
Chapter 17 파이썬과 인터넷

[이번 주 목표]

- 인터넷에서 데이터 가져오기 샘플 구현.
- 내 프로젝트에 검색 기본기능 구현.
사목게임.

- socket : low level 네트워킹 인터페이스

import socket

```
HOST = "          #호스트를 지정하지 않으면 가능한 모든 IP의 접속을 받음
# 1024~49151:server용 port (0~1023:well-known port)
PORT = 49001
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #accept 소켓 생성
s.bind((HOST, PORT))      # 소켓에 주소를 할당
s.listen(1)               # 연결 요청이 있을 때까지 기다림
conn, addr = s.accept()    # 연결 요청을 수락 -> (client socket, 상대방의 주소) 얻음
print('Connected by', addr)
while True:
    data = conn.recv(1024) # client socket으로부터 읽기.
    if not data:
        break
    conn.send(data)        # 받은 데이터를 그대로 클라이언트에 전송
conn.close()              # client socket 닫기
s.close()                 # server socket 닫기
```

- socket : low level 네트워킹 인터페이스

```
import socket
```

```
HOST = '127.0.0.1'
```

```
#localhost
```

```
PORT = 49001
```

```
#서버와 같은 포트를 사용함
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #소켓 생성
```

```
s.connect((HOST,PORT))
```

```
s.send(b'Hello, python')
```

```
#문자를 보냄
```

```
data = s.recv(1024)
```

```
#서버로 부터 정보를 받음
```

```
s.close()
```

```
print('Received:', repr(data))
```

socket

- socket 예제 실행

- socket_server.py 를 cmd 창에서 실행시키기 (client보다 먼저 실행!)


```
C:\\source\\17ch>python socket_server.py  
Connected by ('127.0.0.1', 9532)
```

- socket_client.py 를 cmd 창에서 실행시키기

```
C:\\source\\17ch>python socket_client.py  
Received: b'Hello, Python.'
```

json 데이터 다루기

▪ json(JavaScript Object Notation) 모듈

- 가벼운 데이터 교환을 위한 자바스크립트 문법
- **dump, dumps** (파이썬 데이터를 json 형식으로 변환하여 파일/문자열로 출력)
load, loads (파일/문자열로부터 읽어서 json을 파이썬 객체로 변환)
-  객체는 중괄호와 콜론으로 구성됨
`{"age":30}` ← json 객체 이름은 age 이고 값은 30

```
>>> import json
>>> json.dumps([1,2,3,{ '4':5, '6':7}], separators=(',', ':'))
'[1,2,3,{"4":5,"6":7}]'
>>> json.loads('[1, 2, 3, {"4": 5, "6": 7}])
[1, 2, 3, {'4': 5, '6': 7}]
```

```

strJson = ""
{
    "resultCode": 0,
    "resultMsg": "NORMAL_CODE",
    "numOfRows": 5,
    "pageNo": 3,
    "totalCount": 1361,
    "items": [
        {
            "no": 10,
            "pbanc_ymd": "2021-12-08",
            "tot_cases": 9527,
            "qrntn": 1936,
            "qrntn_rmv": 7534,
            "dth_cnt": 57,
            "pblcn_dt": "2021-12-08 08:00:00"
        },
        {
            "no": 11,
            "pbanc_ymd": "2022-01-21",
            "tot_cases": 1111,
            "qrntn": 2222,
            "qrntn_rmv": 3333,
            "dth_cnt": 55,
            "pblcn_dt": "2022-01-21 08:00:00"
        }
    ]
}

```

json 데이터 다루기

```

import json
import pprint

jsonOb = json.loads(strJson)
pprint.pprint(jsonOb)

strJsonOb = json.dumps(jsonOb)
print(strJsonOb)

```

파이썬이 지원하는 네트워크, 인터넷 모듈

- “16장 도서 관리 프로그램”에 인터넷 기능 추가하기
 - launcher.py (메뉴 출력하고 사용자 명령 입력 받기)
 - xmlbook.py (xml 관련 기능 구현)
 - internetbook.py (인터넷 관련 기능 구현)
 - book.xml (도서 정보 xml 문서)

OpenAPI를 이용하여 책 정보 가져오기

- **OpenAPI**

- 인터넷 서비스 업체(flickr, google, amazon, naver, daum 등)에서 제공하는 서비스를 외부에서 사용할 수 있도록 하는 API
- 서비스 요청 (HTTP 프로토콜의 GET, POST 메서드 이용)
- REST API 통해서 XML 문서 형태로 결과값 반환.

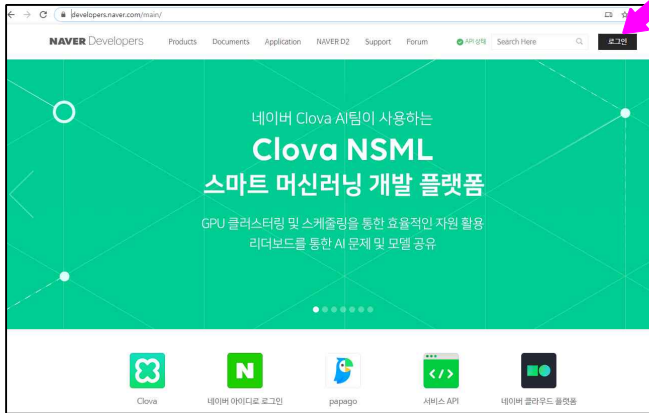
Naver OpenAPI를 이용하여 책 정보 가져오기

- **Naver OpenAPI (<https://developers.naver.com>)**
 - 네이버 OpenAPI 중에서 책 검색 서비스 활용 (책 정보를 가져와서 XML DOM에 추가)
 - 파이썬에서 OpenAPI 이용방법 ([http.client](#), [http.server](#) 2개 모듈 필요)
이 중 읽어오는데는 [http.client](#) 사용!

Naver OpenAPI를 이용하여 책 정보 가져오기

- Naver OpenAPI (<https://developers.naver.com>)

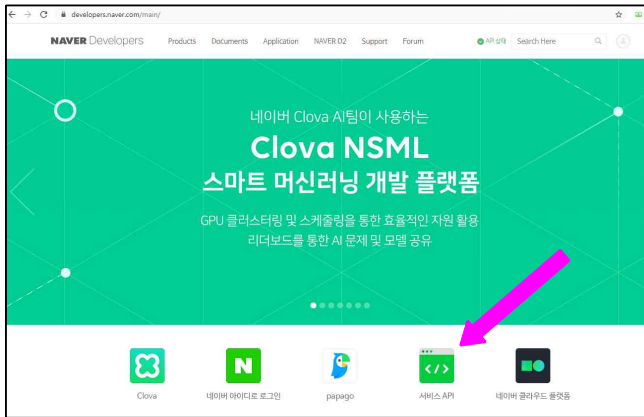
1단계 : 네이버 개발자센터 로그인



Naver OpenAPI를 이용하여 책 정보 가져오기

- Naver OpenAPI (<https://developers.naver.com>)

2단계 : 서비스 둘러보기



Naver OpenAPI를 이용하여 책 정보 가져오기

■ Naver OpenAPI (<https://developers.naver.com>)

3단계 : “오픈 API 이용신청” 선택 후 애플리케이션 정보입력

The screenshot shows the Naver Developers website. The top navigation bar includes 'NAVER Developers', 'Products', 'Documents', 'Application', 'NAVER D2', and 'Support'. A search bar is on the right. The main header is green with the word 'Products' and a sub-header '네이버에서 제공하는 다양한 서비스와 콘텐츠를 소개합니다.'.

Below the header, there's a section 'Products > 서비스 API > 검색'. On the left, a sidebar lists various services: 'API 이용안내', 'Clova', '네이버 아이디로 로그인', '지도', '피미고', '서비스 API', '데이터랩', '검색' (highlighted), '단축URL', '웹과', '네이버 공유하기', '모바일 앱 연동', and '네이버 오픈해인'.

The main content area has a search bar and a list of services. The first service is '검색' (Search). It includes a description: '웹, 뉴스, 블로그 등 분야별 네이버 검색 결과를 웹 서비스 또는 모바일 앱에서 바로 보여 줄 수 있습니다. 또한 "OO역간집"과 같은 지역 검색을 할 수도 있으며, 부가적으로 성인검색어 판별과 한영키 오류 변환 기능을 이용하실 수 있습니다.' Below this, there are icons for '지역 검색' (Local Search) and '검색 부가 기능' (Search Additional Features).

At the bottom, there are two buttons: '오픈 API 이용 신청' (Apply for Open API) and '개발 가이드 보기' (View Development Guide). A pink arrow points to the '오픈 API 이용 신청' button.

Naver OpenAPI를 이용하여 책 정보 가져오기

■ Naver OpenAPI (<https://developers.naver.com>)

약관동의 → 계정정보등록 → 애플리케이션 등록

4단계 : “애플리케이션 이름”, “WEB설정”, “http://tukorea.ac.kr” 입력 후 등록하기

애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

애플리케이션 이름	<input type="text" value="python_kpu"/>	■ 사용 API: “검색”
사용 API	<div>선택하세요. 검색</div>	■ 환경추가: “WEB설정”
환경 추가	<div>환경 추가</div>	■ 검색, 지도 등 사용API 확인하고 ■ 플랫폼(웹서비스 URL, IOS bundle ID, 안드로이드 패키지 이름) 명시
WEB 설정	<div>웹 서비스 URL (최대 10개) <input type="text" value="http://kpu.ac.kr"/></div>	
비로그인 오픈 API 서비스 환경	<div>텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다. • http와 https는 구분하지 않습니다. • www는 빼고 입력해 주세요. 예) http://naver.com • 서버 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)</div>	

Naver OpenAPI를 이용하여 책 정보 가져오기

■ Naver OpenAPI (<https://developers.naver.com>)

5단계 : “Client ID”, “Client Secret” 확인

The screenshot shows the Naver Developers console for an application named 'python_kpu'. The 'API 설정' (API Settings) tab is selected. The 'Client ID' is 'rU3c6jq1ZsLKB8i4As73o' and the 'Client Secret' is 'll6OTwplHT'. A green '개발금' (Development Fee) button is visible next to the Client Secret. The left sidebar shows the application name 'python_kpu' in green. The bottom of the page shows a search bar and the text 'API 호출량/일일허용량' (API Call Count/Daily Allowance) with a value of '0/25000'.

Naver OpenAPI를 이용하여 책 정보 가져오기

■ Naver OpenAPI (<https://developers.naver.com>)

6단계 : “Documents > 서비스API > 검색 > 책” 선택

이 페이지의 1, 2, 3, 6 번 항목을 차례로 확인할 예정.

NAVER Developers Products Documents Application NAVER D2 Support API 상태 Search Here

Documents 네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.

Documents > 서비스 API > 검색 > 책

API 공통 가이드
SDK & Tools
Clova
네이버 아이디로 로그인
지도
파파고
서비스 API

데이터랩
검색
- 블로그
- 뉴스
- 책

검색 > 책

네이버 책 검색 결과를 출력해주는 REST API입니다. 비로그인 오픈 API이므로 GET으로 호출할 때 HTTP Header에 애플리케이션 등록 시 발급받은 Client ID와 Client Secret 값을 같이 전송해 주시면 활용 가능합니다.

오픈 API 이용 신청

1. 준비사항

- 애플리케이션 등록: 네이버 오픈 API로 개발하시려면 먼저 'Application-애플리케이션 등록' 메뉴에서 애플리케이션을 등록하셔야 합니다. [자세한 방법 보기]
- 클라이언트 ID와 secret 확인: '내 애플리케이션'에서 등록된 애플리케이션을 선택하면 Client ID와 Client Secret 값을 확인할 수 있습니다.
- API 권한 설정: '내 애플리케이션'의 'API 권한관리' 탭에서 사용하려는 API가 체크되어 있는지 확인합니다. 체크되어 있지 않을 경우 API에

■ “Documents” → “서비스API” → “검색” 선택 후
■ 왼쪽에서 “책” 선택

Naver OpenAPI를 이용하여 책 정보 가져오기

- Naver OpenAPI (<https://developers.naver.com>)

7단계 : API 기본 정보 확인

2. API 기본 정보

메서드	인증	요청 URL	출력 포맷	설명
GET	-	https://openapi.naver.com/v1/search/book.xml	XML	책 기본 검색
GET	-	https://openapi.naver.com/v1/search/book_adv.xml	XML	책 상세 검색
GET	-	https://openapi.naver.com/v1/search/book.json	JSON	책 기본 검색

Naver OpenAPI를 이용하여 책 정보 가져오기

■ Naver OpenAPI (<https://developers.naver.com>)

8단계 : 요청변수 확인 (query/display/start/d_isbn, ...)

3. 요청 변수

요청 변수명	타입	필수 여부	기본값	설명	비고
query	string	-	-	검색을 원하는 문자열로서 UTF-8로 인코딩한다.	상세검색시 생략가능
display	integer	N	10(기본값), 100(최대)	검색 결과 출력 건수 지정	-
start	integer	N	1(기본값), 1000(최대)	검색 시작 위치로 최대 1000까지 가능	-
sort	string	N	sim(기본값), date	정렬 옵션: sim(유사도순), date(출판일순), count(판매량순)	↑
d_title	string	N	-	책 제목 검색	상세 검색만 해당
d_auth	string	N	-	저자명 검색	상세 검색만 해당
d_cont	string	N	-	목차 검색	상세 검색만 해당
d_isbn	string	N	-	isbn 검색	상세 검색만 해당

Naver OpenAPI : cmd 창에서 url참조

- **naver OpenAPI** (<https://developers.naver.com/docs/search/book>)
 - **9단계 : 호출 예시 확인 -> 적용해 보기**
 - 네이버 OpenAPI 중에서 책 검색 서비스를 활용 (책 정보를 가져와서) 등록하는(XML DOM에 추가) 애플리케이션
 - **Naver Client ID = XXXXXXXXXXXXXXXX**
 - **Naver Client Secret = YYYYYYYYYY**
 - **Love 가 포함된 책 검색해 보기**

6. 예시

호출

```
curl "https://openapi.naver.com/v1/search/book.xml?query=love&display=10&start=1" \
-H "X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 Client ID}" \
-H "X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 Client Secret}"
```

책 기본검색

```
curl "https://openapi.naver.com/v1/search/book.xml?query=love&display=10&start=1" \
-H "X-Naver-Client-Id: XXXXXXXXXXXXXXXX" \
-H "X-Naver-Client-Secret: YYYYYYYYYY" -v
```

- cmd 창에서 수행해 보기!
- 각자 자신의 Client ID, Client Secret 사용하기.

파이썬의 http 관련 모듈

- http.client 모듈을 이용하여 URL 요청
- **http.client.HTTPSConnection** 이용
 - Urllib보다 쓰기가 까다롭지만 더 세세한 부분까지 제어 가능.
- urllib.request.Request 이용

Naver OpenAPI를 이용하여 책 정보 가져오기

openapi_httpclient.py

- HTTPConnection : HTTP 프로토콜을 사용하기 위한 핸들러

HTTPConnection.request(<method>, <url>[, <body>[, [headers>]])

- URL 요청 메서드, <method>= GET, POST, PUT, DELETE

```
import http.client
import urllib.request
from xml.dom.minidom import parseString
```

```
server = "openapi.naver.com"
client_id = "c7HPM052dSRbl7CFtPw"
client_secret = "cpj_ZFsmDi"
conn = http.client.HTTPSConnection(server)
encText = urllib.parse.quote("사랑")
conn.request(      #서버에 GET 요청
```

요청변수 : <https://developers.naver.com/>의
=> "Documents > 서비스API > 검색 > 책"의
=> "3. 요청 변수"들 사용 가능.

```
"GET",
"/v1/search/book.xml?dispaly=10&start=1&query="+encText,
None,
{"X-Naver-Client-Id": client_id, "X-Naver-Client-Secret": client_secret})
res = conn.getresponse()      #openAPI 서버의 답을 받아옴.
if int(res.status) == 200:
    #cLen = res.getheader("Content-Length")      #가져온 데이터 길이
    print(parseString(res.read().decode('utf-8')).toprettyxml())
else:
    print('HTTP request failed: ', res.reason)
```

Naver OpenAPI를 이용하여 책 정보 가져오기

openapi_httpclient.py

- HTTPConnection : HTTP 프로토콜을 사용하기 위한 핸들러

HTTPConnection.request(<method>, <url>[, <body>[, [headers>]])

- URL 요청 메서드, <method>= GET, POST, PUT, DELETE

```
import http.client
import urllib.request
from xml.dom.minidom import parseString
```

```
server = "openapi.naver.com"
client_id = "c7HPM052dSRbl7CFtPw"
client_secret = "cpj_ZFsmDi"
conn = http.client.HTTPSConnection(server)
```

```
encText = urllib.parse.quote("사랑")
```

```
conn.request(      #서버에 GET 요청
    "GET",
```

```
    "/v1/search/book.xml?dispaly=10&start=1&query="+encText,
    None,
```

```
    {"X-Naver-Client-Id": client_id, "X-Naver-Client-Secret": client_secret})
```

```
res = conn.getresponse()      #openAPI 서버의 답을 받아옴.
```

```
if int(res.status) == 200:
```

```
    #cLen = res.getheader("Content-Length")      #가져온 데이터 길이
```

```
    print(parseString(res.read().decode('utf-8')).toprettyxml())
```

```
else:
```

```
    print('HTTP request failed: ', res.reason)
```

“query”는 utf-8로 인코딩하여 전달해야 함.
한글 처리 위해서.

Naver OpenAPI를 이용하여 책 정보 가져오기

openapi_isbntest.py

- 책 isbn 검색 : 사용자가 ISBN 코드를 입력하면 해당 책 정보를 가져와 책 제목만 XML에 저장하는 기능 테스트.

```
import http.client
import urllib.request
from xml.dom.minidom import parseString
```

openapi_httpclient.py 파일을 복사하여
빨간색 부분만 수정하자.

```
server = "openapi.naver.com"
client_id = "c7HPM052dSRrbI7CFtPw"
client_secret = "cpj_ZFsmDi"
conn = http.client.HTTPSConnection(server)
isbn = '0596513984'
conn.request( #서버에 GET 요청
    "GET",
    "/v1/search/book_adv.xml?d_isbn="+isbn,
    None,
    {"X-Naver-Client-Id": client_id, "X-Naver-Client-Secret": client_secret})
res = conn.getresponse() #openAPI 서버의 답을 받아옴.
if int(res.status) == 200:
    #cLen = res.getheader("Content-Length") #가져온 데이터 길이
    print(parseString(res.read().decode('utf-8')).toprettyxml())
else:
    print('HTTP request failed: ' res.reason)
```

d_isbn은
상세검색에서만
제공되므로
book_adv.xml 사용!

“도서관리 프로그램”에 Naver OpenAPI 적용

- **기능 추가** : 사용자가 ISBN을 입력하면 해당 책 정보를 OpenAPI 검색으로 가져와 책 제목만 XML에 저장하는 기능
- 네이버 OpenAPI 출력 결과 XML 문서 : 출력결과 필드 (response filed) <https://developers.naver.com/docs/serviceapi/search/book/book.md#%EC%B1%85> 의 “4. 출력 결과” 부분 참조

필드	값	설명
rss		RSS 리더기 이용하도록 만든 RSS 포맷 컨테이너
channel		검색 결과를 포함하는 컨테이너
title	String	검색결과 문서의 제목. 검색어와 일치하는 부분은 태그로 감싸져 있음

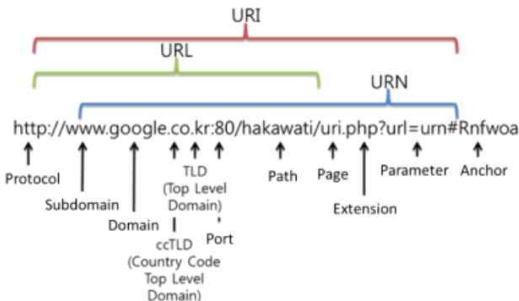
[두산백과] RSS [rich site summary/really simple syndication/RDF site summary]
업데이트가 빈번한 웹사이트의 정보를 사용자에게 보다 쉽게 제공하기 위하여 만들어진 XML 기반의 콘텐츠 배급 포맷

“도서관리 프로그램”에 Naver OpenAPI 적용

- (1) book.py
=> laubcher.py, xmlbook.py, internetbook.py 로 나누기
- (2) 메뉴에 추가
 - printMenu(), launcherFunction() 함수 수정
- (3) 유틸리티 함수 추가
 - HTTPSConnection 서버 연결 함수 : connectOpenAPIServer()
 - URI 생성 함수 : userURIBuilder()
- (4) 서버에 필요한 정보를 URL로 요청하는 함수
 - Get BookDataFromISBN()
- (5) 가져온 XML 문서에서 isbn과 title을 추출하는 함수
 - extractBookData()

URI, URN 구분

- https://inpa.tistory.com/entry/WEB-%F0%9F%8C%90-URL-URI-%EC%B0%A8%EC%9D%B4 참조.



book.py로부터 가져오기 (1/2)

- 16장에서 작성한 book.py 파일을 복사하여 아래 2개 파일로 나누기
 - launcher.py : 메뉴 driven 방식 운영 부분

loopFlag 전역변수

from internetbook import *

printMenu, launcherFunction, QuitBookMgr 함수

메인 프로그램 “run” 부분

- xmlbook.py : xml로 된 책 정보를 다루는 부분
위에 포함되지 않은 나머지 부분들.
- 소스 파일 추가
 - internetbook.py : 인터넷 사용 부분
from xmlbook import *

book.py로부터 가져오기 (2/2)

- BookDoc 전역 변수 정리
 - launcher.py : 사용하지 않도록 삭제하기
 - xmlbook.py 의 LoadXMLFromFile() 함수에서 :
 - (1) 전역 변수 사용을 명시.
global BooksDoc
 - (2) 파일 파싱에 성공했을 때 BookDoc에 등록.
BooksDoc = dom

여기까지 정리한 상태에서
launcher.py 실행 확인!

```
def printMenu():  
    print("\n\nWelcome! Book Manager Program (xml version)")  
    print("====Menu====")  
    print("Load xml: l")  
    print("Print dom to xml: p")  
    print("Quit program: q")  
    print("print Book list: b")  
    print("Add new book: a")  
    print("sEarch Book Title: e")  
    print("Make html: m")  
    print("-----")  
    print("Get book data from isbn: g")  
    print("send mail: i") # 이후에 구현  
    print("sTart Web Service: t") # 이후에 구현  
    print("====Menu====")
```

```
def launcherFunction(menu):
    if menu == 'l':
        LoadXMLFromFile()
        .....
    elif menu == 'g':
        isbn = str(input('input isbn to get :'))
        #isbn = '0596513984'
        ret = getBookDataFromISBN(isbn)
        AddBook(ret)
    elif menu == 'i':          # 이후에 구현
        sendMain()
    elif menu == 't':          # 이후에 구현
        startWebService()
    else:
        print("error : unknow menu key")
```

- **connectOpenAPIServer()** 함수 :

```
from http.client import HTTPSConnection
```

```
conn = None
```

```
server = "openapi.naver.com" # 네이버 OpenAPI 서버
```

```
def connectOpenAPIServer():
```

```
    global conn, server
```

```
    conn = HTTPSConnection(server)
```

```
    conn.set_debuglevel(1)
```

- 길이가 정해지지 않은 사전형 함수인자로 uri 생성

```
def userURIBuilder(uri, **user):  
    str = uri + "?"  
    for key in user.keys():  
        str += key + "=" + user[key] + "&"  
    return str
```

```
def getBookDataFromISBN(isbn):
    global server, conn
    client_id = "XXXXXXXXXXXXXX"
    client_secret = "YYYYYYYYY"
    if conn == None :
        connectOpenAPIServer()          # OpenAPI 접속
        #네이버에서 ISBN에 의한 도서정보 가져올 URL 생성
        uri = userURIBuilder("/v1/search/book_adv.xml", display="1", start="1", d_isbn=isbn)
        conn.request("GET", uri, None, #GET 요청
            {"X-Naver-Client-Id": client_id, "X-Naver-Client-Secret": client_secret})

    req = conn.getresponse()
    print (req.status)
    if int(req.status) == 200 :
        print("Book data downloading complete!")
        return extractBookData(req.read())    #요청이 성공이면 book 정보 추출
    else:
        print ("OpenAPI request has been failed!! please retry")
        return None
```


가져온 XML 문서에서 isbn과 title을 추출

internetbook.py

```
def extractBookData(strXml): #strXml은 OpenAPI 검색 결과 XML 문자열
    from xml.dom.minidom import parseString
    from xml.etree import ElementTree

    tree = ElementTree.fromstring(strXml)

    print (parseString(strXml.decode('utf-8')).toprettyxml())    # 내용 확인용

    # "item" 하위의 "title" 찾기.
    # Note! 그냥 "title"은 찾으면 상단의 "title"도 검색되므로.
    itemElements = tree.iter("item")    # item 엘리먼트 리스트 추출
    for item in itemElements:
        isbn = item.find( " isbn " )    #isbn 검색
        title = item.find( " title " )    #title 검색
        if len(title.text) > 0 :
            # AddBook()에 줄 수 있는 사전형식 반환
            return {"ISBN":isbn.text,"title":title.text}
```

실행 (launcher.py 파일 실행)

=====Menu=====

Load xml: l
Print dom to xml: p
Quit program: q
print Book list: b
Add new book: a
sEarch Book Title: e
Make html: m

Get book data from isbn: g
send mail : i
sTart Web Service: t

=====Menu=====

select menu : **g**
input isbn to get : **0596513984**
200

Book data downloading complete!

b'<?xml version="1.0" encoding="UTF-8"?>\n<channel>\n<title>Search

[키 사용 순서]

l (load xml)

→ b (print book list)

→ g (get book data from isbn)

→ b (print book data)

실행 (launcher.py 파일 실행)

- 'b' 메뉴 이용하여 BooksDoc 에 등록된 것을 확인.

=====Menu=====

Load xml: l
Print dom to xml: p
Quit program: q
print Book list: b
Add new book: a
sEarch Book Title: e
Make html: m

Get book data from isbn: g
send mail : i
sTart Web Service: t

=====Menu=====

select menu : **b**

title= The Very Hungry Caterpillar Pop-Up Book

title= The Shack

title= You Can Negotiate Anything

title= Learning Python (3/E)

[키 사용 순서]

→ l (load xml)

→ b (print book list)

→ g (get book data from isbn)

→ b (print book data)

11주 기획발표 준비

- 발표내용
 - 프로그램 기능 소개 : UI는 실행시켜 보여도 되고 구상한 것을 그림으로 보여도 됨.
 - 사용할 Open API (공공포탈 서비스 이름과 주소 (예: 병원정보서비스 주소), 그 외 사용한 API가 있다면 그의 주소)
 - 개발 일정 (Todo)과 이미 구현된 기능 (Done)
 - Tic-Tac-Toe 게임 시연
- 발표영상
 - 분량 : 5분 이내
 - 제목 : “한국공학-게임-스크립트언어-*{학번1}*-*{학번2}*-기획발표”
- 발표 ppt (프로젝트 문서화)
 - 프로그램 기능 설명, GUI 화면 설명, 개발일정 등 “발표내용”에 나오는 모든 것 포함.
- 프로젝트 진행 시 유의
 - Github Commit 사용자명은 본명으로!
 - Github에 매주 팀원 각각의 commit이 있어야 함.
- 수업 전 제출
 - 유튜브주소 : 공유문서에 등록.
 - 발표 ppt : 과제게시판에 제출 + github에 올려 놓기.