

owllook在线小说搜索引擎使用指南

原创 howie6879 老胡的储物柜 2018-12-29

owllook是一个在线小说搜索引擎，其目的是让阅读更简单、优雅，让每位读者都有舒适的阅读体验



owllook是我大学时期编写的一个Python开源项目，至今维护将近两年了，以前关于owllook介绍的原文章已经年久失修，再加上一些github用户不知道怎么安装使用，因此我将文章重新编辑一番，以作使用指南

## 介绍

owllook是一个基于其他搜索引擎构建的垂直小说搜索引擎，提供搜书、阅读、收藏、追更、推荐等功能

- owllook 演示网址: <https://www.owllook.net/>
- github开源地址: <https://github.com/howie6879/owllook>

owllook不保存数据，所有信息全部利用爬虫技术基于第三方网站搜索发现，再经过在线解析展示给用户，这样对使用者的好处在于：

- 多网站聚合，可选择性高
- 优质网站来源提供，可靠性强
- 页面解析会经过过滤，安全性高
- 自动去广告，可读性强
- 定时更新，自动追更
- 发现同类书友
- .....

owllook基于python3.6，后端采用Sanic（对这个web框架感兴趣的朋友可在公众号后台回复sanic，看看我写的Sanic教程），前端使用了 bootstrap 和 mdui，数据库方面则用了 MongoDB 储存用户使用过程中的产生的基本信息，诸如注册信息、搜索小说信息、收藏小说数据等，对于某些必要的缓存，则利用 Redis 进行缓存处理，如小说缓存、session缓存，注意，对于限制数据：都将在24小时删除

对于不同网站的小说，页面规则都不尽相同，我希望能够在代码解析后再统一展示出来，这样方便且美观，而不是仅仅跳转到对应网站就完事，清新简洁的阅读体验才是最重要的

目前采用的是直接在搜索引擎上进行结果检索，我尽量写少量的规则来完成解析，具体见项目里面的规则定义，遇到自己喜欢的小说网站，你也可以自己添加解析，owllook目前解析了超过 200 + 网站，追更网站解析了 50 +

有一些地方需要用到爬虫，比如说排行榜，一些书籍信息等，我不想动用重量级爬虫框架来写，于是我在owllook里面编写了一个很轻量的爬虫框架来做这件事，见 ruia 异步爬虫框架

目前实现功能如下：

- 多搜索源
- 丰富的解析源
- 界面统一解析
- 完善的阅读体验
- 搜索记录
- 缓存
- 书架
- 书签
- 登录
- 初步兼容手机
- 注册（开放注册）
- 上次阅读记录
- 最新章节
- 书友推荐（简单地基于相似度进行推荐）
- 目录获取
- 翻页
- 搜索排行
- 章节异步加载 感谢@[mscststs](<https://github.com/mscststs>)
- 排行榜 - 起点+owllook
- 自带爬虫框架，统一爬虫规范，对爬虫感兴趣的可以看看 - [ruia](<https://github.com/howie6879/ruia>)

## 安装

接下来的安装步骤，默认以 Centos 7为主，默认在 root用户下操作，首先Python版本，最好是Python3 . 6，然后项目的数据库需要 MongoDB Redis

请先安装：

```
yum -y install bzip2 wget vim git
```

## 环境搭建

owllook的安装很简单，但是对于一些Python不熟悉或者新手来说确实有点困难，下面详细介绍了安装流程

## 数据库

owllook需要你在服务器上安装 MongoDB 和 Redis，你不必管理数据库的创建，用这个的好处就在于可以自动生成

安装MongoDB：

```
vim /etc/yum.repos.d/mongodb-org-3.6.repo
```

输入：

```
[mongodb-org-3.6]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.6/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.6.asc
```

可以看到 repolist 已经存在：



```
# 安装
sudo yum install mongodb-org
# 启动
sudo systemctl start mongod
# 查看 MongoDB 后台进程
ps -aux |grep mongod
# 或者
sudo systemctl status mongod
```

可以看到MongoDB正在后台运行：

```
root@kali:~# sudo systemctl status mongod
● mongod.service - SYSV: Mongo is a scalable, document-oriented database.
   Loaded: loaded (/etc/rc.d/init.d/mongod; bad; vendor preset: disabled)
   Active: active (running) since Sat 2018-12-29 00:47:34 EST; 48s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 27548 ExecStart=/etc/rc.d/init.d/mongod start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/mongod.service
           └─27551 /usr/bin/mongod -f /etc/mongod.conf

Dec 29 00:47:33 host.localdomain systemd[1]: Starting SYSV: Mongo is a scalable, document-oriented database....
Dec 29 00:47:33 host.localdomain runuser[27547]: pam_unix(runuser:session): session opened for user root(uid=0) by root(uid=0).
Dec 29 00:47:34 host.localdomain mongod[27548]: Starting mongod: [ OK ]
Dec 29 00:47:34 host.localdomain systemd[1]: Started SYSV: Mongo is a scalable, document-oriented database..
```

进入 MongoDB：

```
root@kali:~# mongo
MongoDB shell version v3.6.9
connecting to mongo://127.0.0.1:27027
Explicit session session ("id" : UUID("2a84bc76-4b37-4b32-9476-4954585ff4d0"))
MongoDB server version 3.6.9
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
http://docs.mongodb.org/
Questions? Try the support group
http://groups.google.com/group/mongodb-user

Server has startup warnings:
2018-12-29T00:47:34.823+0800 I STORAGE [initandlisten] == WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2018-12-29T00:47:34.823+0800 I STORAGE [initandlisten] ==          See http://wiki.mongodb.org/wiki/wiredtiger
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] == WARNING: Access control is not enabled for the database.
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==          Read and write access to data and configuration is unrestricted.
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] == WARNING: /sys/kernel/new/transparent_hugepage/enabled is 'always'.
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==          We suggest setting it to 'never'
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] == WARNING: soft limits too low. Limits set to 4096 processes, 65536 files. Number of processes
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==          number of files.
2018-12-29T00:47:34.929+0800 I CONTROL [initandlisten] ==
```

安装Redis：

```
yum install epel-release yum-utils
yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum-config-manager --enable remi

yum install redis

vim /etc/redis.conf
# 使 redis 能在后台运行
daemonize yes
```

启动 redis 服务

```
# 启动
systemctl restart redis
# 查看状态
systemctl status redis
# 查看端口
ss -an | grep 6379
```

```
root@kali:~# systemctl status redis
● redis.service - Redis persistent key-value database
   Loaded: loaded (/usr/lib/systemd/system/redis.service; disabled; vendor preset: disabled)
   Drop-In: /etc/systemd/system/redis.service.d
           └─limit.conf
   Active: active (running) since Sat 2018-12-29 01:28:39 EST; 7s ago
     Main PID: 4159 (redis-server)
       CGroup: /system.slice/redis.service
               └─4159 /usr/bin/redis-server 127.0.0.1:6379

Dec 29 01:28:39 host.localdomain systemd[1]: Starting Redis persistent key-value database....
Dec 29 01:28:39 host.localdomain systemd[1]: Started Redis persistent key-value database..
```

启动 redis 客户端

```
redis-cli
# ctrl + d 退出客户端
```

## 下载代码

请先安装git然后克隆代码

```
cd ~
# 下载代码
git clone https://github.com/howie6879/owllook
```

搭建Python3.6环境：

```
# 下载anaconda管理 Python 环境
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-5.3.0-Linux-x86_64.
# 安装
chmod -R a+x Anaconda3-5.3.0-Linux-x86_64.sh
./Anaconda3-5.3.0-Linux-x86_64.sh
# 刷新终端
# 创建 Python3.6 环境
conda create -n python36 python=3.6
```

配置项目：

```
cd ~/owllook
# 安装 pipenv
pip install pipenv
# --python 后面的路径是上面conda创建的路径地址
pipenv install --python ~/anaconda3/envs/python36/bin/python3.6
# 如果出错 继续往下执行
pipenv run pip install pip==18.0
pipenv install
```

如下表示安装成功：

```
root@kali:~/owllook# pipenv install
Installing dependencies from Pipfile.lock (39f5cc)...
52/52 -- 00:01:14
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

运行owllook：

```
cd owllook
python run.py
```

成功如下图：

```
root@kali:~/owllook# cd owllook/
root@kali:~/owllook# python run.py
[2018-12-29 01:45:46,130] 6114-INFO __init__::load_config():124: owllook 启动模式: DEV
[2018-12-29 01:45:46 -0500] [6113] [INFO] Starting unicorn 19.9.0
[2018-12-29 01:45:46 -0500] [6113] [INFO] Listening at: http://0.0.0.0:8001 [6113]
[2018-12-29 01:45:46 -0500] [6113] [INFO] Using worker: sanic.worker.GunicornWorker
[2018-12-29 01:45:46 -0500] [6129] [INFO] Booting worker with pid: 6129
[2018-12-29 01:45:46 -0500] [6130] [INFO] Booting worker with pid: 6130
2018-12-29 01:45:47,009 WARNING aiocache.log(13) | cPickle module not found, using pickle
2018-12-29 01:45:47,043 WARNING aiocache.log(13) | cPickle module not found, using pickle
2018-12-29 01:45:47,629 INFO root(24) | owllook 启动模式: DEV
2018-12-29 01:45:47,646 INFO root(24) | owllook 启动模式: DEV
2018-12-29 01:45:48,558 INFO root(27) | Starting aiocache
2018-12-29 01:45:48,561 INFO root(27) | Starting aiocache
2018-12-29 01:45:48,561 INFO root(38) | Starting redis pool
2018-12-29 01:45:48,558 INFO root(38) | Starting redis pool
```

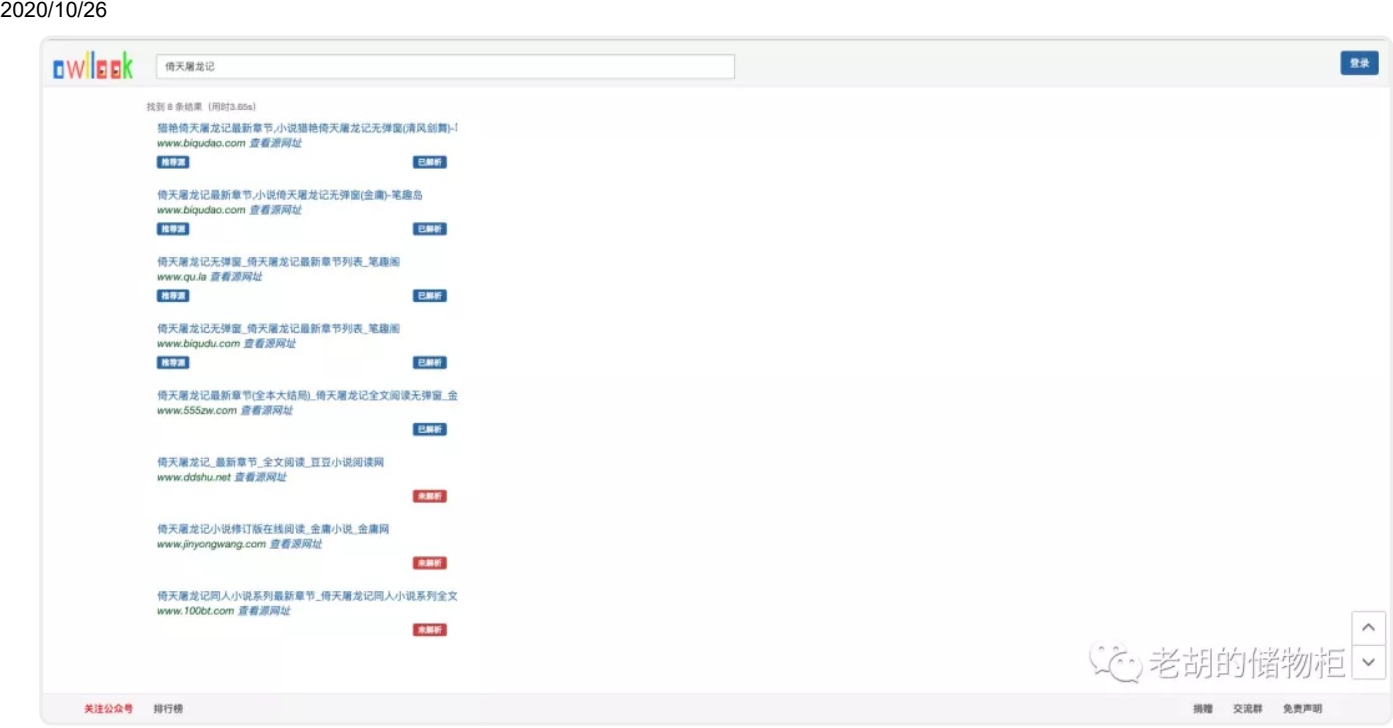
## 项目截图

首页

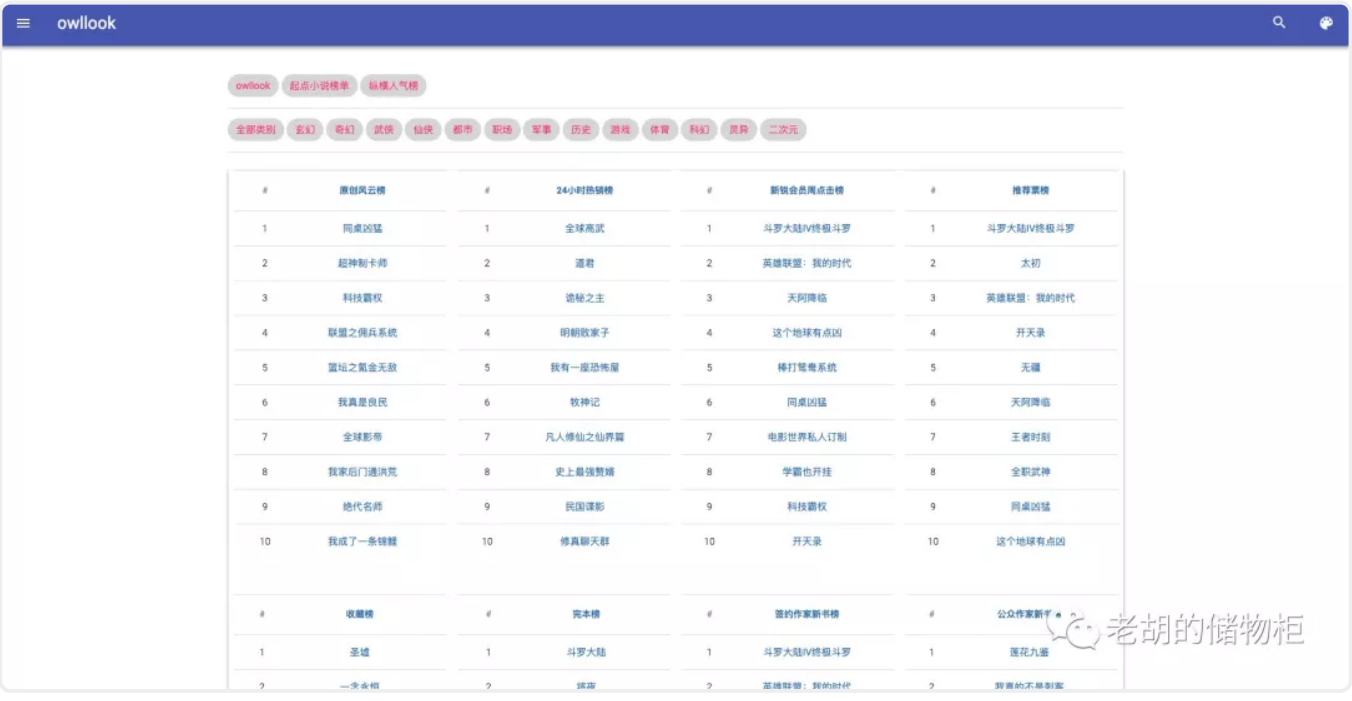


搜索





## 榜单



## 书架



## 解析



## 阅读页



有什么问题欢迎留言，喜欢的话请点赞转发

往期推荐：

Python之朴素贝叶斯对展会数据分类

如何用PEP 8编写优雅的Python代码

Sanic中文教程合集：免费下载

如何用Python创建一个简单的神经网络

Python之装饰器

