**程式碼元件**

**OMFLOW 使用手冊**

**文件版本：1.1.6.3**

**文件產生日期：112/10/03**



# 目錄

[目錄 1](#_Toc147503806)

[通用備註 2](#_Toc147503807)

[前言 2](#_Toc147503808)

[1 表單 2](#_Toc147503809)

[1.1 開單 3](#_Toc147503810)

[1.2 推單 4](#_Toc147503811)

[1.3 列表 4](#_Toc147503812)

[1.4 刪單 6](#_Toc147503813)

[1.5 進階推單 6](#_Toc147503814)

[2 使用者 8](#_Toc147503815)

[2.1 建立使用者1.1.6.0 8](#_Toc147503816)

[2.2 更新使用者 9](#_Toc147503817)

[2.3 查詢使用者(列表) 10](#_Toc147503818)

[2.4 查詢使用者(單一) 11](#_Toc147503819)

[2.5 查詢使用者Email(列表) 12](#_Toc147503820)

[2.6 刪除使用者 12](#_Toc147503821)

[2.7 使用者加入部門 13](#_Toc147503822)

[2.8 使用者移出部門 13](#_Toc147503823)

[3 部門/角色 14](#_Toc147503824)

[3.1 建立部門 14](#_Toc147503825)

[3.2 取得部門資訊 15](#_Toc147503826)

[3.3 加入使用者 16](#_Toc147503827)

[3.4 移出使用者 16](#_Toc147503828)

[3.5 查詢部門職務權責 17](#_Toc147503829)

[3.6 更新部門職務權責 17](#_Toc147503830)

[4 組織圖 18](#_Toc147503831)

[4.1 取得組織圖資訊 18](#_Toc147503832)

[4.2 更新/新增組織圖 20](#_Toc147503833)

[4.3 刪除組織圖 21](#_Toc147503834)

[4.4 透過【同系職務/權責】尋找人員 22](#_Toc147503835)

[4.5 透過【部門職務/權責】尋找人員 23](#_Toc147503836)

# 通用備註

1. 點擊：需要您進行滑鼠點擊的部分，會以粗體方式呈現
2. 變數：提到動態或非固定的文字項目時，會以［］作為提示，如［使用者名稱］
3. 元件：流程設計中每一個物件，通稱為元件
4. 欄位：一般指為可輸入的介面，本系統中也特指表單設計的可輸入欄位
5. 頁籤：介面或視窗上的分頁選項
6. 區塊：指多個欄位的集合，通常會有一個名稱於上方

# 前言

位置：［**主選單**］＞**應用管理**＞**應用設計**＞［**任意應用**］＞［**任意流程**］＞**流程設計**頁籤 ＞**程式碼**元件＞［**齒輪**］按鈕＞［Python］頁籤

此篇主要介紹使用者在設計「程式碼元件」時，可呼叫OMFLOW額外提供的內建function來進行資料查詢或變更，主要針對常用的使用者、部門、組織圖等等相關應用，以便簡化程式碼並確保系統穩定性。

# 表單

位置：［**主選單**］＞Apps＞［**任意應用**］＞［**任意流程**］＞［**任意表單**］

位置：［**主選單**］＞我的任務＞［**任意表單**］

「表單」的相關應用為OMFLOW的主要功能，在流程中也時常需要查詢其他表單資料作為後續參數資料使用。抑或是滿足條件後需要同時推動其他表單繼續按照流程前進，因此針對「表單」設計了以下function供呼叫：

## 開單

開立表單是最常見的功能，一般在OMFLOW介面上操作如下面位置：

位置：［**主選單**］＞**Apps**＞［**任意應用**］＞［**任意流程**］＞**新增**

位置：［**主選單**］＞**服務請求**＞［**任意服務請求**］＞填寫完**送出**

雖然流程設計已提供「外部流程、內部流程」元件進行開單動作，但以上元件會等待新開的流程結束才會回到原流程上繼續前進，如設計者需要在不等待新開的流程結束便直接前進，則需要用到此代碼：

#匯入

from omflow.syscom.tools import OmData

#宣告一個流程的物件

api\_path = 'incident-managment'

inc\_flow\_obj = OmData(api\_path)

#填入表單資料(非必要，視流程需求)

inc\_flow\_obj.setFormData('formitm\_1', 'Apache服務異常')

inc\_flow\_obj.setFormData('formitm\_2', 'red')

inc\_flow\_obj.setFormData('formitm\_3', '1')

#填入流程變數(非必要，視流程需求)

inc\_flow\_obj.setFlowVariable('message', 'success')

inc\_flow\_obj.setFlowVariable('password', '1234567')

#送出開單

user\_id = '1' #使用者編號

files = None #檔案

result = inc\_flow\_obj.create(user\_id, files)

#取得回傳

data\_no = result['data\_no'] #單號

status = result['status'] #狀態True/False

message = result['message'] #錯誤訊息，開單成功則為空字串

## 推單

當表單流程行進「人工處理」元件時會停下，直到使用者手動或呼叫web API對此表單進行「推單」動作才會繼續前進，一般在OMFLOW介面操作如下：

位置：［**主選單**］＞**Apps**＞［**任意應用**］＞［**任意流程**］＞［**任意表單**］＞**送出**

位置：［**主選單**］＞**我的任務**＞［**任意表單**］＞［**快速操作**］

除「手動、呼叫web API」方式以外，也可以在流程行進過程中對特定表單進行「推單」，code代碼如下：

#匯入

from omflow.syscom.tools import OmData

#宣告一個流程的物件

api\_path = 'incident-managment'

inc\_flow\_obj = OmData(api\_path)

#填入表單資料(非必要，視流程需求)

inc\_flow\_obj.setFormData('formitm\_1', 'Apache服務異常')

inc\_flow\_obj.setFormData('formitm\_2', 'red')

inc\_flow\_obj.setFormData('formitm\_3', '1')

#推單

data\_id = '345' #資料編號(可透過列表取得)

user\_id = '1' #使用者編號

files = None #檔案

result = inc\_flow\_obj.update(data\_id, user\_id, files)

#取得回傳

data\_no = result['data\_no'] #單號

status = result['status'] #狀態True/False

message = result['message'] #錯誤訊息，推單成功則為空字串

## 列表

當流程需要查詢特定的表單資料時，可使用code代碼如下：

位置：［**主選單**］＞**Apps**＞［**任意應用**］＞［**任意流程**］＞**篩選**

#匯入

from omflow.syscom.tools import OmData

#宣告一個流程的物件

api\_path = 'incident-managment'

inc\_flow\_obj = OmData(api\_path)

#列表

search\_conditions = [

{

'column':'history',

'condition':'=',

'value':False

}, #查詢條件，查詢非歷史資料

{

'column':'running',

'condition':'=',

'value':False

}, #查詢條件，查詢非執行中資料

{

'column':'error',

'condition':'=',

'value':False

}, #查詢條件，查詢非異常資料

]

search\_columns = [

'data\_no',

'formitm\_1'

] #取得欄位，取得單號以及欄位1

exclude\_conditions = [

{

'column':'closed',

'condition':'=',

'value':True

}

] #排除條件，排除已關單資料

order\_columns = ['-data\_no'] #使用單號逆排序

limit = 0 #取得所有資料(若值給100代表取得前100筆資料)

start = 0 #從第1筆資料開始

result = inc\_flow\_obj.list(search\_conditions, search\_columns, exclude\_conditions, order\_columns, limit, start)

#取得回傳

#回傳詳細範例請參閱 REST API介紹>查詢表單

## 刪單

當表單流程過程中需要刪除特定表單時可使用code代碼，一般在OMFLOW介面操作如下：

位置：［**主選單**］＞**Apps**＞［**任意應用**］＞［**任意流程**］＞［**任意表單**］＞**刪除**

#匯入

from omflow.syscom.tools import OmData

#宣告一個流程的物件

api\_path = 'incident-managment'

inc\_flow\_obj = OmData(api\_path)

#刪單

data\_no\_list = [] #單號陣列

result = inc\_flow\_obj.delete(data\_no\_list)

#取得回傳

status = result['status'] #狀態True/False

message = result['message'] #錯誤訊息，推單成功則為空字

## 進階推單

此方法無須先取得表單data\_id，而是透過「條件查詢」直接將取得的表單依照後續設定進行「推單」動作，使用code代碼如下：

#匯入

from omflow.syscom.tools import OmData

#宣告一個流程的物件

api\_path = 'incident-managment'

inc\_flow\_obj = OmData(api\_path)

#填入表單資料(非必要，視流程需求)

inc\_flow\_obj.setFormData('formitm\_1', 'Apache服務異常')

inc\_flow\_obj.setFormData('formitm\_2', 'red')

inc\_flow\_obj.setFormData('formitm\_3', '1')

#接續以下

#推單

condition = {} #查詢條件

'''

condition可接受兩種資料型態(list/dict)

情境一、查詢 單號等於5 的單並推進

list寫法：

condition = []

con = {'column':'data\_no','condition':'=','value':5}

condition.append(con)

dict寫法：

condition = {'data\_no':5}

情境二、查詢 單號大於5 的單並推進

list寫法：

condition = []

con = {'column':'data\_no','condition':'>','value':5}

condition.append(con)

dict寫法：

condition = {'data\_no\_\_gt':5}

情境三、查詢 單號大於5且某個輸入欄位的值等於test 的單並推進

list寫法：

condition = []

con1 = {'column':'data\_no','condition':'>','value':5}

condition.append(con1)

con2 = {'column':'formitm\_1','condition':'=','value':'test'}

condition.append(con2)

dict寫法：

condition = {'data\_no\_\_gt':5, 'formitm\_1':'test'}

'''

user\_id = '1' #使用者編號

files = None #檔案

update\_duplicate = False

#當條件查詢回來的資料為複數筆時，是否將其全部推進。若否，則全部不推進並回傳失敗訊息。

update\_duplicate\_interval = 0

#當推進的資料為複數筆時，每一筆推進的間隔時間(秒)

wait\_time\_max = 0

#當查詢回來的資料筆數為0時，要進行n次重新查詢，n次查詢都為0後回傳錯誤訊息。

wait\_time\_seconds = 0 #每次重新查詢的間隔時間(秒)

result = inc\_flow\_obj.advanced\_update(condition, user\_id, files, update\_duplicate, update\_duplicate\_interval, wait\_time\_seconds, wait\_time\_max)

#取得回傳

status = result['status'] #狀態True/False

message = result['message'] #回傳訊息。成功推單的data\_id或是失敗訊息。

# 使用者

位置：［**主選單**］＞**人員管理**＞**使用者管理**＞［**任意使用者**］

在流程中如需要「查詢使用者」資料或對使用者相關資料進行「新增、修改、刪除」動作，因此設計了以下function供呼叫：

## 建立使用者**1.1.6.0**

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#填入使用者資料(必填)

username = 'a12345'

token = 'aA!123456'

nick\_name ='王小明'

email = 'a12345@gmail.com'

#填入使用者資料(選填)

#有下列欄位可填

#birthday, gender, phone1, phone2, company, default\_group, ad\_no, extension\_no

other\_param\_dict = {}

other\_param\_dict['ad\_no'] = '001234'

other\_param\_dict['phone1'] = '0911111111'

#建立使用者

result = user\_obj.create(username, token, nick\_name, email, other\_param\_dict)

#取得回傳

user\_id = result['user\_id'] #使用者id

status = result['status'] #狀態True/False

message = result['message'] #錯誤訊息，開單成功則為空字串

## 更新使用者

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#選擇要修改的使用者(必填，user\_id與username二選一填入)

user\_id = None

username = None

#更改密碼(選填)

token = 'aA!123456'

#填入要更新的使用者欄位(選填)

#有下列欄位可填

#email, birthday, gender, phone1, phone2, company, default\_group, ad\_no, extension\_no

other\_param\_dict = {}

other\_param\_dict['phone1'] = '0911111111'

#更新使用者

result = user\_obj.update(user\_id, username, token, other\_param\_dict)

#若 result = 'success' 代表成功，失敗則會回傳錯誤訊息

## 查詢使用者(列表)

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#查詢條件(下為範例，請依照實際需求修改)

search\_conditions = [

{

'column':'gender',

'condition':'=',

'value':'female'

}, #查詢條件，查詢女性使用者

]

search\_columns = [

'username',

'email'

] #取得欄位，取得帳號以及信箱

exclude\_conditions = [

{

'column':'is\_active',

'condition':'=',

'value':False

}

] #排除條件，排除停用的使用者

order\_columns = ['-username'] #使用帳號逆排序

limit = 0 #取得所有資料(若值給100代表取得前100筆資料)

start = 0 #從第1筆資料開始

#查詢使用者

result = user\_obj.list(search\_conditions, search\_columns, exclude\_conditions, order\_columns, limit, start)

#取得回傳

#回傳詳細範例請參閱 REST API介紹>使用者

## 查詢使用者(單一)

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#查詢條件(必填，下列三選一填入)

user\_id = None

username = None

nick\_name = None

#填入要查詢的使用者欄位

#有下列欄位可填

#id, username, nick\_name, email, birthday, gender, phone1, phone2, company, default\_group, ad\_no, extension\_no

#若留空代表查詢全部欄位

user\_attr = []

#查詢使用者

result = user\_obj.load(user\_id, username, nick\_name, user\_attr)

#取得回傳(請依照查詢的欄位取得資料)

#result資料結構如下

{

'id':123,

'username':'12312',

'nick\_name':'王小明',

'email':'123@gamil.com', #上述欄位會依照user\_attr變數不同而有所不同

'groups':[], #groups與roles則是固定提供之回傳，與user\_attr填入與否無關

'roles':[], #groups回傳使用者所在部門id陣列, roles回傳使用者所在的角色id陣列

}

## 查詢使用者Email(列表)

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#填入要查詢的使用者id(陣列)

user\_id\_list = []

#查詢多位使用者email

result = user\_obj.listEmail(user\_id\_list)

#result資料格式如下

['aaa@gmail.com','bbb@gmail.com','ccc@gmail.com','ddd@gmail.com']

## 刪除使用者

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#填入要查詢的使用者帳號(陣列)

username\_list = []

#刪除使用者

result = user\_obj.delete(username\_list)

#result若等於空字串代表刪除成功，回傳任何文字訊息代表是錯誤訊息

## 使用者加入部門

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#選擇使用者(必填)

user\_id = None

#填入要加入的部門id陣列(必填)

group\_id\_list = []

#使用者加入部門

result = user\_obj.addtoGroup(user\_id, group\_id\_list)

#result為boolean，True代表成功，False代表失敗

## 使用者移出部門

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#選擇使用者(必填)

user\_id = None

#填入要移出的部門id陣列(必填)

group\_id\_list = []

#使用者移出部門

result = user\_obj.removefromGroup(user\_id, group\_id\_list)

#result為boolean，True代表成功，False代表失敗

# 部門/角色

位置：［**主選單**］＞**人員管理**＞**部門管理**＞［**任意部門**］

位置：［**主選單**］＞**人員管理**＞**角色管理**＞［**任意角色**］

OMFLOW 1.1.6.0版本後，在流程中如需要查詢「部門、角色」資料或對其進行「新增、修改、刪除」動作，因此設計了以下function供呼叫：

## 建立部門

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_obj = Group()

#填入資料(必填)

group\_name = 'test' #部門名稱

is\_role = False #填入True為角色，False為部門

#填入資料(選填)

parent\_group\_id = None #父部門id，角色不可填

description = '' #說明欄位

group\_no = '' #部門代號

order = 0 #部門排序，數字越小排序越靠前，請填入整數(可以為負)

#建立部門

result = group\_obj.create(parent\_group\_id, group\_name, description, group\_no, is\_role, order)

#取得回傳

group\_id = result['group\_id'] #部門/角色id

status = result['status'] #狀態True/False

message = result['message'] #錯誤訊息，開單成功則為空字串

## 取得部門資訊

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#讀取部門資訊

result = group\_obj.load()

#取得回傳

group\_id = result.get('id','') #部門id

display\_name = result.get('display\_name','') #部門名稱

parent\_group\_id = result.get('parent\_group\_id','') #父部門id

description = result.get('description','') #說明

group\_no = result.get('group\_no','') #部門代號

group\_user\_list = result.get('group\_user\_list','') #該部門底下有的使用者id

permissions = result.get('permissions','') #該角色有的權限，查詢部門時回傳為空陣列

order = result.get('order ','') #部門排序

## 加入使用者

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#填入使用者id列表

user\_id\_list = []

#部門加入使用者，result為boolean，True代表成功，False代表失敗

result = group\_obj.addUsers(user\_id\_list)

## 移出使用者

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#填入使用者id列表

user\_id\_list = []

#部門移出使用者，result為boolean，True代表成功，False代表失敗

result = group\_obj.removeUsers(user\_id\_list)

## 查詢部門職務權責

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#填入使用者id列表

user\_id\_list = []

#部門移出使用者，result為boolean，True代表成功，False代表失敗

result = group\_obj.removeUsers(user\_id\_list)

## 更新部門職務權責

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#填入使用者id列表

user\_id\_list = []

#部門移出使用者，result為boolean，True代表成功，False代表失敗

result = group\_obj.removeUsers(user\_id\_list)

# 組織圖

位置：［**主選單**］＞**人員管理**＞**組織圖**＞［**任意組織圖**］

OMFLOW 1.1.6.0版本後，在流程中如需要查詢「組織圖」資料或對其進行「新增、修改、刪除」動作，因此設計了以下function供呼叫：

## 取得組織圖資訊

OMFLOW版本 **1.1.6.3** 功能：

#匯入

from omflow.syscom.tools import OrgChart

#宣告一個組織圖的物件

org\_name = ''

org\_config\_id = ''

#組織圖名稱/組織圖編號，請擇一填入值

org\_obj = OrgChart(org\_name, org\_config\_id)

#讀取組織圖資訊

result = org\_obj.load()

result = group\_obj.removeUsers(user\_id\_list)

回傳資料範例如下所示：

{

"org\_name": <組織圖名稱>,

"value": [{

"org\_id": <組織元件編號>,

"group\_no": <組織元件代號>,

"group\_name": <組織元件名稱>,

"parent\_group\_no": <父組織元件代號>,

"order": <組織元件排序>,

"host\_user": <託管人>, #使用者編號

"host": <是否被託管>, #True/False

"is\_host": <是否屬於託管組織>, #True/False

"job\_role": [{

"job\_role\_id": <資料編號>,

"position\_id": <職務編號>,

"position\_no": <職務代號>,

"position\_name": <職務名稱>,

"user\_id": <使用者編號>,

"nick\_name": <使用者顯示名稱>,

"is\_active": <使用者是否啟用>, #True/False

"responsibility": <權責>,

"data\_index": <排序>

}

],

"sub\_item": <託管組織>,

"parent\_group\_name": <父組織元件名稱>

},........#依此類推

],

"org\_approver": {

"group": <審核人部門編號>,

"user": <審核人使用者編號>

},

"is\_search\_dept": <組織圖查詢不到時，查詢部門架構>, #True/False

"is\_sub": <是否為託管的組織圖>, #True/False

"parent\_chart\_name": <主圖名稱>,

"uid": <uid>,

"pending\_status": <託管組織圖是否為審核狀態>, #True/False

"host\_user": <託管人>,

"org\_config\_id": <組織圖編號>

}

## 更新/新增組織圖

from omflow.syscom.tools import OrgChart #匯入

org\_obj = OrgChart() #宣告一個組織圖的物件

org\_chart\_dict = {

'org\_name': <組織圖名稱>,

'value': [{

'org\_id': <組織元件編號>, #修改舊資料時填寫

'group\_no': <組織元件代號>,

'group\_name': <組織元件名稱>,

'parent\_group\_no': <父組織元件代號>,

'order': <組織元件排序>,

'host\_user': <託管人>, #使用者編號

'host': <是否被託管>, #True/False

'is\_host': <是否屬於託管組織>, #True/False

'job\_role': [{

'job\_role\_id': <資料編號>, #修改舊資料時填寫

'position\_no': <職務代號>,

'user\_id': <使用者編號>,

'responsibility': <權責>,

'data\_index': <排序>

}]

},........#依此類推

],

'org\_approver': {

'group': <審核人部門編號>,

'user': <審核人使用者編號>

},

'is\_search\_dept': <組織圖查詢不到時，查詢部門架構>, #True/False

'is\_sub': <是否為託管的組織圖>, #True/False

'uid': <uid>,

'org\_config\_id': <組織圖編號> #修改舊資料時填寫

}

result = org\_obj.update(org\_chart\_dict) #更新/新增組織圖資訊

#取得回傳

status = result['status'] #狀態success/update error

message = result['message'] #錯誤訊息，執行成功則為空字串

op\_result = result['result'] #執行結果，執行成功則為空字串

## 刪除組織圖

#匯入

from omflow.syscom.tools import OrgChart

#宣告一個組織圖的物件

org\_obj = OrgChart()

org\_config\_id\_list = [<組織圖編號>,........]

#刪除組織圖

result = org\_obj.delete(org\_config\_id\_list )

#取得回傳

status = result['status'] #狀態success/delete error

message = result['message'] #錯誤訊息，執行成功則為空字串

op\_result = result['result'] #執行結果，執行成功則為空字串

注意要點：

1. 若要刪除託管圖，應更新主圖資訊解除託管，而非以此方法。

## 透過【同系職務/權責】尋找人員

OMFLOW版本 **1.1.6.3** 更動後，可輸入「使用者編號、職務/權責」並透過「組織圖、部門」查詢符合的使用者，此方法也可在程式碼元件使用，code代碼如下：

#匯入

from omflow.syscom.tools import User

#宣告一個流程的物件

user\_obj = User()

#選擇使用者(必填)

user\_id = None

#部門代號(選填)，指定要從哪個部門開始查詢，未填寫則從使用者的預設部門開始查詢

group\_no=None

#填入要查詢對象的職務名稱或職務代號或權責名稱(必填，下列三選一填入)

position\_name = None

position\_no = None

responsibilitie\_name = None

#選擇要從組織圖或部門架構查詢，如果dept\_search是False，請填寫組織圖名稱

dept\_search=False

org\_name=''

#查詢組織圖

result = user\_obj.getPosition(user\_id, group\_no, org\_name, position\_name, position\_no, responsibilitie\_name, dept\_search)

#取得回傳

manager\_user\_id = result.get('user\_id','')

manager\_group\_id = result.get('group\_id','')

## 透過【部門職務/權責】尋找人員

OMFLOW版本 **1.1.6.3** 更動後，可輸入「部門代號、職務/權責」並透過「組織圖、部門」查詢符合的使用者，此方法也可在程式碼元件使用，code代碼如下：

#匯入

from omflow.syscom.tools import Group

#宣告一個部門(角色)的物件

group\_id = ''

group\_no = ''

group\_name = ''

#部門id/部門代號/部門名稱請三選一填入值

group\_obj = Group(group\_id, group\_no, group\_name)

#填入要查詢對象的職務名稱或職務代號或權責名稱(必填，下列三選一填入)

position\_name = None

position\_no = None

responsibilitie\_name = None

#選擇要從組織圖或部門架構查詢，如果dept\_search是False，請填寫組織圖名稱

org\_name=''

dept\_search=False

#查詢組織圖

result = group\_obj.getPosition(org\_name='', position\_name=None, position\_no=None, responsibilitie\_name=None, dept\_search=False)

#取得回傳

manager\_user\_id = result.get('user\_id','')

manager\_group\_id = result.get('group\_id','')