

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

-최종 발표-

2018184011 서길재 / 2018184036 최재준

# 목차

1. 프로그램 기능 소개	03
2. 사용한 Open API	04
3. 간략한 소스 설명	07
4. 예외 처리	11

5. 개발일정	12
6. Commit 내역	13
7. 프로그램 시현	14

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 프로그램 기능 소개

지역화폐 가입되어 있는 558,491개의 가맹점 중

- 사용자 원하는 가맹점 정보 검색
- 검색한 가맹점 정보 출력
- 가맹점 정보 메일 전송
- 가맹점 위치 지도에서 출력

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 사용한 Open API



**지역화폐 가맹점 현황**

경기도 각 시군의 지역화폐 가맹점 현황입니다.

총평점 ★★★★★

내용	<p>경기도 각 시군의 지역화폐 2021년 4분기 가맹점 현황입니다.</p> <p>지역화폐 가맹점은 재난기본소득 사용처와 일부 차이가 있을 수 있습니다.</p> <ul style="list-style-type: none"><li>- 경기지역화폐가 아닌 타 신용카드로 재난기본소득을 이용하는 경우 각 신용카드 홈페이지에서 사용처를 확인해주시기 바랍니다.</li><li>- 백화점, 대형마트, 기업형 슈퍼마켓(SSM), 프랜차이즈 직영점 및 유흥 사행업소는 사용이 제한되며, 연 매출 10억 이하인 소상공인 점포에서만 사용 가능합니다.</li><li>※ 소유 점포가 여러 개일 경우, 합산 매출 10억 이하인 소상공인 점포에서만 사용 가능</li></ul> <p>지역화폐 가맹점 및 재난기본소득과 관련된 자세한 설명들은 아래의 URL들을 참고해주시기 바랍니다.</p> <ul style="list-style-type: none"><li>※ 경기지역화폐 가맹점 : <a href="http://www.gmoney.or.kr/PageLink.do?link=forward:/PageContent.do&amp;tempParam1=&amp;menuNo=040000&amp;subMenuNo=040500&amp;thirdMenuNo=">http://www.gmoney.or.kr/PageLink.do?link=forward:/PageContent.do&amp;tempParam1=&amp;menuNo=040000&amp;subMenuNo=040500&amp;thirdMenuNo=</a></li><li>※ 재난기본소득(경기도) : <a href="https://www.gg.go.kr/contents/contents.do?cildx=1249&amp;menuId=3028">https://www.gg.go.kr/contents/contents.do?cildx=1249&amp;menuId=3028</a></li><li>※ 재난기본소득(FAQ) : <a href="https://basicincome.gg.go.kr/faq">https://basicincome.gg.go.kr/faq</a></li><li>※※※ 재난기본소득 온라인 상담(Q&amp;A) : <a href="https://advice.gg.go.kr/main.do;jsessionid=8C45536F277D94897BF20457AFFD35AF">https://advice.gg.go.kr/main.do;jsessionid=8C45536F277D94897BF20457AFFD35AF</a></li></ul>
----	---

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 사용한 Open API

출력값		
No	출력명	출력 설명
1	SIGUN_NM	시군명
2	CMPNM_NM	상호명
3	INDUTYPE_NM	업종명(종목명)
4	REFINE_ROADNM_ADDR	소재지도로명주소
5	REFINE_LOTNO_ADDR	소재지지번주소
6	DATA_STD_DE	데이터기준일자
7	ZIP_CD	우편번호
8	LAT	위도
9	LOGT	경도
10	BIZREGNO	사업자등록번호
11	INDUTYPE_CD	업종코드
12	FRCS_NO	가맹점번호
13	LIVE_YN	생존여부

```

<row>
  <CMPNM_NM>꼬까꼬까</CMPNM_NM>
  <INDUTYPE_NM>소매업</INDUTYPE_NM>
  <DATA_STD_DE>2021-12-31</DATA_STD_DE>
  <REFINE_LOTNO_ADDR>경기도 성남시 중원구 박석로 72</REFINE_LOTNO_ADDR>
  <REFINE_ROADNM_ADDR>경기도 성남시 중원구 상대원동 860-1번지</REFINE_ROADNM_ADDR>
  <SIGUN_NM>성남시</SIGUN_NM>
  <LIVE_YN>Y</LIVE_YN>
  <LOGT/>
  <LAT/>
  <BIZREGNO>5312601181</BIZREGNO>
  <INDUTYPE_CD>N</INDUTYPE_CD>
  <FRCS_NO/>
  <ZIP_CD/>
</row>

```

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 사용한 Open API

출력값		
No	출력명	출력 설명
1	SIGUN_NM	시군명
2	CMPNM_NM	상호명
3	INDUTYPE_NM	업종명(종목명)
4	REFINE_ROADNM_ADDR	소재지 도로명주소
5	REFINE_LOTNO_ADDR	소재지 지번 주소

시(군)명 / 상호명 / 업종명을 입력 받아  
해당 업체 정보 출력

소재지 도로명주소를 이용하여 지도 출력

시(군)명 / 상호명 / 업종명 /  
소재지 도로명 주소 / 소재지 지번 주소를  
이용하여 메일 전송 및 텔레그램 연동

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 간략한 소스 설명

```
def getData(Name):
    global conn, checkEnd
    if conn == None:
        connectOpenAPIServer()
    req = []
    for i in range(559):
        if checkEnd == True:
            checkEnd = False
            return
        count = 0
        uri = "/RegionMnyFacItStus?KEY=8608449b435244de807e2eb607f8e793&pIndex=" + str(i + 1) + "&pSize=1000&SIGUN_NM=" + quote(Name)
        conn.request("GET", uri)
        req.append(conn.getresponse())
        if int(req[i].status) == 200 :
            SaveData(req[i].read())
```

getData

: OpenApi에서 데이터를 가져오는 함수

```
def SaveData(strXml):
    from xml.dom.minidom import parseString
    from xml.etree import ElementTree
    tree = ElementTree.fromstring(strXml)
    global ListMaket, checkEnd
    itemElements = tree.iter("row")
    Head = tree.iter("RESULT")
    checkIN = 0

    for head in Head:
        if head.find("CODE").text != "INFO-000":
            checkEnd = True
            return

    for row in itemElements:
        if row.find("LIVE_YN").text == 'Y':
            CMPNM_NM = row.find("CMPNM_NM")
            INDUTYPE_NM = row.find("INDUTYPE_NM")
            REFINES_LOTNO_ADDR = row.find("REFINE_LOTNO_ADDR")
            REFINES_ROADNM_ADDR = row.find("REFINE_ROADNM_ADDR")
            SIGUN_NM = row.find("SIGUN_NM")
            if CMPNM_NM.text == None:
                break
            ListMaket.append({'Name' : CMPNM_NM.text + ' ',
                              'category' : INDUTYPE_NM.text, 'address_01' : REFINES_ROADNM_ADDR.text,
                              'address_02' : REFINES_LOTNO_ADDR.text, 'local' : SIGUN_NM.text})
```

SaveData

: 가져온 데이터를 저장하는 함수

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 간략한 소스 설명

```
def setmail():
    global new_m, INPUT_MAIL_WIDGET, rightListBox

    if rightListBox.size() == 0:
        messagebox.showinfo(title='잘못된 명령', message='전송할 메시지가 존재하지 않습니다.')
        return

    new_m = Toplevel()
    new_m.geometry("200x100+750+450")

    INPUT_MAIL_Text = Label(new_m, text='수신하실 메일 주소를 입력하세요', width=10, height=2)
    INPUT_MAIL_Text.pack(side="top", padx=5, expand=True, fill='x')
    INPUT_MAIL_WIDGET = Text(new_m, width=10, height=2)
    INPUT_MAIL_WIDGET.pack(side="top", padx=5, expand=True, fill='x')
    SendButton = Button(new_m, text='메일 전송', command=sendMail)
    SendButton.pack(side="right", padx=10, expand=True)

def setsend(fromAddr, toAddr, msg):
    import smtplib
    s = smtplib.SMTP("smtp.gmail.com", 587)
    s.starttls()
    s.login('sgj9946@gmail.com', 'xahqeslixbobxmww')
    s.sendmail(fromAddr, [toAddr], msg.as_string())
    s.close()

def sendMail():
    global content, INPUT_MAIL_WIDGET, new_m

    receiver = INPUT_MAIL_WIDGET.get("1.0", "end")
    msg = MIMEText('\n'.join(content), _charset="utf8")
    msg['Subject'] = "가맹점 정보"
    setsend('sgj9946@naver.com', receiver, msg)
    content.clear()
    INPUT_MAIL_WIDGET = None
    new_m.destroy()
```

```
def OpenPage():
    import webbrowser
    url = "https://www.gmoney.or.kr/base/main/view"
    webbrowser.open(url)
```

```
def printmap():
    global new
    global currentData

    try:
        new = Toplevel()
        if currentData != {}:
            tempaddress = currentData["address"].split("(")
            tempaddress2 = tempaddress[0].split(",")
            map_widget = tkintermapview.TkinterMapView(new, width=800, height=500, corner_radius=0)
            map_widget.pack()
            marker_1 = map_widget.set_address(tempaddress2[0], marker=True)
            result = currentData["Name"].replace(" ", "")
            marker_1.set_text(result)
            map_widget.set_zoom(19)
    except:
        new.destroy()
        messagebox.showinfo(title='알림', message='해당 지역은 지도 출력이 불가능합니다.')
```

setmail, setsend, sendmail : 메일 전송 함수  
printmap : 지도 출력 함수  
OpenPage : 경기도 지역화폐 홈페이지로  
이동하는 함수



# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 간략한 소스 설명

```
def onSearch():
    global SIGUN_NM_Combo, INDUTYPE_NM_Combo, INPUT_CMPNM_NM, leftListBox, rightListBox, ListMaket, currentData, alreadyCallRegion
    leftListBox.delete(0, leftListBox.size())
    rightListBox.delete(0, rightListBox.size())
    currentData = {}
    num = 0
    if SIGUN_NM_Combo.get() == "시/군":
        messagebox.showinfo(title='알림', message='지역을 선택해주세요.')
        return

    if not ListMaket:
        getData(SIGUN_NM_Combo.get())
    else:
        if SIGUN_NM_Combo.get() != ListMaket[0]['local']:
            ListMaket.clear()
            getData(SIGUN_NM_Combo.get())

    for i in ListMaket:
        if INPUT_CMPNM_NM.get() != '':
            if INPUT_CMPNM_NM.get() in str(i['Name']):
                _text = str(i['Name']) + "|" + str(i['local']) + "|" + str(i['category']) + "|" + str(i['address_01']) + "|" + str(i['address_02'])
                leftListBox.insert(num, _text)

        else:
            if str(INDUTYPE_NM_Combo.get()) != "모두표시" and str(INDUTYPE_NM_Combo.get()) != '업종분류':
                if str(i['category']) in classification[INDUTYPE_NM_Combo.get()]:
                    _text = str(i['Name']) + "|" + str(i['local']) + "|" + str(i['category']) + "|" + str(i['address_01']) + "|" + str(i['address_02'])
                    leftListBox.insert(num, _text)
            else:
                _text = str(i['Name']) + "|" + str(i['local']) + "|" + str(i['category']) + "|" + str(i['address_01']) + "|" + str(i['address_02'])
                leftListBox.insert(num, _text)

    leftListBox.insert(0, "")
    leftListBox.insert(0, spam.count(str(leftListBox.size() - 1)))

    if INPUT_CMPNM_NM.get() != '':
        if leftListBox.size() == 0:
            messagebox.showinfo(title='알림', message='입력하신 업체는 지역화폐 사용이 불가능하거나 해당 시/군에 존재하지 않습니다.')
```

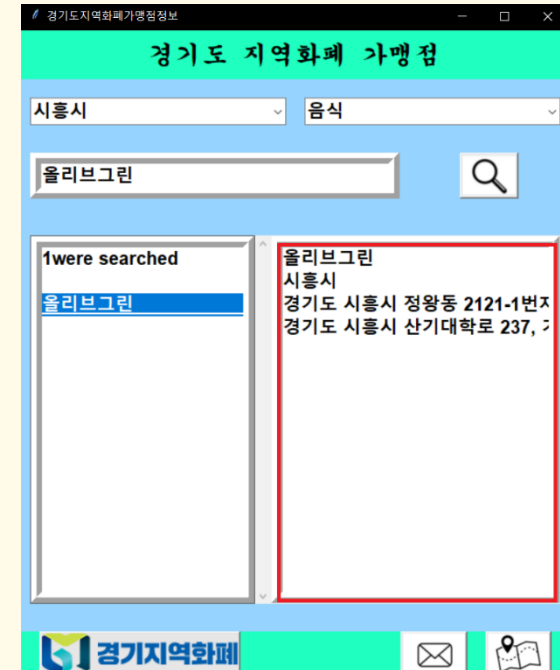


onSearch : 검색한 정보에 대하여 해당하는 데이터들의 업종명 출력

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 간략한 소스 설명

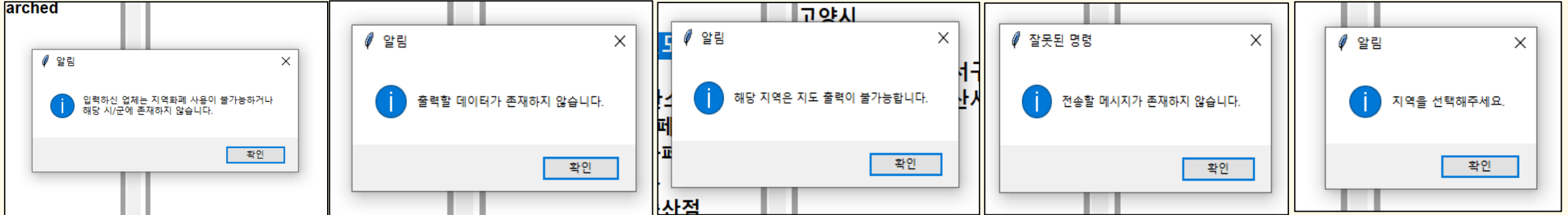
```
def event_for_listbox(event):  
    global rightListBox, content, currentData  
  
    temp = []  
    selection = event.widget.curselection()  
    if selection:  
        currentData = {}  
        index = selection[0]  
        data = event.widget.get(index)  
        temp = data.split("/")  
        if len(temp) == 1:  
            return  
        content.clear()  
        rightListBox.delete(0, rightListBox.size())  
        for i in range(0, 5):  
            if temp[i] != "None":  
                rightListBox.insert(i, temp[i])  
                content.append(temp[i])  
        currentData = {'Name' : str(temp[0]), 'address' : str(temp[3])}
```



event\_for\_listbox  
: 선택한 업체의 정보를 출력하는 함수

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 예외 처리



1. 검색한 업체가 없을 시, 해당 업체가 없다고 알려준다.
2. 지도를 선택하였을 때, 선택된 데이터가 없으면 출력할 데이터가 없다고 알려준다.
3. 해당 지역이 tkmapview로 출력이 불가능한 지역이면 해당 지역은 출력이 불가능하다고 알려준다.
4. 메일을 눌렀을 때, 선택된 데이터가 없으면 전송할 데이터가 없다고 알려준다.
5. 지역을 선택하지 않고 검색 버튼을 클릭하면, 지역을 선택해야 한다고 알려준다.

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## 개발 일정

일정	개발 계획	실제 개발 과정
1~2주차	애플리케이션 기획	<ul style="list-style-type: none"><li>• OpenAPI 선정 및 UI 기획</li></ul>
3주차	XML 파일 기반 기본 기능 구현	<ul style="list-style-type: none"><li>• GUI 제작</li><li>• OpenAPI를 이용하여 데이터를 저장하는 기능 구현</li><li>• 받아온 데이터를 활용하여 검색 기능 구현</li></ul>
4주차	Open API를 이용하여 데이터 가져오기	<ul style="list-style-type: none"><li>• 선택한 데이터에 대한 이메일 전송</li><li>• 선택한 가맹점 위치를 지도에 출력하는 기능 구현</li><li>• OpenAPI 호출 key 변경 및 key 변경에 따른 검색 방식 변경</li></ul>
5주차	C/C++ 연동	<ul style="list-style-type: none"><li>• 기존 검색 기능 개선 및 업체 분류 세분화</li><li>• 사용자 편의를 위한 알림 팝업 생성</li><li>• C/C++ 연동 및 배포본을 만들기 위한 파일 제작</li></ul>
6주차	Telegram Bot 연동 및 배포본 제작	<ul style="list-style-type: none"><li>• Telegram Bot 연동</li><li>• C/C++ 연동을 활용한 함수 작성</li><li>• OpenAPI 변경으로 인하여 발생한 오류 처리</li></ul>

# 경기도 지역화폐 가맹점 정보 조회 애플리케이션

## Commit 내역

🕒 30 commits

<div>Commits on Jun 12, 2022</div> <div>C/C++ 연동 SeoGJ committed yesterday</div>	<div>Commits on May 28, 2022</div> <div>발표자료 및 오류 수정 OmarChoi committed 16 days ago</div> <div>SendMail_GetInput_ReceiverMail SeoGJ committed 16 days ago</div>	<div>Commits on May 16, 2022</div> <div>기획발표 OmarChoi committed 28 days ago</div>
<div>Commits on Jun 7, 2022</div> <div>최종발표 자료 SeoGJ committed 6 days ago</div> <div>텔레그램 연동 OmarChoi committed 6 days ago</div>	<div>Commits on May 26, 2022</div> <div>OpenAPI_CallSystem_Modifying OmarChoi committed 19 days ago</div>	<div>Commits on May 15, 2022</div> <div>Delete 구도잡기.pdf SeoGJ committed on 15 May</div> <div>발표자료 OmarChoi committed on 15 May</div> <div>Delete Project SeoGJ committed on 15 May</div> <div>Create Project SeoGJ committed on 15 May</div>
<div>Commits on Jun 4, 2022</div> <div>C/C++ 연동 SeoGJ committed 9 days ago</div> <div>지역화폐 가맹점 분류 세분화 및 검색결과 존재하지 않을 시 알림창 생성 OmarChoi committed 10 days ago</div> <div>지역화폐 분류 세분화 및 검색결과 존재하지 않을 시 알림창 생성 OmarChoi committed 10 days ago</div>	<div>Commits on May 25, 2022</div> <div>ThirdWeek OmarChoi committed 20 days ago</div>	<div>Commits on May 13, 2022</div> <div>stone.py OmarChoi committed on 13 May</div> <div>Add files via upload SeoGJ committed on 13 May</div>
<div>Commits on Jun 1, 2022</div> <div>Merge branch 'main' of <a href="https://github.com/OmarChoi/scriptlang">https://github.com/OmarChoi/scriptlang</a> OmarChoi committed 13 days ago</div> <div>yahtzee 게임 제작 OmarChoi committed 13 days ago</div> <div>Add Scrollbar SeoGJ committed 13 days ago</div> <div>시/군을 바꾸지 않고 검색했을 때, 검색방법 개선 및 이미지 폴더 생성 OmarChoi committed 13 days ago</div>	<div>Commits on May 24, 2022</div> <div>SecondWeek SeoGJ committed 20 days ago</div>	<div>Commits on May 8, 2022</div> <div>구도잡기 OmarChoi committed on 8 May</div> <div>구도잡기 OmarChoi committed on 8 May</div> <div>tic-tac-toe OmarChoi committed on 8 May</div>
	<div>Commits on May 22, 2022</div> <div>SecondWeek SeoGJ committed 22 days ago</div> <div>SecondWeek OmarChoi committed 23 days ago</div>	
	<div>Commits on May 17, 2022</div> <div>hangman OmarChoi committed 27 days ago</div>	

2018184011 서길재  
Commit Name - SeoGJ

2018184036 최재준  
Commit Name - OmarChoi

# 프로그램 시현

감사합니다