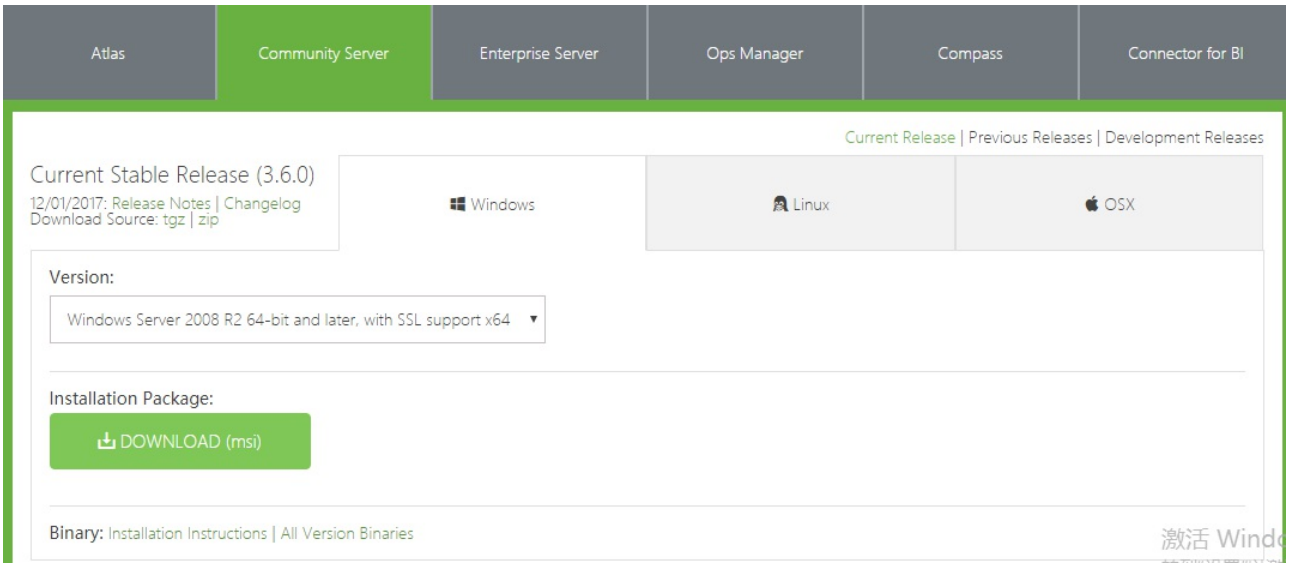
将爬取的文章存储到MongoDB

关于数据的存储有很多选择，最简单的方式就是直接保存到 CSV 文件中，这种方式操作简单，适合数据量少的情况，Python的标准库 csv 模块就可以直接支持。如果遇到数据量非常大的情况，就必须要用到专业的数据库系统，你既可以使用 MySQL 这样的关系型数据库，也可以使用 MongoDB 一类的文档型数据库。用Python 操作 MongoDB 非常方便，无需定义表结构就可以直接将数据插入，所以我们在这一节采用 MongoDB 来存储数据。

MongoDB 安装

MongoDB 目前最新版本是3.6，在官网地址https://www.mongodb.com/download-center#community
选择相应平台下载安装。



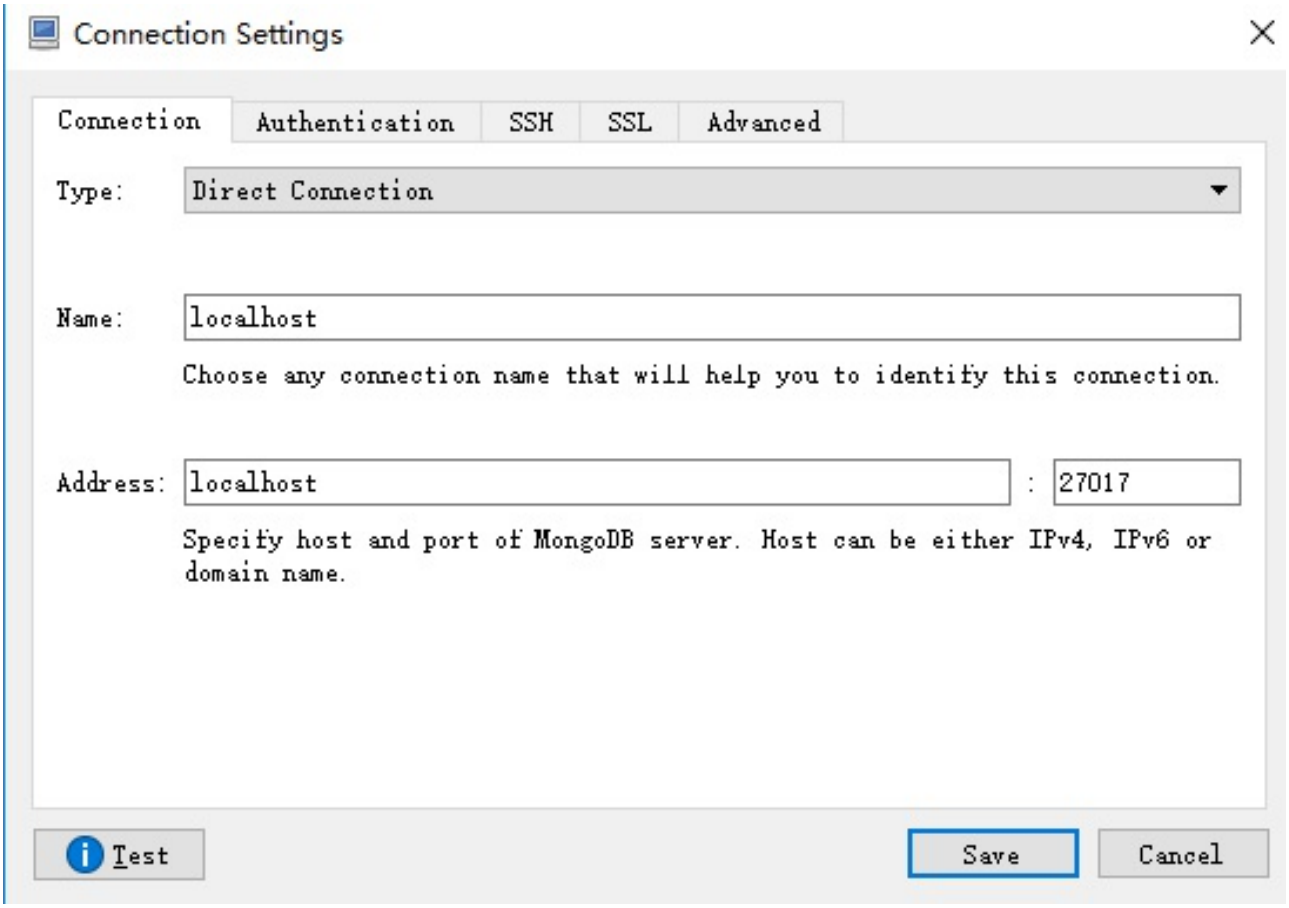
Windows 默认安装在 C:\Program Files\MongoDB\Server\3.6\，macOS 也可以直接通过 brew 命令安装，Linux平台直接下载压缩包解压即可。

```
brew install mongodb --with-openssl
```

启动 MongoDB

```
mongod --dbpath <path to data directory>
```

默认端口是 27017，为了更好的查看数据，我们可以装一个 MongoDB 客户端， 官方自带有 compass，也可以下载第三方工具 Robo 3T https://robomongo.org/ (https://robomongo.org/)，这里推荐大家使用免费的 Robo 3T。



MongoEngine

MongoEngine 是 MongoDB 的 DOM（Document-Object Mapper）框架，一种类似于关系型数据库中的ORM框架 ， 使用它可以更方便并写出简洁的代码

安装

```
$ pip install mongoengine
```

连接

```
from mongoengine import connect
# 连接 mongodb，无需事先创建数据库
connect('weixin', host='localhost', port=27017)
```

定义数据模型

```
# -*- coding: utf-8 -*-
from datetime import datetime

from mongoengine import DateTimeField
from mongoengine import Document
from mongoengine import IntField
from mongoengine import StringField
from mongoengine import URLField
from mongoengine import connect

__author__ = "liuzhijun"

# 连接 mongodb
connect('weixin2', host='localhost', port=27017)


class Post(Document):
    """
    文章信息
    """
    title = StringField()  # 文章标题
    content_url = StringField()  # 文章链接
```

```
    content = StringField()  # 文章内容
    source_url = StringField()  # 原文链接
    digest = StringField()  # 文章摘要
    cover = URLField(validation=None)  # 封面图
    p_date = DateTimeField()  # 推送时间

    read_num = IntField(default=0)  # 阅读数
    like_num = IntField(default=0)  # 点赞数
    comment_num = IntField(default=0)  # 评论数
    reward_num = IntField(default=0)  # 赞赏数
    author = StringField()  # 作者

    c_date = DateTimeField(default=datetime.now)
# 数据生成时间
    u_date = DateTimeField(default=datetime.now)
# 最后更新时间
```

数据保存

在第五小节中，我们只是把抓取的数据简单的打印出来，现在我们就把它存数据库，因为抓取的数据中有很多无用的字段，所以，这里我们写一个工具函数叫 sub_dict 用于获取指定字段信息。

```
import html
def sub_dict(d, keys):
    return {k: html.unescape(d[k]) for k in d if
k in keys}

d = {"a": "1", "b": 2, "c": 3}
sub_dict(d, ["a", "b"]) # {"a":"1", "b": "2"}
```

获取字典的子字典可以用字典推导式实现，我这里还导入了html.unescape方法是希望保存到数据库的数据都是经过反转义处理的。

```
    @staticmethod
    def save(msg_list):

        msg_list = msg_list.replace("\/", "/")
        data = json.loads(msg_list)
        msg_list = data.get("list")
        for msg in msg_list:
            p_date =
msg.get("comm_msg_info").get("datetime")
            msg_info =
msg.get("app_msg_ext_info")  # 非图文消息没有此字段
            if msg_info:
                WeiXinCrawler._insert(msg_info,
p_date)
                multi_msg_info =
msg_info.get("multi_app_msg_item_list") # 多图文推送，把第二条第三条也保存
                for msg_item in multi_msg_info:

WeiXinCrawler._insert(msg_item, p_date)
            else:
                logger.warning(u"此消息不是图文推送,
data=%s" % json.dumps(msg.get("comm_msg_info")))

    @staticmethod
    def _insert(item, p_date):
        keys = ('title', 'author', 'content_url',
'digest', 'cover', 'source_url')
        sub_data = utils.sub_dict(item, keys)
```

```
            post = Post(**sub_data)
            p_date = datetime.fromtimestamp(p_date)
            post["p_date"] = p_date
            logger.info('save data %s ' % post.title)
            try:
                post.save()
            except Exception as e:
                logger.error("保存失败 data=%s" %
post.to_json(), exc_info=True)
```

如果是文字推送就没有app_msg_ext_info字段，无需保存，multi_app_msg_item_list是多图文推送字段，而且和外层的app_msg_ext_info字段是一致的，有标题、封面图、摘要、链接等信息，所以我们把插入数据库的代码_insert作为私有方法抽离出来共用。

最后我们看一下保存的数据。

```
db.getCollection('post').find({}).sort({"p_date":-1})
```

post 0.006 sec.

Key	Value	Type
(1) ObjectId("5a4539c7a54d75940d090fd...	{ 14 fields }	Object
_id	ObjectId("5a4539c7a54d75940d090fd3")	ObjectId
title	5个酷毙的Python工具	String
content_url	http://mp.weixin.qq.com/s?__biz=MjM5MzgyODQ...	String
source_url		String
digest	工欲善其事必先利其器，一个好的工具能让起到事半...	String
cover	http://mmbiz.qpic.cn/mmbiz_jpg/rO1ibUkmNGMm...	String
p_date	2017-12-27 08:00:00.000Z	Date
read_num	0	Int32
like_num	0	Int32
comment_num	0	Int32
reward_num	0	Int32
author	刘志军	String
c_date	2017-12-29 02:36:55.787Z	Date
u_date	2017-12-29 02:36:55.787Z	Date
(2) ObjectId("5a4539c7a54d75940d090fd...	{ 14 fields }	Object
(3) ObjectId("5a4539c7a54d75940d090fd...	{ 14 fields }	Object
(4) ObjectId("5a4539c7a54d75940d090fd...	{ 14 fields }	Object

小结

本节完成代码在GitHub v0.3 (https://github.com/pythonzhichan/weixincrawler/tree/v0.3) 这小节我们主要熟悉 Mongodb 的安装以及如何用Python连接 Mongodb 进行数据存储，推荐两个资源，第一个是：《MongoDB 入门指南》(https://jockchou.gitbooks.io/getting-started-with-mongodb/content/)，第二个是 MongoEngine 教程 (http://docs.mongoengine.org/index.html)，如果你想进行系统的学习 MongoDB，推荐两本书籍《MongoDB权威指南》和《MongoDB实战》。