# 數位服務個人化

# 身分驗證與授權管理機制技術文件

v2.0

國家發展委員會

中華民國 107 年 6 月

# 目錄

# 壹、目的

本文件主要描述國發會「MyData 平台之身分驗證及授權管理機制」內容。

MyData 之身分驗證及授權機制實作符合 OpenID Connect (OIDC)規範 OIDC 是基於 RFC 6749 The OAuth 2.0 Authorization Framework 標準之上的一種 OAuth 2.0 協議。它使客戶端可以根據授權服務器執行的身分驗證來驗證最終用戶的身分，及以可互操作和類似 REST 的方式獲取有關最終用戶的基本配置文件信息。

OAuth 2.0 規範中提及四種授權作業流程，MyData 授權平台以其中的授權碼作業流程（Authorization Code Grant Type Flow）為主。

MyData 身分認證除了提供「eGov 帳號密碼驗證」外，亦提供「自然人憑證驗證」以增加用戶實名驗證的強度。

# 貳、 應用範圍

## 一、 身分驗證

服務提供者欲取得用戶同意授權前需引導用戶完成 MyData 身分驗證。內容細節請參考本文件章節「肆、伍、MyData 整合協作流程示意」。

## 二、 取得用戶同意授權

服務提供者欲取得資料提供者的使用者資源前，需先取得使用者同意授權。內容細節請參考本文件章節「肆、伍、MyData 整合協作流程示意」

## 三、 檢核授權 access_token

資料提供者於處理服務提供者之資源存取請求時，需檢核服務提供者是否夾帶使用者同意授權 token 及其是否為合法的 token。資料提供者透過 MyData 提供的 Introspection Endpoint 來完成這項檢核工作。內容細節請參考本文件章節「伍、五、Introspection Endpoints」。

### 四、 取得用戶資訊

服務提供者 SP 及資料提供者 DP，可透過 MyData 平台提供的 UserInfo Endpoint 取得用戶資訊，其中包括 eGov 帳號及用戶身分證字號，SP 及 DP 可藉此對應及綁定自身系統的會員帳號。內容細節請參考本文件章節「伍、六、UserInfo Endpoint」。

## 參、 名詞定義

| 名稱 | 定義 |
|---|---|
| OAuth 2.0 | 系統授權流程規範，定義於 RFC 6749<br>The OAuth 2.0 Authorization Framework<br>https://tools.ietf.org/html/rfc6749 |
| OpenID Connect | OAuth 2.0 的補充規範，強調身分驗證流程<br>http://openid.net/connect/ |
| eGov 帳號 | 我的 E 政府提供的會員帳號 |
| Data Provider, DP | 資料提供者，存放或保管民眾個人資料之機關單位。 |
| Service Provider, SP | 服務提供者，提供民眾進行個人資料之加值服務機關單位。 |
| Authorization Server, AS | 授權管理者，執行身分驗證與授權管理機制，本規範之授權管理者為本會政府服務平台(GSP)。 |
| Resource Owner, RO | 資料擁有者/使用者，泛指用戶或民眾。 |
| access_token | AS 發的用戶同意授權 |

# 肆、 身分驗證及授權流程說明

## 一、規範依據

MyData 平台之身分驗證機制實作 OpenID Connect 規範。OpenID Connect (OIDC)補充說明了 OAuth 2.0 中沒有特別說明的身分驗證（Authentication）部份，並擴展定義了 ID Token 及 UserInfo Endpoint 來解決第三方客戶端標識用戶身分認證的問題。

## 二、擴充定義

MyData 平台之身分驗證，除了 eGov 帳密驗證外，另提供以自然人憑證驗證方式，藉此加強帳戶實名制特性。

## 三、授權項目的描述

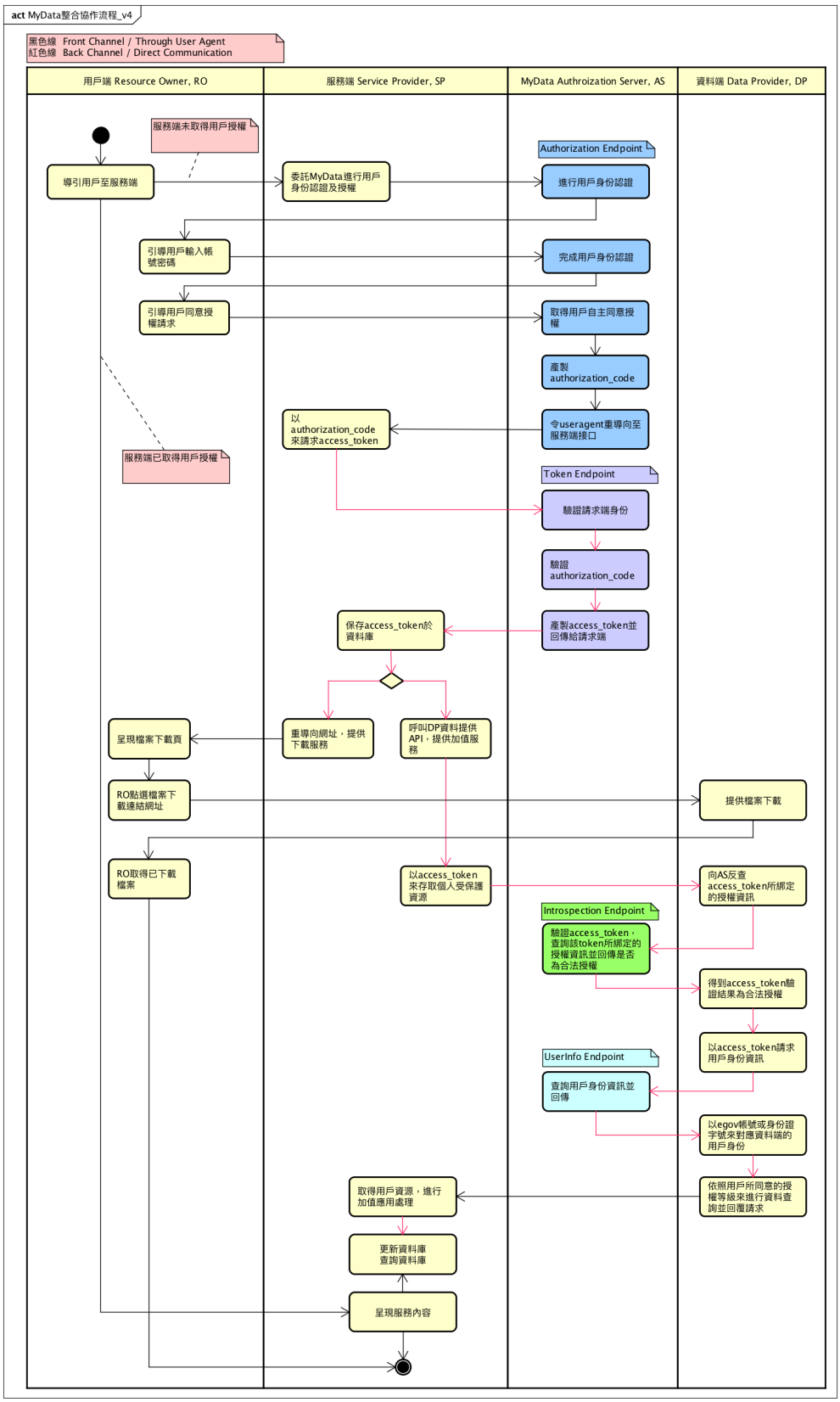
MyData 使用 scope 值來描述授權項目。scope 的使用方式被定義於 OAuth 2.0 規範中，如：openid, profile, offline_access...等。

針對 MyData 的授權管理需求，MyData 平台提供授權項目管理功能介面給資料提供者（DP）使用進行授權項目的管理。

## 四、請求授權的時機

服務提供者（SP）要求用戶同意授權的時機，應於必要取得該資源時才要求用戶同意給予存取該資源的授權，而非於登入時就一次性的要求用戶同意所有需要的授權。

# 五、MyData 整合協作流程示意



# 伍、 身分驗證及授權流程 API endpoints 規格說明

## 一、系統環境與條件

所有的 API endpoint 皆以 RESTful Service 方式提供介面，且皆基於 TLS v1.1 以上提供加密傳輸管道。

## 二、well-known/openid-configuration

MyData 授權主機提供設定資訊檔，說明系統參數如已支援的 endpoint, scope,...等等。

設定資訊檔網址，如下：

https://login.cp.gov.tw/v01/.well-known/openid-configuration

SP 應於呼叫任何 API 之前，先取得 MyData 授權主機提供的設定資訊。

## 三、Authorization Endpoint

此 endpoint 主要目的為讓 SP 從 RO 取得「同意授權許可」（authorization grant），同時令 RO 完成 DP 的身分驗證。

請參考 OAuth 2.0 及 OpenID Connect 規範內容。

### （一）身分驗證請求

請求網址示意：

```
GET /authorize?
    response_type=code
    &scope=openid%20profile%20email
    &client_id=s6BhdRkqt3
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1
```

參數/欄位說明：

| 參數/欄位 | 說明 |
| --- | --- |
| scope | 必要。<br>多筆以空格分隔，區分大小寫，順序不重要。<br>scope 值是由授權主機所定義或是經由「資源註冊作業」所管理。<br>支援的 scope 請參考 well-known/openid- |

| | |
|---|---|
| | configuration |
| response_type | 必要。<br>支援的 response_type 請參考 well-known/openid-configuration |
| client_id | 必要。<br>SP 身分識別值，經由「SP 註冊作業」取得。 |
| redirect_uri | 必要。<br>SP 所指定的重導向 URI，此 URI 必需符合「SP 註冊作業」中所輸入的 redirect_uri。 |
| state | 建議有。<br>協助 Client 識別發出請求時的狀態，授權主機會保持原樣的於重導向至 redirect_uri 時夾帶於參數中。<br>如 &state=xxx |

註：以上未包含所有於 OAuth 2.0 規範中提及的所有參數。僅列出目前需要的。

## （二）身分驗證及授權請求對話框

### 登入驗證對話框：

當請求參數檢核通過後，授權主機會要求用戶進行登入身分驗證。MyData 平台除了提供 eGov 密帳驗證外，也提供自然人憑證驗證。

### 同意授權對話框：

當完成用戶身分驗證後，授權主機依照 scope 值中載明的資源項目來詢問用戶是否同意授權「依賴方」可取得受保護資源。

## （三）回覆身分驗證請求成功

用戶同意授權後，AS 依此產製「授權碼 authorization code」，並藉由重導向至 redirect_uri 時夾帶於參數中。

重導向 redirect_uri 示意：

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb?
      code=SplxlOBeZQQYbYS6WxSbIA
      &state=af0ifjsldkj
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| code | 授權主機針對此次授權所產生的授權碼。 |

| | |
|---|---|
| state | 與身分驗證請求所帶入的參數 state 值相同。 |

### （四）回覆身分驗證請求失敗

參考規範 OpenID Connect section 3.1.2.6. Authentication Error Response 內容。 http://openid.net/specs/openid-connect-core-1_0.html#AuthError

失敗回覆網址示意：

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb?
      error=invalid_request
      &error_description=Unsupported%20response_type%20value
      &state=af0ifjsldkj
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| error | 必要。錯誤代碼。<br>請參考 RFC6749 section 4.1.2.1<br>https://tools.ietf.org/html/rfc6749#section-4.1.2.1 |
| error_description | 非必要。錯誤描述。 |
| error_uri | 非必要。錯誤描述頁面網址。 |
| state | 必要。<br>與身分驗證請求所帶入的參數 state 值相同。 |

## 四、Token Endpoint

Token endpoint 主要目的在令「依賴方」取得 access_token, id_token 以及 refresh_token。

### （一）核發 access_token 請求

SP 使用從 Authorization Endpoint 中取得的授權碼 authorization code 來請求核發 access_token。

此 Endpoint 只接受 HTTP POST 請求。參數的傳遞方式為 application/x-www-form-urlencoded。

請求網址示意：

```
POST /token HTTP/1.1
Host: login.cp.gov.tw
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
      &code=SplxlOBeZQQYbYS6WxSbIA
      &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| client_id | 必要。<br>SP 身分識別值，於 SP 註冊作業取得。 |
| client_secret | 必要。<br>SP 身分識別密碼，於 SP 註冊作業取得。 |
| grant_type | 必要。<br>authorization_code：授權碼模式。 |
| code | 必要。<br>授權主機 AS 所核發的授權碼 authorization code。 |
| redirect_uri | 必要。<br>帶入與請求授權碼時相同的 redirect_uri，且需符合 SP 註冊時所提供的 redirect_uri 清單中的項目。 |

Java Code Example：

```
List<NameValuePair> pairList = new ArrayList<>();
pairList.add(new BasicNameValuePair("client_id", getClientId()));
pairList.add(new BasicNameValuePair("client_secret", getClientSecret()));
pairList.add(new BasicNameValuePair("grant_type",
OidcGrantType.authorization_code.toString()));
pairList.add(new BasicNameValuePair("code", code));
pairList.add(new BasicNameValuePair("redirect_uri", getRedirectUri()));
CloseableHttpClient httpClient = HttpClientBuilder.create().build();
HttpPost post = new HttpPost(config.getTokenEndpoint());
post.setEntity(new StringEntity(URLEncodedUtils.format(pairList, "UTF-8")));
post.addHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-
8");
TokenEntity tokenEntity = null;
HttpResponse response = httpClient.execute(post);
```

## （二）回覆核發 access_token 請求成功

核發 access_token 請求成功後，授權主機回覆給 SP 的資訊，如：id_token, access_token, refresh_token, ...等。

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
 "access_token": "SlAV32hkKG",
 "token_type": "Bearer",
 "refresh_token": "8xLOxBtZp8",
 "expires_in": 3600,
 "id_token":"eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4
Mjg5
NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wU
zZ
fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyOD
A5Nz
AKfQ.ggW8hZ1EuVLuxNuuIJKX_V8a_OMXzR0EHR9R6jgdqrOOF4daGU96S
r_P6q Jp6IcmD3HP99Obi1PRs-cwh3LO-
p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ NqeGpe-
gccMg4vfKjkM8FcGvnzZUN4_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd
QyHE5lcMiKPXfEIQILVq0pc_E2DzL7emopWoaoZTF_m0_N0YzFC6g6EJbO
EoRoS
K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3_osjzw2PAithfubEEBLuVV
k4 XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"
}
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Content-Type | application/json |
| Cache-Control | no-store |
| Pragma | no-cache |

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| access_token | 必要。授權主機核發的 token。 |

| | |
|---|---|
| token_type | 必要的。<br>固定回覆字串"Bearer"。說明 access_token 符合 RFC 6750 OAuth 2.0 Bearer Token Usage 的規範。<br>https://tools.ietf.org/html/rfc6750 |
| refresh_token | 可選的。<br>用來進行更新 access_token 所需的 token。<br>當身分驗證請求時 scope 值中有指定 offline_access 時才會回覆 refresh_token。 |
| expires_in | 必要的。<br>access_token 的有效期限，單位秒。 |
| id_token | 必要的。<br>代表用戶訊息的 JWT 格式資料，並使用 JWS 簽章。<br>RFC 7519 JWT<br>https://tools.ietf.org/html/rfc7519 |

## （三）回覆核發 access_token 請求失敗

參考規範 OpenID Connect section 3.1.3.4. Token Error Response 內容。
http://openid.net/specs/openid-connect-core-1_0.html#TokenErrorResponse

失敗回覆網址示意：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
 "error": "invalid_request"
}
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Content-Type | application/json |
| Cache-Control | no-store |
| Pragma | no-cache |

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| error | 必要。錯誤代碼。<br>請參考 RFC6749 section 4.2.2.1<br>https://tools.ietf.org/html/rfc6749#section-4.2.2.1 |
| error_description | 非必要。錯誤描述。 |
| error_uri | 非必要。錯誤描述頁面網址。 |
| state | 必要。<br>與身分驗證請求所帶入的參數 state 值相同。 |

## （四）Refresh Token 請求

與請求核發 access_token 相同，Client 須以 HTTP POST 方法進行請求，參數的傳遞方式為 application/x-www-form-urlencoded。

refresh_token 只能使用一次，使用後除了拿到新的 access_token 之外也應該拿到新的 refresh_token。

Refresh Token 請求網址示意：

```
POST /token HTTP/1.1
Host: login.cp.gov.tw
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
      &refresh_token= 8xLOxBtZp8
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| client_id | 必要。<br>SP 身分識別值，於 SP 註冊作業取得。 |
| client_secret | 必要。<br>SP 身分識別密碼，於 SP 註冊作業取得。 |
| grant_type | 必要。<br>refresh_token：刷新 access_token 的有效期。 |
| refresh_token | 必要。<br>核發 access token 時一起核發的 refresh token。 |

Java Code Example：

```
List<NameValuePair> pairList = new ArrayList<>();
pairList.add(new BasicNameValuePair("client_id", getClientId()));
pairList.add(new BasicNameValuePair("client_secret", getClientSecret()));
pairList.add(new BasicNameValuePair("grant_type",
OidcGrantType.refresh_token.toString()));
pairList.add(new BasicNameValuePair("refresh_token", refreshToken));
CloseableHttpClient httpClient = HttpClientBuilder.create().build();
HttpPost post = new HttpPost(config.getTokenEndpoint());
post.setEntity(new StringEntity(URLEncodedUtils.format(pairList, "UTF-8")));
post.addHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-
8");
TokenEntity tokenEntity = null;
HttpResponse response = httpClient.execute(post);
```

## （五）回覆 Refresh Token 請求成功

回覆內容除了沒有 id_token 之外，其它的部份會與核發 access_token 請求回覆的內容相同。

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "TlBN45jURg",
  "token_type": "Bearer",
  "refresh_token": "9yNOxJtZa5",
  "expires_in": 3600
}
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Content-Type | application/json |
| Cache-Control | no-store |
| Pragma | no-cache |

參數/欄位說明：

| 參數/欄位 | 說明 |
| --- | --- |
| access_token | 必要。授權主機核發的 token。 |
| token_type | 必要的。<br>固定回覆字串"Bearer"。說明 access_token 符合 RFC 6750 OAuth 2.0 Bearer Token Usage 的規範。<br>https://tools.ietf.org/html/rfc6750 |
| refresh_token | 必要。<br>會拿到新的 refresh token。 |
| expires_in | 必要的。<br>access_token 的有效期限，單位秒。 |

## （六）回覆 Refresh Token 請求失敗

請參考「（三）核發 Access Token 請求回覆（失敗）」內容。

## （七）ID Token 規格說明

ID Token 是一個 JWT（JSON Web Token）格式的字符串。使用 JWS 進行數位簽章，以確保內容的不可篡改性。

ID Token 就是一個 JWS 其結構包含三個部份：header, payload, signature，每個部份皆以 base64encoder 進行編碼，三個部份以點號（.）串接後組成一個 ID Token。

ID Token 示意：

eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9
.
eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMDA4MTkzODAsDQogImh0dHA6Ly9leGFtcGxlLmNvbS9pc19yb290Ijp0cnVlfQ
.
dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk

ID Token, 編碼前的 header 示意：

```
{
  "alg": "HS256"
}
```

參數/欄位說明：

| 參數/欄位 | 說明 |
| --- | --- |
| alg | 必要。指定演算法。<br>HS256：HMAC SHA-256 |

ID Token, 編碼前的 payload 示意：

```
{
 "iss": "https://server.example.com",
 "sub": "24400320",
 "aud": "s6BhdRkqt3",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1311281970,
 "iat": 1311280970,
 "auth_time": 1311280969,
 "acr": "urn:mace:incommon:iap:silver"
}
```

參數/欄位說明：

| 參數/欄位 | 說明 |
| --- | --- |
| iss | 要必的。<br>Issuer Identifier, 用以識別提供認證信息者的唯一性，標示這個認證訊息是誰核發的。<br>格式為以 https 開頭且區分大小寫的 URL，但不包含 query parameters 及 fragment。 |
| sub | 必要的。<br>Subject Identifier, 由 Issuer Identifier 所提供的用戶的識別值（在 iss 的範圍內唯一），也就是這個 iss 所提供的用戶鍵值。<br>最大長度為 255 個 ASCII 字符。 |
| aud | 必要的。<br>Audience(s), 用來標示誰接收此 ID Token。必需包含 OAuth2 的 client_id。<br>當為單一值時以單一字串表示，若為多值時以字串陣列表示。 |
| exp | 必要的。<br>Expiration time, 過期時間，超過此時間的 ID Token 會作廢不再被驗證通過。<br>值為數值，從 1970-01-01T0:0:0Z 開始至過期時 |

| | |
|---|---|
| | 間的秒數。 |
| iat | 必要的。<br>這個 ID Token 的建立時間。<br>值為數值，從 1970-01-01T0:0:0Z 開始至建立時間的秒數。 |
| auth_time | Authentication time, 用戶完成登入驗證的時間。<br>如果登入驗證請求有帶入參數 max_age，則 auth_time 是必要的，否則是可選的。<br>值為數值，從 1970-01-01T0:0:0Z 開始至完成登入驗證時間的秒數。 |
| nonce | 可選的。<br>client 發送身分驗證請求時提供的隨機字符串，可令 client 方便關聯 client session 與 id token 及防止 replay attacks。<br>如果驗證請求時有提供 nonce，此處就會拿到一樣的字符串。 |
| acr | 可選的。<br>Authentication Context Class Reference, 表示一個驗證上下文（authentication context）引用值，用來標示驗證上下文。 |
| amr | 可選的。<br>Authentication Methods References, 表示完成身分驗證的方法，格式為字符串的陣列。<br>password：帳密驗證。<br>cert：自然人憑證驗證。 |
| at_hash | 可選的。<br>如有值，client 應用來驗證 access token。<br>at_hash 產生的方式為將 access token 以 HMAC SHA-256 演算法 client_secret 為 key 加密後，再以 base64url 編碼後產生。 |

ID Token, signature（數位簽章）：

簽章所包含的內容包括 JWS header 及 payload，做法是先以 base64encoder 編碼 header，再以 base64encoder 編碼 payload，再將編碼後的 header, payload 以點號（.）合併為一個字串，再將此字串先後以 HMAC SHA-256 演算法進行簽章及以 base64encoder 編碼後產生。

授權主機使用 client 註冊時產生的 client_secret 來當成進行簽章所需的 key。

JWT：

JWT（JSON Web Token, RFC 7519）的形式是一種以點號（.）分隔多個區塊的字符串，而每個區塊皆是 base64encoder 編碼後的字符串。JWT 只是描述一種結構形式，實際應用則依照 JWS（JSON Web Signature, RFC 7515）或是 JWE（JSON Web Encryption, RFC 7516）的規範而定。
https://tools.ietf.org/html/rfc7519
https://tools.ietf.org/html/rfc7516

## 五、Introspection Endpoint

Introspection Endpoint 主要令「資源方」向 MyData 授權主機反查「依賴方」送來的 access_token，如：要求存取的資源項目是否正確，以及用戶是否已同意授權等。

### （一）Introspection 請求

Introspection Endpoint 支援 Http POST 呼叫，參數的傳遞方式為 application/x-www-form-urlencoded，同時需以 HTTP Basic authentication 方式帶入身分驗證資訊。

Credential 的形式為將 client_id 及 client_secret 以冒號（:）合併為一個字串後（如：client_id:client_secret）再以 Base64 編碼，並將編碼後的字符串放置於 http header 中。
如：Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

請求網址示意：

```
POST /token HTTP/1.1
Host: login.cp.gov.tw
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

token=TlBN45jURg
```

參數/欄位說明：

| 參數/欄位 | 說明 |
| --- | --- |
| token | 必要。<br>核發的 access token。 |

Java Code Example：

```
List<NameValuePair> pairList = new ArrayList<>();
pairList.add(new BasicNameValuePair("token", accessToken));
CloseableHttpClient httpClient = HttpClientBuilder.create().build();
HttpPost post = new HttpPost(config.getIntrospectionEndpoint());
post.setEntity(new StringEntity(URLEncodedUtils.format(pairList, "UTF-8")));
post.addHeader("Accept", "application/json");
post.addHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-
8");
post.addHeader("Authorization", "Basic
"+basicAuthenticationSchema(clientId,clientSecret));
IntrospectEntity introspectEntity = null;
HttpResponse response = httpClient.execute(post);

private String basicAuthenticationSchema(String clientId, String clientSecret) {
        StringBuilder sb = new StringBuilder();
        sb.append(clientId).append(":").append(clientSecret);
        try {
                return encoder.encodeToString(sb.toString().getBytes());
        } catch (Exception e) {
                e.printStackTrace();
                return "";
        }
}
```

## （二）回覆 Introspection 請求成功

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
 "nbf": "",
 "exp": "",
 "iss": "",
```

```
  "aud": "",
  "client_id": "",
  "sub": "",
  "auth_time": "",
  "scope": "",
  "active": "",
}
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| nbf | 非必要。<br>整數時間戳，以 1970 年 1 月 1 日 UTC 之後的秒數測量，表明此標記何時不被使用。 |
| exp | 非必要。<br>整數時間戳，以 1970 年 1 月 1 日 UTC 之後的秒數測量，表明此令牌何時過期 |
| iss | 非必要。<br>表示該令牌的發行者的字符串 |
| aud | 非必要。<br>特定於服務的字符串標識符或表示此令牌的預期受眾的字符串標識符列表 |
| client_id | 非必要。<br>依賴方註冊作業中取得 |
| sub | 非必要。<br>令牌的主題，通常是授權此令牌的資源所有者的機器可讀標識符 |
| auth_time | 非必要。<br>授權此令牌的時間（unix timestamp）。 |
| scope | 非必要。<br>與此令牌相關聯的範圍列表的空格分隔。 |
| active | 必要。<br>指示是否所呈現的令牌當前處於活動狀態 |

## （三）回覆 Introspection 請求失敗

回覆失敗網址示意：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
```

Pragma: no-cache

```
{
 "error": "invalid_request"
}
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Content-Type | application/json |
| Cache-Control | no-store |
| Pragma | no-cache |

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| error | 必要。錯誤代碼。<br>請參考 RFC6749 section 5.2<br>https://tools.ietf.org/html/rfc6749#section-5.2 |
| error_description | 非必要。錯誤描述。 |
| error_uri | 非必要。錯誤描述頁面網址。 |

## 六、UserInfo Endpoint

UserInfo endpoint 主要令 SP 取得 RO 用戶資訊。

### （一）UserInfo 請求

使用 HTTP GET 方法來進行請求，並採用 Bearer token 進行身分驗證。

請求網址示意：

```
GET /userinfo HTTP/1.1
     Host: login.cp.gov.tw
     Authorization: Bearer SlAV32hkKG
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Authorization | Bearer token<br>例：Bearer SlAV32hkKG |

Java Code Example：

```
CloseableHttpClient httpClient = HttpClientBuilder.create().build();
HttpGet get = new HttpGet(config.getUserinfoEndpoint());
get.addHeader("Content-Type", "application/json");
get.addHeader("Authorization", "Bearer "+accessToken);
HttpResponse response = httpClient.execute(get);
UserInfoEntity userInfo = null;
if(response.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
        String responseString = EntityUtils.toString(response.getEntity(), "UTF-8");
        logger.debug("以 access_token 請求 user_info 成功！ -> {}", responseString);
        if(StringUtils.isNotEmpty(responseString)) {
                ObjectMapper om = new ObjectMapper();
                userInfo = om.readValue(responseString, UserInfoEntity.class);
        }
}else {
        logger.warn("以 access_token 請求 user_info 失敗！  HttpStatus -> {}",
response.getStatusLine().getStatusCode());
}
```

## （二）回覆 UserInfo 請求成功

UserInfo 實際回傳的欄位若少於規範已列示的欄位，以不出現該欄位為原則，而非將該欄位值付予 null 或是空字串。

SP 取得 UserInfo 後，應比對 sub 值是否與 ID Token 中的 sub 值相同，若不相同則不應該使用取得的 UserInfo 資料。

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json

{
 "sub": "GSP USER ID",
 "cn": "王小明",
 "uid": "身分證字號",
 "uid_verified": "True|False 身分證字號是否已驗證",
 "birthdate": "1973/07/14",
 "gender": "M|F 性別",
 "email": "janedoe@example.com",
 "account": "eGov 帳號"
}
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| sub | 此次授權用來代表帳戶的唯一識別值 |
| cn | 中文姓名 |
| uid | 身分證字號 |
| uid_verified | 身分證字號是否已驗證 |
| birthdate | 生日 |
| gender | 性別。<br>male/female |
| email | 電子郵件 |
| account | eGov 帳號 |

## （三）回覆 UserInfo 請求失敗

回覆請求示意：

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: error="invalid_token",
        error_description="The access token expired"
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| error | 錯誤代碼。<br>invalid_request：<br>沒提供必要的參數、提供了不支援的參數、提供了錯誤的參數值、同樣的參數出現多次、使用一種以上的方法來出示 Access Token （如放在 header 裡又放在 form 裡）、或是其他無法解讀 request 的情況。<br>invalid_token：<br>Access Token 過期、被收回授權、無法解讀、或其他 Access Token 不合法的情況。這種情況下, Client 可以重新申請一個 Access Token 並且用新的 Access Token 來重試 request。<br>insufficient_scope：<br>這個 request 需要出示比 Client 出示的 Access Token 代表的 scopes 還要更多的 scopes。這種情 |

| | |
|---|---|
| | 況下，可以另外提供 scope auth-param 來具體指出需要哪些 scopes。 |
| error_description | 錯誤說明。 |

# 七、Log Endpoint

Log endpoint 主要令 Mydata Portal 紀錄 SP 或 DP 請求資料或傳送資料日誌。

## （一）Log 請求

請求網址示意：

```
POST /v01/log HTTP/1.1
Host: log.mydata.nat.gov.tw/mydatalog
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
```

http header：

| 參數/欄位 | 說明 |
|---|---|
| Authorization | Basic (clientId:clientSecret base64 Encoding)<br>例：Basic SlAV32hkKG |

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| providerKey | egov 帳號。 |
| userName | 姓名。 |
| uid | 身份證字號。 |
| clientId | 帳號登入時的 SP 端鍵值。 |
| auditEvent | 授權狀態 1.登入(AS) 2.授權(AS) 3.登出(AS) 4. 要求資料(SP) 5. 送資料(DP) 6.接收資料(SP) |
| scope | 授權範圍. 多筆以空格或逗號隔開 |

Java Code Example：

```
List<NameValuePair> pairList = new ArrayList<>();
pairList.add(new BasicNameValuePair("providerKey", providerKey));
pairList.add(new BasicNameValuePair("userName", userName));
```

```
pairList.add(new BasicNameValuePair("uid", uid));
pairList.add(new BasicNameValuePair("clientId", clientId));
pairList.add(new BasicNameValuePair("auditEvent", auditEvent));
pairList.add(new BasicNameValuePair("scope", scope));
CloseableHttpClient httpClient = HttpClientBuilder.create().build();
HttpPost post = new HttpPost(config.getIntrospectionEndpoint());
post.setEntity(new StringEntity(URLEncodedUtils.format(pairList, "UTF-8")));
post.addHeader("Accept", "application/json");
post.addHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-
8");
post.addHeader("Authorization", "Basic
"+basicAuthenticationSchema(clientId,clientSecret));
IntrospectEntity introspectEntity = null;
HttpResponse response = httpClient.execute(post);

private String basicAuthenticationSchema(String clientId, String clientSecret) {
        StringBuilder sb = new StringBuilder();
        sb.append(clientId).append(":").append(clientSecret);
        try {
                return encoder.encodeToString(sb.toString().getBytes());
        } catch (Exception e) {
                e.printStackTrace();
                return "";
        }
}
```

## （二）回覆 Log 請求成功

UserInfo 實際回傳的欄位若少於規範已列示的欄位，以不出現該欄位為原則，而非將該欄位值付予 null 或是空字串。

SP 取得 UserInfo 後，應比對 sub 值是否與 ID Token 中的 sub 值相同，若不相同則不應該使用取得的 UserInfo 資料。

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json

{
 "code": "0",
 "text": "Ok"
}
```

### （三）回覆 Log 請求失敗

回覆請求示意：

```
HTTP/1.1 200 OK
Content-Type: application/json

{
 "code": "-1105",
 "text": "AuthenticateFail"
}
```

參數/欄位說明：

| 參數/欄位 | 說明 |
|---|---|
| code | 錯誤代碼。0 為正常<br>AuthenticateFail：-1105<br>身份驗證錯誤。<br>AccessDenied：-1111<br>存取被拒絕<br>NotAllowedIp：-1112<br>不是被允許的 IP |
| text | 錯誤說明。 |

## 陸、 授權記錄查詢及取消授權作業

MyData 平台提供令會員用戶可於登入 MyData 平台後，查詢同意授權記錄及取消已同意授權之記錄。

授權記錄欄位說明：

| 欄位 | 說明 |
|---|---|
| 授權時間 | 核發授權的時間。 |
| SP 服務名稱 | 請求獲取資源的服務名稱。 |
| 授權項目 | 授權項目名稱。等於受保護資源項目。<br>雖然核發授權時可能會一次賦予多筆授權項目，但授權記錄列表時應將各別的授權項目 |

| | |
|---|---|
| | 分別列示，如此用戶可各別取消單一的授權項目。 |
| 狀態 | 授權狀態。 |