

从0开始学习Typora---好用的本地markdown编辑器

原创

Hal白夜  已于 2022-04-06 23:13:52 修改  281  收藏 1

分类专栏: [教程知识](#) 文章标签: [typora markdown](#)

于 2021-01-03 23:38:43 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_38958597/article/details/112163092

版权



[教程知识](#) 专栏收录该内容

16 篇文章 1 订阅

订阅专栏

从0开始学习Typora—好用的本地markdown编辑器

在写文这方面, markdown绝对的多平台发布的利器。因为不同平台, 不同编辑器写出来的笔记, 格式是不一样的。这点我就深有体会, 之前在有道云写的笔记, 每次想发布到博客时, 都会因为懒而放弃。因为重新排一次格式真是太麻烦了, 非常的繁琐。万一有一个地方需要改格式, 还不得不把两个地方的格式都改了, 因为编辑器样式类型不同, 所以改的方式还不一样。。

当然不可否认, md的格式还是没有有道云直接编辑来的方便和好看, 但是md在跨平台发布的优势实在太大了, 最后我也屈服了, 并找到了这款比较好用的本地编辑器。

一、Markdown介绍

Markdown, 一种轻量级标记语言, 它允许人们使用易读易写的纯文本格式编写文档。——百度百科

注意, md不是编辑器, 它本质上是一门语言, 只是专精于文本格式的简单标记语言。就像html一样, 不是非要是java、c++那样的才算是语言的哦。

跨平台性

只要有过几次跨平台发布的经验, 就会明白, 一款能跨平台发布的编辑器有多好用。md最大的优势就是它所有的格式样式都是纯文本格式的。既然是纯文本, 只要迁移的平台支持这种语言, 就可以使用它。

原本我们在各种编辑器写文、调样式, 实际上样式都是由平台实现的, 不同的样式在它们的后台可能对应不同的语法。这种情况下, 换一个平台就相当于换了一种语言语法, 想要换平台发布自然只能重新按照另一个平台的语法去改样式。

而现在大部分平台都对md这种语法提供了支持, 这就意味着我们写好的md文章, 可以随意的复制粘贴到各个平台发布, 而无需担心样式问题了。

二、Markdown编辑器

可能有人会觉得, 既然md是一种语言而不是软件, 那为什么我们还需要md编辑器之类的呢? 既然是语言, 用记事本都是可以的吧?

实际上编辑器对于md就像IDEA对于java, 一款好的工具能让你的写作(开发)事半功倍。

写过CSDN的可能都知道，一般的md编辑器，都是两个页面的，左边用来编辑，右边用来预览。因为我们在将文字和代码结合为md语法时，如果没有预览功能，看起来会很难受，显得乱糟糟的，也难以判断整体的样式。所以预览是必要的。

但是现在还有更好用的编辑器，也就是本文的核心——Typora

三、Typora下载安装

Typora是一款支持实时预览的Markdown文本编辑器，它有OS X、Windows、Linus三个平台的版本，并且是免费的。

它的下载安装也是非常简便的

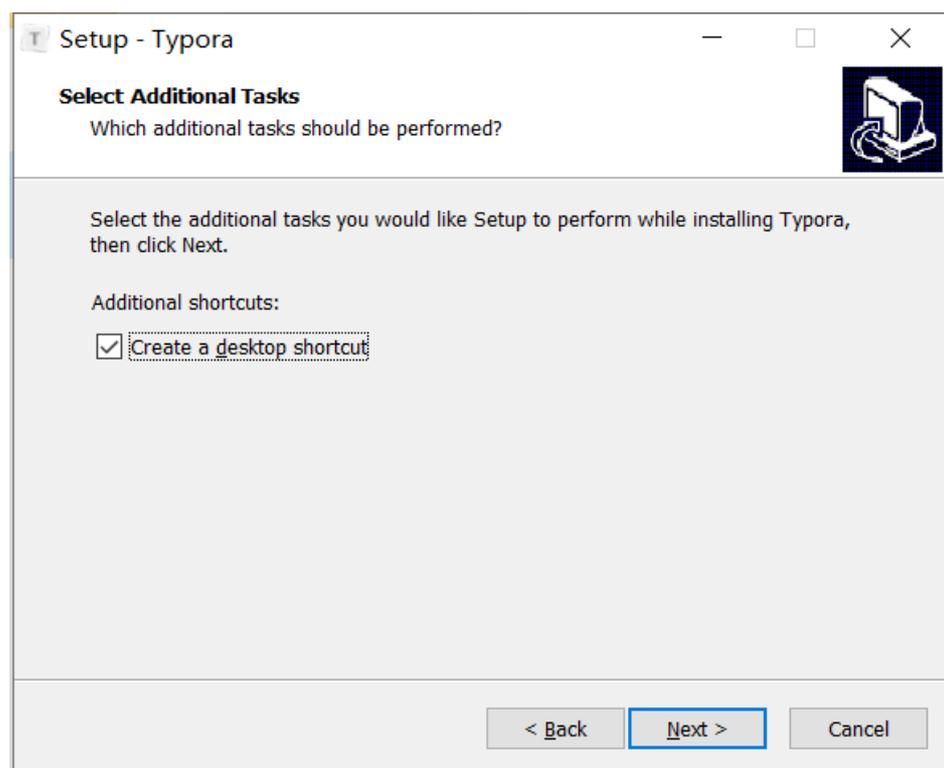
首先官方下载：<https://www.typora.io/#windows>

2022.4.6更新：

typora终于也要收费了，官网也被墙了打不开了。。。虽然提倡大家支持正版，但是也提供一下到目前为止仍然可以用的免费版：0.11.18版本，我已经搜集到资源了，大家直接下载安装覆盖就可以了：<https://www.aliyundrive.com/s/gBVYh7bEB2S>

选64位的，如果下载很慢可以用复制下载链接到迅雷下载（经验之谈）

然后安装，选择安装路径，勾选生成桌面快捷方式，点安装就好了



四、使用Typora

1.实时渲染

其他优势不说，光凭一个优势，就足以让它封神了，那就是：**实时渲染**

这个实时可不是左边编辑右边预览的那种“实时”，而是在我们使用md语法时，它会实时渲染我们的文本编辑框，隐藏md语法的代码，并将格式效果显现出来，达到看起来和有道云笔记一样的效果。

尽管格式效果和方便程度上，可能和真正的有道云那种编辑器还差了一丢丢。但是md语言的跨平台性带来的优势实在是太大了，足以忽略这微不足道的缺陷。（另外，有道云笔记也是支持md文章的，但是很可惜也是左右两边的编辑器）。

如图，是我正在进行md语法码字的效果：

一、Markdown介绍

Markdown，一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。——百度百科

注意，md不是编辑器，它本质上是一门语言，只是专精于文本格式的简单标记语言。就像html一样，不是非要是java、c++那样的才算是语言的哦。

跨平台性

只要有过几次跨平台发布的经验，就会明白，一款能跨平台发布的编辑器有多好用。md最大的优势就是它所有的格式样式都是纯文本格式的。既然是纯文本，只要迁移的平台支持这种语言，就可以使用它。

原本我们在各种编辑器写文、调样式，实际上样式都是由平台实现的，不同的样式在它们的后台可能对应不同的语法。这种情况下，换一个平台就相当于换了一种语言语法，想要换平台发布自然只能重新按照另一个平台的语法去改样式。

而现在大部分平台都对md这种语法提供了支持，这就意味着我们写好的md文章，可以随意的复制粘贴到各个平台发布，而无需担心样式问题了。

二、Markdown编辑器

可能有人会觉得，既然md是一种语言而不是软件，那为什么我们还需要md编辑器之类的呢？既然是语言，用记事本都是可以的吧？

2.简洁

简短的两个字，却是大部分软件都做不到的。

打开Typora的编辑器页面，可以看到，整个编辑器只有一个导航栏，和白色的页面（也可以调成黑色），除此之外再无它物。这个简洁度已经和sublime差不多了。

T 从0开始学习Typora---好用的本地markdown编辑器.md - Typora

文件(E) 编辑(E) 段落(P) 格式(O) 视图(V) 主题(T) 帮助(H)

文件

大纲

从0开始学习Typora---好用的本地markdown编辑器

- 一、Markdown介绍
- 二、Markdown编辑器
- 三、Typora下载安装
- 四、使用Typora

四、使用Typora

1.实时渲染

其他优势不说，光凭一个优势，就

3.视图功能

通过开启视图功能的侧边栏，可以使用文件树功能。我们可以像有道云一样的笔记格式管理各个文章，甚至可以在这里快速的新建一篇文章或目录。



视图里的另一个功能是大纲，其实就是目录，通过我们使用各级标题读取目录，完整展现出来，还可以通过点击快速的定位到具体一节的内容。



我们还可以在视图里切换为**源代码格式**，查看标准的md源格式，以便解决一些格式问题。

另外**专注模式**和**打字机模式**就可以根据需要使用了，专注模式会在你编辑某行时将其他行变成灰色，让你专注于当前行。打字机模式则是让的光标始终自动的处于屏幕中央了。这两个模式甚至可以同时开启。

4.偏好设置

通过**文件/偏好设置**，我们可以按自己的操作习惯定制Typora，不要小看这个偏好设置，它可以操作的空间可就多了。

节选一些我改过的配置：

①启动选项

可以设置启动选项为打开指定目录，或打开上一次的**文件/目录**，我们可以指定一个目录专门用于存放笔记。

②自动保存

勾选自动保存，在我们切换到其他窗口时，它就会自动帮你保存当前的更改了。

③窗口样式

我比较习惯经典，虽然一体化看起来更具高级感。

④字体大小

编辑器的字太小或太大？在这里自定义吧。

⑤状态栏

可以在文章下方新增一行状态栏，主要是字数统计等等

⑥侧边栏大纲允许折叠

可以勾选，让目录结构可以折叠或展开（原本是固定全展开的）

还有一些其他配置，就不一一列举了。新版本还有语法检查的功能（虽然似乎只能检查英语），这些留给自己去探索吧，下面要说一个更重要的使用方式。

五、关于图片

到此，我们似乎克服了种种难关，能轻松的一次编辑，到处发布了，但是这里仍然有个大问题，就是图片。

1.图片去哪了

要知道，我们之前在有道云、CSDN、简书甚至QQ空间这些不同平台写文章时，可能最常用的就是在外部截图，然后粘贴进文本编辑器中。实际上这种方式是将图片上传到了该平台，然后用它们返回的http链接渲染图片，也就是这个图片实际上是存储在该平台中，任由他们宰割。

我们平时其实没有意识到这点，以为插入了图片，图片就上传上去嵌入在文章中中了，没有考虑图片具体的情况。所以偶尔我们会看到有人发布的文章，文本没事，图片却都“挂了”。又或者这些图片变成了统一的一张图，内容是仅限于**百度app**查看。

这都是平台搞得鬼，我们控制不了我们的图片。虽然有点夸张，但是兴许哪天他们想节省内存，删了一些图片，这些图片就有你文章中的。最糟糕的是你还没有备份，无了就无了。

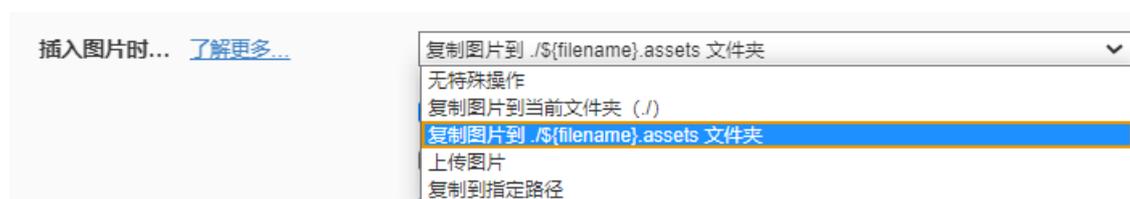
2.markdown中的图片

在markdown中，图片更像是一个地址，它不像word将图片真正的插入到了文档中混合起来，而是用一个引用来表示图片，这个引用就是图片的地址，不管是网络地址还是本地路径，它都会按照地址加载解析这个图片。

这样的话，图片就真正的是由我们自己掌控的了，该如何存储，存储到本地还是网络，都由我们自己控制。

3.typora配置图片存储

在偏好设置中，我们可以方便的配置图片的操作，在插入图片时，我们可以选择：



①无特殊操作

也就啥都不干。

②复制图片到当前文件夹(/)

将插入的图片，复制一份到当前文档所在的目录下。

③复制图片到./\$(filename).assets文件夹

也就是在当前目录下创建一个和文档同名但是后缀是assets的文件夹，把图片复制一份进去。

④上传图片

即配置图片上传到远端但是前提是配置好图片上传的相关功能。

⑤复制到指定路径

即使自定义复制图片到哪里，可以用一些特殊符号标记。

我们在下一节配置图片上传，这一小节我们配置本地存储，通过如上的选项，我们可以按照自己的偏好配置图片存储。可以为当前目录专门创建一个image目录保存图片：选择选项⑤，然后路径设置为：**./images**。

不过我喜欢为每篇文章单独设置一个存储目录，所以我的配置如下：



我们也可以单独在图片上，右键进行存储或上传的操作：



4.typora整合PicGo进行图片上传

这里虽然我选择默认是图片保存到本地，但是最终还是要将图片上传到存储库，才可能实现一篇文章，到处发布的目标。对我而言，保存到本地是备份，然后可以手动操作每张图片上传到远端。

还是在偏好设置里：



① 下载PicGO

首先我们得下载一个PicGO，图片上传需要通过它。而且由于我们对它不熟，所以要下载的是可视化的APP，而不是命令行。刚好typora甚至还帮我们提供了它的下载地址：<https://github.com/Molunerfinn/PicGo/releases/tag/v2.3.0-beta.4>

这里我下的是尝鲜版，你们可以考虑下个稳定版，windows系统，所以选择exe后缀的下载：

Others

PicGo-Core 升级到 v1.4.14

Assets 12

latest-linux.yml	381 Bytes
latest-mac.yml	531 Bytes
latest.yml	359 Bytes
PicGo-2.3.0-beta.4-mac.zip	62.5 MB
PicGo-2.3.0-beta.4.ApplImage	65.6 MB
PicGo-2.3.0-beta.4.dmg	64.7 MB
PicGo-2.3.0-beta.4.dmg.blockmap	69.8 KB
PicGo-Setup-2.3.0-beta.4.exe	44.7 MB
PicGo-Setup-2.3.0-beta.4.exe.blockmap	48.4 KB
picgo_2.3.0-beta.4_amd64.snap	55.2 MB
Source code (zip)	
Source code (tar.gz)	

② 安装 PicGO

下载好了直接打开，选好安装目录点安装即可，打开后：



③ Typora配置 PicGO

如图，typora的偏好设置里，配置好PicGO的安装位置：



④配置阿里云OSS

虽然typora和PicGO接好了，但是PicGO还没配置好我们要上传图片的地方，即图床。



这里我个人推荐阿里云的，阿里就不多说了肯定靠谱些，而且我记得应该是不贵的。由于我之前已经申请过阿里云的对象存储OSS了，所以可惜不能带领大家一步步来了，只能给一些配置信息。

阿里云创建子用户

由于我们只是给图床使用的，所以可以创建一个子用户专门访问这些，提高安全性，登录阿里云，点击头像下的访问控制。



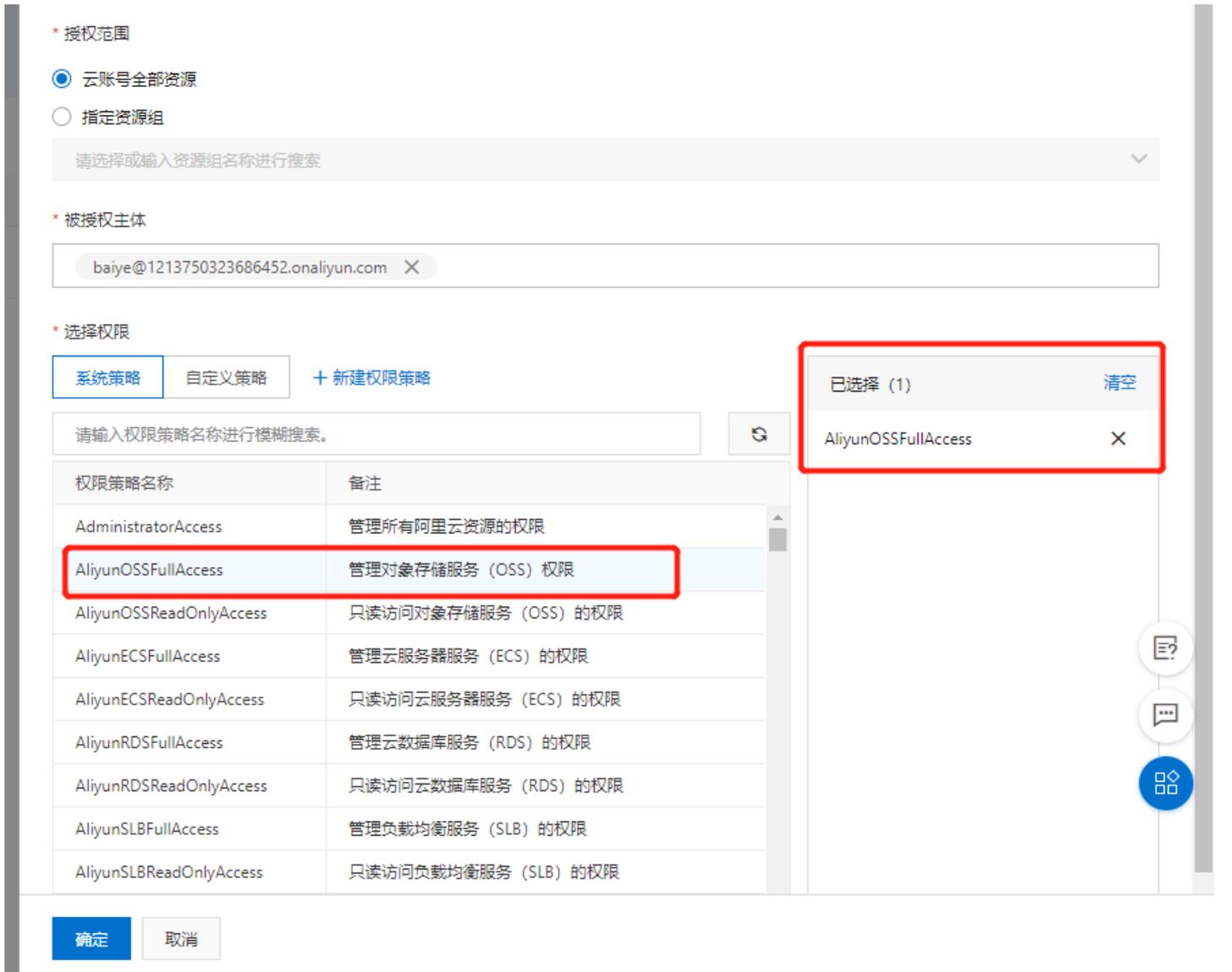
咱随便创一个，但是编程访问记得勾选



创建好的时候，我们可以看到AccessKey ID和AccessKey Secret，这个得先复制出来，因为它只能看一次，如果没复制到，那就只有创建新的AccessKey



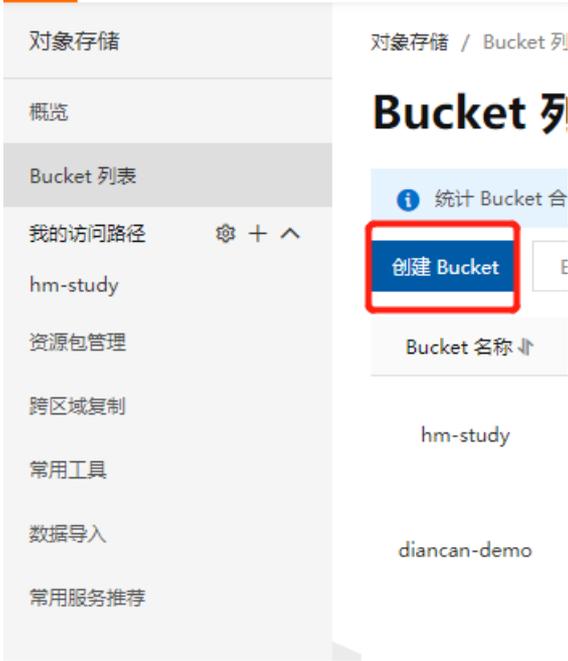
然后授予权限，只授予OSS的全部权限就够了



阿里云 OSS 创建 Bucket

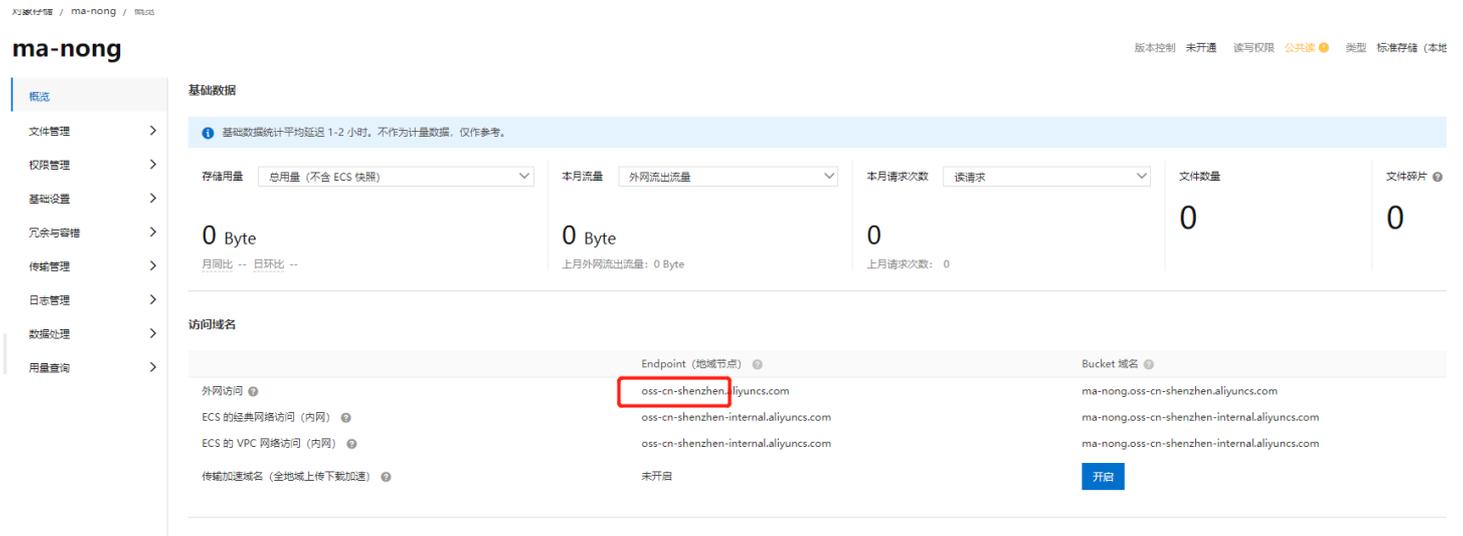
虽然我之前已经有了OSS，但是针对不同功能，还是得重新创建一个Bucket存储桶

回到OSS控制台，找到点击创建Bucket



创建桶的参数就按照自己的喜好进行配置吧，但是有个项一定要配置为公共读

创建好后如下



回到PicGO配置，前两项就是一开始创建的子用户的AccessKey ID和AccessKey Secret，存储空间就是Bucket桶名，存储区域可以参照上图访问域名，如下即可：



然后可以在PicGO中上传一张试试，成功的话，typora也应该没啥问题，右键文章内的图片上传，会发现它上传成功了，并且还会自动将图片地址替换为OSS返回的图片链接：

名，存储区域可以参照上图访问域名，如下即可：

 ![image-20210103225242769] (https://ma-nong.oss-cn-shenzhen.aliyuncs.com/typoraImages/image-20210103225242769.png)

六、总结

至此，我们可以好好的用Typora写文，无需担心跨平台的问题了，以后如果我发现了什么Typora更方便的功能，也会分享到这里。

参考资料：

Typora+PicGo+阿里云OSS实现图片上传功能

https://blog.csdn.net/weixin_44497705/article/details/108996963



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)