Pharming Certificate

BOB DF 7TH 오태규

1. Option 명령어

```python
def main():
    try:
    # 여기서 입력을 인자를 받는 파라미터는 단일문자일 경우 ':' 긴문자일경우 '='을끝에 붙여주면됨
        opts, args = getopt.getopt(sys.argv[1:],"p:i:",["input=","help"])
    except getopt.GetoptError as err:
        print (str(err))
        help()
        sys.exit(1)
    proxy_option = 0
    cert_path = None
    if opts == [] :
        help()
        sys.exit(1)
    for opt,arg in opts:
        if (opt == '-p'):
            '''
                using selinum download zip file
            '''
            proxy_option = arg
            work.selinumDownload(proxy_option)
            #sys.exit(1)
        elif (opt == '-i' or opt =='--input'):
            cert_path = arg
            '''
                unzip and insert data into database
            '''
            work.search(cert_path)
            dbquery.inputIPDB()
            #sys.exit(1)
        elif ( opt == "-h") or ( opt == "--help"):
            help()
            sys.exit(1)
```

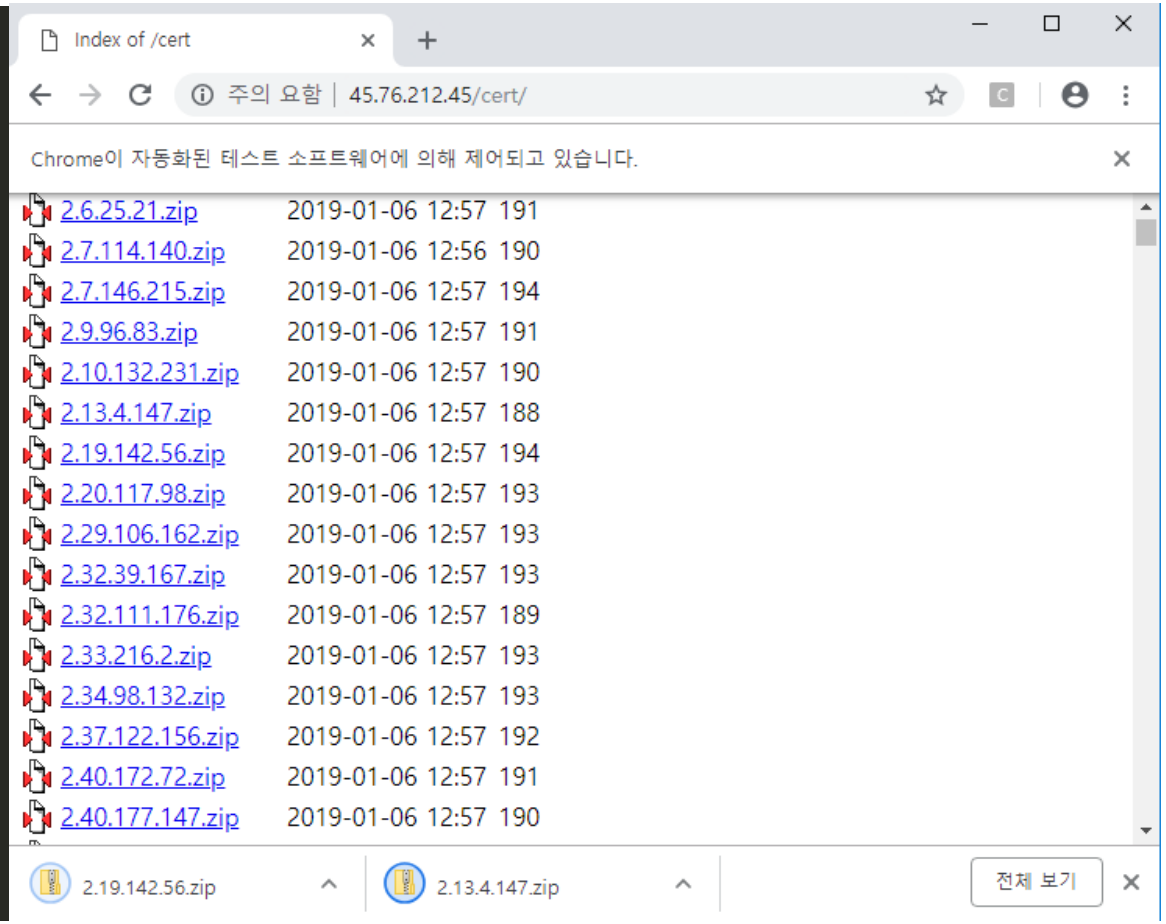
```
C:₩Users₩taegue₩Downloads₩pharming-master₩pharming-master>python

C:₩Users₩taegue₩Downloads₩pharming-master₩pharming-master>python main.py -h
option -h not recognized
print help usage
[-p] is proxy option and download cert
    use proxy 1
    None use proxy 0
[-i][--input] is input cert info insert databse path
[-h][--help] is help option
```

2. selinum download

```python
def selinumDownload(prox_flag):
    #사이트 정보 다운로드
    url = "http://45.76.212.45/cert/"
    ipcheck_url = 'https://httpbin.org/ip'
    i = 4
    while True :
        #Get a proxy from the pool
        if (prox_flag == 1):
            try:
                proxies = get_proxies()
                proxy_pool = cycle(proxies)
                proxy = next(proxy_pool)
                if (i>=37000):
                    break
                response = requests.get(ipcheck_url,proxies={"http": proxy, "https": proxy})
                print(response.json())
                chrome_options = webdriver.ChromeOptions()
                chrome_options.add_argument('--proxy-server=%s' % proxy)
                driver = webdriver.Chrome(options=chrome_options)
                driver.get(url)
                for j in range(i, i+1000):
                    try:

                        download_url = '/html/body/table/tbody/tr[' + str(
                            j) + ']/td[2]/a'
                        driver.find_element_by_xpath(download_url).click()
                        i = i+1
                    except :
                        print("end")
                        break
            except:
                print("Skipping. Connnection error")
```



3. 압축해제 및 DB 파일저장

```
def search(dirname):
    #다운로드 받은 파일 압축 해제 및 DB저장
    sql_insertList =[]
    count = 0
    try:
        filenames = os.listdir(dirname)
        for filename in filenames:
            full_filename = os.path.join(dirname, filename)
            if os.path.isdir(full_filename):
                search(full_filename)
            else:
                ext = os.path.splitext(full_filename)[-1]
                if ext == '.zip':

                    cert = unZip(full_filename)
                    sql_insertList.append(cert)
                    count = count + 1

                    if len(sql_insertList) >= 2000 or count >= len(filenames):
                        # print (sql_insertList)

                        dbquery.inputDB(sql_insertList)
                        sql_insertList = []

        return 1
    except PermissionError:
        print ("permission Denided")
        return 0
```

4. 인증서 정보 파싱

```python
def find_people_info(text):
    #공인인증서 정보 추출
    '''
        return type
        ['name', '83230778832671167227', 'woori', 'personal', 'yessign', 'country']
    '''
    reg_info = '^cn=([가-힣]+)\(\)([0-z]+),ou=([a-zA-Z]+),ou=([a-zA-Z]+),o=([a-zA-Z]+),c=([a-zA-Z]+)'

    cert_value = re.findall(reg_info, text)
    cert_list =[]

    for i in range(0, len(cert_value[0])):
        cert_list.append(cert_value[0][i])
    # print (cert_list)
    #exit(1)
    return cert_list
```

5. DB 데이터 저장

```
def inputDB(sql_list) : #insert log file into mysql database
    conn = pymysql.connect(host='localhost', user='root', password='1234',
                           db='cert_info', charset='utf8')

    curs = conn.cursor()
    sql = """insert into cert(date, name, account_number, bank_info,  country, IP) values (%s, %s, %s, %s, %s,

    for i in range(0, len(sql_list)):

        #date, bank_info, accountnumber, ip, country
        curs.execute(sql, (sql_list[i][7][:-2], sql_list[i][0], sql_list[i][1], sql_list[i][2], sql_list[i][5]
        #curs.execute(sql, ('2018-08-26', 'name', 1, 'kr', 'ip'))
    print("dataBase Insert")
    conn.commit()
```

5. DB 데이터 저장

```
def inputIPDB():
    fp = open('ipinfo.txt', 'r')

    conn = pymysql.connect(host='localhost', user='root', password='1234',
                           db='cert_info', charset='utf8')
    sql = """insert into ip_info(ip_addr, date, country) values (%s, %s, %s)"""
    curs = conn.cursor()
    input_IPdbList = []
    count = 0
    while (True):
        iptime = fp.readline()

        count = count + 1

        if (len(input_IPdbList) >= 3000 or iptime==''):
            for i in range(0, len(input_IPdbList)):
                curs.execute(sql, (input_IPdbList[i][0], input_IPdbList[i][1], input_IPdbList[i][2]))

            input_IPdbList = []
            print ('input db')
            conn.commit()
            if (iptime == ''):
                break
        ipList = retIPList(iptime)
        input_IPdbList.append(ipList)
```

6. Whois API 국가조회

```python
def getCountry(ip_info):
    api_key = '2018083111331754935052'

    url = 'http://whois.kisa.or.kr/openapi/ipascc.jsp?query=' + ip_info + '&key=' + api_key + '&answer=json'
    # url = 'http://whois.kisa.or.kr/openapi/ipascc.jsp?query=211.101.23.45&key=2018083111331754935052&answer
    req = requests.get(url)
    text = req.text

    data = json.loads(text)
    return data["whois"]["countryCode"]
```

seq	ip_addr	date	country
1	2.6.202.40	2018-06-26 17:54:00	FR
2	2.8.29.49	2018-06-26 17:53:00	FR
3	2.8.226.42	2018-06-26 17:55:00	FR
4	2.9.186.198	2018-06-26 17:55:00	FR
5	2.11.166.3	2018-06-26 17:52:00	FR
6	2.11.225.47	2018-06-26 17:55:00	FR
7	2.12.35.126	2018-06-26 17:52:00	FR

count(*)
36834

7. 저장 결과

```
38 • select bank_info, count(*) from cert_info.cert group by bank_info;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

bank_info	count(*)
shinhan	2853
ieonbuk	2723
daegu	2920
gwangju	2891
kookmin	2725
ibk	2734
busan	2823
gyunggnam	2895
nh	2950
woori	2729
citi	2808
ieiu	2879
sc	2904

```
41 • select cert.*, ip_info.country from cert join ip_info on cert.IP = ip_info.ip_addr;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

seq	date	name	account_number	bank_info	country	IP	country
22019	2018-06-26 17:05:00	문선용	17802241101544211721	nh	kr	100.124.22.63	none
22020	2018-06-26 17:05:00	한종철	10083425254784674272	ieiu	kr	100.126.97.213	none
22021	2018-06-26 17:05:00	권선영	83146201554651817386	kookmin	kr	100.127.178.173	none
22022	2018-06-26 17:05:00	채재섭	75831716708730578745	ibk	kr	100.128.39.40	US
22023	2018-06-26 17:05:00	강경희	14665002668163372172	citi	kr	100.129.134.69	US
22024	2018-06-26 17:05:00	이준현	33440171642037282783	ieiu	kr	100.129.202.125	US
22025	2018-06-26 17:05:00	소현	58602747708531058821	sc	kr	100.13.70.208	US
22026	2018-06-26 17:05:00	장규리	35264530673358715657	nh	kr	100.132.116.118	US
22027	2018-06-26 17:05:00	방순영	22251501026652507126	shinhan	kr	100.132.46.186	US
22028	2018-06-26 17:05:00	홍은민	55736562786722606841	citi	kr	100.134.56.220	US
22029	2018-06-26 17:05:00	박연주	81342221408333035376	gwangju	kr	100.137.150.238	US
22030	2018-06-26 17:05:00	백영진	40801544352531165872	ieiu	kr	100.143.174.21	US
22031	2018-06-26 17:05:00	기제성	35437260066563616707	sc	kr	100.144.181.62	US
22032	2018-06-26 17:05:00	조하진	43843220065001278047	ieiu	kr	100.145.94.100	US
22033	2018-06-26 17:05:00	권유정	23872055617550625028	sc	kr	100.152.232.87	US
22034	2018-06-26 17:05:00	탁수지	34436226685881620871	nh	kr	100.156.110.180	US
22035	2018-06-26 17:05:00	강정석	45611876873001518024	ieonbuk	kr	100.156.249.166	US