl
i@th0.me

Madoko (**M**arkdown in **K**oka) 是 Microsoft Research 搞出来的一个运行在浏览器中的 Markdown 编辑器，其宣传最大的优势是更加学术，最近进行了一些试用。

0.1. Markdown + LaTeX ?

Madoko 之所以更为学术，是因为其融合了 LaTeX 的一些功能。可以说是对 LaTeX 语法的简化再配合上 Markdown 的语法。

但是这样的融合真的是否能够结合二者的好处，我个人的观点是基本肯定但保留一点怀疑的。因为 LaTeX 本身作为一种排版语言，较为接近于底层，这样的好处在于，你可以排版出任何你想要的版式，再加上各种宏包的应用，你甚至可以实现你想做的任何事。而 Markdown 作为最近几年兴起的一种 HTML 标记语言的简化形式，其最大的好处就是清晰易读、易书写，只要掌握一些简单的规则就可以免去写传统网页时的复杂排版，而只需要把注意力放到内容上。

所以 Markdown+LaTeX 的初衷很不错，其概念我认为也有一定的可取之处。毕竟 LaTeX 这种东西学习曲线还是挺陡的，光是配置环境可能就能难倒一批人。而 Markdown 简单易学，如果 Markdown 能排版出 LaTeX 的效果，自然很不错。事实上 Madoko 也确实做到了可以生成 LaTeX 效果的 PDF。你再也不用费劲去记各种 LaTeX 的命令块了，引用文献也不用费了半天劲还编译不出来所要的结果，真正所见即所得。当然这些好处的前提是你本身对排版要求不高的情况下。刚才已经说过 LaTeX 最大的好处就是较为接近底层。如果说 Knuth 原始的 TeX 是汇编的话，LaTeX 就是 C 语言一样的存在，既能底层，又能封装成上层。但是 Madoko 的定位就相当于 Python 一类的高级脚本语言，用 Python 去搞驱动和嵌入式不是完全不

可能但是不合适。（这里没有黑任何一门语言的意思）

所以，如果你对排版有较高的要求、或者学术文章有特别的排版需求，就最好不要用 Madoko 了。而对于初学者，或者纯 Markdown 使用者，Madoko 还是值得去尝试的。

0.2. 中文之殇

如果你写英文文档或者学术文章，Madoko 默认提供的模板足够你用甚至拿去投稿。但是，如果你是一个中文用户，你又要写中文的文章亦或是你要写学术文章，Madoko 显得就比较尴尬了。用 Madoko 写中文就是我上面提到的“较高要求”的排版。为什么？

因为中文支持的问题是从 LaTeX 上古时期就留下来的问题，具体渊源这里不细说，但是好在现在已经有了 `xeCJK` 以及高级封装宏包 `ctex`。LaTeX 渲染中文必定需要引用这些宏包（或是之一）。Madoko 确实也提供了 LaTeX 引用宏包的高级功能，在引言部分放一句

```
Package : ctex
```

或者

```
Package : xeCJK
```

就可以了。但是如果我要指定字体呢？Madoko 倒是提供了对宏包进行设置的方法：

```
Package : [fontset=windowsnew]ctex
```

但是如果要进行更为高级的设置可能就没有办法直接用 LaTeX 的源代码直接改，例如：

```
\ctexset{fontset=adobe}
```

总之，对宏包的引用和基本设置是没有问题的，可以看出设计者至少是考虑到了高级用户的使用。但是对于高级用户要定义自己的命令可能就比较麻烦了。而且服务器的环境也不是透明的，很有可能造成一定的问题。

0.3. 本地 or 在线

Madoko 在官方网页上提供的编辑器是调用的其后端的服务器进行 LaTeX 进行渲染的。在线的东西总归不太放心，万一断个网，又要写点东西；或者

突然把浏览器关了，数据就没了。本地存储确实不如在线云来的安全，但是至少我能控制。好在 Madoko 提供了本地版，你需要的只是本地的 Node.js 环境以及 npm。对于 Linux 来说这不是问题，包管理器 Terminal 下面几行命令就搞定了。

```
sudo npm install -g madoko-local
```

Windows 也不会更麻烦。接着执行

```
madoko-local -l .
```

就会以当前目录作为工作目录，启动服务，然后 `-l` 参数自动打开浏览器。所以相当于是你开了一个本地服务器。对于 Linux 来说，监听端口是只有 root 权限才能做到的，所以你得 `sudo`。

然而，这并没有结束，最终渲染成 LaTeX 还是要连接到远程的服务器进行。至于隐私问题，反正说是生成完就删除。

另外这里插一点，最开始我引用了一些包，很不幸报错。报错信息里面直接就是 `C:\User\xxx\` 的 Windows 路径，所以猜想这后端肯定是挂在 Windows Azure 上的。事实上在后面的实验中，发现确实 Windows 下的 Madoko+TeXLive 工作得很好。

如果要纯本地生成，要求你的本地必须要有 LaTeX 环境，官方推荐是 TeXLive。另外还要再装一个名叫 madoko 的包

```
npm install -g madoko
```

在 Windows 下可能会比较容易，事实上也成功了。但是在 Linux 下，由 `sudo` 启动的 Madoko 可能会找不到你的 LaTeX 环境，但是这是 Linux 本身的问题。但是就算是解决了这一个问题。个人尝试数次仍然不能在 Linux 下进行本地编译，会出现一些错误，还有待进一步研究。设计者应该是在 Windows 下开发的，对于 Linux 的兼容性有待提高。

0.4. 总结

不管怎样，Madoko 在 Linux 下的表现不是太好，在 Window 下基本上就是两条命令就能搞定。或许是有待我自己去研究。至于 Madoko 究竟好不好用，还是要亲自去尝试一番。毕竟 Madoko 可以说是对 LaTeX 高级封装的大胆尝试。