# Connection MongoDB


pip install pymongo

## 3.1. Robomongo 설치 (MongoDB 관리 GUI 툴) (실습)

- https://robomongo.org/download
  - 맥에서 처음 실행시 잘 안되면, 삭제 후, 다시 다운로드받아서 재설치하세요
- 실행
  - Click Create
    - In Connection
      - Name: AWS EC2 IP
      - Address: AWS EC2 IP (port는 27017 디폴트)
    - In Authentication
      - Database 이름: admin
      - User Name: 사용자 ID
      - Password: 사용자 암호
    - Click Test & Save button to check connection
  - Connection
    - Click Coonect buttion
  - Check connection
    - Right-click (server name) -> Show Log
  - Create DB
    - Right-click (Server) -> Create Database -> Add dave_db
  - Create Collection
    - Right-click (Collections(0) in created DB) -> Create Collection -> Add test
  - Insert Document
    - Right-click (test collection) -> Insert Document -> Add the following JSON(BSON) Document -> Click Validate & Save buttons
    - Click View Documents in test collection
    - Check *id* in indexes of test collection (인덱스 자동 생성)

# Crawling Attached File


pip install tika

# java-hwp

JitPack a1a799b2c5

본 제품은 한글과컴퓨터의 한글 문서 파일(.hwp) 공개 문서를 참고하여 개발하였습니다.

개발에 많은 도움을 주신 cogniti님과 libhwp Google Group 그룹에 감사드립니다.

HWP 파일에서 텍스트를 추출하는 자바 라이브러리이며 ruby-hwp 의 자바 버전입니다. ruby-hwp의 로직을 대부분 그대로 사용하며 ruby-hwp의 문자매핑 정보(hnc2unicode.rb) 파일을 사용합니다.

HWP 5.0 버전의 Compound File은 Apache-POI의 POIFS File System을 사용하여 처리합니다.

## 사용방법

```
File hwp = new File("hangul.hwp"); // 텍스트를 추출할 HWP 파일
Writer writer = new StringWriter(); // 추출된 텍스트를 출력할 버퍼
HwpTextExtractor.extract(hwp, writer); // 파일로부터 텍스트 추출
String text = writer.toString(); // 추출된 텍스트
```

```
import jpype
import scrapy
import os
import sys
from crawlNKDB.items import CrawlnkdbItem
import re
import pymongo
from pymongo import MongoClient
import gridfs
from tika import parser
from tempfile import NamedTemporaryFile
from itertools import chain
control_chars = ''.join(map(chr, chain(range(0, 9), range(11, 32), range(127, 160))))
CONTROL_CHAR_RE = re.compile('[%s]' % re.escape(control_chars))
```

```
class Boardbotkinu19Spider(scrapy.Spider):
    ###### 수정사항
    name = 'boardbotKinu19'
    allowed_domains = ['www.kinu.or.kr']
    ####### 수정사항
    start_urls = ['http://www.kinu.or.kr/www/jsp/prg/api/dlL.jsp?menuIdx=340&category=41&thisPage=1&searchField=&searchText=']

    def __init__(self):
        scrapy.Spider.__init__(self)
        ####### 수정사항
        self.start_urls = 'http://www.kinu.or.kr/www/jsp/prg/api/dlL.jsp?menuIdx=340&category=41&thisPage=1&searchField=&searchText='
        self.client = pymongo.MongoClient(config['DB']['MONGO_URI'])
        self.db = self.client['attchment']
        self.fs = gridfs.GridFS(self.db)
        jarpath = os.path.join(os.path.abspath('.'), './../lib/hwp-crawl.jar')
        jpype.startJVM(jpype.getDefaultJVMPath(), "-Djava.class.path=%s" % jarpath)
```

```
def start_requests(self):
    yield scrapy.Request(self.start_urls, self.parse)

def parse(self, response):
    page_no = 1
    last_page_text = response.xpath('//*[@id="boardActionFrm"]/div[1]/div[1]/span').extract()
    print(last_page_text)
    last_page_no = re.findall("\d+", str(last_page_text))
    print(last_page_no)
    last_page_no = int(last_page_no[-1])
    print(last_page_no)
    while True:
        if page_no > last_page_no:
            break
        ####### 수정사항
        link = "http://www.kinu.or.kr/www/jsp/prg/api/dlL.jsp?menuIdx=340&category=41&thisPage=" + str(page_no) + "&searchField=title&searchText="
        print(link)
        yield scrapy.Request(link, callback = self.parse_each_pages, meta={'page_no': page_no, 'last_page_no': last_page_no})
        page_no += 1
```

```
def parse_each_pages(self, response):
    page_no = response.meta['page_no']
    last_page_no = response.meta['last_page_no']
    print("###pageno:  ", page_no)
    last = response.xpath('//*[@id="boardActionFrm"]/div[2]/table/tbody/tr[1]/td[1]/text()').get()
    if page_no == last_page_no:
        category_last_no = int(last)
    else:
        first = response.xpath('//*[@id="boardActionFrm"]/div[2]/table/tbody/tr[10]/td[1]/text()').get()
        category_last_no = int(last) - int(first) + 1
    category_no = 1
    while True:
        if(category_no > category_last_no):
            break
        category_link = response.xpath('//*[@id="boardActionFrm"]/div[2]/table/tbody/tr['+ str(category_no) +']/td[2]/a/@href').get()
        url = 'http://www.kinu.or.kr/www/jsp/prg/api/' + category_link
        # print(url)
        number = response.xpath('//*[@id="boardActionFrm"]/div[2]/table/tbody/tr['+str(category_no)+']/td[1]').get()
        #print(number)
        item = CrawlnkdbItem()
        date = response.xpath('//*[@id="boardActionFrm"]/div[2]/table/tbody/tr['+str(category_no)+']/td[3]').xpath('string()').get()
        item[config['VARS']['VAR4']] = date
        yield scrapy.Request(url, callback=self.parse_post, meta={'item':item})
        category_no += 1
```

```python
def parse_post(self, response):
    title = response.xpath('//*[@id="cmsContent"]/div[1]/p').xpath('string()').get()
    body = response.css('#tab_con > div').xpath('string()').get()

    writer = response.css('#cmsContent > div.board_wrap_bbs > table > thead > tr:nth-child(1) > td').xpath('string()').get()
    top_category = response.css('#container > div.content > div.conTop > div > h2').xpath('string()').get()
    item = response.meta['item']
    item[config['VARS']['VAR1']] = title
    item[config['VARS']['VAR3']] = writer
    item[config['VARS']['VAR2']] = body
    item[config['VARS']['VAR5']] = "통일연구원"
    item[config['VARS']['VAR6']] = "http://www.kinu.or.kr/www/jsp/prg/"
    item[config['VARS']['VAR7']] = top_category
    file_name = title
    file_icon = response.xpath('//*[@id="cmsContent"]/div[2]/table/thead/tr[5]/td/a/img').get()
    if file_icon:
        file_download_url = response.xpath('//*[@id="cmsContent"]/div[2]/table/thead/tr[5]/td/a/@href').extract()
        file_download_url = file_download_url[0]
        item[config['VARS']['VAR10']] = file_download_url
        item[config['VARS']['VAR9']] = file_name
        print("@@@@@@file name ", file_name)
        if file_icon.find("hwp") != -1 :
            print('find hwp')
            yield scrapy.Request(file_download_url, callback=self.save_file_hwp, meta={'item':item}) #
        else:
            yield scrapy.Request(file_download_url, callback=self.save_file, meta={'item':item})
    else:
        print("###############file does not exist################")
        yield item
```

```
def save_file(self, response):
    item = response.meta['item']
    file_id = self.fs.put(response.body)
    item[config['VARS']['VAR11']] = file_id

    tempfile = NamedTemporaryFile()
    tempfile.write(response.body)
    tempfile.flush()

    extracted_data = parser.from_file(tempfile.name)
    extracted_data = extracted_data["content"]
    if str(type(extracted_data)) == "<class 'str'>":
        extracted_data = CONTROL_CHAR_RE.sub('', extracted_data)
        extracted_data = extracted_data.replace('\n\n', '')
    tempfile.close()
    item[config['VARS']['VAR12']] = extracted_data
    yield item
```

```
def save_file_hwp(self, response):
    item = response.meta['item']
    file_id = self.fs.put(response.body)
    item[config['VARS']['VAR11']] = file_id

    tempfile = NamedTemporaryFile()
    tempfile.write(response.body)
    tempfile.flush()

    testPkg = jpype.JPackage('com.argo.hwp') # get the package
    JavaCls = testPkg.Main # get the class
    hwp_crawl = JavaCls() # create an instance of the class
    extracted_data = hwp_crawl.getStringTextFromHWP(tempfile.name)
    if str(type(extracted_data)) == "<class 'str'>":
        extracted_data = CONTROL_CHAR_RE.sub('', extracted_data)
        extracted_data = extracted_data.replace('\n\n', '')
    print(extracted_data)
    print("###############get the hwp content###############")
    tempfile.close()
    item[config['VARS']['VAR12']] = extracted_data
    yield item
```

```
def __del__(self):
    jpype.shutdownJVM()
```

master

NKDB / crawlNKDB / lib / **config.cnf**

**0sunzero0** debug crawler

**1** contributor

23 lines (20 sloc) | 497 Bytes

```
[DB]
mongo_uri = mongodb://localhost:27017
mongo_db = NKDB

[VARS]
var1 = post_title
var2 = post_body
var3 = post_writer
var4 = post_date
var5 = published_institution
var6 = published_institution_url
var7 = top_category
var8 = published_date
var9 = file_name
var10 = file_download_url
var11 = file_id_in_fsfiles
var12 = file_extracted_content

[LOCAL]
path_spider = /Users/sunzero/Dropbox/Crawling/NKDB_Crawling/crawlNKDB/spiders

[SERVER]
path_spider = /home/hyeyoung/NKDB/NKDB/crawlNKDB/spiders
```

Thank you