```python
from pymongo import MongoClient
import psycopg2
import dotenv
import os
import json
import datetime
import sys
import urllib
import urllib.request
import xmltodict

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

def get_oldest_keyword(conf):
    cursor.execute( \
        '''SELECT keyword, completed
            FROM keywords_keyword
            WHERE updated IS NULL''')

    result = cursor.fetchone()

    print(result)

    if result != None:
        return result

    pivot_days = conf.get('pivot_days', 30)
    oneMonthAgo = datetime.datetime.now() \
        - datetime.timedelta(days=pivot_days)
    print('one month ago', oneMonthAgo)
    cursor.execute( \
        '''SELECT keyword, completed
            FROM keywords_keyword
            WHERE updated < (%s)
            ORDER BY updated''', [oneMonthAgo])

    result = cursor.fetchone()
    print(result)

    return result

def extract_db_fields(item):
    fields = ['applicantName', 'applicationDate', 'applicationNumber',\
        'astrtCont', 'inventionTitle', 'registerDate', 'registerNumber',
        'registerStatus']

    res = dict()
    for x in fields:
        res[x] = item[x]
    return res

def crawl_patent_iterator(keyword):
    client_key = os.getenv('KIPRIS_KEY')
    print(client_key)
    encText = urllib.parse.quote(keyword)
```

```
58
59      current_page = 0
60      total_count = -1
61      number_of_rows = 10
62
63      while total_count < 0 or current_page * number_of_rows < total_count:
64
65          current_page += 1
66
67          url = "http://plus.kipris.or.kr/kipo-api/kipi" \
68              + "/patUtiModInfoSearchSevice/getWordSearch?word=" \
69              + encText + f"&pageNo={current_page}&numOfRows={number_of_rows}" \
70              + "&ServiceKey=" + client_key
71
72          request = urllib.request.Request(url)
73          response = urllib.request.urlopen(request)
74
75          if response.getcode() == 200:
76              res_dict = xmltodict.parse(response.read().decode('utf-8'))
77
78              items = res_dict['response']['body']['items']['item']
79              total_count = int(res_dict['response']['count']['totalCount'])
80              for item in items:
81                  yield extract_db_fields(item)
82          else:
83              print("Error Code: ", rescode)
84              yield None
85
86  def store_in_db(item, keyword):
87      appNumber = item['applicationNumber']
88      cursor.execute(
89          '''SELECT keywords
90              FROM patents_patent
91              WHERE app_number = (%s)''',
92          [appNumber])
93
94      item_in_db = cursor.fetchone()
95
96      print('item in DB', item_in_db)
97
98      if item_in_db != None:
99          if keyword in item_in_db[0]:
100             print(f"{item['applicationNumber']} is not updated")
101             return False
102         else:
103             new_keywords = item_in_db[0] + [keyword]
104             cursor.execute(
105             '''UPDATE patents_patent
106                 SET keywords = (%s)
107                 WHERE app_number = (%s)''',
108                [new_keywords, appNumber])
109             db.commit()
110             print(f"{item['applicationNumber']} is updated")
111
112             return True
113     else: # item이 DB에 없다면
114         print(item)
```

```
115         cursor.execute(
116             '''INSERT INTO patents_patent VALUES
117                 (%s, %s, %s, %s, %s, %s, %s, %s, %s)
118             ''',
119             [item['applicantName'],
120             item['applicationDate'],
121             item['applicationNumber'],
122             item['astrtCont'],
123             item['inventionTitle'],
124             item['registerDate'],
125             item['registerNumber'],
126             item['registerStatus'],
127             [keyword]])
128         db.commit()
129         print(f"{item['applicationNumber']} is inserted")
130 
131         return True
132 
133 
134 def crawl_patents_into_db(keyword_in_db):
135 
136     keyword = keyword_in_db[0]
137 
138     incremental = keyword_in_db[1]
139 
140     print('keyword', keyword)
141     cursor.execute(
142         '''UPDATE keywords_keyword
143             SET completed = 'f'
144             WHERE keyword = (%s)''',
145         [keyword])
146     db.commit()
147 
148     known_count = 0
149     accepting_known_patents = 50
150     for item in crawl_patent_iterator(keyword):
151         if item == None:
152             print(f"{keyword} 검색중에 문제가 발생하였습니다.")
153             sys.exit(1)
154 
155         updated = store_in_db(item, keyword)
156         if incremental:
157             if not updated:
158                 known_count += 1
159             else:
160                 known_count = 0
161             if known_count >= accepting_known_patents:
162                 break
163 
164     keywords_db_collection.update_one({'keyword': keyword},\
165         {"$set": {"completed": True, \
166             "updated": datetime.datetime.now()}})
167 
168 
169 
170 if __name__ == '__main__':
171     # .env에 있는 환경변수 로드
```

```
172     dotenv.load_dotenv(verbose=True)
173
174     # 설정 파일 불러오기
175     conf = load_config_file()
176
177     # mongoDB 초기화
178     mongo_user = os.getenv('MONGO_INITDB_ROOT_USERNAME')
179     mongo_password = os.getenv('MONGO_INITDB_ROOT_PASSWORD')
180
181     db = psycopg2.connect(host='localhost', dbname='patents',\
182                 port=5432)
183     cursor = db.cursor()
184
185     keywords_per_day = conf.get('keywords_per_day', 1)
186
187     for idx in range(0, keywords_per_day):
188         keyword = get_oldest_keyword(conf)
189         if keyword == None:
190             print('No more keyword remained')
191             sys.exit(0)
192
193         crawl_patents_into_db(keyword)
```