

# aiohttp多任务异步协程

aiohttp是python的一个非常优秀的第三方异步http请求库. 我们可以用aiohttp来编写异步爬虫(协程)

安装:

```
1 pip install aiohttp
```

实例代码:

```
1 import aiohttp
2 import asyncio
3 import time
4 import requests
5
6 # 异步下载
7 async def aiodownload(url, session):
8     name = url.split("/")[-1]
```

```
9      # 发送请求，这里和requests.get()几乎没区别，除了代理换成了proxy
10      async with session.get(url) as resp:
11          # 读取数据。如果想要读取源代码。直接resp.text()即可。比原来多了个()
12          content = await resp.content.read()
13          # 写入文件，有兴趣可以参考aiofiles，我这里根本不需要。
14          with open(name, mode="wb") as f:
15              f.write(content)
16
17
18  async def main():
19      # 创建session对象 -> 相当于requests对象
20      async with aiohttp.ClientSession() as session:
21          # 添加下载任务
22          tasks =
23          [asyncio.create_task(aiodownload(url, session))
24           for url in urls]
25
26          # 等待所有任务下载完成
27          await asyncio.wait(tasks)
28
29  # 同步方式下载图片
30  def download(url):
```

```
29     name = url.split("/")[-1]
30     resp = requests.get(url)
31     content = resp.content
32     with open(name, mode="wb") as f:
33         f.write(content)
34
35 # 我故意弄了一堆url做测试
36 urls = [
37     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/26b7e178e987be6d914bf8d
    1af120890.jpg",
38     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/191468637cab2f0206f7d1d
    9b175ac81.jpg",
39     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62
    0729f367e.jpg",
40     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62
    0729f367e.jpg",
41     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62
    0729f367e.jpg",
42     "http://kr.shanghai-
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62
    0729f367e.jpg",
```





```
59     "http://kr.shanghai-  
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62  
    0729f367e.jpg",  
60     "http://kr.shanghai-  
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62  
    0729f367e.jpg",  
61     "http://kr.shanghai-  
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62  
    0729f367e.jpg",  
62     "http://kr.shanghai-  
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62  
    0729f367e.jpg",  
63     "http://kr.shanghai-  
    jiuxin.com/file/2020/1031/563337d07af599a9ea64e62  
    0729f367e.jpg",  
64 ]  
65  
66 if __name__ == '__main__':  
67     t2 = time.time()  
68     for url in urls:  
69         download(url)  
70     print(time.time() - t2)  
71  
72     t1 = time.time()  
73     # 异步爬虫  
74     asyncio.run(main())
```

```
75     print(time.time() - t1)
```

从最终运行的结果中能非常直观的看到用异步IO完成爬虫的效率明显高了很多