龍華科技大學

校園入口網擴充案

單一登入整合開發指南

V 1.0

中華民國 113 年 11 月

德瑞數位科技股份有限公司

目　　錄

一、 前言 1

二、 單一登入運作架構說明 1

(一) 應用系統使用原有的登入驗證網頁 2

(二) 應用系統使用員工入口網的應用系統選單進行登入驗證 3

三、 應用系統向管理者申請使用單一登入服務 3

四、 整合範例說明 4

(一) 步驟 1 4

(二) 步驟 2 4

(三) 步驟 3 5

(四) 步驟 4 6

(五) 步驟 5 6

(六) 步驟 6 8

(七) 步驟 7 8

(八) 步驟 8 9

(九) 步驟 9 9

(十) 登出 OAuth 9

(十一) 存取網址說明 10

修訂歷史記錄欄

版本	出版日期	修訂章節	修訂原因	承辦者
1.0	113/11/11	All	初版	劉明輝

一、前言

本文件在說明龍華科技大學(以下簡稱本系統)，各應用系統廠商進行單一登入整合時的技術指南。提供單一登入系統基本架構與程式開發範例。

二、單一登入運作架構說明

本系統採用 OAuth2(OpenID Connect)協定，建構單一登入服務，提供院內內部各應用系統認證授權服務。本系統提供各應用系統 2 種不同的使用情境進行登入驗證。應用系統可依業務需求選擇適合的登入驗證方式。

情境一：應用系統希望仍保有自己的登入網頁，並整合本系統所提供的 OAuth2(OpenID Connect)使用者身分驗證程序。

情境二，應用系統捨棄或關閉原有的登入網頁，使用者皆由登入「員工入口網」後，點選該應用系統已配合 OAuth2(OpenID Connect)修改的登入網址(URL)，本系統將依 OAuth2(OpenID Connect)協定提供應用系統該使用者的已經驗證完成與其個人相關資料。

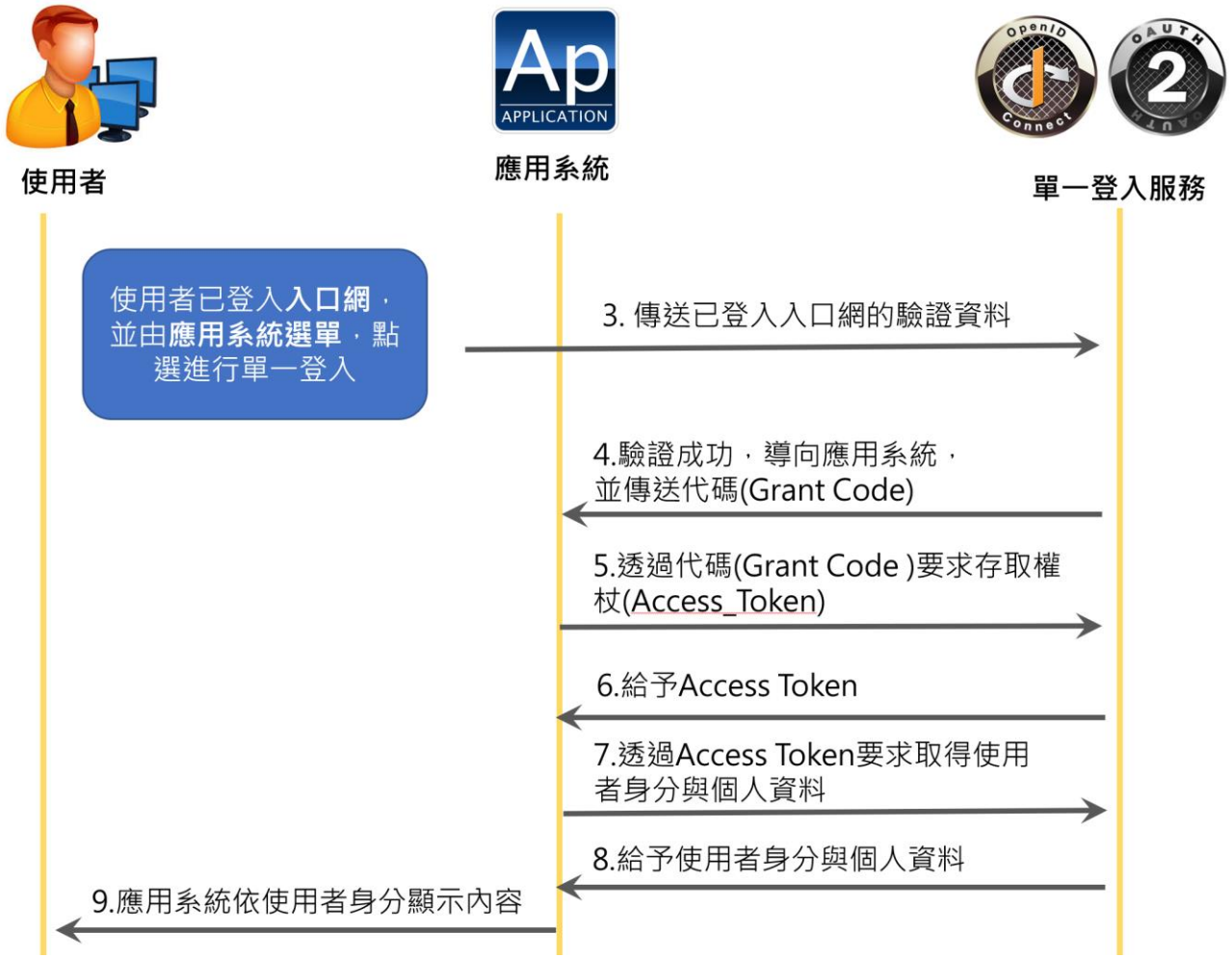
這 2 種情境可以並存使用，也可單獨使用，單一登入的功能完全一樣。以下針對這 2 種情境進行圖解說明。

(一) 應用系統使用原有的登入驗證網頁

使用這種驗證方式，應用系統需要執行以下 9 個步驟方能完成驗證程序並提供使用者應用系統的相關服務。請依下圖說明進行相關程式的修改，第三章將提供步驟 1 到步驟 9 的範例說明。

(二) 應用系統使用校園入口網的應用系統選單進行登入驗證

使用這種驗證方式，應用系統僅需要執行步驟 4 到步驟 9 的驗證程序，即可提供使用者應用系統的相關服務。請依下圖說明進行相關程式的修改，第三章將提供步驟 4 到步驟 9 的範例說明。



三、應用系統向管理者申請使用單一登入服務

應用系統開發廠商需先填寫「單一登入驗證服務-申請單」，待管理者審核後，才能核發應用系統所使用的帳號與密碼後，再進行整合作業。

四、整合範例說明

(一) 步驟1

使用者點選應用系統，並要求登入，應用系統請依以下的範例撰寫登入的按鈕或連結圖式。

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="utf-8" />
  <script>
    function Auth() {
      var URL = '驗證網址';  //驗證網址，由管理者提供，請參考表 1
          URL += '?response_type=id_token';
          URL += '&client_id=應用系統帳號';    //應用系統帳號須向管理者申請
          URL += '&redirect_uri=應用系統撰寫的接收網頁';  //應用系統在步驟 4 接收來自
                                                          //單一登入系統的 Call Back
                                                          //URL，應用系統須提供 URL
                                                          //給單一登入系統
        URL += '&state=' +new Date().getTime();  //state 為一個動態的變數值，防止瀏覽
                                                 //器 cache
        window.location.href = URL;
        }
  </script>
</head>
<body>
   <button onclick="Auth();">應用系統登入按鈕</button>
</body>
</html>
```

(二) 步驟2

如步驟 1 所示，應用系統已經將登入頁面導向單一登入頁面，使用者將在此步驟輸入其驗證的帳號與密碼。範例網頁示意圖如下。

若所傳入的資料有誤時，會回傳錯誤範例如下：

```
{
  "error":"unsupported_response_type",
  "error_description": "your input response_type is xxxxx"
}
```

回傳代碼	回傳訊息	備註
unsupported_response_type	response_type 不合法	目前 reponse_type 的值僅允許 id_token
unauthorized_client	client_id 不存在	
redirect_uri not same	redirect_uri 與申請資料不符	

(三) 步驟3

通過步驟 2 驗證成功後，單一登入系統將產生該使用者執行該應用系統的代碼(Grant Code)，並將該代碼回傳給該應用系統所提供的【應用系統撰寫的接收網頁】(Call Back Url)。

(四) 步驟4

應用系統自行撰寫【應用系統撰寫的接收網頁】(Call Back Url)程式接收代碼(Grant Code)，**參數名稱為 code**。

(五) 步驟5

應用系統需自行撰寫程式利用代碼(Grant Code)向單一登入系統要求要求存取使用者 id_token 資訊與存取進階使用者資訊的權杖(Access Token)。若使用者 id_token 解碼後的資料對應用系統已經足夠，就可以不用再執行步驟 6 以後的操作。以下是單一登入系統提供的要求網址、輸入參數與回傳值說明。

Access Token 要求網址: 請參考表 1

輸入參數：

參數名稱	說明
grant_type	authorization_code (固定值)
client_id	應用系統帳號
client_secret	應用系統密碼
code	單一登入伺服器回傳的資料
redirect_uri	Call Back Url

回傳資料：

參數名稱	型態	說明
id_token	字串	使用者資訊的 JWT
access_token	字串	存取使用者進階資訊的存取權杖

範例：使用 POST 方式來傳遞資料(client_id 與 client_secret 由管理者設定後提供)，並且 Content-Type 設定為 application/x-www-form-urlencoded

輸入參	grant_type: authorization_code client_id: openid1 client_secret: openid1

數	code: 1559813955611 redirect_uri: http://localhost:8080/callBackUrl.do
回傳資料	{ "**id_token**" : "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9. eyJkaXNwbGF5TmFtZSI6IueVoualrTk5IiwiY24iOiJkaXJlazk5IiwiZXhwIjoxNjg1MzI5MDk0fQ. exdUg0su8yTBl0cLnGaWXv-XUIwnECmoLB-meTihxEg ", "**access_token**" : "OFJLVGw2K1U3dXVwMzBrU3czaHhNUT09" }

若所傳入的資料有誤時，會回傳錯誤，範例如下：

{"error":" unauthorized_client", "error_description": "your input client_id is xxxxx"}

回傳代碼	代表意義	備註
unauthorized_client	連線 IP 不允許或者 client_id 錯誤	若是 IP 不允許,系統會回應目前存取的 IP 值。 若是 client_id 錯誤，系統會回應目前使用的 client_id 值
access_denied	client_secret 不正確	
redirect_uri not same	redirect_uri 與申請資料不符	

id_token 資料結構說明如下：

id_token 的值由三個部份組成，由「.」做為分隔，每個部份的資料皆是經由 Base64URL 編碼而成，使用 Base64URL 解碼即可還原資料。Base64URL 的解碼與常見的 Base64 解碼方式有一些差異，詳細可參考 https://base64.guru/standards/base64url。以下針對這三個部份說明：

1. Header: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
    - base64 解碼後 {"typ":"JWT","alg":"HS256"}
    - 代表使用 JWT 格式，並且使用的演算法為 HS256
2. Payload（claim）：

    eyJkaXNwbGF5TmFtZSI6IueVoualrTk5IiwiY24iOiJkaXJlazk5IiwiZXhwIjoxNjg1MzI5MDk0fQ

- base64 解碼後 {"givenName":"0099","cn":"dire000", "exp":1685329094}
- 資料內容：cn:使用者帳號，givenName:使用者姓名，exp: 簽發過期時間

3. Signature: exdUg0su8yTBl0cLnGaWXv-XUIwnECmoLB-meTihxEg
   使用加密演算法 HmacSHA256，與應用系統 client_secret 的值當成 key(範例中使用的 key 為 openid1)，將 Header 與 Payload 使用「.」串起來後(Header+"."+Payload)，並進行加密運算，所得出的值必須與 Signature 相符，才代表此資料是正確的。
4. JWT Library: 可參閱 https://jwt.io/#libraries-io

(六) 步驟6

應用系統利用步驟 5 取得合法的存取權杖(Access_Token)。

(七) 步驟7

使用存取權杖(Access_Token)取得更多的使用者進階的資料。應用系統需自行撰寫程式利用存取權杖(Access_Token)向單一登入系統要求取得使用者進階身分與其他使用者個人資料。

要求網址: 請參考表 1

輸入 HttpHeader 參數：

參數名稱	說明
Authorization	"Bearer "+access_token
Content-Type	application/x-www-form-urlencoded

回傳 JSON 資料：

型態	說明
字串	使用者進階欄位資料

範例：使用 POST 方式來傳遞資料

注意事項：access_token 是放在 HTTP 的 Header 中，非 HTTP 的 Body

中，並且 Bearer 後方有一個空白字串

輸入 Header 參數	Authorization: Bearer OFJLVGw2K1U3dXVwMzBrU3czaHhNUT09

回傳 JSON 欄位資料（由管理者設定）	{
	"givenName" : "OO99",
	"cn" : "direOOO"
	}

- 資料內容：cn:使用者帳號，givenName:使用者姓名

若所傳入的資料有誤或者 access_token 已過期時，會回傳錯誤，範例如下：

{"error":"invalid_request", "error_description": "your access_token xxxx is invalid or timeout"}

回傳代碼	代表意義	備註
invalid_request	所傳入的 access_token 不正確或已過期	
access_token not found	所傳入的 access_token 為空。常見的問題是沒有將 access_token 的值放至 HTTP Header 中來傳送	

(八) 步驟8

應用系統利用步驟 7 取得使用者身分進階相關資料。

(九) 步驟9

應用系統開始提供該使用者相關服務。

(十) 登出OAuth

若需要立即登出 OAuth，則可以在登出網址(參考表 1)後面，附加參數 redirect_uri，則可以立刻登出。登出後，會轉向 redirect_uri 所提供的網址。

範例： 登出網址?redirect_uri=https://ap logout url

(十一) 存取網址說明

表 1

功能說明	正式機網址內容
驗証網址	https://eportal.lhu.edu.tw/openidConnectServer.do
使用者資訊與 Access Token 要求網址	https://eportal.lhu.edu.tw /openidConnectServerToken.do
使用者進階身分資料要求網址	https://eportal.lhu.edu.tw/openidConnectServerInfo.do
登出網址	https://eportal.lhu.edu.tw /openidConnectServerLogout.do