【技術面】
新版 SSO 之應用程式介面（API）、認（驗）證技術規格

數位發展部
114年12月

議　　程

1. OIDC（OpenID Connect）協定簡介
2. 認證服務資訊流簡介
3. 介接範例說明
4. 其他參考資訊

OIDC（OpenID Connect）

協定簡介

OIDC協定簡介

定義

以OAuth 2.0為基礎，OIDC是一種結合身份驗證（Authentication）和授權（Authorization）的技術協議。

Web 1.0

傳統帳密登入驗證

使用者要記住**太多密碼，容易忘記或重複使用**。服務提供者必須儲存密碼，若資料庫被駭則使用者資料可能外洩。

Web 2.0

OAuth 2.0（授權機制誕生）

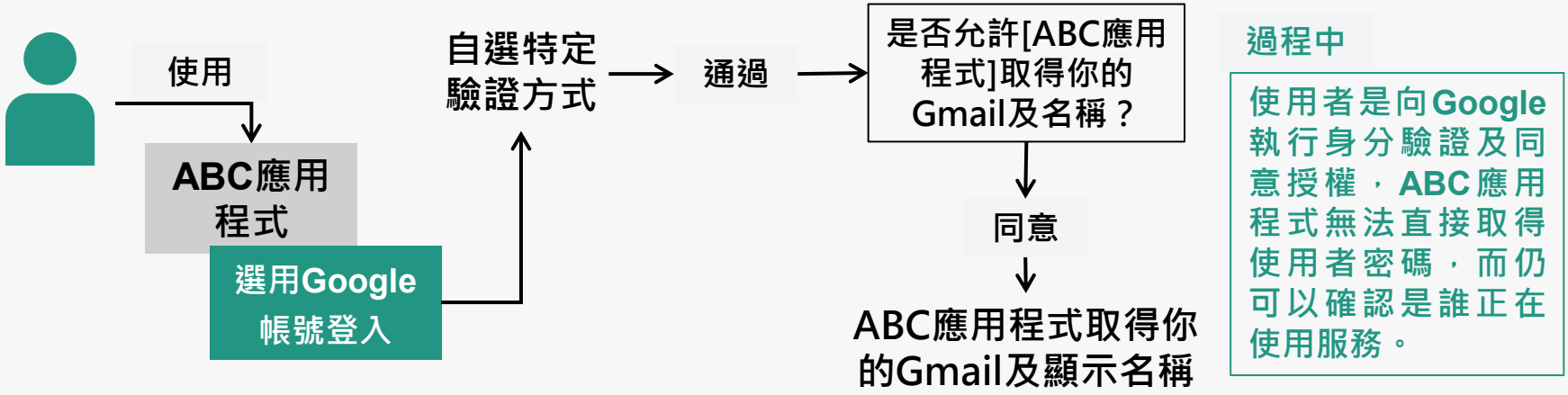
讓**使用者透過授權第三方應用程式存取特定資訊，而無須分享密碼**。例如你可以用「用 Google 帳號登入」Spotify，而 Spotify 只能依照授權取得你的信箱，無法知道你的Calendar內容。

Web 3.0

OpenID Connect（OAuth 2.0 + 身份驗證）

OIDC補足了OAuth 2.0於身份驗證應用上的缺陷。**透過實作ID Token，讓應用程式可以確認使用者的身份**，解決過去無法直接驗證使用者的問題。不僅在無密碼的基礎上強化安全，也使服務應用層面得以拓展。

常見應用

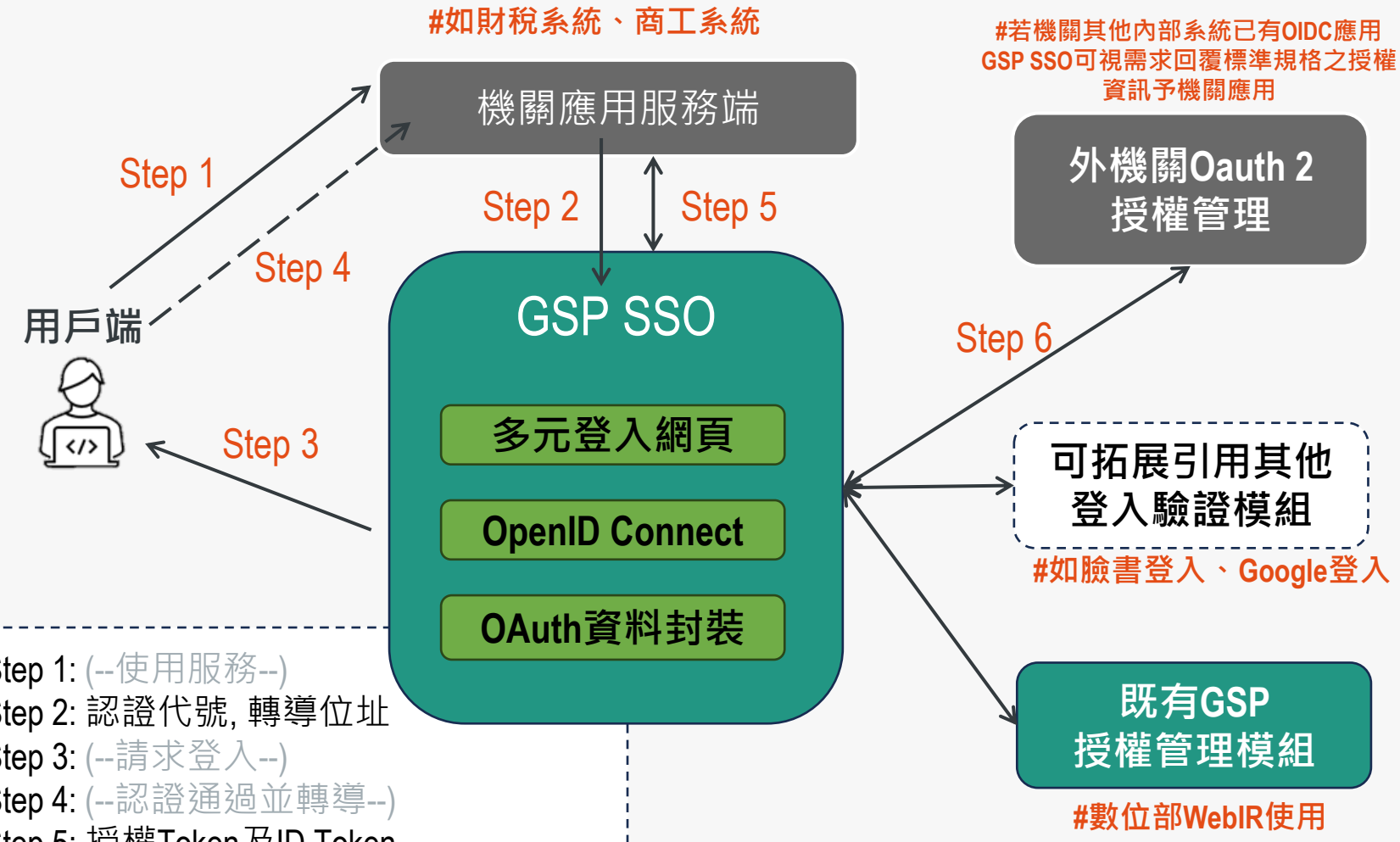


OIDC已是廣泛應用的企業級應用

- **全球科技巨頭與企業應用上皆有應用實例，受國際產業標準認可**：國際知名科技公司，如Google、Microsoft、Apple、Amazon皆有採用此標準格式、或提供此類標準登入認證功能予用戶使用，同時知名雲端SaaS平台，如 Okta、Auth0、Azure AD等亦支援OIDC，受到國際大型企業廣泛應用。
- **符合美國及歐盟等先進國家之安全標準**：OIDC支援無密碼登入（Passwordless），透過減少密碼依賴加以提高資訊安全性，OIDC設計與應用，符合GDPR、CIS、NIST等國際安全標準。
- **強化安全認證機制，2023-2024年間無重大資安漏洞**：基於OAuth 2.0協議且透過JWT（JSON Web Token）提供安全的Token驗證機制。可支援 PKCE原則（Proof Key for Code Exchange）加以防範中間人攔截，建立安全可靠的身分認證機制。

認證服務資訊流簡介

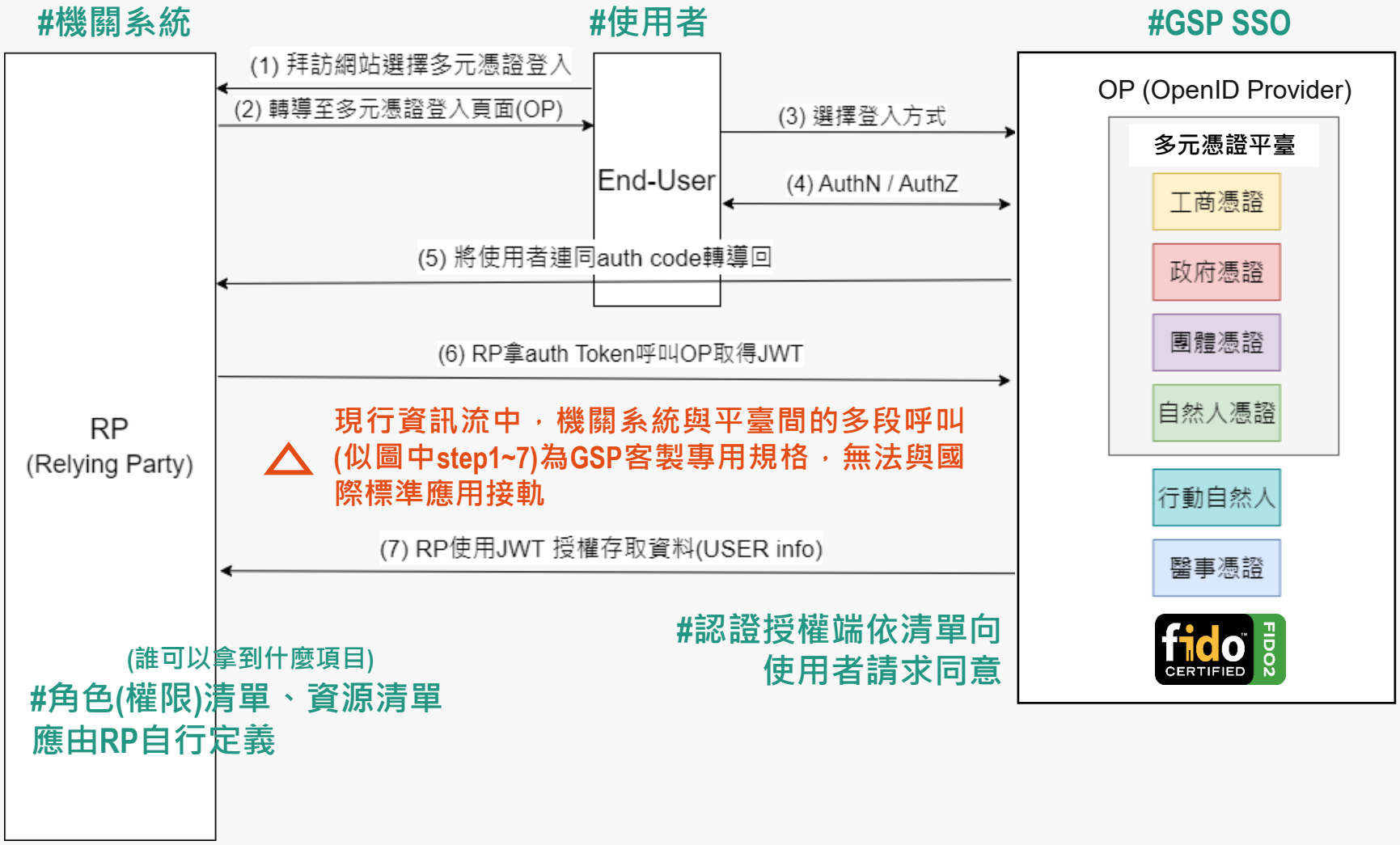
新版SSO服務資訊流摘要 (1/2)



- Step 1: (--使用服務--)
- Step 2: 認證代號, 轉導位址
- Step 3: (--請求登入--)
- Step 4: (--認證通過並轉導--)
- Step 5: 授權Token及ID Token
- Step 6: 授權Token

ID Token含應用服務所需個人識別資訊

新版SSO服務資訊流摘要 (2/2)



新版服務API清單

 # 將於114年12月發布完整API文件與範例程式碼，請以正式版資訊為主

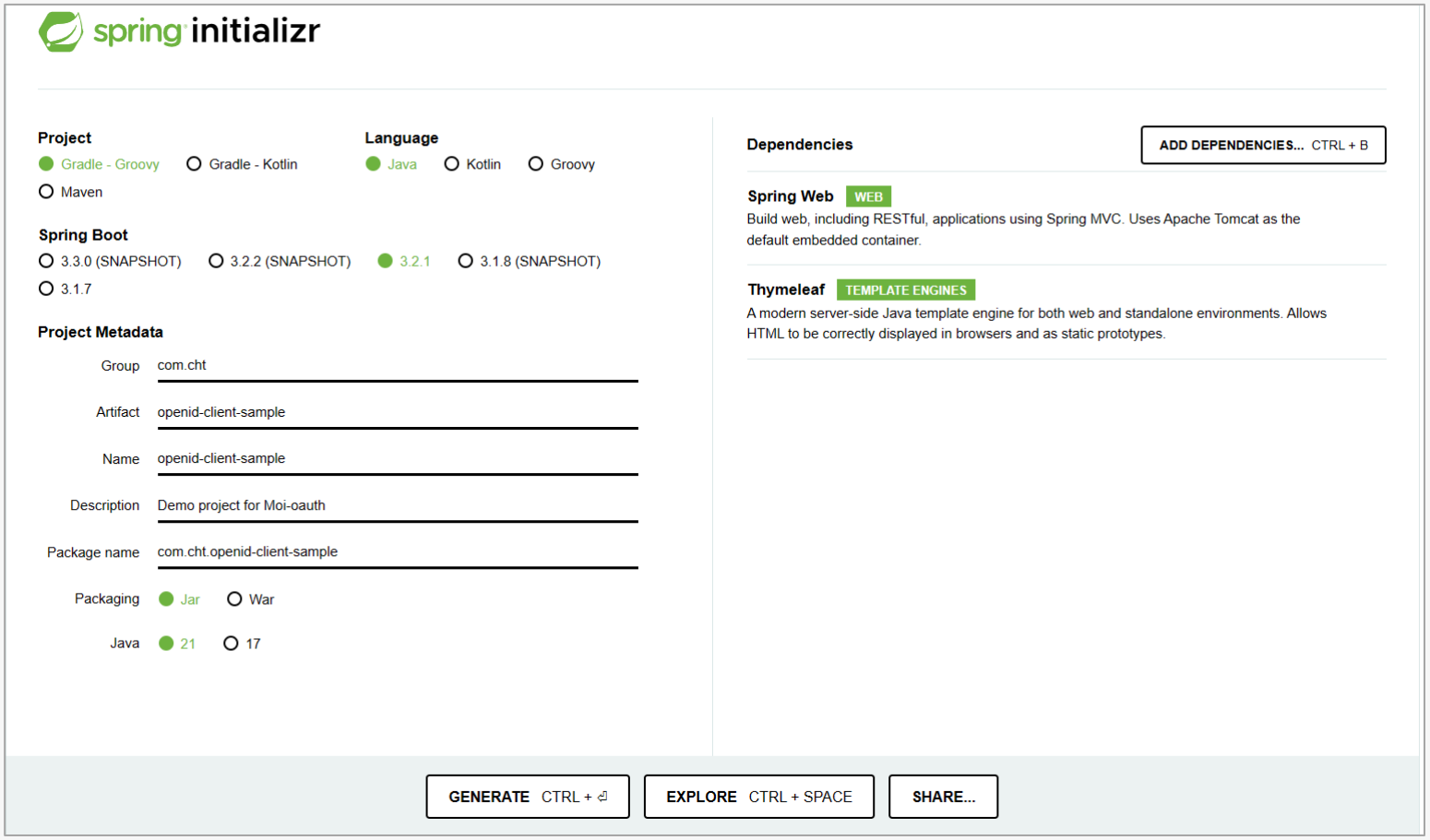
編號	用途說明
01	取得認證伺服器元資料，透過此API可取得認證伺服器之服務端點資訊 https://login.cp.gov.tw/realms/gsp-sso/.well-known/openid-configuration
02	向認證伺服器請求登入認證，會轉導登入頁面。並於登入完成後，由認證伺服器向需用機關系統發送授權碼(Authorization Code)進行後續請求認證與存取令牌使用。 https://login.cp.gov.tw/realms/gsp-sso/protocol/openid-connect/auth? (參數省略)
03	用戶端取得授權碼(code)後，向認證伺服器請求認證與存取令(access_token/id_token)，id_token中包含會員相關資料 https://___(Your website)___?state=___&session_state=___&iss=___&code=____ (參數省略)
04	機關系統以id_token向認證伺服器(GSP新SSO)發出登出請求： https://login.cp.gov.tw/realms/gsp-sso/protocol/openid-connect/logout
05	機關系統以id_token+證號向認證伺服器請求會員證號驗證，並回傳Y/N https://ssoapi.cp.gov.tw/auth/checkUserUid

介接範例說明

以Spring Boot專案為例

- 環境
  - Spring boot 3.2.1
  - Java 21

Spring initializer



引入相依套件

- 使用Maven/Gradle皆可

```
18  dependencies {
19      implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
20      implementation 'org.springframework.boot:spring-boot-starter-web'
21      implementation 'org.springframework.boot:spring-boot-starter-webflux'
22      testImplementation 'org.springframework.boot:spring-boot-starter-test'
23  }
```

實作轉導認證

以form submit進行認證轉導

- 須帶入正確的對應參數

client_id: 須提出申請，由平臺配發專用ID

redirect_uri: 填入指定返回的機關網址，註冊時需提供該資訊至平臺，非法url將被平臺阻擋

```
10  <form method="get" th:action="@{'https://authserver.local:9000/moi-oauth/oauth2/authorize'}">
11      <input type="hidden" name="response_type" value="code" />
12      <input type="hidden" name="client_id" value="cht-sample" />
13      <input type="hidden" name="redirect_uri" value="http://127.0.0.1:2412/authorized" />
14      <input type="hidden" name="scope" value="openid profile" />
15
16      <!-- Generate and include random state and nonce values -->
17      <input type="hidden" name="state" th:value="${T(java.util.UUID).randomUUID().toString().substring(0, 8)}" />
18      <input type="hidden" name="nonce" th:value="${T(java.util.UUID).randomUUID().toString().substring(0, 8)}" />
19
20      <button type="submit">多元憑證登入</button>
21  </form>
```

Demo

測試示範用首頁

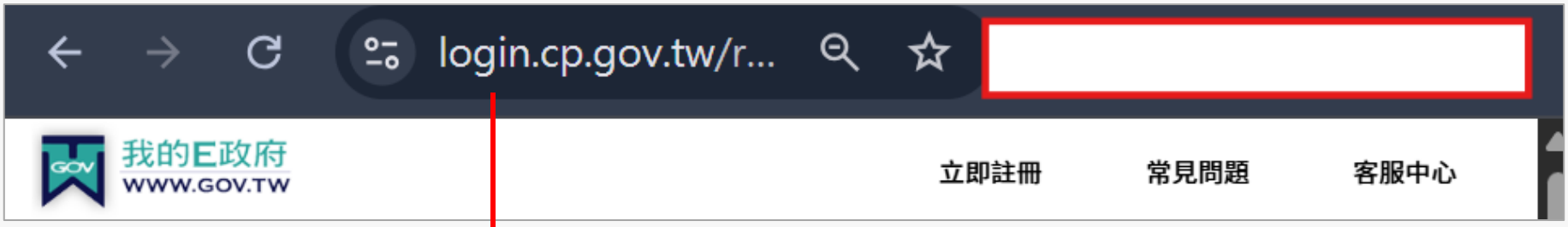
- 以平臺端提供之範例程式包示意
- 範例程式包的前置作業說明，以發佈文件為準

Demo - 轉導至登入頁

- 將使用者轉導至新版SSO的多元選項登入頁

Demo - 注意參數

- 機關系統呼叫時須帶入正確的client id, redirect_uri，否則將被平臺端阻擋



https://login.cp.gov.tw/realms/gsp-sso/protocol/openid-connect/auth?response_type=code&client_id=___&scope=openid&state=____&redirect_uri=___&nonce=____

Demo - 取得Authorization code

- GSP伺服器透過302轉導，會將code參數傳遞至當初註冊服務時提供的redirect_url
- 機關系統應用端須開立對應參數/欄位接取auth code, state等值

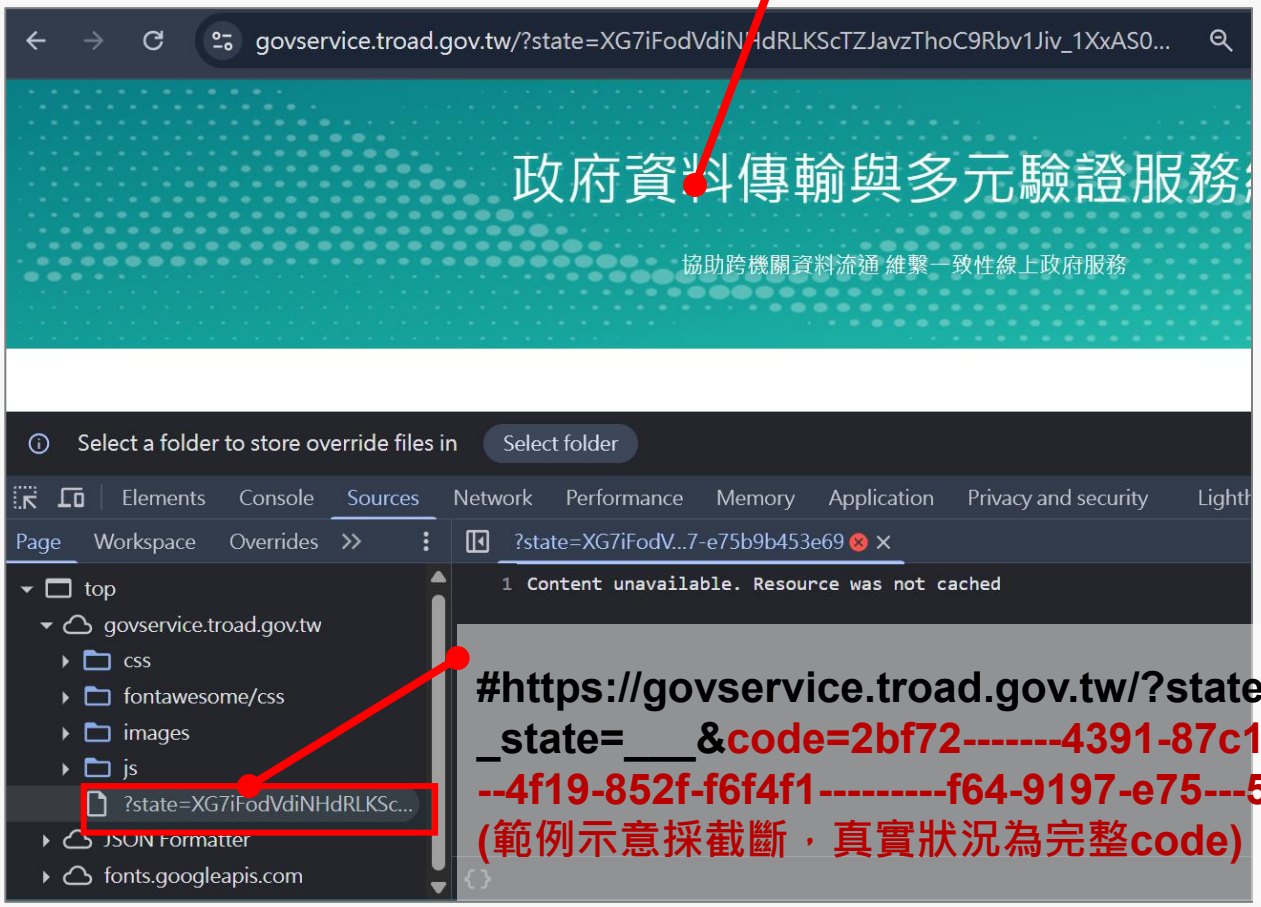
```
@Controller
@RequestMapping("/authorized")
public class OidcController {

    no usages
    @GetMapping
    public String oidcCallback(@RequestParam("code") String code,
                               @RequestParam("state") String state, Model model ) {
        // Add the received parameters to the model
        model.addAttribute( attributeName: "code", code);
        model.addAttribute( attributeName: "state", state);

        // Return the demo.html template
        return "demo";
    }
}
```

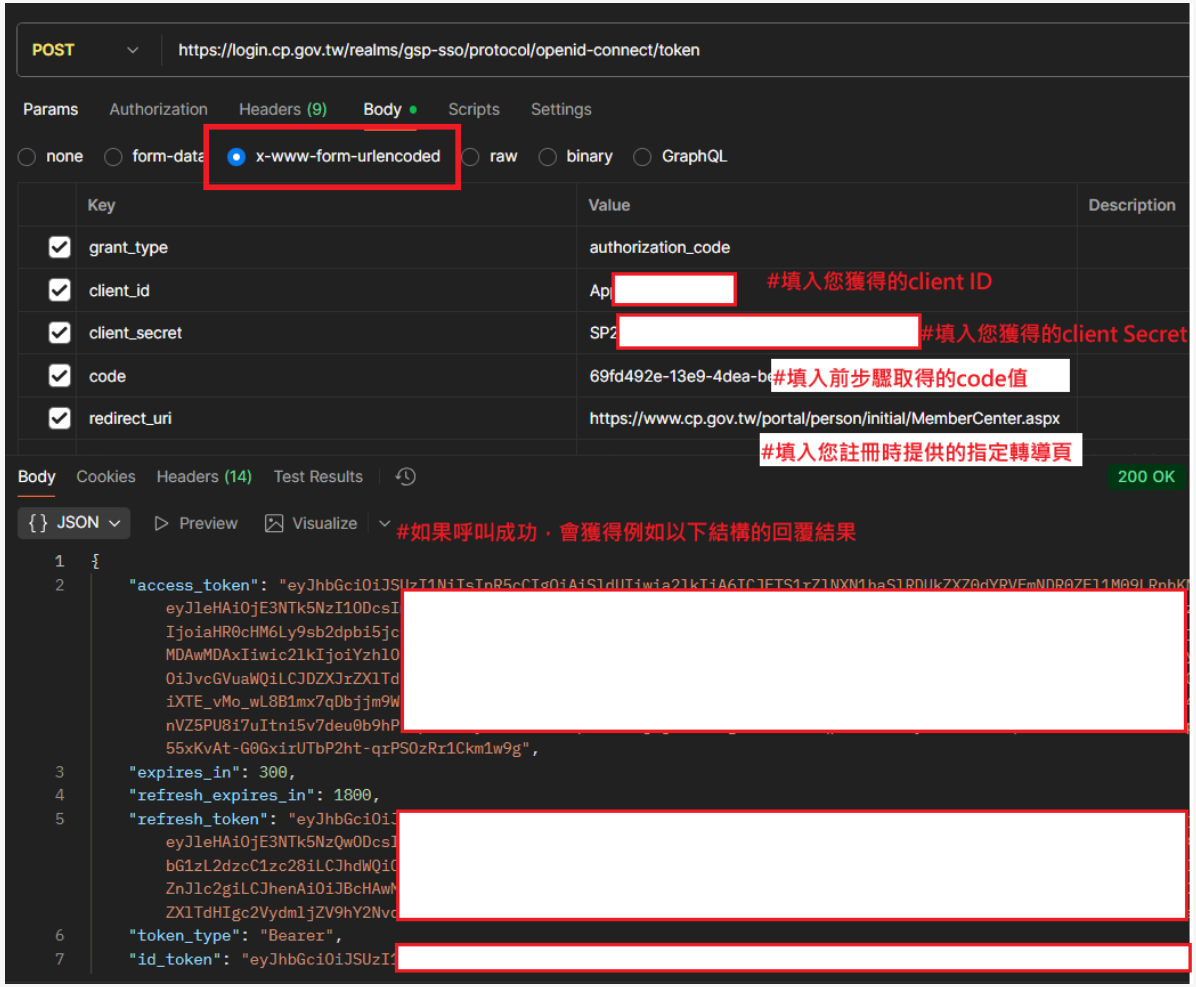
Demo - 取得Authorization code

- 範例: 透過302 redirect回傳code至機關系統



Demo – 取得id_token

- 執行code_to_token作業，機關系統可透過Http POST取得id_token
  - 詳細說明可參考未來(114年12月)發布的API文件
  - 如何發動http post因程式語言/框架等不盡相同，此處以工具postman示範測試



Demo – 取得id_token(續)

- 回覆結果中取得id_token


```
{
    "access_token": "eyJraWQiOiIyMjRjMTJlNi1iMjczLTRiNTctODljMy0wYmEyMWUyNDJlMTIiLCJhbGciOiJSUzI1NiJ9.
        eyJzdWIiOiIyLjUuNC41PSMxMzEwMzAzMDMwMzAzMDMwMzAzMTMxMzYzNTMyMzYzMzM0MzksQ0496YOt5Yag5b-XLEM9VFciLCJhdWQiOiJjaHQtc2FtcGxlIiwibmJmIjoxNzA1OTA3NzQzLCJzY29wZSI6WyJvcGVua
        zcyI6Imh0dHBzOi8vYXV0aHNlcnZlci5sb2NhbDo5MDAwL21vaS1vYXV0aCIsImV4cCI6MTcwNTkwODA0MywiaWF0IjoxNzA1OTA3NzQzLCJqdGkiOiI3MmE2MWQ4My1iYWUzLTQ1ZjEtYmVjMi1jN2Y4NGQxODNhZGY
        TO5A5u9Ufv0QYP1n8ftDQv-xB6Pg7IcJ4aGwyiP2WXaIBVC9uB_enKmH_Zzx16yZQruI9JZKoT03hPZ0g8Ww-Sbkzdv2Febkxaxl7KeIj7jc0KMzl57GQQrNXDLUQH4jeSQ9c943ym9qY8iX8xvCO8QH_hWjtDD_88hdB
        4oshSDlInN7H79cB5A4c0--QWPbaZeeL3nDFIi3enwpcIrKBmw7YpW2M1QYWYLgdPKN0Lv_s4IQU2tzlROvsg3h1TlWSGdxLdMRbtk15HZqk2qZAuAGeOsnxghcHpAPnUHCtbNre2ycv8ISTE0V2GekBCJg",
    "refresh_token": "jqD9ooRu9Kq9XWGdFIH89aYPidTc5xJ2zkvEOVSid7u4yyDSlsqete2QJiaVr5WpHOezH6-ABW0_G-OTGfPqbsrCI-7wKTrU4kG7epbxSNyy_j-OkKXey3xsHnBbozJr",
    "scope": "openid profile",
    "id_token": "eyJraWQiOiIyMjRjMTJlNi1iMjczLTRiNTctODljMy0wYmEyMWUyNDJlMTIiLCJhbGciOiJSUzI1NiJ9.
        eyJzdWIiOiIyLjUuNC41PSMxMzEwMzAzMDMwMzAzMDMwMzAzMTMxMzYzNTMyMzYzMzM0MzksQ0496YOt5Yag5b-XLEM9VFciLCJpc3MiOiJodHRwczovL2F1dGhzZXJ2ZXIubG9jYWw6OTAwMC9tb2ktb2F1dGgiLCJ0
        jZSI6IjZlYzgzZmMwIiwic2lkIjoibUpMc0luOGNabGREekhTRjNEbnYtTGJjdWhnd0lVbzA0RExIVW5hakJfNCIsImF1ZCI6ImNodC1zYW1wbGUiLCJ1aWQiOiJEMTIyMDQ5MzgxIiwiYXpwIjoiY2h0LXNhbXBsZSI
        kwNzU3MiwiZXhwIjoxNzA1OTA5NTQzLCJpYXQiOjE3MDU5MDc3NDMsImp0aSI6IjUwODM0ZTYzLTUyYzMtNDZjYy1iYmNiLTg2ZDg3NjhhNTI4NSJ9.
        b2LtPlvnaejzMl1tYhwkXIwtD012l98W-scTyHBgBgdIHwABW3-x1TiVHE4htq1DcHOcllI5jXFWTpVhAoC6sRYBePamiaMRXuIR-_0zmVBFDPHk0tlnkM_WM3aZJLYTabyApPgnFS1TBc3ycqLB-lVizBhCpqgGfOE3
        QXnhCazjEpr5cAorCF0zJDm-f_faTBYzKaPfP8V7clVF7SQCGNOtAVpkiFPBJw95B_5Od8_3uhJNh1k4LIxPwaYnY6PRZivbQH1scndlh-kYpC_1JRIU5Lny_00IDSWnneKOeaBdAPtXp12hG7BtwnISXjA",
    "token_type": "Bearer",
    "expires_in": 299
}
```

Demo – 取得id_token(續)


```
{
  “account”: “useraccount”, //使用者帳號
  “CerkeyStr”: “01142564…”, //憑證序號@憑證主體
  “ouoid”: “2.16.886.111.222”, //機關OID
  “snou”: “012341112”, // 機關代碼
  “servent”: “2”, // 0:非公務員, 1:正式公務員, 2:廣義公務員
  "type": "moica", // 使用者登入時所使用之憑證類別
  "aud": "oidc-client",
  "azp": "oidc-client",
  "auth_time": 1692781401,
  "iss": "http://127.0.0.1:9000",
  "exp": 1692783222,
  "iat": 1692781422,
  "sid": "czOr5LD8d547nQrIPZtnR5fViSQo2c2U_Asy3f801bk"
}
```


- 取出id_token，依JWT格式將其解碼後可得如圖內容，機關系統自行應用

- 系統回覆欄位依註冊服務時提出之申請為準

其他參考資訊

參考資料

- OpenID基金會 / How it works: https://openid.net/developers/how-connect-works/
- OpenID基金會/ JSON Web Token (JWT) and JSON Object Signing and Encryption (JOSE)各程式語言參考套件清單: https://openid.net/developers/jwt-jws-jwe-jwk-and-jwa-implementations/
- 多家知名科技公司針對OIDC提供說明與同意其安全性
  - 微軟: https://www.microsoft.com/zh-hk/security/business/security-101/what-is-openid-connect-oidc?utm_source=chatgpt.com
  - Google: https://developers.google.com/identity/openid-connect/openid-connect?hl=zh-tw&utm_source=chatgpt.com
  - 美國網安企業Palo Alto Networks: https://www.paloaltonetworks.com/blog/prisma-cloud/openid-connect-oidc-security/?utm_source=chatgpt.com
- 康乃爾大學研究者於2017年針對 OIDC 進行深入的形式化安全分析，研究人員建立詳細的形式模型，證明OIDC在身份驗證、授權和網路會話(session)完整性等各方面具備安全性。Citation: Fett, D., Küsters, R., & Schmitz, G. (2017, August). The web sso standard openid connect: In-depth formal security analysis and security guidelines. In 2017 IEEE 30th Computer Security Foundations Symposium (CSF) (pp. 189-202). IEEE. https://arxiv.org/abs/1704.08539?utm_source=chatgpt.com

報告完畢，敬請指教！