

HW2 Serverless Computing

r08942088 李政旻

1. A report that include some discussions and experiences learned.

網站連結：http://34.70.158.85/cloud/hw2/call_function.html



The screenshot shows a web interface titled "Combine 2 images" in a green header. Below the header, there are two sections for image selection. The first section is labeled "第一個圖片：" and contains a "瀏覽..." button and the text "未選擇檔案。". The second section is labeled "第二個圖片：" and also contains a "瀏覽..." button and the text "未選擇檔案.". Below these sections is a label "貼合方式(圖片一位置)：" followed by a text input field containing the word "left". At the bottom of the form is a green "上傳" button.

作用：將兩張圖片合成一張，可選擇貼合方式，目前只有上下左右四種。

情境：列印學生證、身分證的時候可以少用一張紙，環保愛地球。但沒有提供裁剪圖片，列印的時候證件會變小，所以還是用PPT做可能比較快。

操作：選擇兩張圖片以及貼合方式，點擊上傳，然後稍等一下，結果就會顯示出來了。

Combine 2 images

第一個圖片：

017.jpg

第二個圖片：

025.jpg

貼合方式(圖片一位置)：

left



Combine 2 images

第一個圖片：

017.jpg

第二個圖片：

025.jpg

貼合方式(圖片一位置)：

left



Google cloud function 設定：

Google Cloud Platform

Cloud Computing 2020

Cloud Functions

編輯函式

hw2

us-central1

觸發條件

HTTP

觸發條件類型

HTTP

網址

https://us-central1-inspired-berm-292410.cloudfunctions.net/hw2

儲存

取消

變數、網路及進階設定

進階

環境變數

連線

輸入設定

☒ 允許所有流量

☐ 僅允許內部流量

☐ 允許內部流量和來自 Cloud 負載平衡器的流量

僅能接收相同專案或虛擬私人雲端 SC 範圍內的流量。

可接收相同專案、相同虛擬私人雲端 SC 範圍內或者來自 Cloud 負載平衡器的流量。

輸出設定

根據預設，函式可以向網際網路傳送要求，但無法向虛擬私人雲端網路中的資源傳送要求。如要向虛擬私人雲端網路中的資源傳送要求，請建立或選取虛擬私人雲端連接器。

虛擬私人雲端連接器

建立連接器

☐ 僅透過虛擬私人雲端連接器將要求轉送至私人 IP

☐ 透過虛擬私人雲端連接器轉送所有流量

下一步

取消

main.py:

1. 設定header，開啟跨站請求

```
# Set CORS headers for preflight requests
if request.method == 'OPTIONS':
    # Allows GET requests from origin https://mydomain.com with
    # Authorization header
    headers = {
        'Access-Control-Allow-Origin': '*', #'http://34.70.158.85',
        'Access-Control-Allow-Methods': 'POST',
        'Access-Control-Allow-Headers': 'Authorization',
        'Access-Control-Max-Age': '3600',
        'Access-Control-Allow-Credentials': 'true'
    }
    return ('', 204, headers)

# Set CORS headers for main requests
headers = {
    'Access-Control-Allow-Origin': '*', #'http://34.70.158.85',
    'Access-Control-Allow-Credentials': 'true'
}
```

2. 載入PIL模組，根據option將圖片合成，再編碼成base64格式回傳。

```
from PIL import Image
from io import BytesIO
import base64

if request.files:
    image1 = Image.open(request.files["image1"])
    image2 = Image.open(request.files["image2"])

    if request.form["option"] == "left":
        new_image = Image.new('RGB', (image1.size[0]+image2.size[0], max(image1.size[1],image2.size[1])), (250,250,250))
        new_image.paste(image1, (0,0))
        new_image.paste(image2, (image1.size[0],0))
    elif request.form["option"] == "right":
        new_image = Image.new('RGB', (image1.size[0]+image2.size[0], max(image1.size[1],image2.size[1])), (250,250,250))
        new_image.paste(image2, (0,0))
        new_image.paste(image1, (image2.size[0],0))
    elif request.form["option"] == "top":
        new_image = Image.new('RGB', (max(image1.size[0],image2.size[0]), image1.size[1]+image2.size[1]), (250,250,250))
        new_image.paste(image1, (0,0))
        new_image.paste(image2, (0,image1.size[1]))
    elif request.form["option"] == "bottom":
        new_image = Image.new('RGB', (max(image1.size[0],image2.size[0]), image1.size[1]+image2.size[1]), (250,250,250))
        new_image.paste(image2, (0,0))
        new_image.paste(image1, (0,image2.size[1]))

    buffered = BytesIO()
    new_image.save(buffered, format="jpeg")
    result = base64.b64encode(buffered.getvalue()).decode("utf-8")
    return ({"state":True, "result":result},200,headers)
else:
    return ({"state":False, "msg":"unsupport"},200,headers)
```

requirements.txt: Pillow >= 8.0.0



Html: 根據jquery文件把contentType設定成false才能傳送file物件，並在收到回應的圖片後顯示在網頁中。

```
<script>
function upload(){
    $("#result").attr('src', "/static/loading.gif");

    let image1 = $("#image1").prop("files")[0];
    let image2 = $("#image2").prop("files")[0];
    let option = $("#option").val();
    let form = new FormData();
    form.append("image1", image1);
    form.append("image2", image2);
    form.append("option", option);
    //console.log(option);

    $.ajax({
        url: "https://us-central1-inspired-berm-292410.cloudfunctions.net/hw2", processData:false, contentType:false,
        //url: "", processData:false, contentType:false,
        type: "POST", data: form,
        success: function (data, textStatus) {
            //console.log(data);
            if(data["state"] && data["result"]){
                $("#result").attr('src', 'data:image/jpeg;base64,' + data["result"]);
            }
        },
        error: function (XMLHttpRequest, textStatus, errorThrown) {
            console.log(textStatus + errorThrown);
            return false;
        }
    });
}
```

討論：這次作業使用google cloud function，操作容易，但佈署較慢，必須先把功能都做出來再上傳，如果cloud function出什麼問題比較不好處理。

由於這次作業包含server在內都是自己架設，在思考要做什麼功能的時候，經常會碰到一個問題，「怎麼不用server做就好了？」。漸漸地才了解到，server和function通常不會是由同一方製作，一個單位只會在其中一邊擁有較高的自由。

如果是自己建立server，使用webhook的時機就是呼叫別人的API，比如google vision，提供一個特別強大的功能，自己可能做不來，或是簡訊王，替人發送簡訊，也是一般做不太到的。

如果server是建立在別人的平台，被限制了使用程式的權限，比如wordpress或是其他只提供前端頁面較高自由度的平台，需要特定的程式運算或是處理的時候就可以用cloud function。

2. A github URL which contains your source code of the project.

<https://github.com/a2134666/CloudComputingHW2>