

---

# MarkdownPicPicker Documentation

*Release 1.1.0*

**kingname**

**Jan 05, 2019**



---

contents:

---

<b>1</b>	<b>What can it do</b>	<b>3</b>
<b>2</b>	<b>Windows</b>	<b>5</b>
<b>3</b>	<b>macOS</b>	<b>7</b>
<b>4</b>	<b>Config</b>	<b>9</b>
4.1	Use Qiniu web host . . . . .	9
4.2	Only copy url . . . . .	11
<b>5</b>	<b>Develop your own uploader</b>	<b>13</b>
5.1	Rules . . . . .	13
<b>6</b>	<b>Attention</b>	<b>15</b>
6.1	Fix Pillow bug . . . . .	15
<b>7</b>	<b>TODO</b>	<b>17</b>
<b>8</b>	<b>Thanks to</b>	<b>19</b>



MarkdownPicPicker is an assistant which can help you add picture in Markdown. It will upload the image in your clipboard to web picture host and copy the Markdown-format link() to your clipboard or pasteboard. Now it supports Windows and Mac OS.



# CHAPTER 1

---

## What can it do

---

Version 1.1.0 can do

1. Picture host support Qiniu and SM.MS.
2. You can use it as soon as you download it without any config.
3. Through config, you can add your own picture host uploader.
4. Picture will be saved to local first.
5. Copy the Markdown-format link to clipboard or pasteboard.
6. Easy to add your own uploader.





## CHAPTER 2

---

### Windows

---

1. Download MarkdownPicPicker at [https://github.com/kingname/MarkdownPicPicker/releases/download/v1.0.0/MarkdownPicPicker\\_v1.0.0.zip](https://github.com/kingname/MarkdownPicPicker/releases/download/v1.0.0/MarkdownPicPicker_v1.0.0.zip)
2. Copy image into clipboard
3. Run MarkdownPicPicker.exe



## CHAPTER 3

---

macOS

---

1. Install pngpaste: `brew install pngpaste`
2. Download and unzip MarkdownPicPicker at: [https://github.com/kingname/MarkdownPicPicker/releases/download/v1.0.0/macOS\\_MarkdownPicPicker.zip](https://github.com/kingname/MarkdownPicPicker/releases/download/v1.0.0/macOS_MarkdownPicPicker.zip)
3. Copy image into pasteboard
4. Run MarkdownPicPicker.app



The default web image host is SM.MS, however, you can use Qiniu by just create a config file.

### 4.1 Use Qiniu web host

If there is not a config, MarkdownPicpicker will use SM.SM as the default picture host. But this website may break-down in the future and your data maybe unsafe. If you want to use Qiniu web host, please create a folder called config and write the config.ini like follow:

Please Create/Edit config/config.ini, every item in it means:

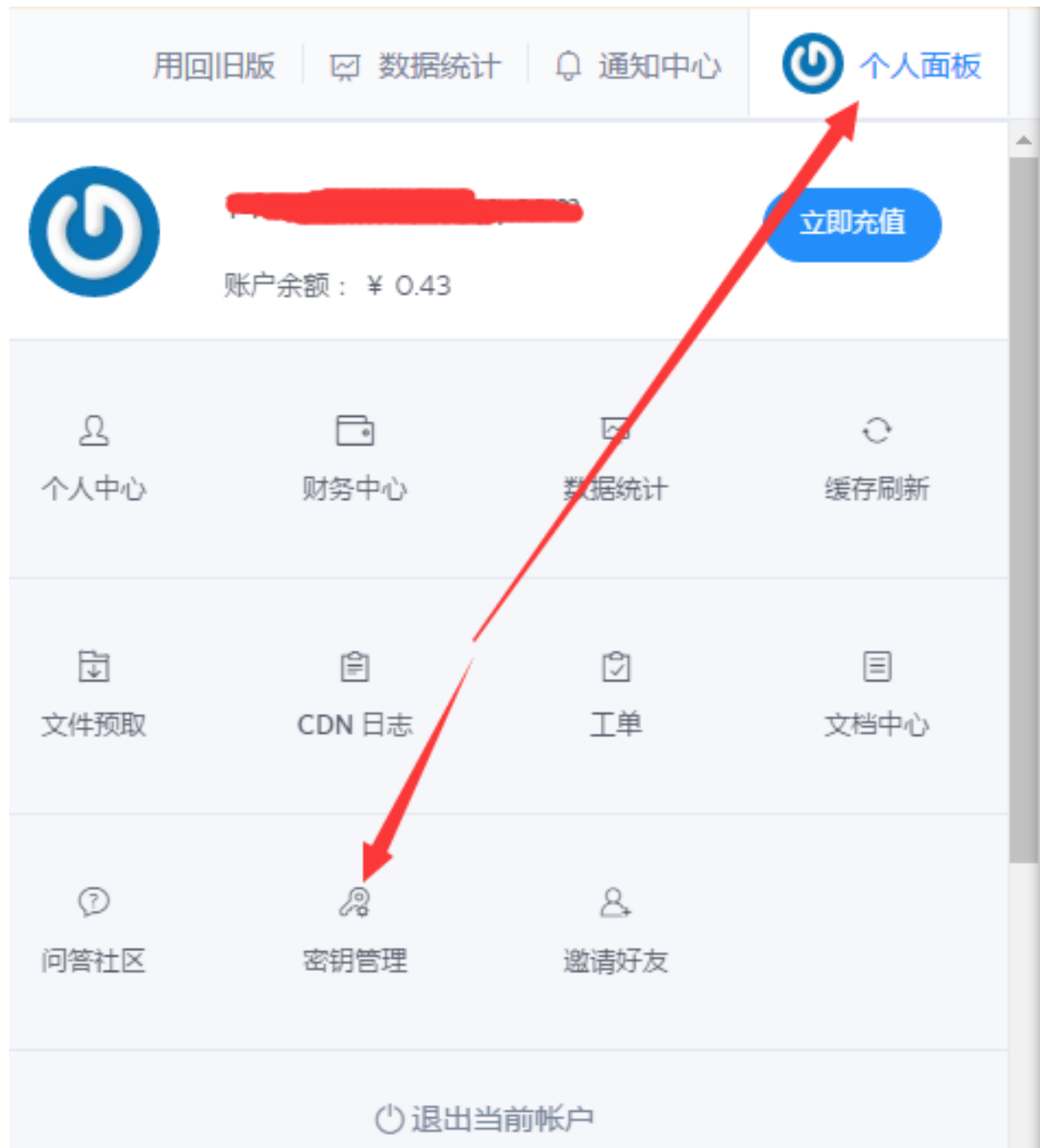
```
[basic]
picture_folder = pic # necessary, the local folder to save a copy of image
picture_suffix = png #necessary, the format of your image, 'png' only

picture_host = QiniuUploader

[QiniuUploader]
access_key = Q6sS422005AasdfasfafgfCcCpF36tqvYQ75Zvzw
secret_key = 6QtAqqTxoSadffadfgewehxPLX2CCmoOaB2aLObM
container_name = picturebed
url = http://7sbpmp.com1.z0.glb.clouddn.com/{}
```

Notice: please remove the comments in config.ini.

In the config, access\_key and secret\_key can be found in Qiniu's controlpanel

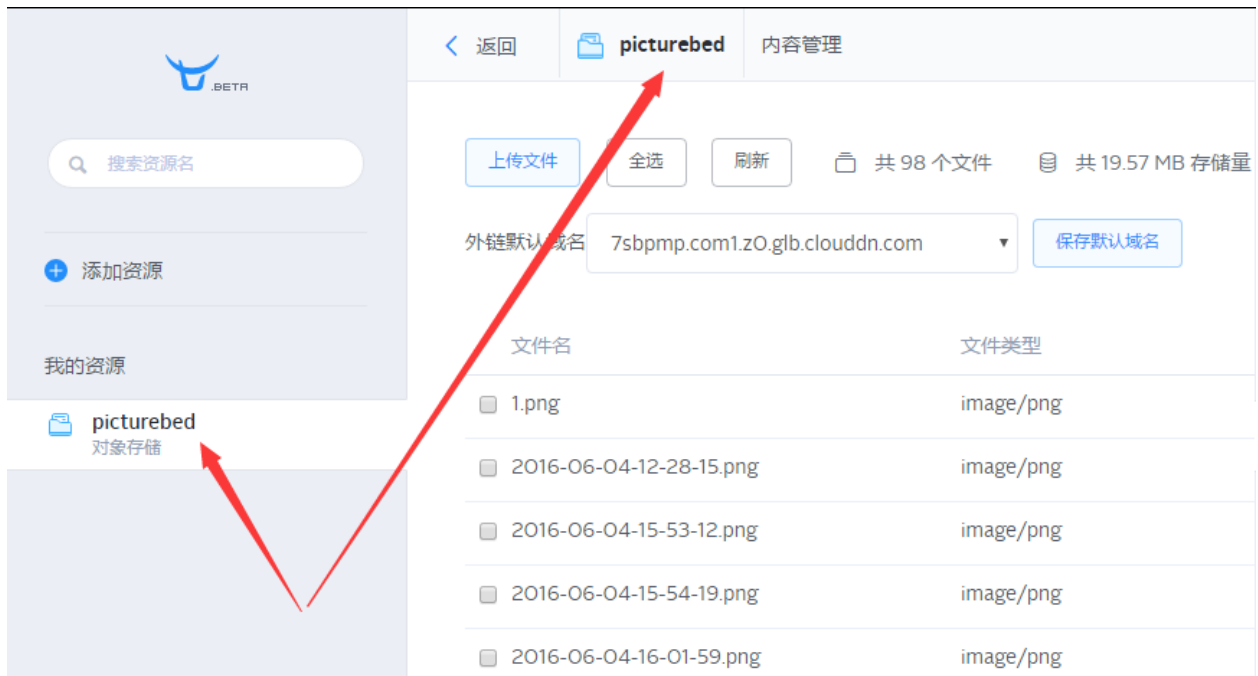


## 密钥管理

一个帐号最多拥有两对密钥(Access/Secret Key)；更换密钥时，请创建第二个密钥；删除密钥前须停用；出于安全考虑，建议您周期性地更换密钥。您可以查看更多 [安全使用密钥建议](#)。

创建时间	AccessKey/SecretKey	状态	操作
2016-04-12	AK: <input type="text" value="Q6sS422O05AwYD5aVqM3FqCcCpF36tqvYQ75Zvzw"/> SK: <input type="text" value="....."/> <input type="button" value="显示"/>	使用中	<input type="button" value="停用"/>

container\_name means



## 4.2 Only copy url

If you want to copy the url of image only instead of ![url] you can use parameter: `-linkonly`:

```
markdownpicpicker.exe -linkonly
```

or:

```
python MarkdownPicPicker.py -linkonly
```

I recommend you to use AutoHotKey to launch MarkdownPicPicker. In this way, the time of add an image can be less than 2 seconds. My AutoHotKey config is

```
#c::Run, G:\github\MarkdownPicPicker\markdownpicpicker.exe -linkonly  
Return  
!c::Run, G:\github\MarkdownPicPicker\markdownpicpicker.exe  
Return
```

What you need to do is:

- Copy image.
- Press the short key.
- Paste the Markdown format url in your article.



---

## Develop your own uploader

---

Now MarkdownPicPicker only support [Qiniu](#) and [SM.MS](#). For different countries, there must be more brilliant picture hosts. I hope you can help me add more picture host uploaders, and submit PR to me. It is really very easy but helpful

### 5.1 Rules

- Put your uploader in the `uploader` folder, file name is unlimited, for example: `ExampleUploader.py`
- The class name **must** be `Uploader`
- If the picture host need token or other parameters, please add a parameter called `config_info` in `__init__`, every parameters in your config will be sent to here in the type of `dict`

```
def __init__(self, config_info=None):
    self.token = config_info['token']
    self.username = config_info['username']
    self.password = config_info['password']
    self.url = config_info['url']
```

- There must be a method called `upload`, there parameter `picture_path` is the local path of the file:

```
def upload(self, picture_path, link_only=False)
```

- Edit `config/config.ini`, add the parameters of your own uploader, the Section is your uploader's filename without `.py`, and the `picture_host` in `[basic]` is the Section, for example:

```
[basic]
picture_folder = pic
picture_suffix = png
picture_host = ExampleUploader

[ExampleUploader]
token = 123456
```

(continues on next page)

(continued from previous page)

```
username = xxx  
password = yy  
url = http://xxx.xxx/upload
```

---

## Attention

---

If you want to use the source code to run MarkdownPicPicker in Windows, please pay attention to follow problem:

### 6.1 Fix Pillow bug

Windows version use Pillow's method `ImageGrab.grabclipboard()` to get the image in clipboard, but there is a official bug, this bug will cause this error:

```
Unsupported BMP bitfields layout
```

This bug begin from Pillow 2.8.0 and it is not fixed until Pillow 3.2.0. Here I will tell you how to fix this bug: Please open `<path to your Python>Libsite-packagesPILBmpImagePlugin.py` there should be these codes:

```
if file_info['bits'] in SUPPORTED:
    if file_info['bits'] == 32 and file_info['rgba_mask'] in SUPPORTED[file_info['bits']
↪']:
        raw_mode = MASK_MODES[(file_info['bits'], file_info['rgba_mask'])]
        self.mode = "RGBA" if raw_mode in ("BGRA",) else self.mode
    elif file_info['bits'] in (24, 16) and file_info['rgb_mask'] in SUPPORTED[file_
↪info['bits']]:
        raw_mode = MASK_MODES[(file_info['bits'], file_info['rgb_mask'])]
    else:
        raise IOError("Unsupported BMP bitfields layout")
else:
    raise IOError("Unsupported BMP bitfields layout")
```

change them to:

```
if file_info['bits'] in SUPPORTED:
    if file_info['bits'] == 32 and file_info['rgba_mask'] in SUPPORTED[file_info['bits']
↪']:
        raw_mode = MASK_MODES[(file_info['bits'], file_info['rgba_mask'])]
        self.mode = "RGBA" if raw_mode in ("BGRA",) else self.mode
```

(continues on next page)

(continued from previous page)

```
    elif file_info['bits'] in (24, 16) and file_info['rgb_mask'] in SUPPORTED[file_
↪info['bits']]:
        raw_mode = MASK_MODES[(file_info['bits'], file_info['rgb_mask'])]
        '''the code to add begin'''
    elif file_info['bits'] == 32 and file_info['rgb_mask'] == (0xff0000, 0xff00, ↪
↪0xff):
        pass
        '''the code to add end'''
    else:
        raise IOError("Unsupported BMP bitfields layout")
else:
    raise IOError("Unsupported BMP bitfields layout")
```

## CHAPTER 7

---

TODO

---

- More screenshots
- Save image from pasteboard without pngpaste
- More Picture web host
- Hide the terminal windows
- Linux Usable



## CHAPTER 8

---

Thanks to

---

laixintao