Python
一條龍專案

待售物件分析

Branden Darren Howard Jeff

Get Started

爬蟲動機
Crawler Motive

主要目標

- 整合開放資料和房仲網
- 幫助消費者在購屋時，找出議價空間

購買房子對大多數人而言，可能是一生中，最大筆的支出交易。如何選擇價格合理且符合需求的房子，是一個重要課題。以往購買房屋的資訊來源， 價格資訊單方面掌握在房仲業者手上。

時程分工

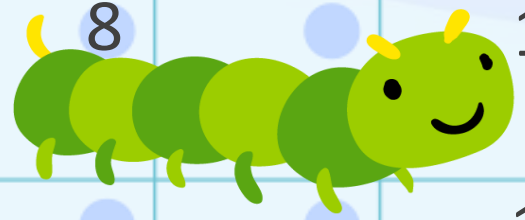
MAY

				6 題目更改		
9			12 			

時程分工

JUNE

6 建立資料庫	7 上傳資料庫	8		10	11 Git hub	
13 視覺化				17 API		
20 Tableau API	21 簡報製作	22 簡報製作		24 Big Day		



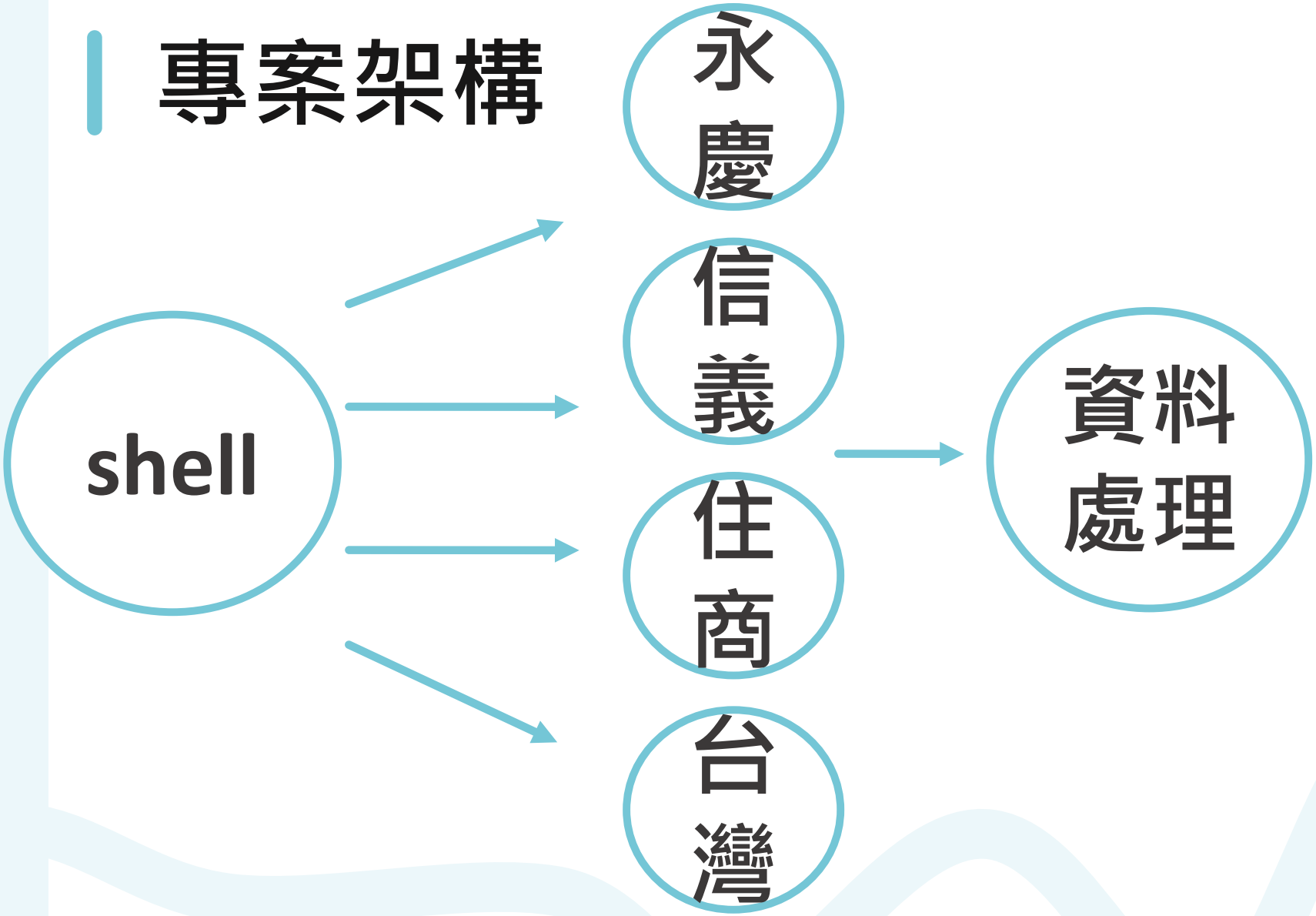
大綱

1. 專案架構
2. 資料蒐集
3. 視覺化分析
4. 困境與回顧
5. 未來工作

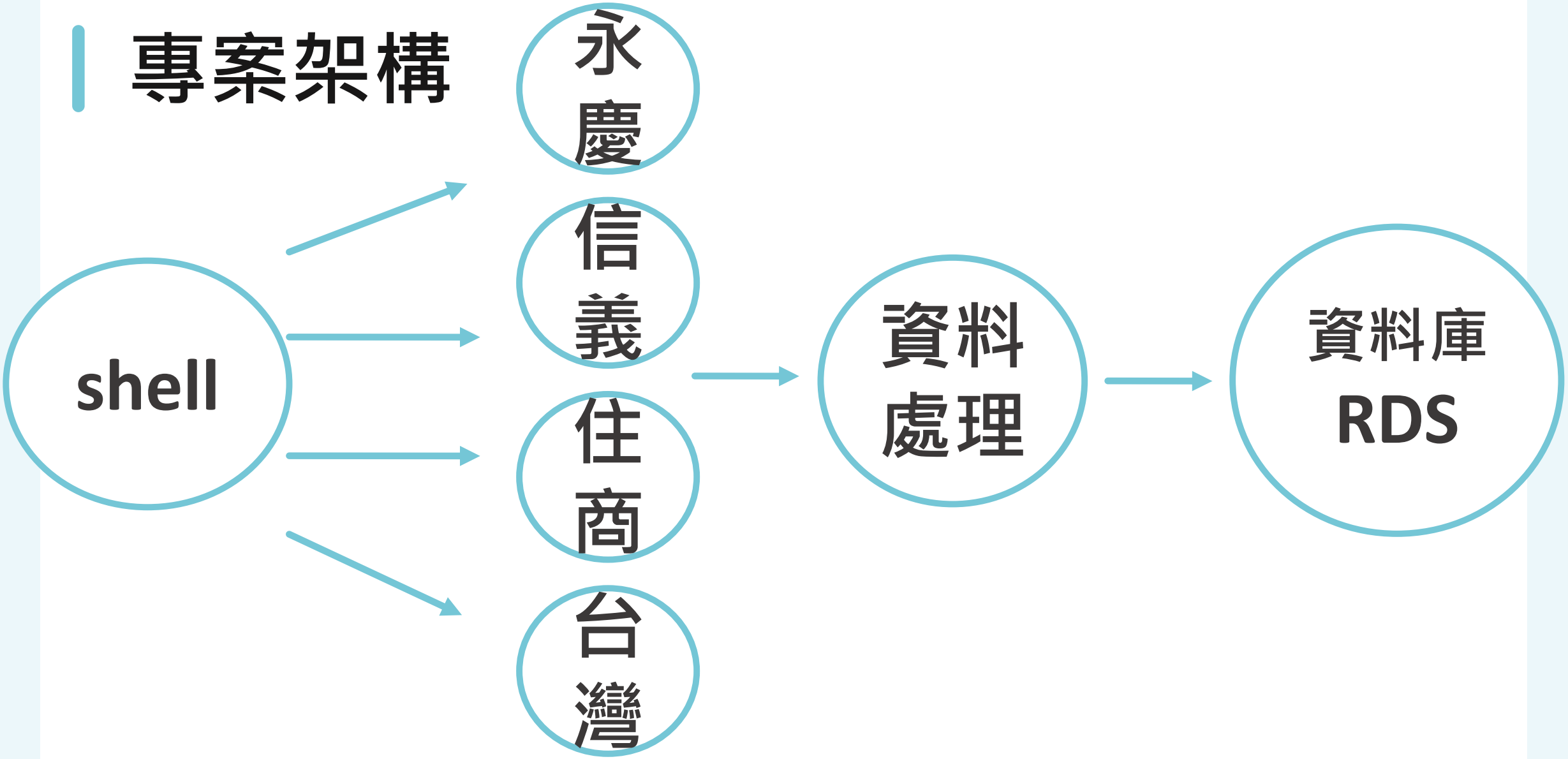
專案架構

專案架構

專案架構



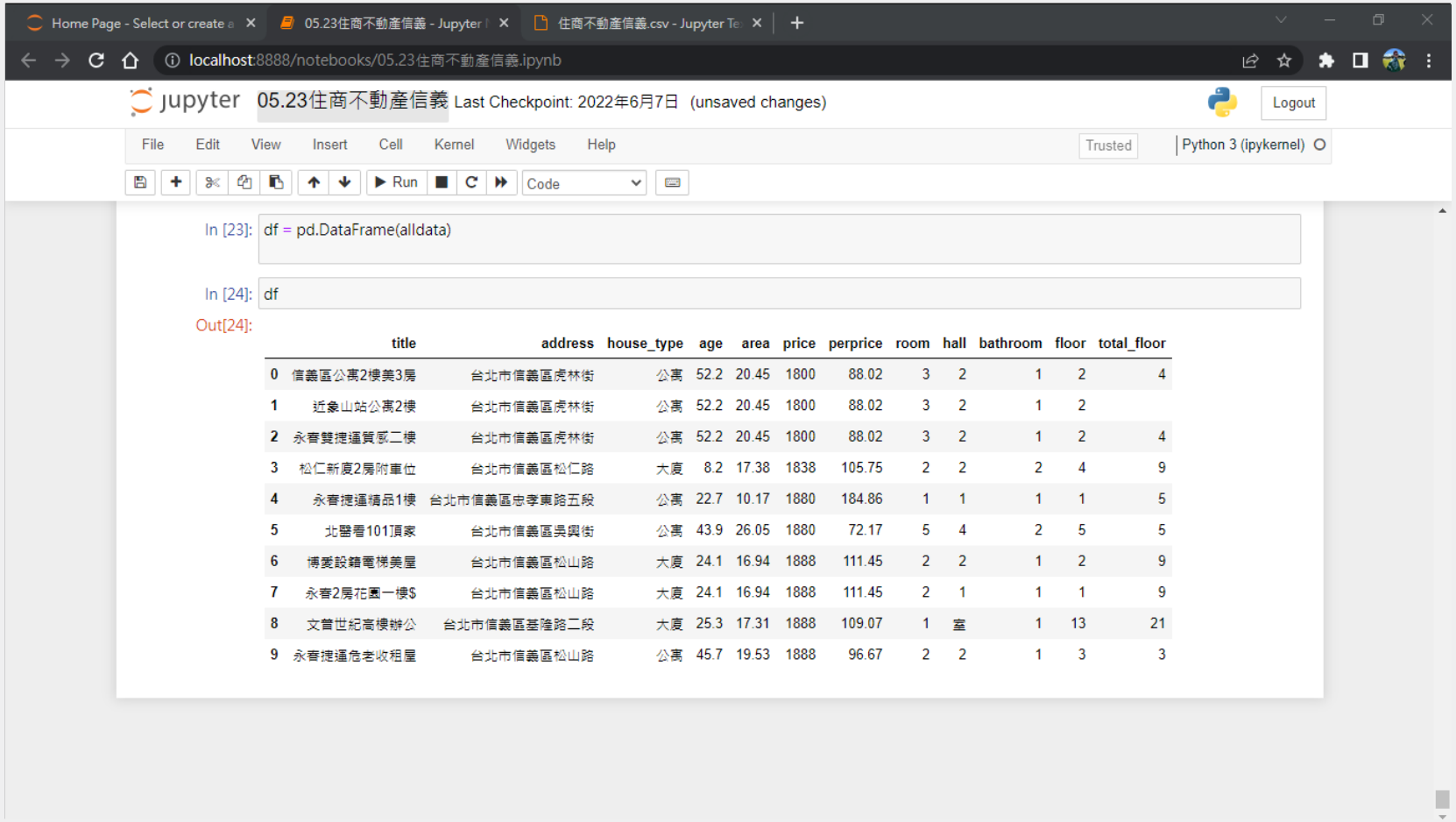
專案架構



資料蒐集

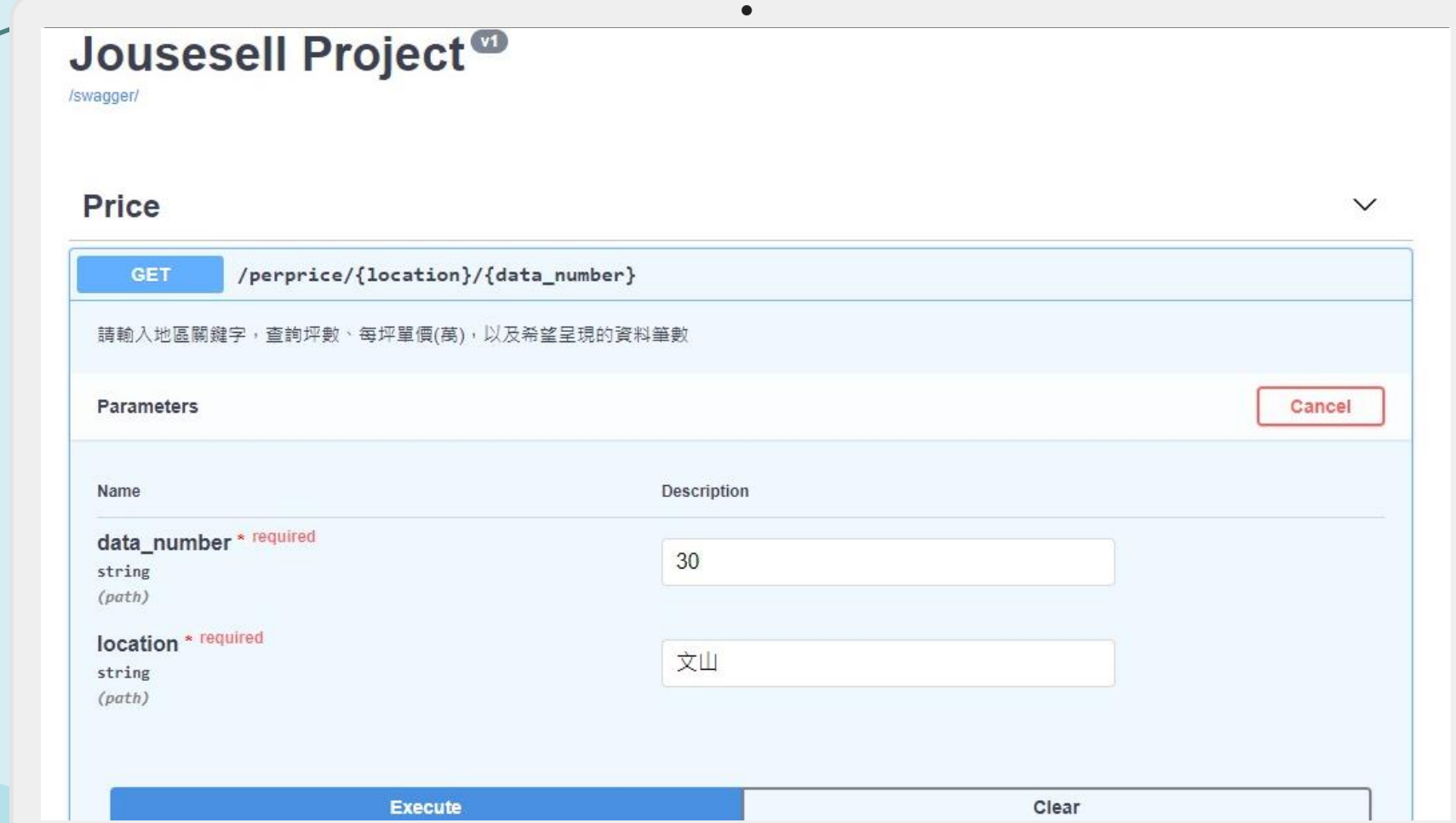
物件名	地址	房屋型態	屋齡
坪數	售價	每坪單價	房
廳	衛	樓層	總樓層數

資料格式

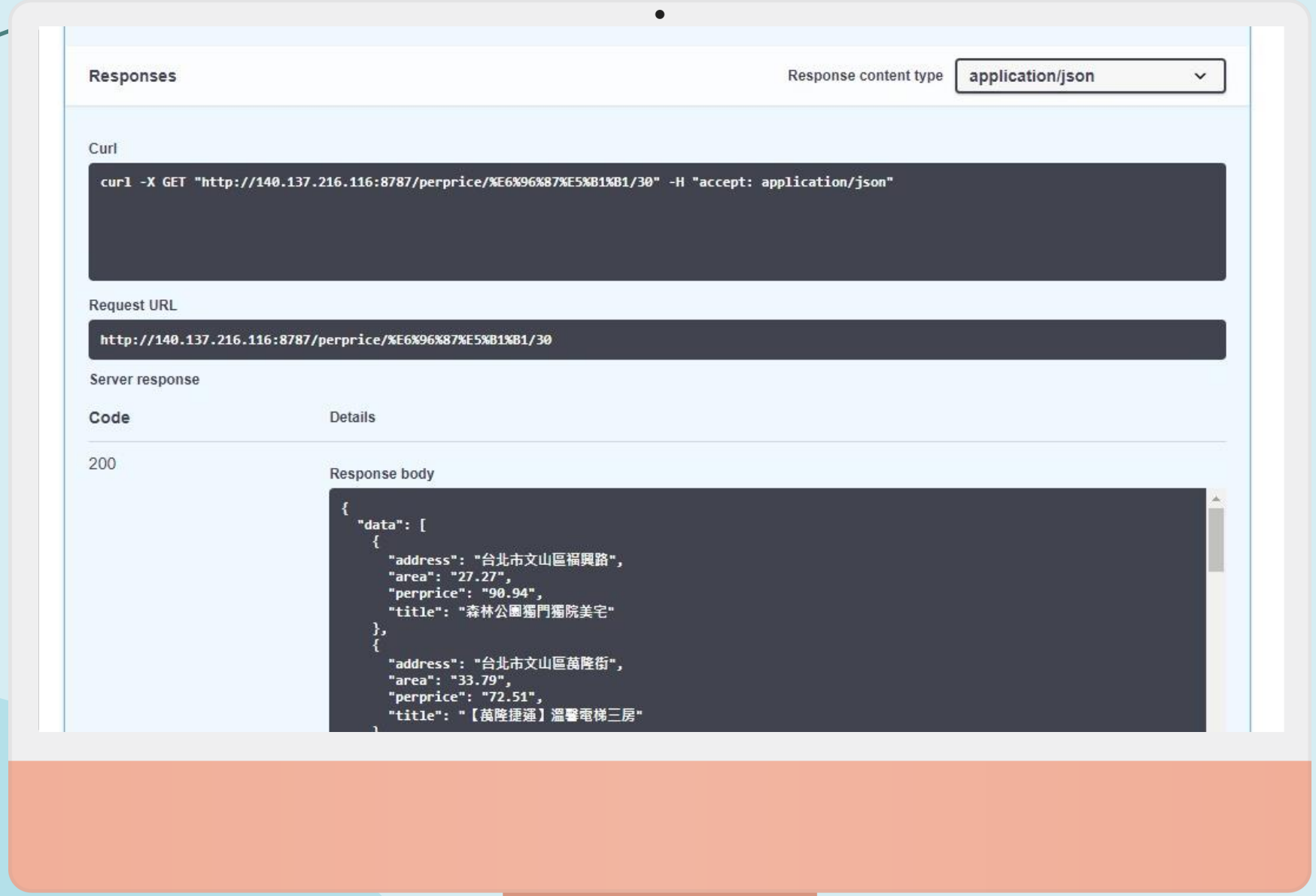


	title	address	house_type	age	area	price	perprice	room	hall	bathroom	floor	total_floor
0	信義區公寓2樓美3房	台北市信義區虎林街	公寓	52.2	20.45	1800	88.02	3	2	1	2	4
1	近象山站公寓2樓	台北市信義區虎林街	公寓	52.2	20.45	1800	88.02	3	2	1	2	
2	永春雙捷邁質感二樓	台北市信義區虎林街	公寓	52.2	20.45	1800	88.02	3	2	1	2	4
3	松仁新廈2房附車位	台北市信義區松仁路	大廈	8.2	17.38	1838	105.75	2	2	2	4	9
4	永春捷運精品1樓	台北市信義區忠孝東路五段	公寓	22.7	10.17	1880	184.86	1	1	1	1	5
5	北醫看101頂家	台北市信義區吳興街	公寓	43.9	26.05	1880	72.17	5	4	2	5	5
6	博愛設籍電梯美屋	台北市信義區松山路	大廈	24.1	16.94	1888	111.45	2	2	1	2	9
7	永春2房花園一樓$	台北市信義區松山路	大廈	24.1	16.94	1888	111.45	2	1	1	1	9
8	文普世紀高樓辦公	台北市信義區基隆路二段	大廈	25.3	17.31	1888	109.07	1	室	1	13	21
9	永春捷運危老收租屋	台北市信義區松山路	公寓	45.7	19.53	1888	96.67	2	2	1	3	3

API


Jousesell Project
v1
/swagger/
Price
GET
/perprice/{location}/{data_number}
請輸入地區關鍵字，查詢坪數、每坪單價(萬)，以及希望呈現的資料筆數
Parameters
Cancel
Name
Description
data_number * required
string
(path)
30
location * required
string
(path)
文山
Execute
Clear

API


Responses
Response content type
application/json
Curl
curl -X GET "http://140.137.216.116:8787/perprice/%E6%96%87%E5%B1%B1/30" -H "accept: application/json"
Request URL
http://140.137.216.116:8787/perprice/%E6%96%87%E5%B1%B1/30
Server response
Code
Details
200
Response body
{
"data": [
{
"address": "台北市文山區福興路",
"area": "27.27",
"perprice": "90.94",
"title": "森林公園獨門獨院美宅"
},
{
"address": "台北市文山區萬隆街",
"area": "33.79",
"perprice": "72.51",
"title": "【萬隆捷運】溫馨電梯三房"

分析

Analysis

Xinyi +

Daan +

Wenshan +

視覺化分析 × 各房型

下降

透天是大樓 7.4 %

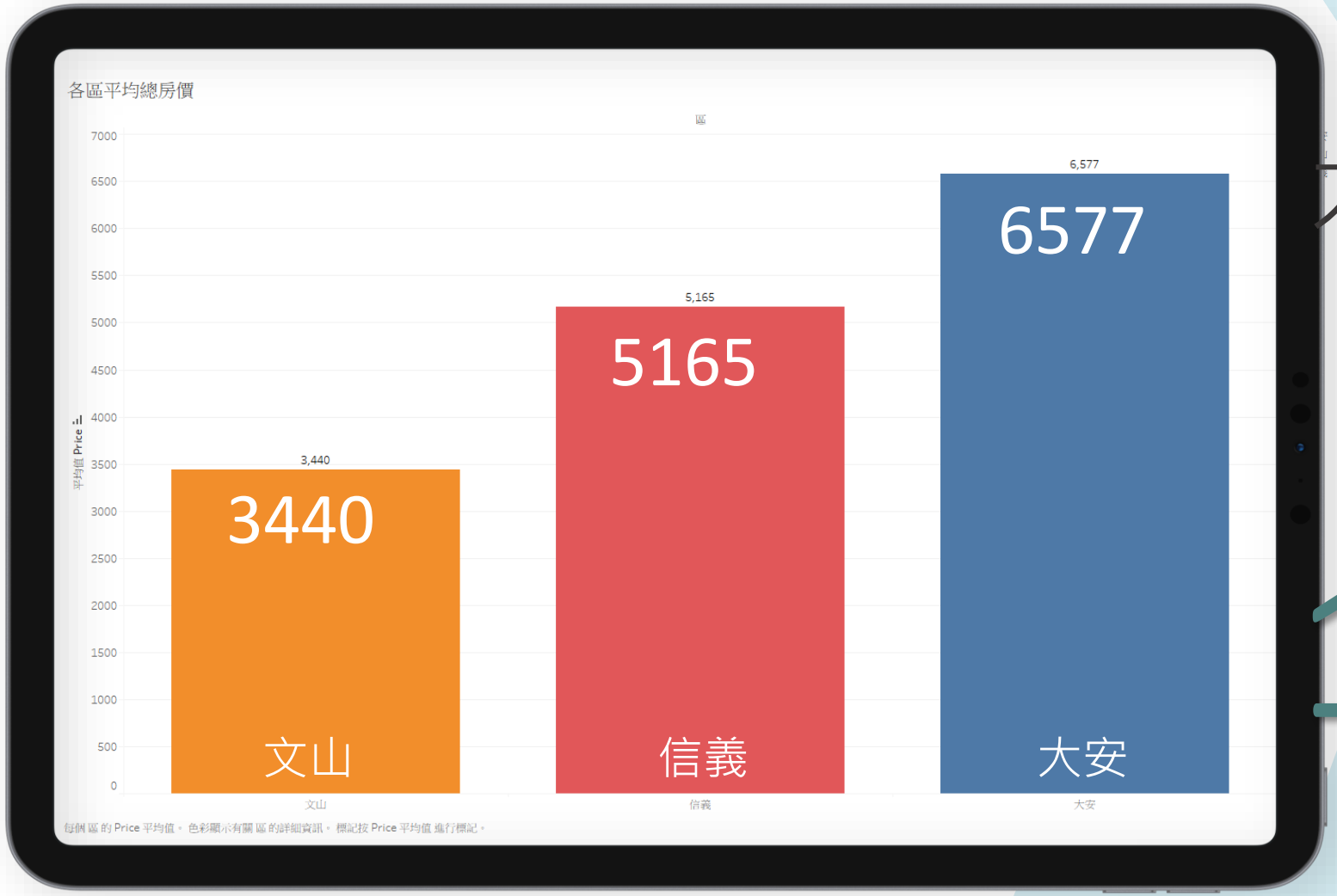
公寓的

華廈 8.5 %

1.46X

公寓 5.9 %

視覺化分析 × 各區/總房價

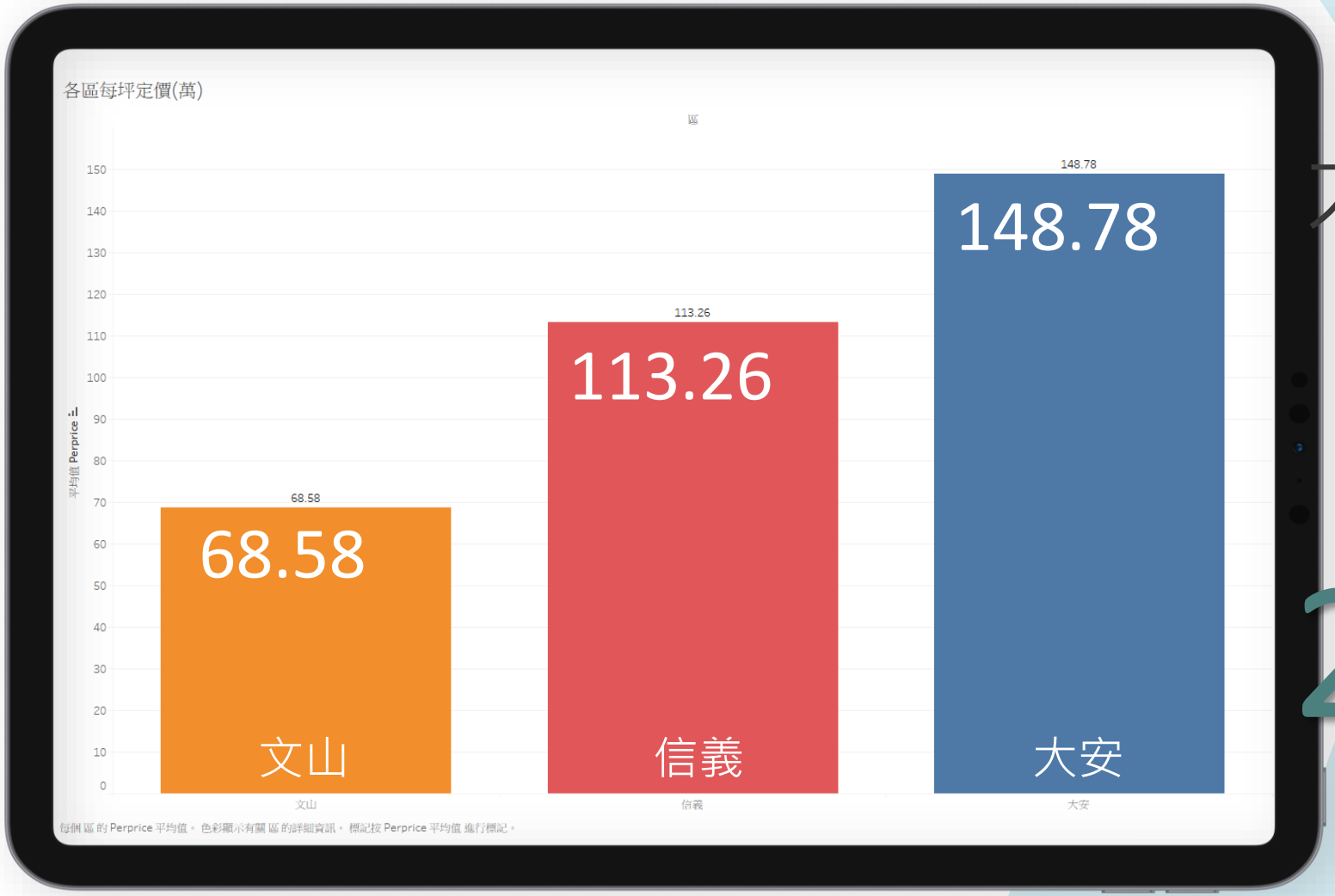


大安是

文山的

1.91 X

視覺化分析 × 各區/每坪

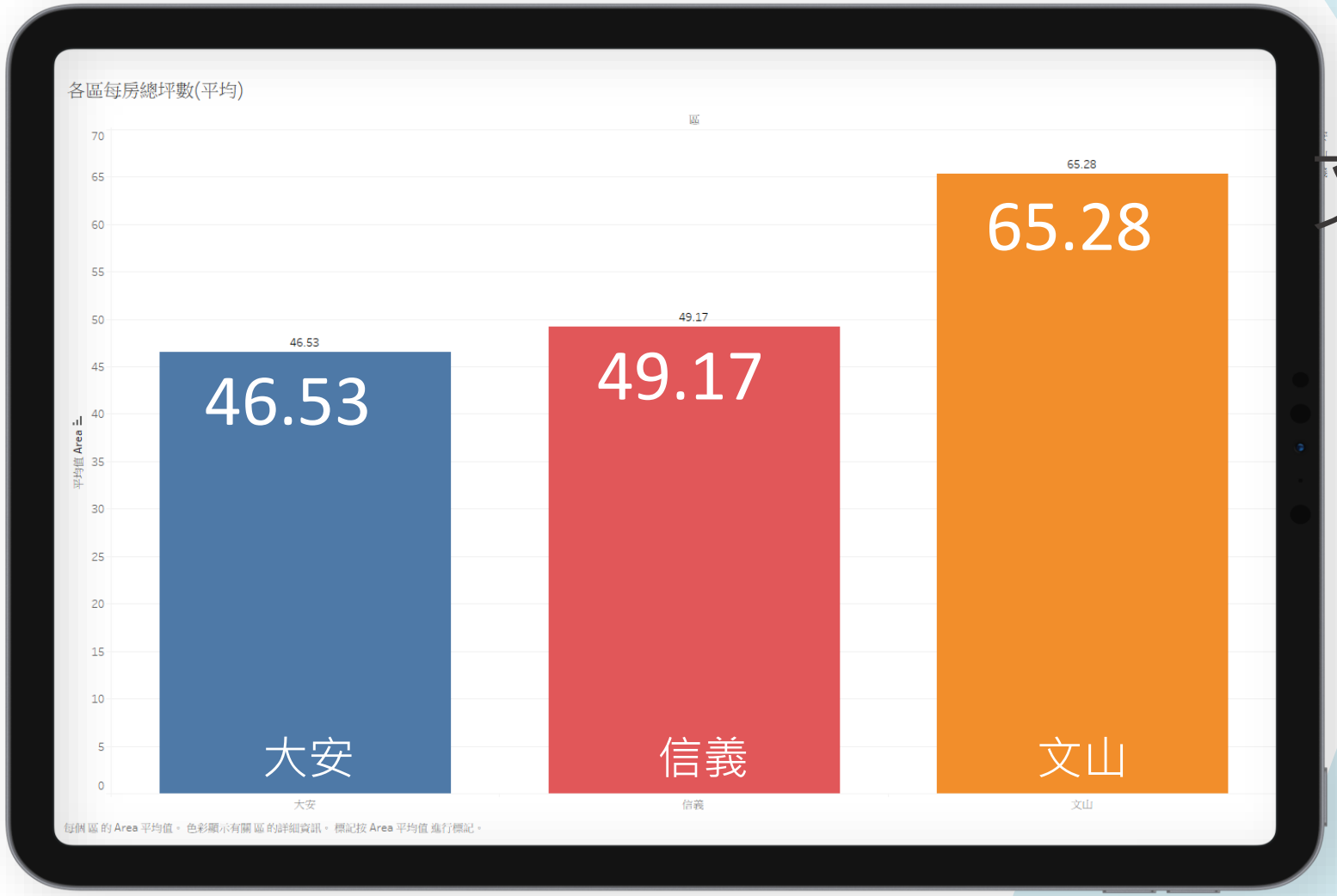


大安是

文山的

2.16 X

視覺化分析 × 各區/坪數

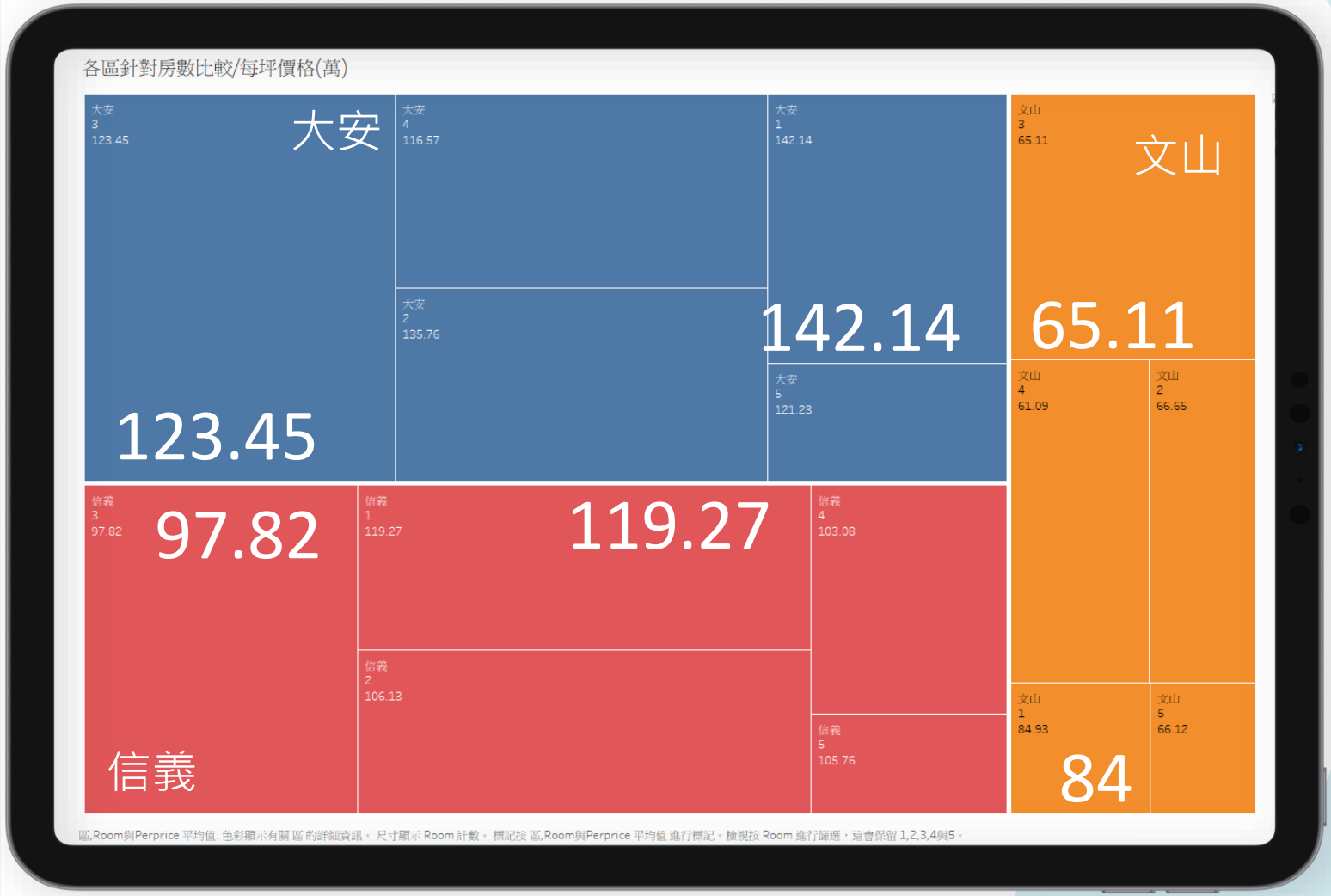


文山是

大山的

1.4 X

視覺化分析 × 各區/房數/單價



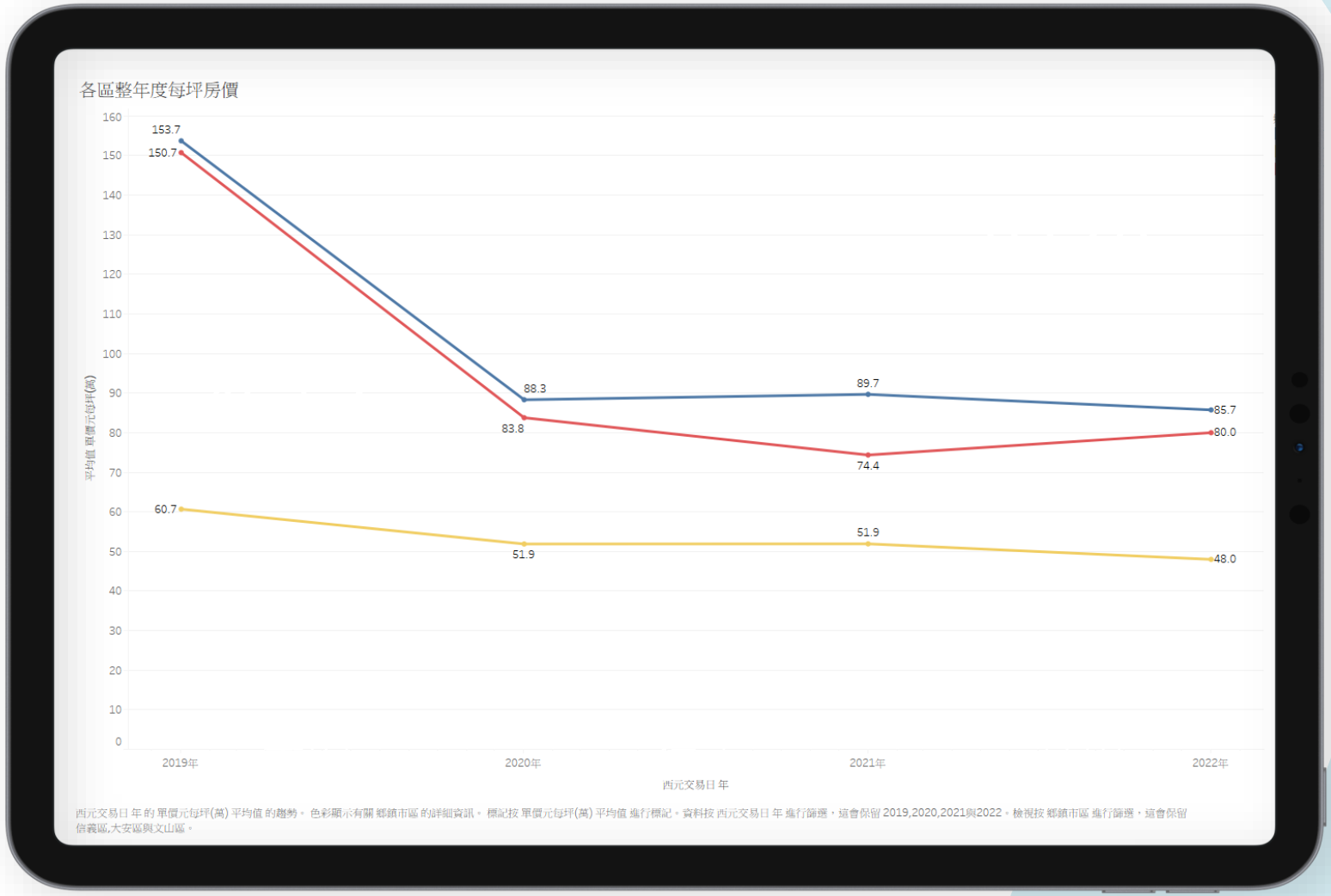
1房

最貴

3房

CP王

視覺化分析 × 各區/單價



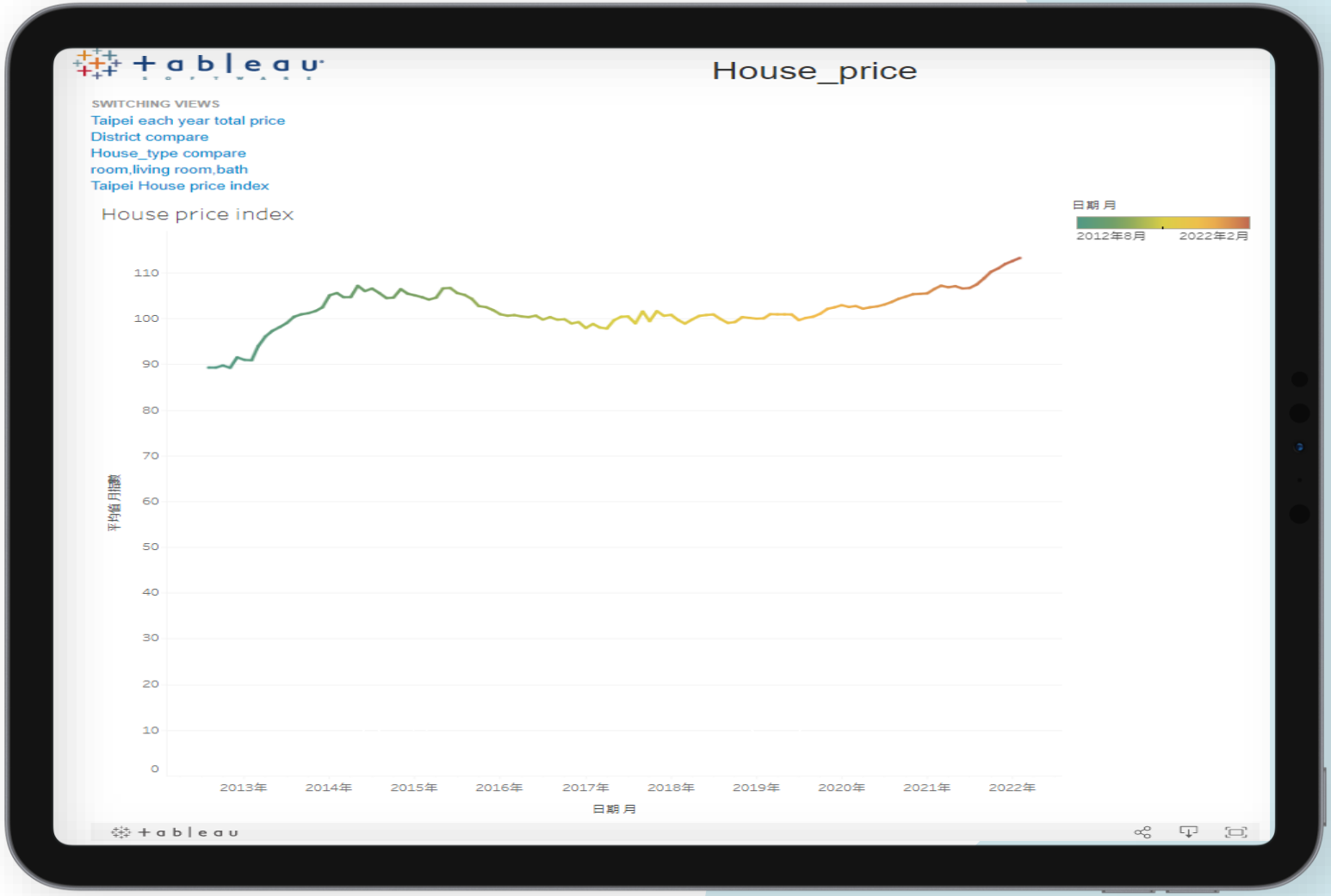
- 大安
- 信義
- 文山

信義/大安

降幅

達 44 %

視覺化分析 × 台北住房指數



困境 Dilemma

Difficult +

Solution ?

How ?

06.24

困境 回顧

57 DAY

Branden

爬蟲

- 樓層欄位的定義以及房/廳/衛的資料的格式切分較為複雜，並且包含許多不需要的雜項字元。
- 因為缺少每坪價位的資訊，因此需要做Dataframe上的計算，過程中產生inf的無窮值。

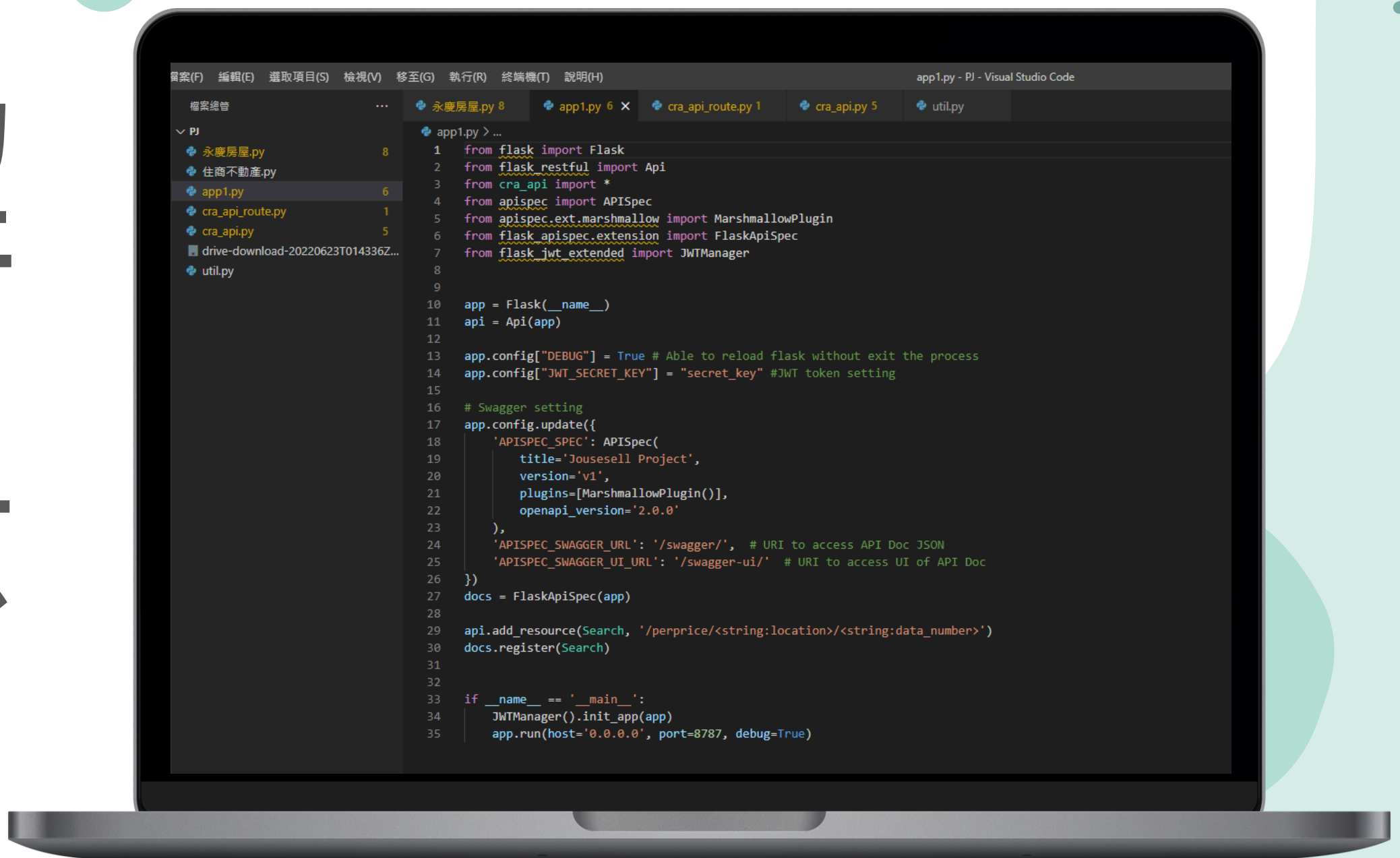
API

- 撰寫API時在JSON格式上有錯誤。

解決方法

- Dataframe的調整，包含計算上，除了做to_numeric的轉型之外，在資料清理時也要檢查錯誤值做取代。
- 撰寫JSON時可以習慣做JSON Formatter的檢查

解決


檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
app1.py - PJ - Visual Studio Code
檔案總管
永慶房屋.py 8
app1.py 6
cra_api_route.py 1
cra_api.py 5
util.py
PJ
永慶房屋.py 8
住商不動產.py
app1.py 6
cra_api_route.py 1
cra_api.py 5
drive-download-20220623T014336Z...
util.py
app1.py > ...
1 from flask import Flask
2 from flask_restful import Api
3 from cra_api import *
4 from apispec import APISpec
5 from apispec.ext.marshmallow import MarshmallowPlugin
6 from flask_apispec.extension import FlaskApiSpec
7 from flask_jwt_extended import JWTManager
8
9
10 app = Flask(__name__)
11 api = Api(app)
12
13 app.config["DEBUG"] = True # Able to reload flask without exit the process
14 app.config["JWT_SECRET_KEY"] = "secret_key" #JWT token setting
15
16 # Swagger setting
17 app.config.update({
18 'APISPEC_SPEC': APISpec(
19 title='Jousesell Project',
20 version='v1',
21 plugins=[MarshmallowPlugin()],
22 openapi_version='2.0.0'
23),
24 'APISPEC_SWAGGER_URL': '/swagger/', # URI to access API Doc JSON
25 'APISPEC_SWAGGER_UI_URL': '/swagger-ui/' # URI to access UI of API Doc
26 })
27 docs = FlaskApiSpec(app)
28
29 api.add_resource(Search, '/perprice/<string:location>/<string:data_number>')
30 docs.register(Search)
31
32
33 if __name__ == '__main__':
34 JWTManager().init_app(app)
35 app.run(host='0.0.0.0', port=8787, debug=True)

06.24

困境回顧

57 DAY

Darren

爬蟲

- 價格的區塊有多種顯示方式，無法正確定位到總價
- 房廳衛的格式有多種可能性，資料切割處理繁雜

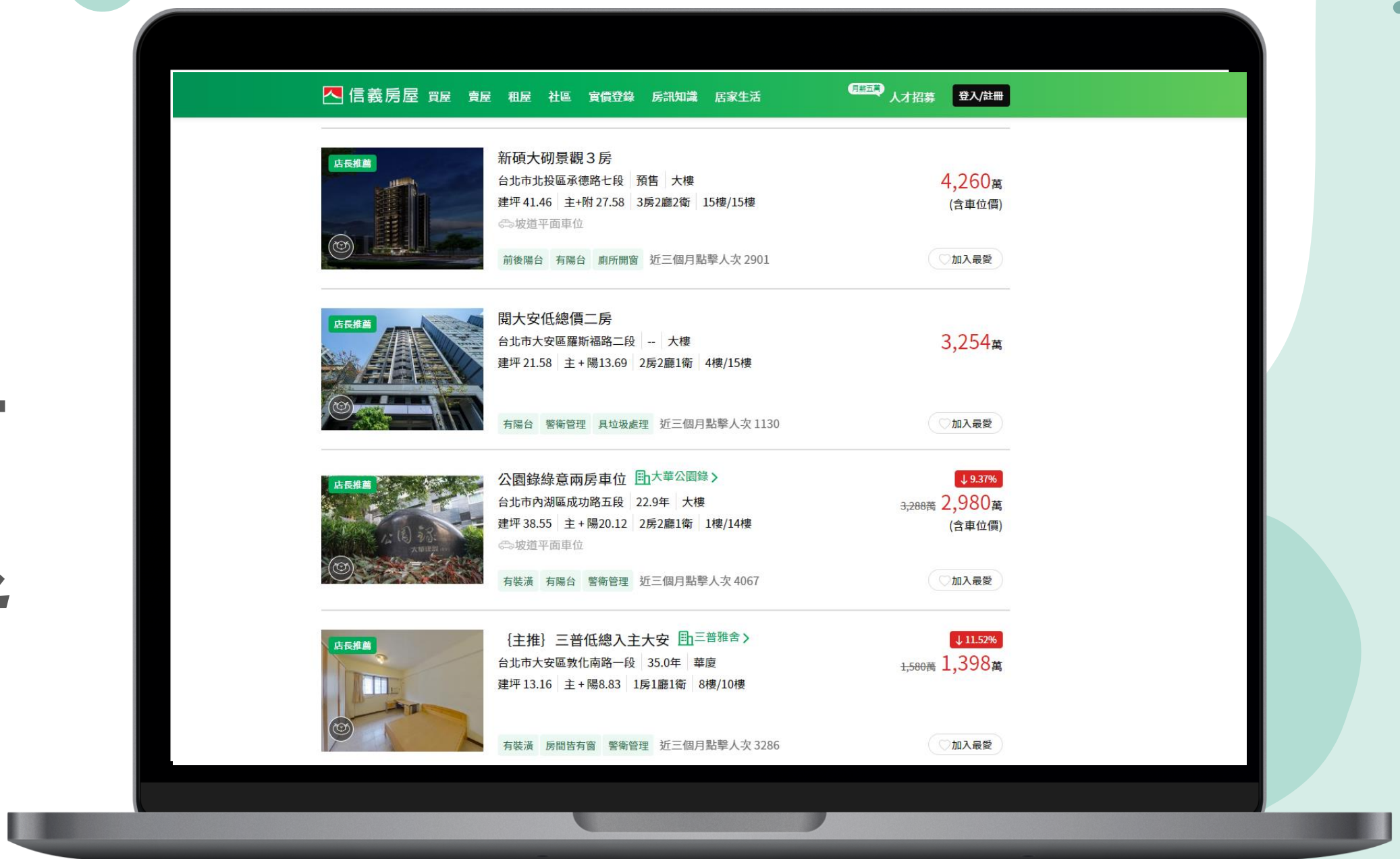
資料庫

- 透過AWS的RDS功能架設，主要的問題只有一開始忘了開RDS的3306 port的權限，導致無法連線登入

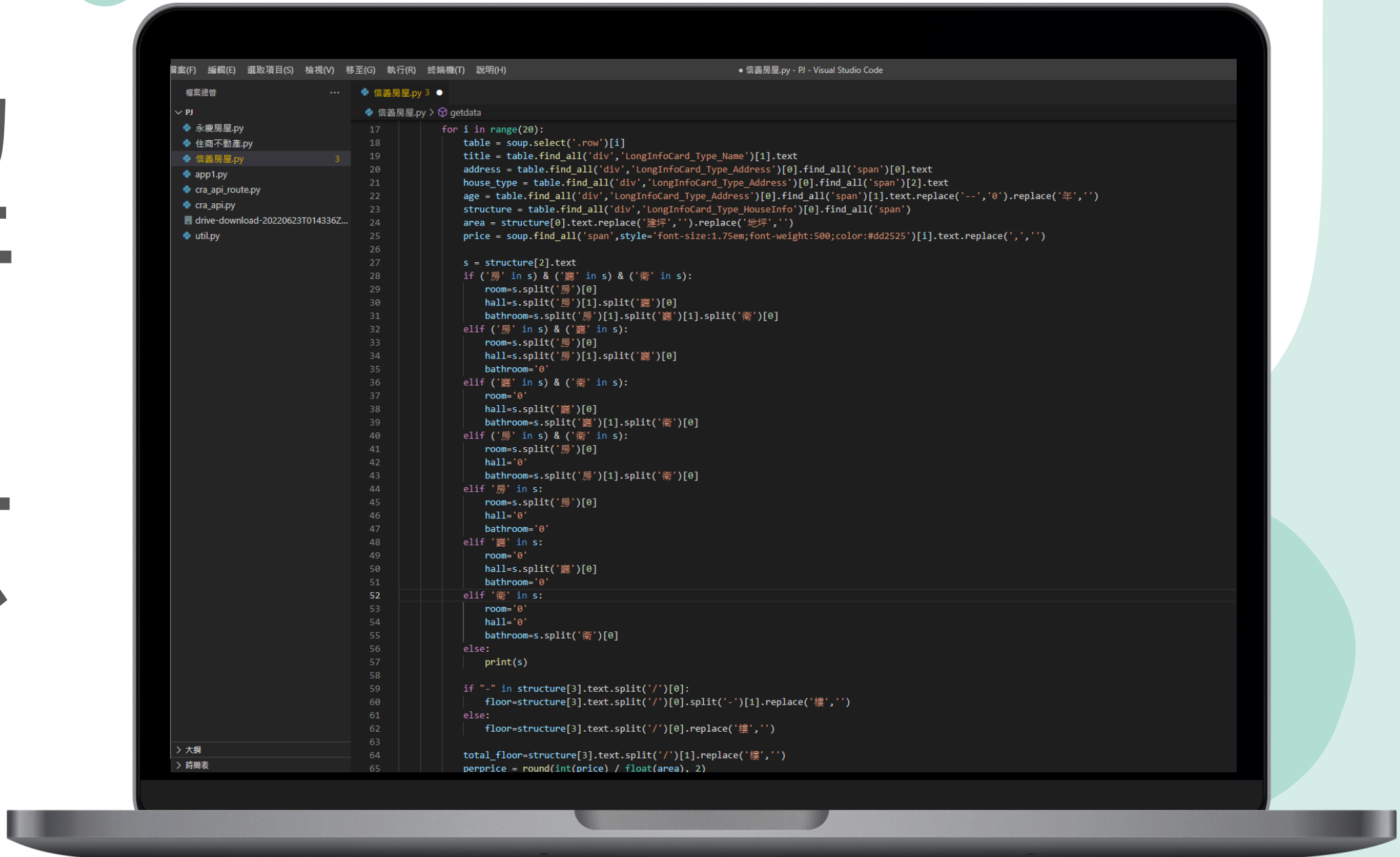
解決方法

- 發現總價的字體色彩與其他字體皆不同，因此透過find_all定位屬性的方法去抓取
- 列舉8項可能性 使用if elif else的判斷式去篩選

問題


信義房屋 買屋 賣屋 租屋 社區 實價登錄 房訊知識 居家生活
月薪五萬 人才招募
登入/註冊
店長推薦
新碩大砌景觀３房
台北市北投區承德路七段 | 預售 | 大樓
建坪 41.46 | 主+附 27.58 | 3房2廳2衛 | 15樓/15樓
坡道平面車位
4,260萬
(含車位價)
前後陽台 有陽台 廁所開窗 近三個月點擊人次 2901
加入最愛
店長推薦
閱大安低總價二房
台北市大安區羅斯福路二段 | -- | 大樓
建坪 21.58 | 主＋陽13.69 | 2房2廳1衛 | 4樓/15樓
3,254萬
有陽台 警衛管理 具垃圾處理 近三個月點擊人次 1130
加入最愛
店長推薦
公園錄綠意兩房車位
大華公園錄
台北市內湖區成功路五段 | 22.9年 | 大樓
建坪 38.55 | 主＋陽20.12 | 2房2廳1衛 | 1樓/14樓
坡道平面車位
↓9.37%
3,288萬 2,980萬
(含車位價)
有裝潢 有陽台 警衛管理 近三個月點擊人次 4067
加入最愛
店長推薦
{主推} 三普低總入主大安
三普雅舍
台北市大安區敦化南路一段 | 35.0年 | 華廈
建坪 13.16 | 主＋陽8.83 | 1房1廳1衛 | 8樓/10樓
↓11.52%
1,580萬 1,398萬
有裝潢 房間皆有窗 警衛管理 近三個月點擊人次 3286
加入最愛

解決


檔案(F) 編輯(E) 選取項目(S) 檢視(V) 移至(G) 執行(R) 終端機(T) 說明(H)
• 信義房屋.py - PJ - Visual Studio Code
檔案總管
信義房屋.py 3
PJ
永慶房屋.py
住商不動產.py
信義房屋.py 3
app1.py
cra_api_route.py
cra_api.py
drive-download-20220623T014336Z...
util.py
信義房屋.py > getdata
大綱
時間表
for i in range(20):
table = soup.select('.row')[i]
title = table.find_all('div','LongInfoCard_Type_Name')[1].text
address = table.find_all('div','LongInfoCard_Type_Address')[0].find_all('span')[0].text
house_type = table.find_all('div','LongInfoCard_Type_Address')[0].find_all('span')[2].text
age = table.find_all('div','LongInfoCard_Type_Address')[0].find_all('span')[1].text.replace('--','0').replace('年','')
structure = table.find_all('div','LongInfoCard_Type_HouseInfo')[0].find_all('span')
area = structure[0].text.replace('建坪','').replace('地坪','')
price = soup.find_all('span',style='font-size:1.75em;font-weight:500;color:#dd2525')[i].text.replace(',','')
s = structure[2].text
if ('房' in s) & ('廳' in s) & ('衛' in s):
room=s.split('房')[0]
hall=s.split('房')[1].split('廳')[0]
bathroom=s.split('房')[1].split('廳')[1].split('衛')[0]
elif ('房' in s) & ('廳' in s):
room=s.split('房')[0]
hall=s.split('房')[1].split('廳')[0]
bathroom='0'
elif ('廳' in s) & ('衛' in s):
room='0'
hall=s.split('廳')[0]
bathroom=s.split('廳')[1].split('衛')[0]
elif ('房' in s) & ('衛' in s):
room=s.split('房')[0]
hall='0'
bathroom=s.split('房')[1].split('衛')[0]
elif '房' in s:
room=s.split('房')[0]
hall='0'
bathroom='0'
elif '廳' in s:
room='0'
hall=s.split('廳')[0]
bathroom='0'
elif '衛' in s:
room='0'
hall='0'
bathroom=s.split('衛')[0]
else:
print(s)
if "-" in structure[3].text.split('/')[0]:
floor=structure[3].text.split('/')[0].split('-')[1].replace('樓','')
else:
floor=structure[3].text.split('/')[0].replace('樓','')
total_floor=structure[3].text.split('/')[1].replace('樓','')
perprice = round(int(price) / float(area), 2)

06.24

困境回顧

57 DAY

Howard

爬蟲

- 爬蟲時遇到網頁有透過ajax技術，因此沒辦法用 requests html 來爬到需要的資料。

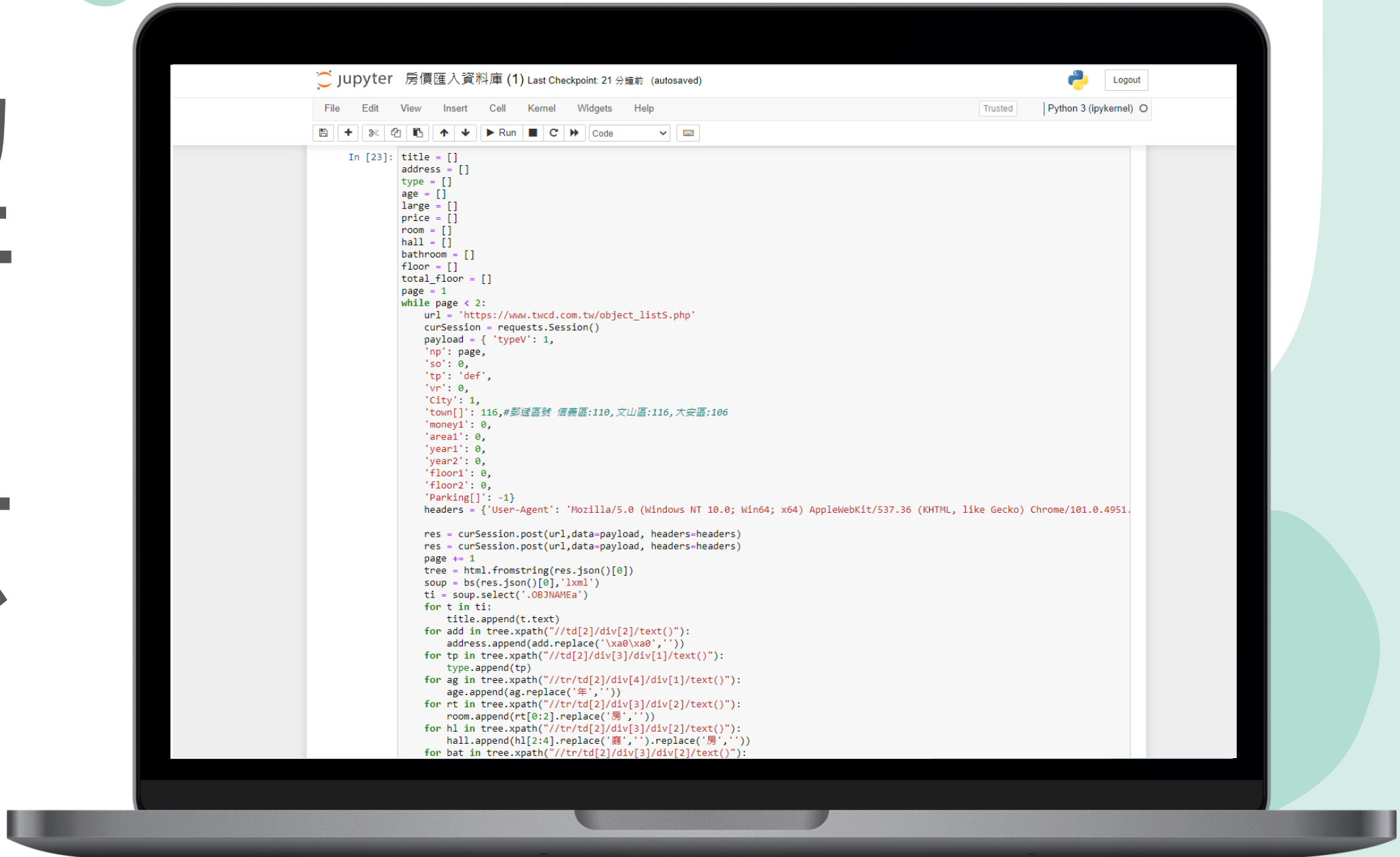
資料整理

- 由於爬下來的資料很雜，有出現空值或是格式不符等錯誤，影響視覺化的呈現。

解決方法

- 在Network 可以找到資料需要的Url，透過post或get 請求，來取得資料。
- 空值的部分填入整欄的平均值，民國的日期改以西元方式呈現。

解決


jupyter 房價匯入資料庫 (1) Last Checkpoint: 21 分鐘前 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [23]:
title = []
address = []
type = []
age = []
large = []
price = []
room = []
hall = []
bathroom = []
floor = []
total_floor = []
page = 1
while page < 2:
url = 'https://www.twcd.com.tw/object_listS.php'
curSession = requests.Session()
payload = { 'typeV': 1,
'np': page,
'so': 0,
'tp': 'def',
'vr': 0,
'City': 1,
'town[]': 116,#鄉鎮區號 信義區:110,文山區:116,大安區:106
'money1': 0,
'area1': 0,
'year1': 0,
'year2': 0,
'floor1': 0,
'floor2': 0,
'Parking[]': -1}
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.
res = curSession.post(url,data=payload, headers=headers)
res = curSession.post(url,data=payload, headers=headers)
page += 1
tree = html.fromstring(res.json()[0])
soup = bs(res.json()[0],'lxml')
ti = soup.select('.OBJNAMEa')
for t in ti:
title.append(t.text)
for add in tree.xpath("//td[2]/div[2]/text()"):
address.append(add.replace('\xa0\xa0',''))
for tp in tree.xpath("//td[2]/div[3]/div[1]/text()"):
type.append(tp)
for ag in tree.xpath("//tr/td[2]/div[4]/div[1]/text()"):
age.append(ag.replace('年',''))
for rt in tree.xpath("//tr/td[2]/div[3]/div[2]/text()"):
room.append(rt[0:2].replace('房',''))
for hl in tree.xpath("//tr/td[2]/div[3]/div[2]/text()"):
hall.append(hl[2:4].replace('廳','').replace('房',''))
for bat in tree.xpath("//tr/td[2]/div[3]/div[2]/text()"):

06.24

困境

回顧

57 DAY

Jeff

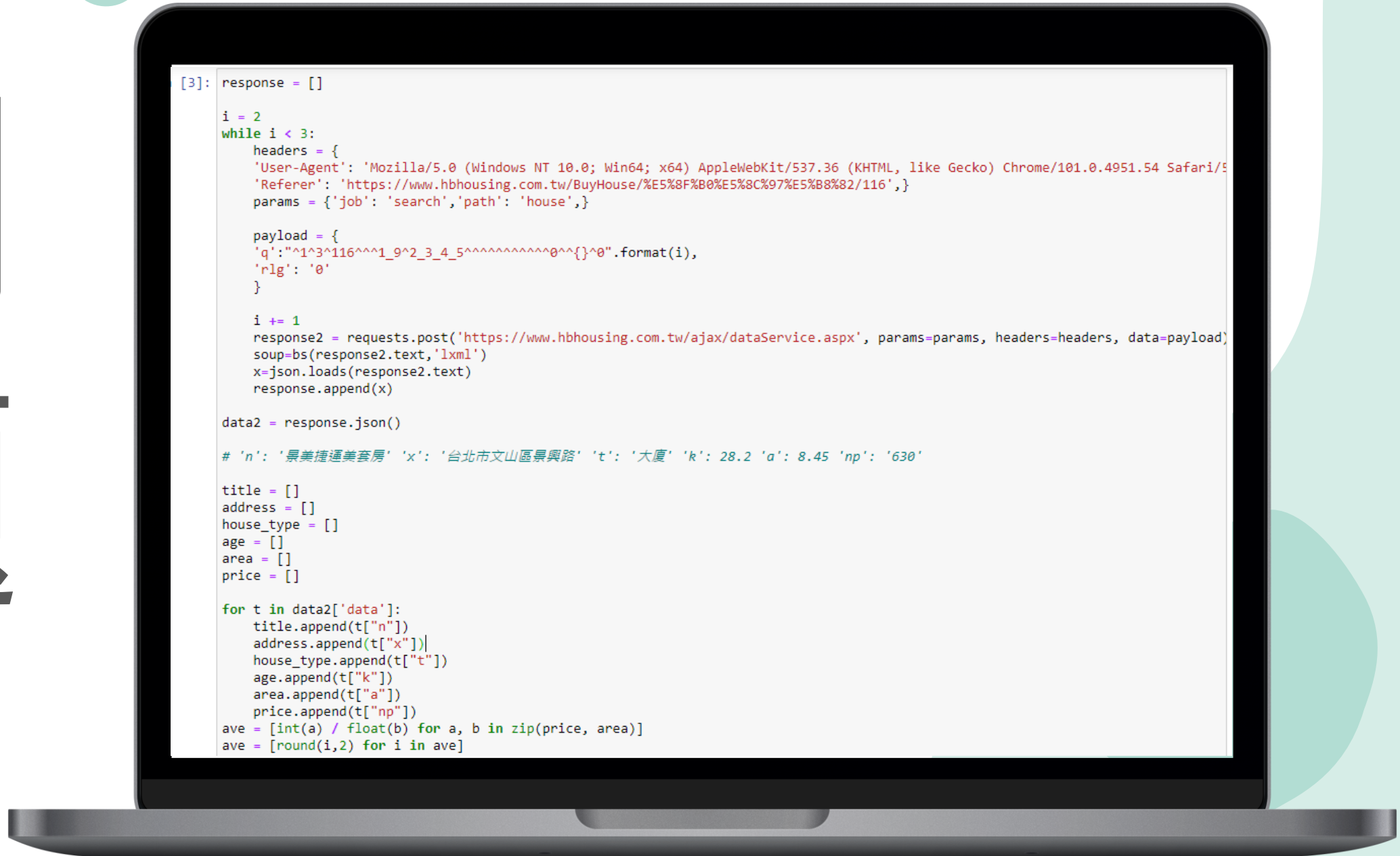
爬蟲

- 無法使用while迴圈把每一頁資料Append到同列表
- 無法使用Request.get的方式來取得資料

解決方法

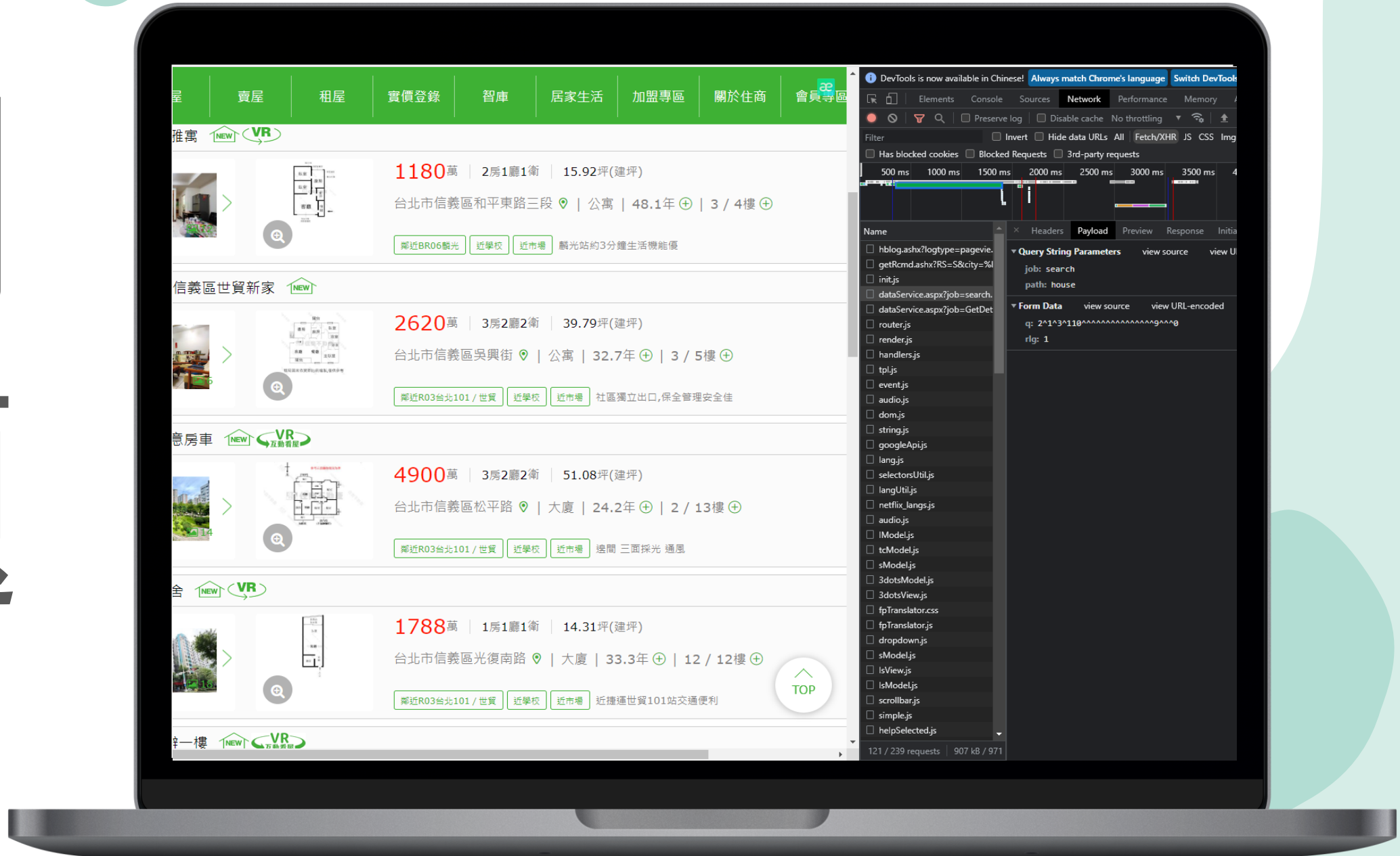
- 使用雙For迴圈把資料取出來，就可以Append到列表。
- 從Data-Service中的Payload來去GET

問題

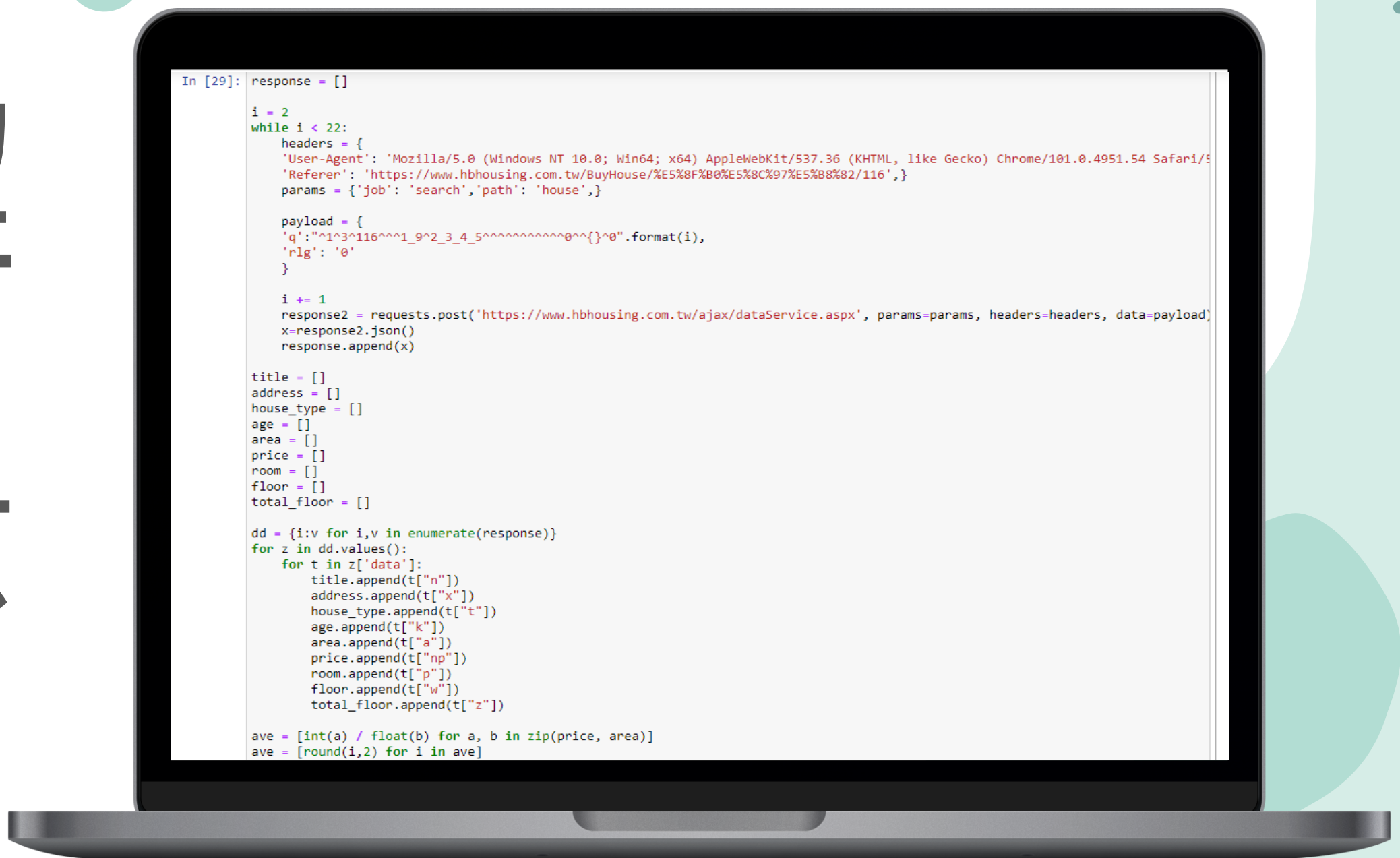


```
[3]: response = []

     i = 2
     while i < 3:
         headers = {
         'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/5
         'Referer': 'https://www.hbhousing.com.tw/BuyHouse/%E5%8F%B0%E5%8C%97%E5%B8%82/116',}
         params = {'job': 'search','path': 'house',}

         payload = {
         'q':"^1^3^116^^^1_9^2_3_4_5^^^^^^^^^^0^^{}^0".format(i),
         'rlg': '0'
         }

         i += 1
         response2 = requests.post('https://www.hbhousing.com.tw/ajax/dataService.aspx', params=params, headers=headers, data=payload)
         soup=bs(response2.text,'lxml')
         x=json.loads(response2.text)
         response.append(x)

     data2 = response.json()

     # 'n': '景美捷運美賽房' 'x': '台北市文山區景興路' 't': '大廈' 'k': 28.2 'a': 8.45 'np': '630'

     title = []
     address = []
     house_type = []
     age = []
     area = []
     price = []

     for t in data2['data']:
         title.append(t["n"])
         address.append(t["x"])
         house_type.append(t["t"])
         age.append(t["k"])
         area.append(t["a"])
         price.append(t["np"])
     ave = [int(a) / float(b) for a, b in zip(price, area)]
     ave = [round(i,2) for i in ave]
```

問題


賣屋
租屋
實價登錄
智庫
居家生活
加盟專區
關於住商
雅寓
NEW
VR
1180萬 | 2房1廳1衛 | 15.92坪(建坪)
台北市信義區和平東路三段 | 公寓 | 48.1年 | 3 / 4樓
鄰近BR06酪光
近學校
近市場
酪光站約3分鐘生活機能優
信義區世貿新家
NEW
2620萬 | 3房2廳2衛 | 39.79坪(建坪)
台北市信義區吳興街 | 公寓 | 32.7年 | 3 / 5樓
鄰近R03台北101 / 世貿
近學校
近市場
社區獨立出口,保全管理安全佳
意房車
NEW
VR
互動看屋
4900萬 | 3房2廳2衛 | 51.08坪(建坪)
台北市信義區松平路 | 大廈 | 24.2年 | 2 / 13樓
鄰近R03台北101 / 世貿
近學校
近市場
邊間 三面採光 通風
舍
NEW
VR
1788萬 | 1房1廳1衛 | 14.31坪(建坪)
台北市信義區光復南路 | 大廈 | 33.3年 | 12 / 12樓
鄰近R03台北101 / 世貿
近學校
近市場
近捷運世貿101站交通便利
TOP
一樓
NEW
VR
互動看屋
DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools
Elements
Console
Sources
Network
Performance
Memory
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Has blocked cookies
Blocked Requests
3rd-party requests
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
Name
hblog.ashx?logtype=pagevie...
getRcmd.ashx?RS=S&city=%E...
init.js
dataService.aspx?job=search...
dataService.aspx?job=GetDet...
router.js
render.js
handlers.js
tpl.js
event.js
audio.js
dom.js
string.js
googleApi.js
lang.js
selectorsUtil.js
langUtil.js
netflix_langs.js
audio.js
IModel.js
tcModel.js
sModel.js
3dotsModel.js
3dotsView.js
fpTranslator.css
fpTranslator.js
dropdown.js
sModel.js
lsView.js
lsModel.js
scrollbar.js
simple.js
helpSelected.js
121 / 239 requests
907 kB / 971
Headers
Payload
Preview
Response
Initia
Query String Parameters
view source
view U
job: search
path: house
Form Data
view source
view URL-encoded
q: 2^1^3^110^^^^^^^^^^^^^^^^g^^^0
rlg: 1

解決



In [29]:

```
response = []

i = 2
while i < 22:
    headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/5
    'Referer': 'https://www.hbhousing.com.tw/BuyHouse/%E5%8F%B0%E5%8C%97%E5%B8%82/116',}
    params = {'job': 'search','path': 'house',}

    payload = {
    'q':"^1^3^116^^^1_9^2_3_4_5^^^^^^^^^^0^^{}^0".format(i),
    'rlg': '0'
    }

    i += 1
    response2 = requests.post('https://www.hbhousing.com.tw/ajax/dataService.aspx', params=params, headers=headers, data=payload)
    x=response2.json()
    response.append(x)

title = []
address = []
house_type = []
age = []
area = []
price = []
room = []
floor = []
total_floor = []

dd = {i:v for i,v in enumerate(response)}
for z in dd.values():
    for t in z['data']:
        title.append(t["n"])
        address.append(t["x"])
        house_type.append(t["t"])
        age.append(t["k"])
        area.append(t["a"])
        price.append(t["np"])
        room.append(t["p"])
        floor.append(t["w"])
        total_floor.append(t["z"])

ave = [int(a) / float(b) for a, b in zip(price, area)]
ave = [round(i,2) for i in ave]
```

未来

Future

MORE ?

Enrich ?

Vision ?

未來工作


台北市
信義區
電梯大廈
公寓
套房
111年04月 行情
90 萬/坪
跟上月比較
4%
跟去年同期比較
9%
成交單價
成交件數
96萬
94萬
92萬
90萬
88萬
86萬
84萬
82萬
80萬
78萬
均價(萬元/坪)
120件
110件
100件
90件
80件
70件
60件
110年11月
110年12月
111年01月
111年02月
111年03月
111年04月
成交均價
成交件數
行情演算說明

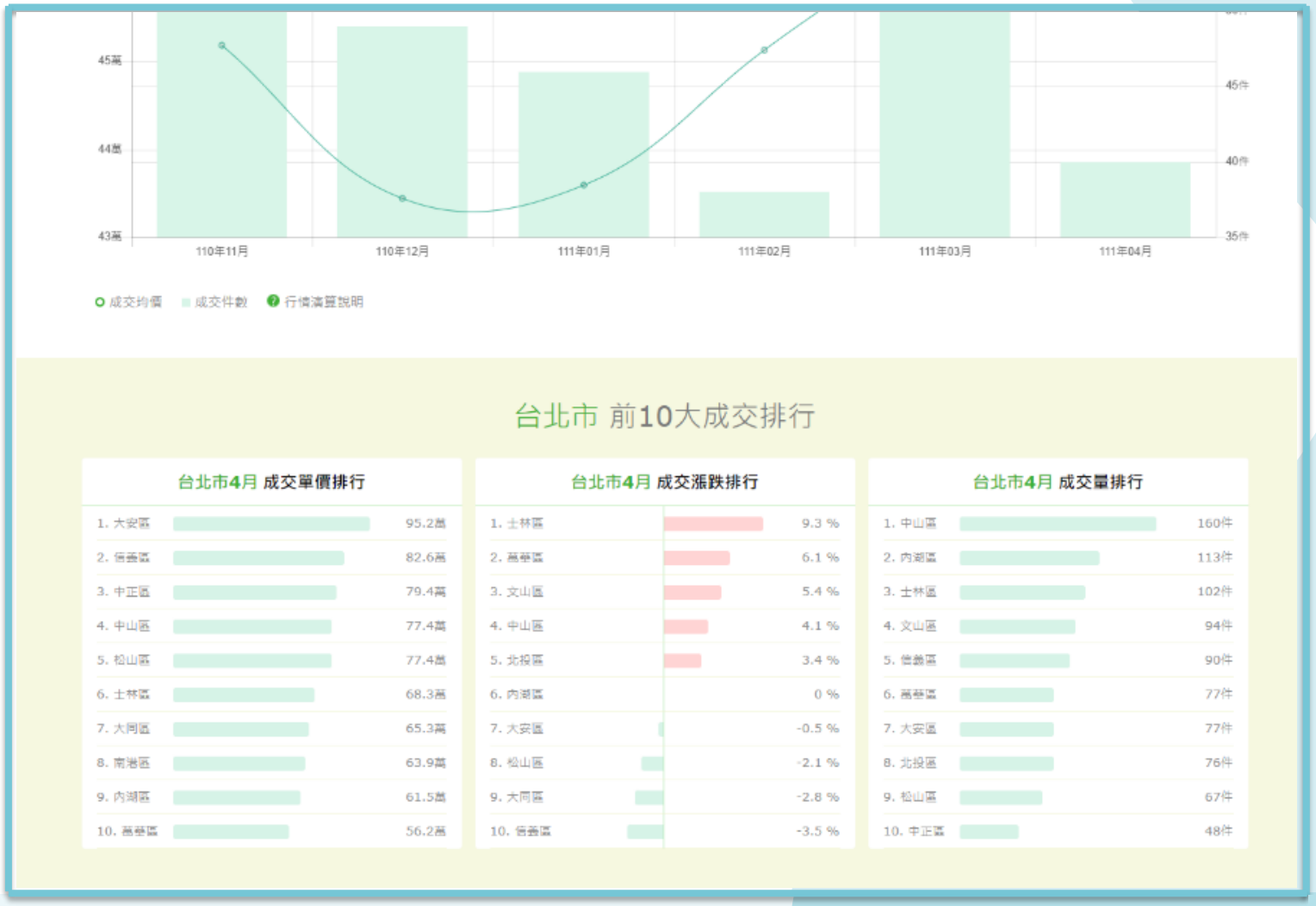
未來工作


實價登錄
依區域查詢
依捷運查詢
依地圖查詢
台北市 文山區
請輸入欲查詢地址
進階篩選
立即搜尋
台北市 文山區 實價登錄
資料來源：內政部實價登錄 基準月份：111年04月
1個月內
3個月內
近半年
近1年
近2年
公寓
平均：44萬/坪
跟上月比較：-10%
跟去年同期比較：+1%
電梯大樓
平均：53.3萬/坪
跟上月比較：-12%
跟去年同期比較：-4%
透天
平均：92.7萬/坪
跟上月比較：+42%
跟去年同期比較：+92%
套房
平均：55.6萬/坪
跟上月比較：--
跟去年同期比較：--
TOP
台北市 文山區 實價登錄走勢
電梯大廈
公寓
套房
其他

未來工作

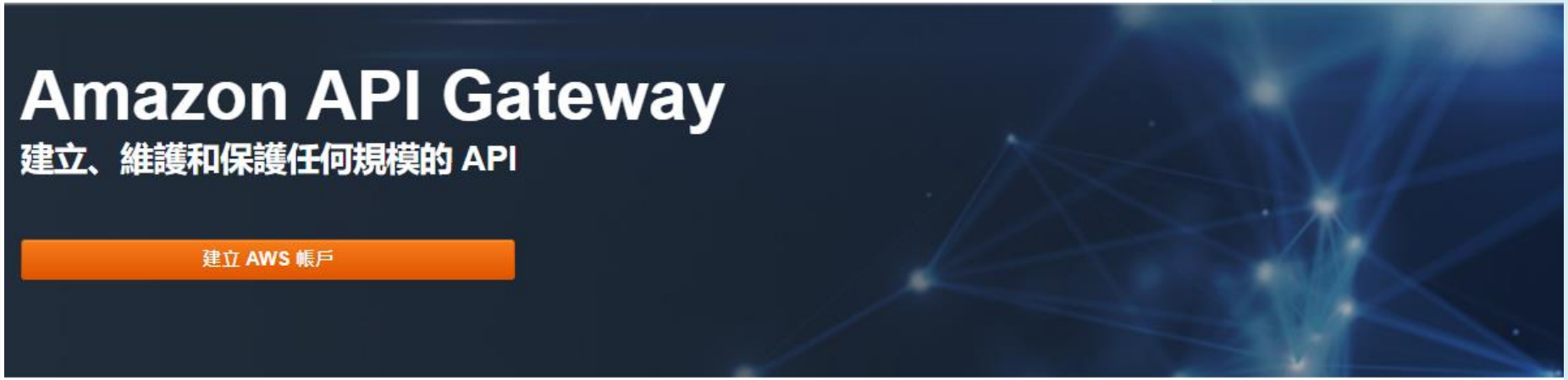

台北市 文山區 實價登錄走勢
電梯大廈
公寓
套房
其他
成交單價
成交件數
49萬
48萬
47萬
46萬
45萬
44萬
43萬
70件
65件
60件
55件
50件
45件
40件
35件
110年11月
110年12月
111年01月
111年02月
111年03月
111年04月
成交均價
成交件數
行情測量說明
台北市 前10大成交排行

未來工作


45萬
44萬
43萬
45件
40件
35件
110年11月
110年12月
111年01月
111年02月
111年03月
111年04月
成交均價
成交件數
行情演算說明
台北市 前10大成交排行
台北市4月 成交單價排行
1. 大安區 95.2萬
2. 信義區 82.6萬
3. 中正區 79.4萬
4. 中山區 77.4萬
5. 松山區 77.4萬
6. 士林區 68.3萬
7. 大同區 65.3萬
8. 南港區 63.9萬
9. 內湖區 61.5萬
10. 萬華區 56.2萬
台北市4月 成交漲跌排行
1. 士林區 9.3 %
2. 萬華區 6.1 %
3. 文山區 5.4 %
4. 中山區 4.1 %
5. 北投區 3.4 %
6. 內湖區 0 %
7. 大安區 -0.5 %
8. 松山區 -2.1 %
9. 大同區 -2.8 %
10. 信義區 -3.5 %
台北市4月 成交量排行
1. 中山區 160件
2. 內湖區 113件
3. 士林區 102件
4. 文山區 94件
5. 信義區 90件
6. 萬華區 77件
7. 大安區 77件
8. 北投區 76件
9. 松山區 67件
10. 中正區 48件

Darren X aws

未來工作



Amazon API Gateway 是一種全受管的服務，可讓開發人員輕鬆地建立、發佈、維護、監控和保護任何規模的 API。API 可作為應用程式的「前門」，以便從後端服務存取資料、商業邏輯或功能。使用 API Gateway 時，您可以建立 RESTful API 和 WebSocket API，以啟用即時雙向通訊應用程式。API Gateway 支援容器化、無伺服器工作負載和 Web 應用程式。

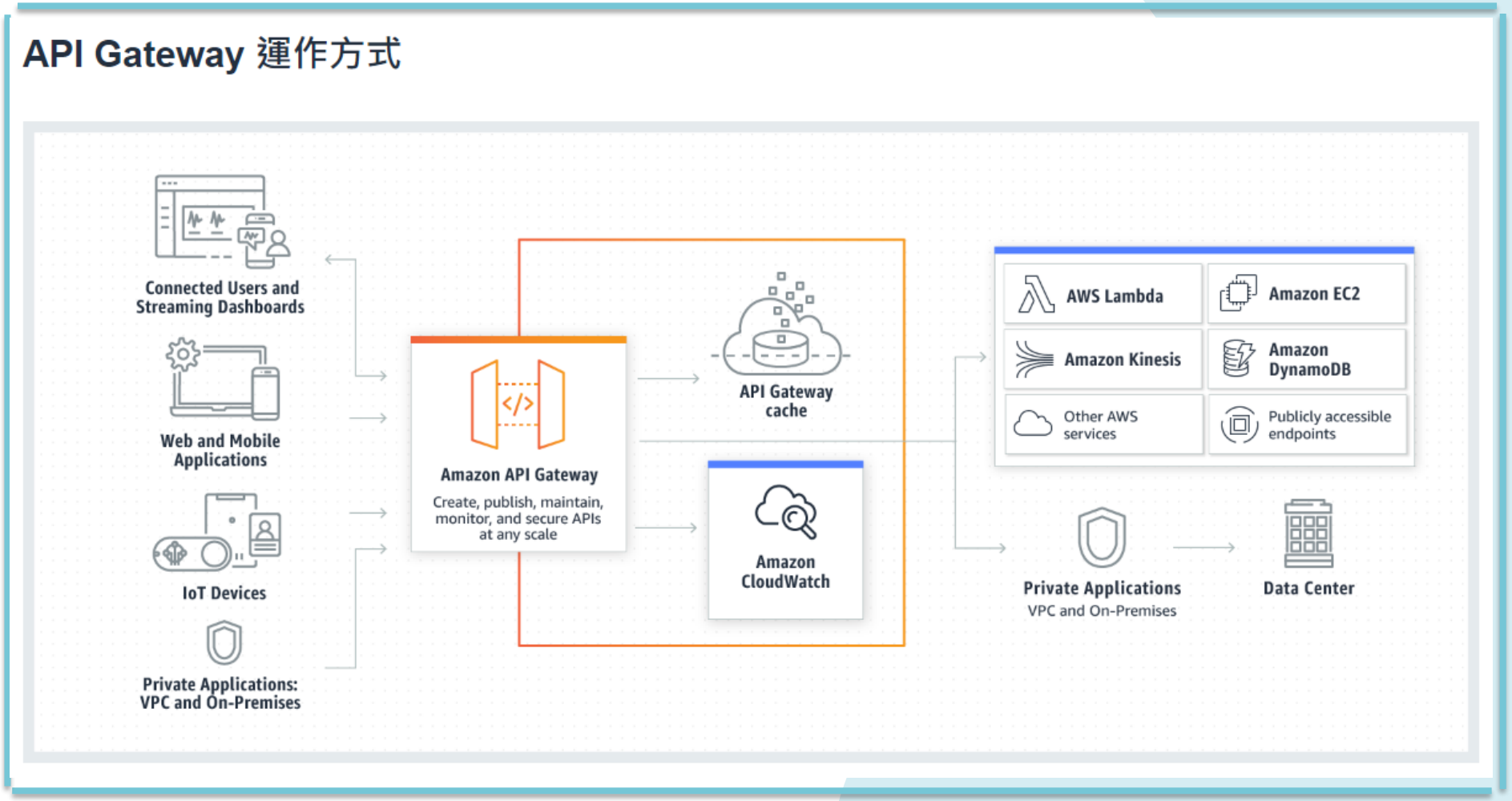
API Gateway 負責處理有關接受和處理多達數十萬個並行 API 呼叫的所有工作，包括流量管理、CORS 支援、授權和存取控制、調節、監控和 API 版本管理。API Gateway 沒有最低費用或啟動成本。您要為收到的 API 呼叫和資料傳輸量支付費用，而使用 API Gateway 分級定價模型，可在 API 用量擴展時減少成本。

100 萬個 API 呼叫接收免費

(使用 AWS 免費方案 12 個月的每個月)

免費試用 »

未來工作


API Gateway 運作方式
Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center

未來工作

優勢

有效率的 API 開發

使用 API Gateway 同時執行相同 API 的多個版本，可讓您快速重複執行、測試和發行新的版本。您要為 API 呼叫和資料傳輸付費；既沒有最低費用，也沒有前期承諾。

任何規模的效能

使用 Amazon CloudFront 來妥善利用節點的全球網路，為最終使用者提供 API 請求和回應的最低可能延遲。調節流量和授權 API 呼叫，以確保後端操作可以承受流量高峰，而且不需要呼叫後端系統。

大規模節省成本

API Gateway 提供 API 請求的分級定價模式。在最高級之中，每百萬請求的 API 請求價格低至 0.90 USD，這是因為在各個 AWS 帳戶中，每個區域的 API 用量都有所增加，所以您的成本會下降。

輕鬆監控

從 API Gateway 儀表板監控效能指標，以及 API 呼叫、資料延遲和錯誤率的相關資訊，這可讓您使用 Amazon CloudWatch，以視覺化方式監控對您服務的呼叫。

靈活的安全性控制

利用 AWS Identity and Access Management (IAM) 和 Amazon Cognito，授與對 API 的存取權。如果您使用 OAuth 字符，則 API Gateway 會提供原生 OIDC 和 OAuth2 支援。為了支援自訂的授權要求，您可以從 AWS Lambda 執行 Lambda 授權方。

RESTful API 選項

使用 HTTP API 或 REST API 建立 RESTful API。對大多數使用案例而言，HTTP API 是建置 HTTP API 的最佳方法，最高可比 REST API 便宜 71%。如果您的使用案例需要單一解決方案中有 API 代理功能和管理功能，您就可以使用 REST API。

The End