```python
import pymongo
import json
import dotenv
import os
import sys
import urllib
import urllib.request
from pymongo import MongoClient
import datetime
import xmltodict


def load_config_file():
    with open("crawl.conf", 'r') as conf_file:
        return json.load(conf_file)
    return dict()

def get_oldest_keyword():
# 가장 crawl되지 않은 keyword를 반환한다.
    # 한번도 crawling한 적이 없는 keyword를 crawling한다.
    result = keywords_db_collection.find_one({"updated": None})
    if result != None:
        print(f"keyword: {result['keyword']}")
        return result

    # 모두 crawling한 적이 있다면 이곳에 도달한다.
    # 가장 옛날에 crawling한 녀석을 찾는다
    # 단 한달 안에 crawling한 녀석은 제외한다.
    oneMonthAgo = datetime.datetime.now() - datetime.timedelta(days=30)
    results = keywords_db_collection.find({ "updated": {"$lt": oneMonthAgo}}) \
        .sort("updated", 1).limit(1)

    for item in results:
        print(f"keyword: {item['keyword']}")
        return item

    return None

def extract_db_fields(item):
# 크롤링해 온 dict에서 DB에 저장할 fileld만 빼내어 반환한다.
# @param  item 크롤링해 온 item
    res = dict()
    fields = ['applicantName', 'applicationDate', 'applicationNumber', 'astrtCont', \
        'inventionTitle', 'registerDate', 'registerNumber', 'registerStatus']

    for x in fields:
        res[x] = item[x]

    return res

def crawl_patent_iterator(keyword):
# keyword로 검색한 특허를 키프리스에서 가져다 하나씩 반환하는 Iterator
#
# @param keyword 검색하고자하는 keyword
# @return 매 iteration마다 patent dict를 반환. 단 문제가 생기면 None을 반환

    client_key = os.getenv('KIPRIS_KEY')
```

```
58      number_of_rows = 10 # 한번에 crawling하는 특허 갯수. 500이 이상적이나 테스트 목적으로 10으로 설
    정
59      total_count = -1    # 총 특허 갯수. 처음에는 알 수 없으므로 -1로 설정
60      current_page = 0    # 현재까지 crawling한 페이지 수
61
62      while total_count < 0 or current_page * number_of_rows <= total_count:
63
64          encText = urllib.parse.quote(keyword)
65          current_page += 1 # API에서는 1페이지부터 시작하므로 미리 1 증가시킨다
66
67          url = "http://plus.kipris.or.kr/kipo-
    api/kipi/patUtiModInfoSearchSevice/getWordSearch?word=" \
68              + encText + f"&year=10&pageNo={current_page}&numOfRows={number_of_rows}"
    \
69              + "&ServiceKey=" + client_key
70
71          request = urllib.request.Request(url)
72          response = urllib.request.urlopen(request)
73          rescode = response.getcode()
74
75          if(rescode==200):
76              response_body = response.read()
77              response_dict = xmltodict.parse(response_body.decode('utf-8'))
78
79              items = response_dict['response']['body']['items']['item']
80              for item in items:
81                  # DB에 저장할 field만 골라내어 반환
82                  yield extract_db_fields(item)
83
84              total_count = int(response_dict['response']['count']['totalCount'])
85              print(f"crawl: {current_page*number_of_rows}/{total_count}")
86          else:
87              # 문제가 생길경우 None을 반환함으로써 문제가 생겼음을 알린다.
88              print("Error Code:" + rescode)
89              yield None
90
91  def store_in_db(item, keyword):
92  # DB에 item을 등록한다.
93  # 이미 등록된 item이라면 새로운 keyword도 등록한다.
94  # keyword도 등록되어 있다면 수정사항이 없음을 알린다.
95  # @param item      등록할 특허
96  # @param keyword   검색한 keyword
97  # @return          수정사항이 있으면 True 아니면 False
98      # applicationNumber를 id로 활용한다.
99      item['_id'] = item['applicationNumber']
100
101     # 해당 특허가 이미 있는지 점검한다.
102     item_in_db = patents_db_collection.find_one({"_id": item['_id']})
103     if item_in_db != None:
104         if keyword in item_in_db['keywords']:
105             # keyword까지 이미 등록되어 있다면 update할 내용이 없다.
106             return False
107         else:
108             # keyword가 등록되어 있지 않다면 keyword를 update한다.
109             new_keywords = item_in_db['keywords'] + keyword
110             patents_db_collection.update_one({"_id": item['_id']}, {'$set':
    {'keywords':new_keywords}})
111             return True
```

```
112     else:
113         # 특허가 등록되어 있지 않다면 특허를 등록한다.
114         item['keywords'] = [keyword]
115         patents_db_collection.insert_one(item)
116         return True
117 
118 def crawl_patents_into_db(keyword_in_db):
119     keyword = keyword_in_db['keyword']
120 
121     # 앞서 완전히 검색을 완료한 keyword는 (completed), incremental하게 검색한다.
122     incremental = keyword_in_db.get('completed', False)
123     # 이번 검색을 시작하기 전에 해당 keyword가 completed되지 않았음을 적는다.
124     keywords_db_collection.update_one({"keyword": keyword}, \
125         {'$set': {'completed': False}})
126 
127     # incremental 검색에서
128     known_count = 0
129     for item in crawl_patent_iterator(keyword):
130         if item == None: # 검색에 문제가 있었다면
131             print(f"{keyword} 키워드검색 중 문제가 발생하였습니다.")
132             sys.exit(1)
133 
134         #검색에 문제가 없었다면
135         updated = store_in_db(item, keyword)
136         # incremental 검색에서 이전 검색이 반복 됐다면
137         if incremental:
138             if not updated:
139                 known_count += 1
140             else:
141                 known_count = 0
142             if known_count >= accepting_known_patents:
143                 # 충분히 아는 patent를 만났다면 완료를 저장하고 반환한다
144                 patents_db_collection.update_one({"keyword": keyword}, \
145                     {'$set': {'completed': True, 'updated':
    datetime.datetime.now()}})
146                 return
147 
148     #모든 patent들을 다 crawling했다면 완료를 저장하고 반환한다.
149     keywords_db_collection.update_one({"keyword": keyword}, \
150         {'$set': {'completed': True, 'updated': datetime.datetime.now()}})
151     return
152 
153 
154 if __name__ == '__main__':
155     # 환경변수 로드하기 (API 키 등을 위해서)
156     dotenv.load_dotenv(verbose=True)
157 
158     # mongoDB 연결하기
159     cluster = MongoClient("mongodb://root:example@localhost/")
160     db = cluster['cosmax']
161     keywords_db_collection = db['keywords']
162     patents_db_collection = db['patents']
163 
164     # 한번에 Crawling할 keyword의 갯수를 가져 온다. 없다면 기본값으로 1을 반환
165     conf = load_config_file()
166     crawls_per_day = conf.get('keyword_per_day', 1)
167     accepting_known_patents = conf.get('accepting_known_patents', 10)
```

```
168
169     for idx in range(0, crawls_per_day):
170         keyword_from_db = get_oldest_keyword()
171         if keyword_from_db != None:
172             crawl_patents_into_db(keyword_from_db)
```