任务：爬取以太坊市值前1000 project的所有token图片

任务步骤：

1. 计算市值并提取前1000project（截至2022-05-18）的基本信息（Project\_ID、address、name）✔

已保存至 top1000\_nft.csv

1. 获取每个project的meatadata保存链接格式 ✔

通过etherscan readcontract页面（https://etherscan.io/token/address#readContract）的abi接口，获取每个project的token id 为1的token meatadata保存链接，其中出现多种异常情况，添加URI\_status变量记录，取值及含义如下：

|  |  |  |
| --- | --- | --- |
| 取值 | 含义 | 频率 |
| 1 | 爬取完毕 | 766 |
| -1 | etherscan未记录contract | 76 |
| -2 | 该contract不含tokenURI方法 | 9 |
| -3 | 爬取过程出现超时或其他异常 | 104 |
| -4 | 该project无id为1的token | 45 |

爬取内容及异常检验已保存至URI\_df.csv

爬虫代码已附至retrieve\_uri.py

1. 获取每个project的image保存链接格式 (待完成)

根据前文所获取的tokenURI,来定位image\_url，并做出分类，大致来说应会有以下类型：

ipfs存储(boredapeyachtclub 0xbc4ca0eda7647a8ab7c2061c2e118a18a936f13d)；

中心化服务器存储(meebits 0x7bd29408f11d2bfc23c34f18275bbf23bb716bc7)；

链上编码svg存储(terraforms 0x4e1f41613c9084fdb9e34e11fae9412427480e56)；

不存储(cryptopunks 0xb47e3cd837ddf8e4c57f05d70ab865de6e193bbb)

类型后为代表项目及address，该分类参考文章(https://www.tuoniaox.com/news/p-348076.html)

该阶段产出：

对每个URI\_status为1的project，找出他们的image\_url，同时根据上述类别进行分类以便后续处理（若有更多类别，请注释）

由于后续将会对URI异常的project进行处理后更新URI\_df.csv，故需注意保证流程的可复现性

1. 针对不同存储方式的project进行分别处理

目前已完成对IPFS存储的图像爬取代码

待第3阶段完成后，再一起商讨对其他类型的处理方式