**软件使用说明书**

目录

[软件使用说明书 1](#_Toc23048)

[1介绍 2](#_Toc11194)

[2系统架构 2](#_Toc29888)

[3主要功能 3](#_Toc12297)

[4名词解释 3](#_Toc2319)

[5安装流程 4](#_Toc13198)

[5.1服务端的安装 4](#_Toc20685)

[5.1.1环境准备 4](#_Toc10887)

[5.1.2启动服务端 5](#_Toc12777)

[5.2客户端的安装 7](#_Toc9029)

[6使用流程 7](#_Toc10525)

[6.1服务端的使用 7](#_Toc30848)

[6.1.1登录 7](#_Toc23427)

[6.1.2图片管理 8](#_Toc9001)

[6.1.3页面管理 9](#_Toc3485)

[6.2客户端的使用 14](#_Toc4032)

[6.2.1修改配置文件 14](#_Toc29641)

[6.2.2启动客户端 15](#_Toc15235)

[7客户端的文件说明 15](#_Toc12099)

[7.1日志文件说明 15](#_Toc16368)

[7.2执行文件说明 15](#_Toc9232)

[7.3配置文件说明 16](#_Toc26475)

[config.json配置文件说明 16](#_Toc30615)

# 1介绍

众所周知,当今时代已经是一个大数据时代，我们无时无刻都在和数据打交道，我们既是数据的生产者，也是它的消费者。数据的采集及使用是促进大数据和人工智能这两个领域发展不可避免的条件。特别是在人工智能领域，数据的重要性更是不可言说。没有数据人工智能就只是“人工智障”，没有数据的人工智能的一切都是空谈。可以说，掌握数据就掌握了资源，而有效利用数据就是促进发展的密码。由于原始数据的信息量巨大，操作专业性较强，大部分人都面临着获取原始数据难，处理数据时间长，数据有效利用率低等这些问题，而这些都是阻碍发展的主要因素。面对大量化的信息，我们无法或难以从中快速地提取有效信息为己所用。因此我们团队依靠自身独特的设计方案和技术支持，通过“关键词匹配”和“以图搜图”的方式，能够让使用者有针对性地快速提取有效信息，这不仅增强了数据的使用力，提高了效率，更在很大程度上节约了人力物力，满足了大量的市场需求，有利于促进数据和人工智能领域的发展，实现了产品的社会价值。

# 2系统架构

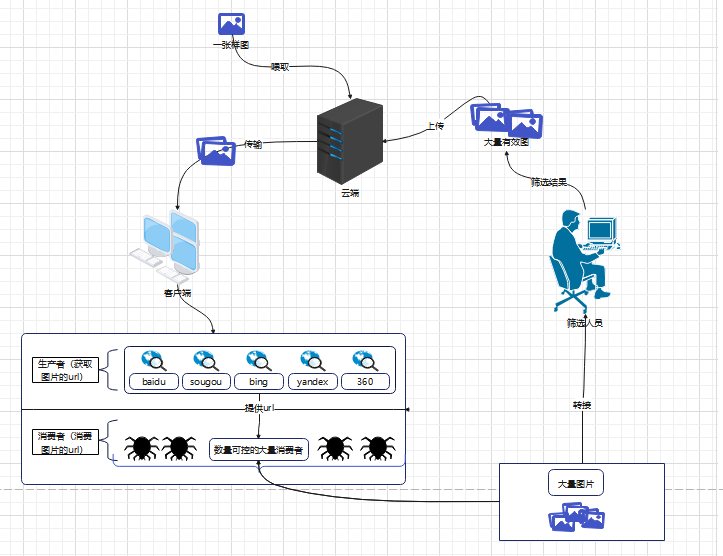
本软件采用云端分布式架构，主要由采集器客户端、采集器服务端、数据存储和分析系统组成。

1. 客户端：主要用于用户图片采集的控制和管理，包括采集器启停、采集规则和策略的配置、采集器节点的监控等功能。

（2）服务端：主要用于采集器节点的管理、数据处理和分发，包括采集器节点的动态调度、

数据去重和过滤、数据分析和存储等功能。

1. 数据存储和分析系统：主要用于存储采集器采集的图片数据和进行数据分析，包括数据清洗、数据建模、数据可视化等功能。



系统架构图

# 3主要功能

（1）支持多种图片资源的采集：包括但不限于静态图片、动态图片、SVG图片等。

（2）高并发采集：采集器采用多节点分布式架构，可实现高并发图片采集，提高采集效率。

（3）可定制化：采集器支持用户自定义采集规则和策略，可根据用户需求进行灵活配置。

（4）数据去重和过滤：采集器具有去重和过滤功能，可以避免重复采集和采集无效图片，提高采集效率和准确性。

（5）监控和报警：采集器具有实时监控和报警功能，可以及时发现和解决采集过程中的问题，保证采集器的稳定性和可靠性。

# 4名词解释

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **默认值** | **能否修改** |
| 根目录 | 项目的根目录 | /img-spider-server | 否 |
| host | 服务器的ip地址，本地访问使用默认值，其他请使用服务器的真实ip或域名替换。 | 127.0.0.1 | 是 |
| port | 服务端占用的端口 | 8000 | 否 |
| **img\_spider** | **图片爬虫，其包含了“以图搜图”和抽取页面上图片两个功能** | **无** | 否 |
| **keyword\_spider** | **关键字爬虫，通过给定的关键字，访问特定的API，获取相应的图片信息；img\_spider和keyword\_spider是整个客户端的核心，keyword\_spider为img\_spider提供“以图搜图”的原始图片，因此keyword\_spider可以看做是img\_spider的生产者之一。** | **无** | 否 |

# 5安装流程

**注意：在下面的安装步骤中，部分特定的命令只能在特定的目录下执行成功或有效，因此，在安装过程中请勿随机切换目录；同时，服务端的安装和客户端的安装是在不同设备上进行的，在启动客户端之前，应首先完成服务端的安装。**

## 5.1服务端的安装

### 5.1.1环境准备

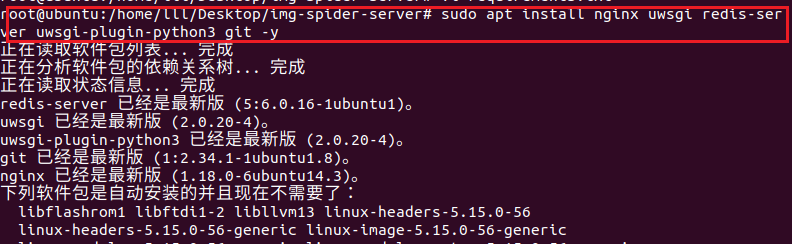
1.切换到root权限，命令如下：

|  |
| --- |
| $ su |



|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install nginx uwsgi redis uwsgi-plugin-python3 git -y |

2.安装依赖项，命令如下：



下载安装包，命令如下：

|  |
| --- |
| $ git clone <https://gitee.com/max-LLL/img-spider-server.git>  $ cd img-spider-server |

下载完成后，会在当前目录下看到一个img-spider-server文件夹，此为项目文件夹，在执行完上面的命令后，会切换到项目的根目录下。

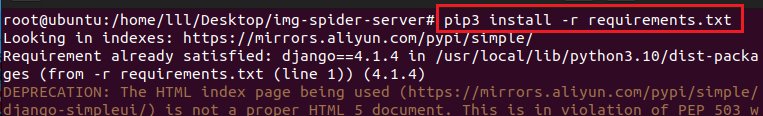
在项目的根目录下创建虚拟环境并激活，命令如下：

|  |
| --- |
| $ python3 -m venv venv  $ source venv/bin/activate |



5.在项目的根目录下安装第三方库，命令如下：

|  |
| --- |
| $ pip3 install -r requirements.txt |

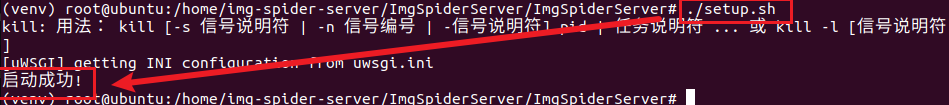


### 5.1.2启动服务端

**注意：在步骤5.1.1是服务端的初始化过程，仅需执行一遍即可，在后续需要对服务端进行重启或启动操作时，只需要执行步骤5.1.2即可。**

1.切换到ImgSpiderServer/ImgSpiderServer目录下并启动项目，命令如下：

|  |
| --- |
| $ cd /ImgSpiderServer/ImgSpiderServer  $ ./setup.sh |



使用如上命令启动项目后，如果看到终端输出”启动成功!”的字样，说明项目启动成功，能够打开浏览器，访问地址:http://{host}/img-spider-server/admin/

进行验证。（**注意：如果在本地访问，host的值可以为127.0.0.1或服务器的真实地址，其他请使用服务器的真实地址**）



后台登录界面

打开后会看到登录界面，输入用户名和密码即可进行登录（默认用户名：admin，默认密码：admin）；登录成功后会看到如下界面：



后台界面

至此，服务端的安装就完成。

## 5.2客户端的安装

注意：目前客户端仅支持windows10及以上的系统。

1. 下载安装包

|  |
| --- |
| 下载地址：https://gitee.com/max-LLL/img-spider-exe/repository/archive/master.zip |

1. 解压文件

将下载好的压缩包进行解压，解压后进入到/img-spider-exe-master/img-spider-exe-master目录下可以看到如下内容，其中main.exe为客户端的执行文件，config.json为客户端的配置文件，chromedriver.exe为脚本文件，三个文件缺一不可。



客户端解压后的文件内容

# 6使用流程

## 6.1服务端的使用

**注意，该演示为本地演示，因此在以下的地址中，host都可以使用127.0.0.1来进行替代。**

### 6.1.1登录

地址：<http://{host}/img-spider-server/admin/>

访问该地址后会看到如下界面，在用户名和密码输入框中输入用户名和密码，点击“登录”按钮即可完成登录。（注意：服务端初始化后的初始用户名和密码都为admin，您在实际登录过程中也可以使用您新建的用户名和密码）

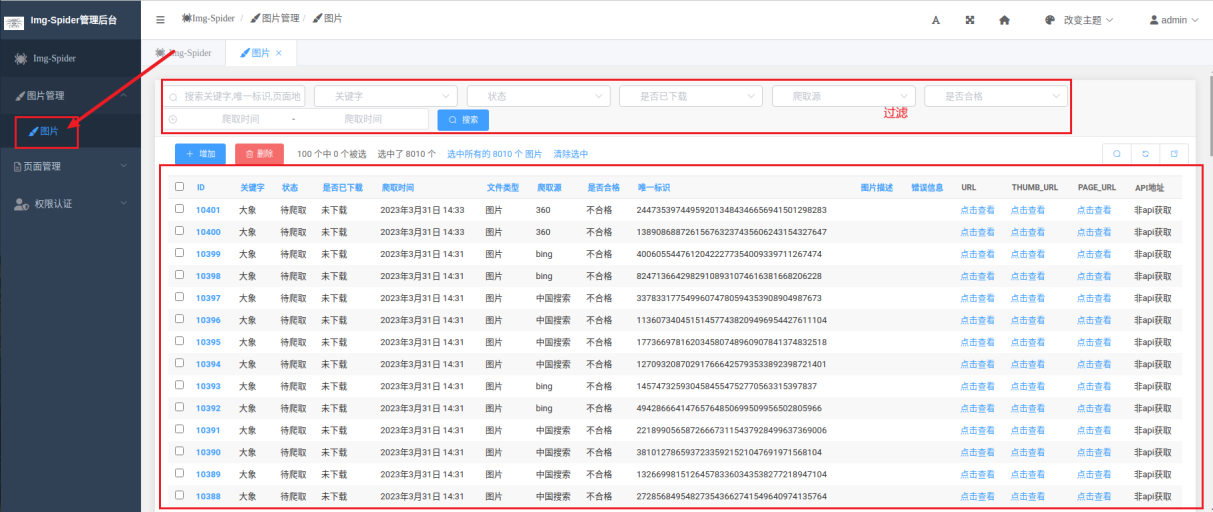


后台登录界面

登录成功后会跳转到管理页面，在管理页面的左侧可以看到“图片管理”，“页面管理”以及“权限验证”选项，点击不同的选项可以展示相关的操作页面。

### 6.1.2图片管理

点击【图片管理】->【图片】，即可看到如下页面：



图片管理

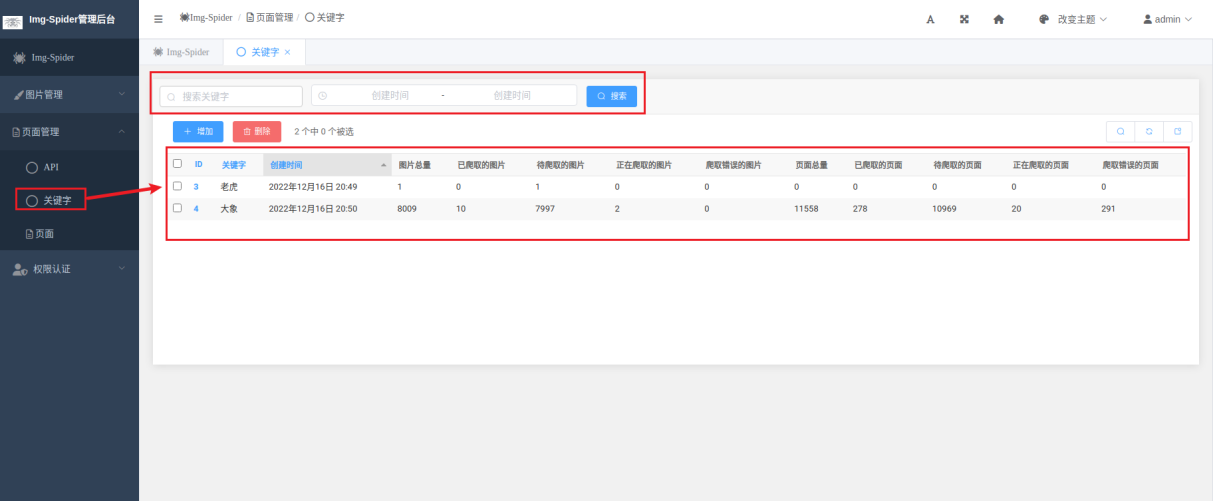
**在该页面中，能够对图片进行筛选和查看，图片中各字段的含义如下：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **默认值** |
| ID | 图片的id | 无 |
| 关键字 | 搜索时的关键字 | 无 |
| 状态 | 用于“以图搜图”的，可选值有：待爬取，爬取中，已爬取，爬取错误，在以图搜图中，正在使用的状态为：爬取中，爬取成功的状态为：已爬取，爬取错误的状态为：爬取错误，其他状态的图片对应未爬取。 | 待爬取 |
| 是否已下载 | 图片是否已下载到本地 | 无 |
| 文件类型 | 可选类型有：图片，视频 | 图片 |
| 爬取源 | 图片的来源，可选值有：360，bing，yandex，中国搜索，搜狗，百度；表示从哪个搜索引擎获取 | 无 |
| 是否合格 | 表示当前图片是否符合要求 | 否 |
| 唯一标识 | 每个图片的唯一标识，用于去重 | 无 |
| 图片描述 | 图片的描述信息 | 无 |
| 错误信息 | 图片爬取失败时的错误信息 | 无 |
| url | 图片原图的地址 | 无 |
| thumb\_url | 图片缩略图的地址 | 无 |
| page\_url | 图片所在的页面的地址 | 无 |
| api地址 | 如果图片是从特定的api中获取的，则显示该api的地址；否者显示:非api获取 | 非api获取 |

### 6.1.3页面管理

#### 6.1.3.1关键字

点击【页面管理】->【关键字】，即可看到如下页面：



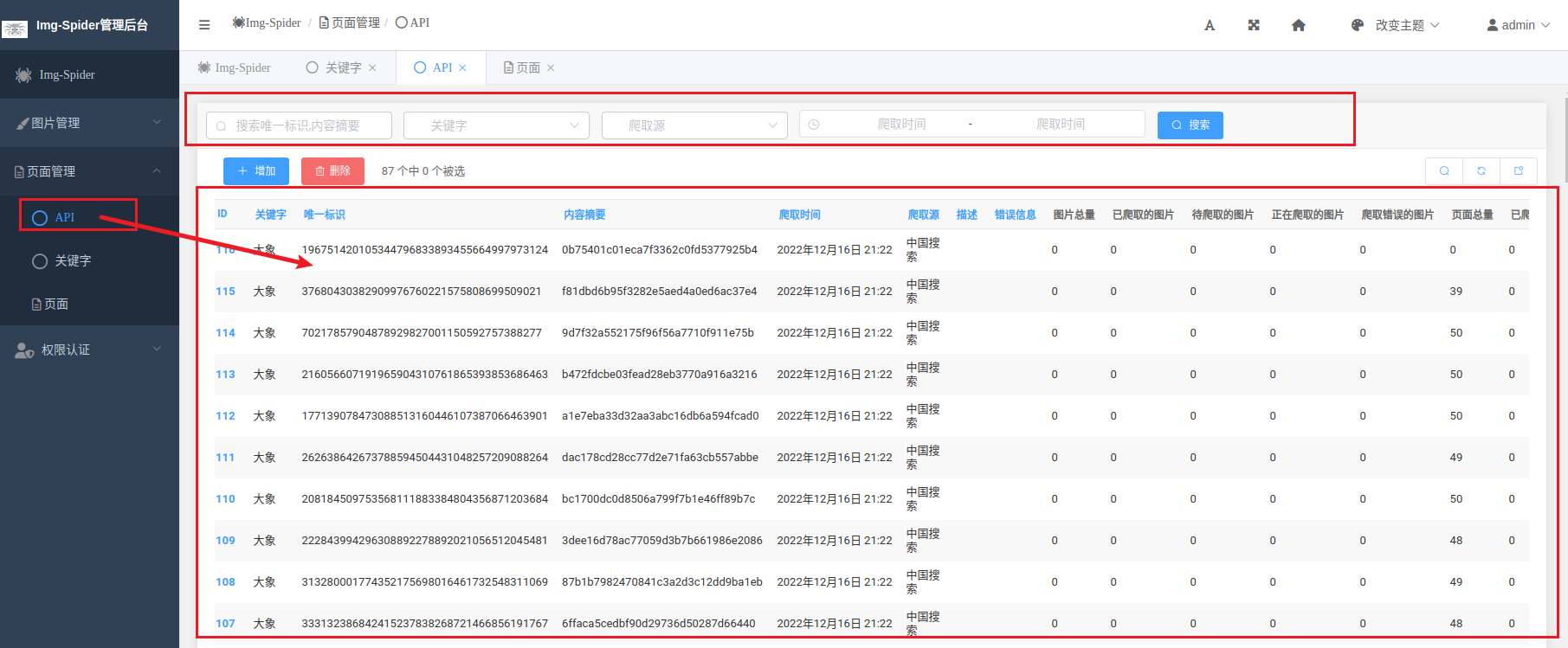
关键字管理

**在该页面中，能够对关键字进行筛选和查看，关键字中各字段的含义如下：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **默认值** |
| ID | 关键字的id | 无 |
| 关键字 | 搜索时的关键字 | 无 |
| 创建时间 | 关键字的创建时间 | 无 |
| 图片总量 | 该关键字下图片的总量 | 0 |
| 已爬取的图片 | 已用于“以图搜图”的图片数量 | 0 |
| 待爬取的图片 | 还未用于“以图搜图”的图片数量 | 0 |
| 正在爬取的图片 | 正在用于“以图搜图”的图片数量 | 0 |
| 爬取错误的图片 | 在“以图搜图”中爬取出错的图片数量 | 0 |
| 页面数量 | 该关键字下页面的总量 | 0 |
| 已爬取的页面 | 已经爬取的页面的数量 | 0 |
| 待爬取的页面 | 待爬取的页面的数量 | 0 |
| 正在爬取的页面 | 正在爬取的页面的数量 | 0 |
| 爬取错误的页面 | 爬取错误的页面的数量 | 0 |

#### 6.1.3.2 API

点击【页面管理】->【API】，即可看到如下页面：



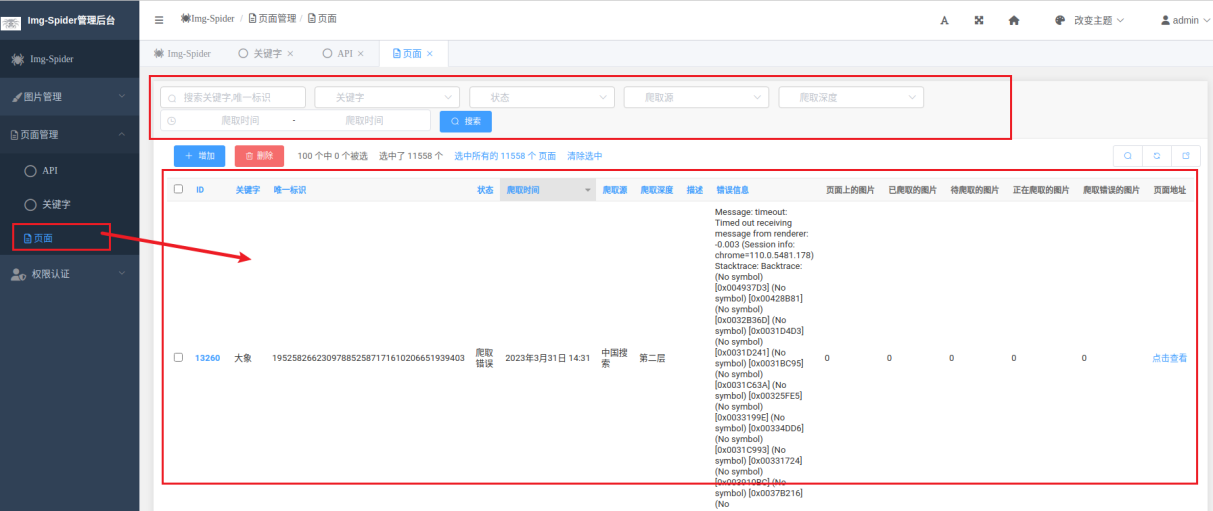
API管理

**在该页面中，能够对API进行筛选和查看，关键字中各字段的含义如下：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **默认值** |
| ID | API的id | 无 |
| 关键字 | 搜索时的关键字 | 无 |
| 唯一标识 | api的唯一标识，用于去重，防止重复爬取 | 无 |
| 内容摘要 | api内容的md5值，再次爬取时，如果内容摘要的值发生改变就说明页面内容发生改变，若发生改变才会重新爬取 | 无 |
| 爬取时间 | api的爬取时间 | 无 |
| 爬取源 | api的来源，可选值有：360，bing，yandex，中国搜索，搜狗，百度；表示从哪个搜索引擎获取 | 无 |
| 描述 | 对api内容的描述信息 | 无 |
| 错误信息 | api内容爬取失败的错误信息 | 无 |
| 图片总量 | 该api下图片的总量 | 0 |
| 已爬取的图片 | 该api下已用于“以图搜图”的图片数量 | 0 |
| 待爬取的图片 | 该api下未用于“以图搜图”的图片数量 | 0 |
| 正在爬取的图片 | 该api下正在用于“以图搜图”的图片数量 | 0 |
| 爬取错误的图片 | 该api下用于“以图搜图”过程中爬取失败的图片数量 | 0 |
| 页面总量 | 该api下页面的总量 | 0 |
| 已爬取的页面 | 该api下已爬取的页面的数量 | 0 |
| 待爬取的页面 | 该api下待爬取的页面的数量 | 0 |
| 正在爬取的页面 | 该api下正在爬取的页面的数量 | 0 |
| 爬取错误的页面 | 该api下爬取错误的页面的数量 | 0 |
| API地址 | 当前api的地址 | 无 |

#### 6.1.3.3页面

点击【页面管理】->【页面】，即可看到如下页面：



页面管理

**在该页面中，能够对页面进行筛选和查看，页面中各字段的含义如下：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **默认值** |
| ID | 页面的id | 无 |
| 关键字 | 搜索时的关键字 | 无 |
| 唯一标识 | 页面的唯一标识，用于去重，防止重复爬取 | 无 |
| 状态 | 页面的状态，可选值有：待爬取，爬取中，已爬取和爬取错误 | 待爬取 |
| 爬取时间 | 页面的爬取时间 | 无 |
| 爬取源 | 页面的来源，可选值有：360，bing，yandex，中国搜索，搜狗，百度；表示从哪个搜索引擎获取 | 无 |
| 爬取深度 | 页面的递归深度，最大深度为3，即最多爬取第三层的页面（例如：页面a为首次获得，那么页面a的爬取深度为第一层；页面b从页面a获取，那么页面b的爬取深度为第二层，依次类推） | 第一层 |
| 描述 | 页面的描述信息 | 无 |
| 错误信息 | 页面爬取失败的错误信息 | 无 |
| 页面上的图片 | 页面上的图片总量 | 0 |
| 已爬取的图片 | 该页面已爬取的图片的数量 | 0 |
| 待爬取的图片 | 该页面待爬取的图片的数量 | 0 |
| 正在爬取的图片 | 该页面正在爬取的图片的数量 | 0 |
| 爬取错误的图片 | 该页面爬取错误的图片的数量 | 0 |
| 页面地址 | 该页面的地址 | 无 |
| API地址 | 如果该页面是通过指定api获取的，则会显示api的地址；否则显示非api获取 | 无 |

## 6.2客户端的使用

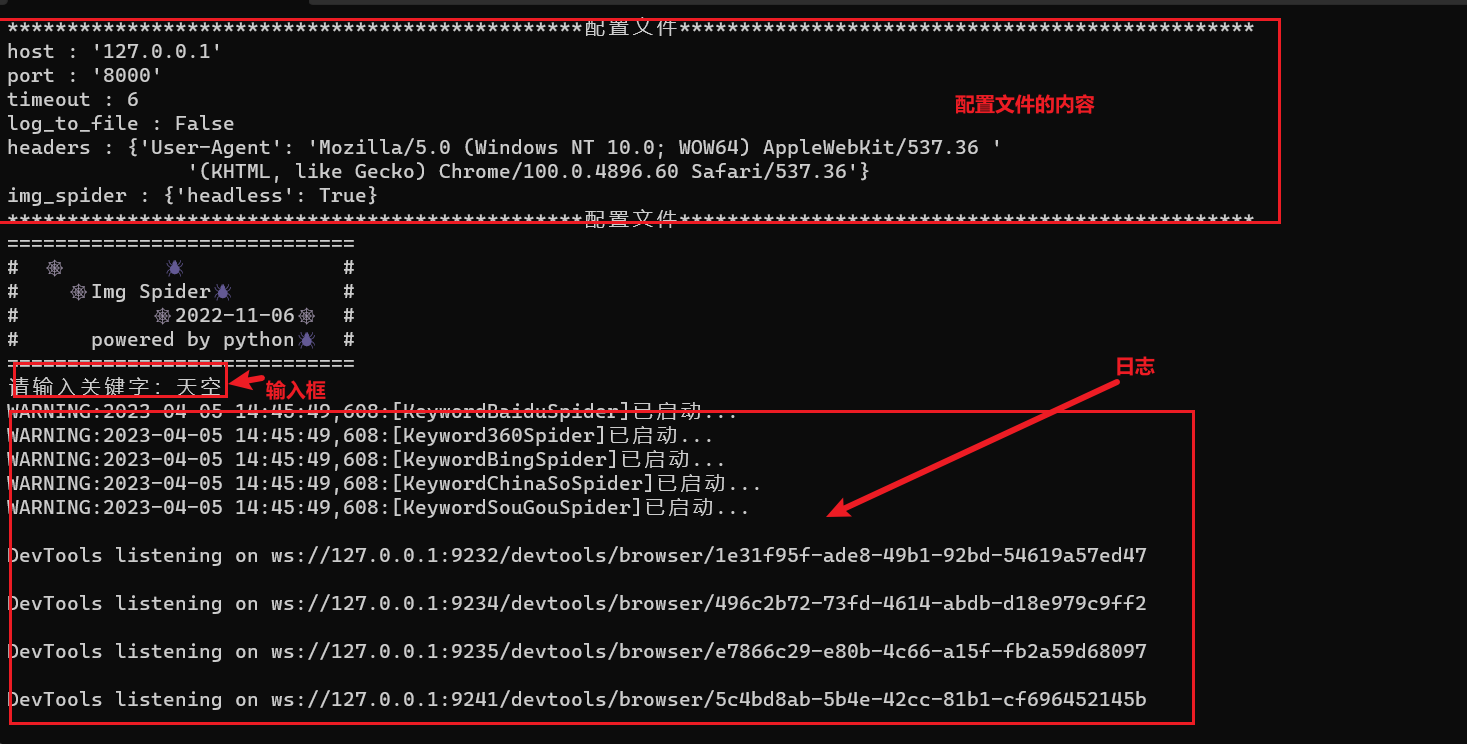
在执行以下操作前，请确保您已将客户端压缩包解压，并已进入到img-spider-exe-master/img-spider-exe-master目录下，在改目录下您可以看到三个文件，分别是：chromedriver.exe，config.json和main.exe文件。

### 6.2.1修改配置文件

双击config.json文件进行配置文件的修改，修改请参考”配置文件说明”进行。

### 6.2.2启动客户端

完成配置文件的修改并保存后，双击main.exe即可启动客户端。



客户端启动成功图

# 7客户端的文件说明

## 7.1日志文件说明

日志文件保存在log目录中，如果在配置文件中设置了log\_to\_file的值为true，那么程序在执行过程中产生的日志就会输出到此目录下，同时，会根据操作的日期生成相应的子目录。

## 7.2执行文件说明

执行文件为main.exe，双击该文件即可启动客户端，执行该文件即可实现功能要求。

## 7.3配置文件说明

配置文件保存了客户端在执行过程中依赖的一些参数，我们以json的格式来保存配置项，因此在修改的过程中一定要保持文件内容符合json格式的规范，否则配置文件读取失败会导致程序无法启动。

### config.json配置文件说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **默认值** | **能否修改** |
| host | 服务端的ip | 127.0.0.1 | 是 |
| port | 服务端使用的端口 | 8000 | 是 |
| timeout | 一次请求过程中的最大等待时间 | 6 | 是 |
| log\_to\_file | 日志是否输出到文件 | false | 是 |
| headers | 请求头的配置 | {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.60 Safari/537.36"} | 是 |
| img\_spider | 图片爬虫的配置 | {"headless":true} | 是 |