數位服務個人化

資料提供者技術文件

V1.5

國家發展委員會
中華民國 108 年 6 月

版本修正紀錄

項次	版本	修正內容	時間	頁次
1	1.1	移除原章節 捌、單筆資料提介面規格準則 玖、批量資料提介面規格準則 增加章節 捌、DP-API Endpoint 規格準則	108/05/10	P24
2	1.2	調整章節「陸、一、流程示意圖」。 修正章節「捌、二、DP-API 請求及回覆規格說明」之內容。 新增章節「捌、三、DP-API Heartbeat 機制說明」。	108/05/11	P14 P25 P26
3	1.3	修正章節「貳、二、完成 MyData 線上註冊作業」之內容。 調整章節「捌、二、（四）DP 回覆請求」，增加狀態碼 504。 修正章節「玖、DP 資料打包檔案規格準則」之內容。	108/05/20	P3 P24 P27
4	1.4	修正章節「柒、四、（二）回覆 UserInfo 請求成功」	108/05/27	P21
5	1.5	調整章節「玖、一、DP 資料打包檔案規格說明」	108/6/10	P27

目錄

壹、目的

本文件目的主要描述「資料提供者」於實作「MyData 平臺之資料提供者」時應依循的作業流程、準則、技術規格及相關注意事項。

貳、如何成為資料提供者

一、完成 MyData 資料提供者資格申請作業

機關單位如欲加入 MyData 成為「資料提供者」，需先完成資格申請。

內容細節請參考本文件章節「肆、資料提供者資格申請作業」。

二、完成 MyData 線上註冊作業

註冊作業包括：

- 身分註冊：登錄機關單位基本資料。
- 資料集註冊：登入資料集資訊。

內容細節請參考本文件章節「伍、資料提供者管理作業」。

三、完成 GSP OAuth2 授權機制之授權驗證等相關系統整合介接

資料提供者需配合 MyData 實作「身分驗證及授權流程」之系統整合介接。

相關之 Endpoint 規格，請參考章節「柒、授權主機 API Endpoint 規格說明」。

四、實作資料提供介面規格

資料提供者應保有彈性並定義資料提供介面規格，但於發展實作資料提供介面規格時，應參考並依循國發會發佈之「共通性應用程式介面規範（OAS）」所提及之要點實作，使 API 具有共通性之特性，為擴大政府資訊服務效益。

資料提供介面規格準則，請參考本文件章節「捌、DP-API Endpoint 規格準則」及「玖、DP 資料打包檔案規格準則」。

參、名詞定義

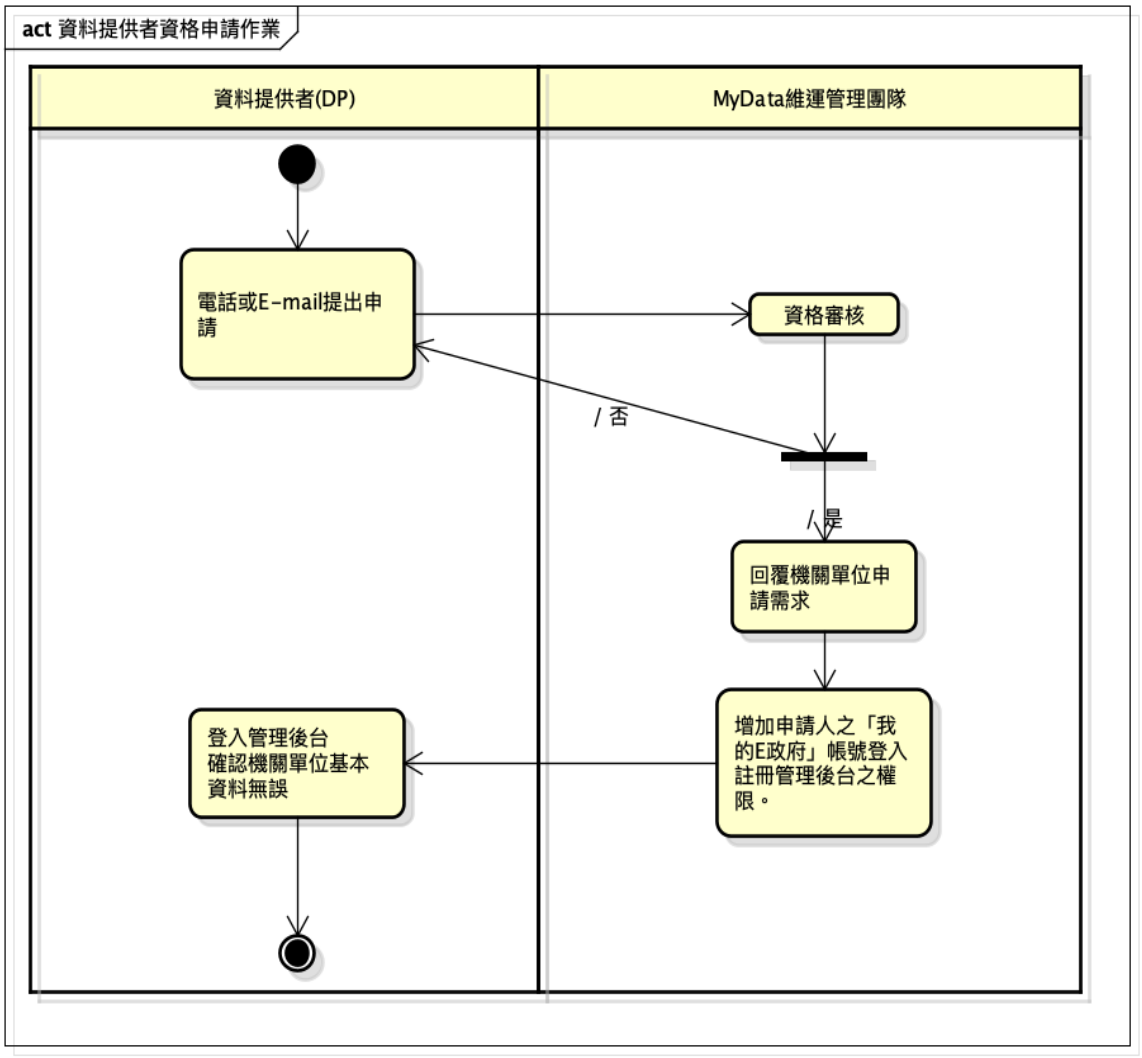
名稱	定義
OAS	共通性應用程式介面規範。
Data Provider, DP	資料提供者，存放或保管民眾個人資料之機關單位。
Service Provider, SP	服務提供者，提供民眾進行個人資料之加值服務機關單位。
Authorization Server, AS	授權管理者，執行身分驗證與授權管理機制，本規範之授權管理者為本會政府服務平臺(GSP)。
Resource Owner, RO	資料擁有者/使用者，泛指用戶或民眾。
OAuth 2.0	系統授權流程規範，定義於 RFC 6749 The OAuth 2.0 Authorization Framework https://tools.ietf.org/html/rfc6749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

肆、資料提供者資格申請作業

機關單位欲使用 MyData 機制成為資料提供者角色，應先完成資格申請，步驟說明如下：

步驟項次	流程內容
1	機關單位以電話或 E-mail 聯繫管理團隊，提出 MyData 註冊管理後台使用權限申請。（聯絡資訊 Tel:02-86925588#5555, E-mail: mydata@ndc.gov.tw ）

2	管理團隊回覆機關單位申請需求，增加機關單位申請人之「我的 E 政府」帳號登入註冊管理後台之權限。
3	機關單位申請人以「我的 E 政府」帳號登入註冊管理後台並確認機關單位基本資料無誤後，使用資料提供者管理功能項目。
4	完成。機關單位已成為資料提供者，可使用後台資料集註冊相關功能。



機關單位以電話（號碼）、電子郵件（信箱）聯繫 MyData 維運團隊申辦註冊管理後台機關帳號，並於申辦時提供「機關單位名稱」及「機關單位地址」、「聯絡人姓名」、「聯絡電話」、「電子郵件信箱」與申請人之「我的 E 政府註冊帳號」，由 MyData 維運人員協助完成帳

號註冊作業。完成機關單位註冊後，MyData 維運人員將透過註冊時機關單位提供之「聯絡電話」及「電子郵件信箱」通知機關單位聯絡人。

伍、資料提供者管理作業

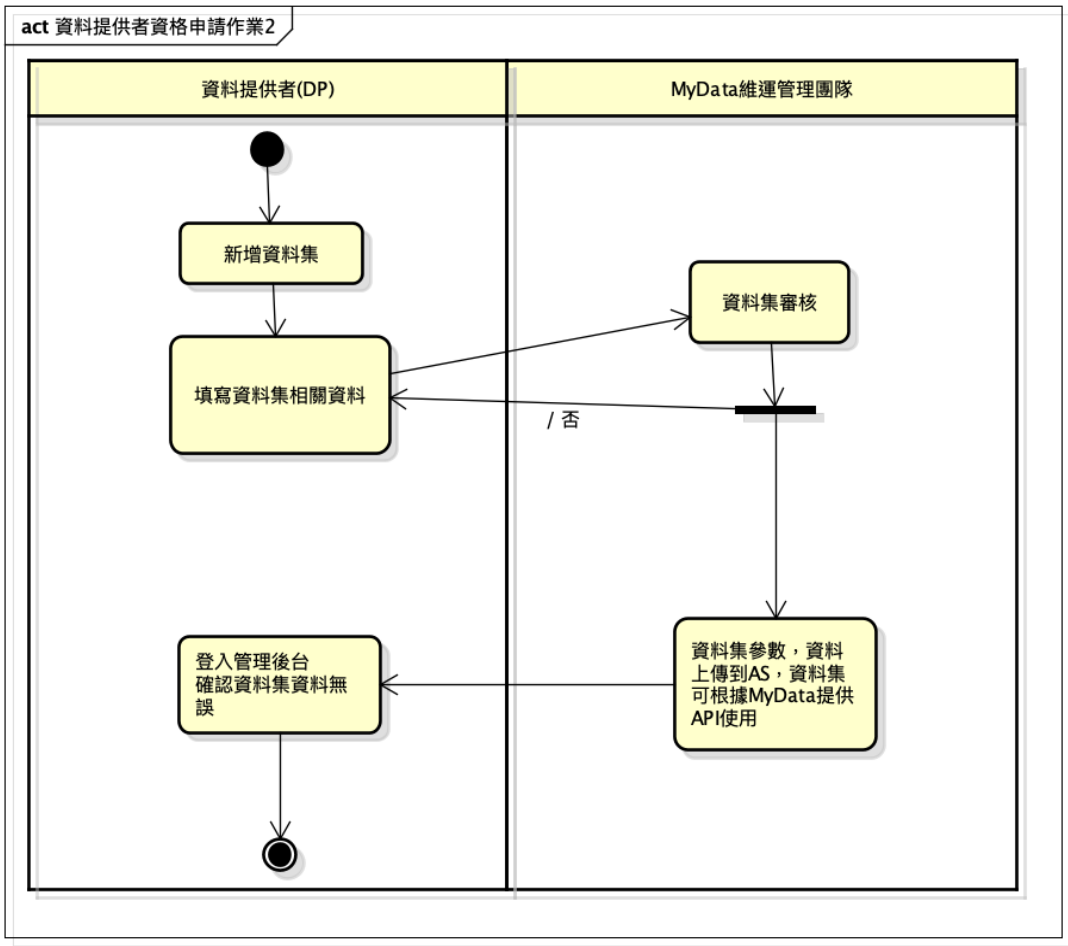
一、基本資料編輯

機關單位登入管理平臺後，點選「機關單位管理」功能項目，可自行編輯機關單位基本資料，包含「聯絡人姓名」、「聯絡電話」、「聯絡 E-mail」、「E 政府帳號」（管理後台登入使用）。於此功能頁面中，可瀏覽目前機關單位已建立之資料集與加值服務項目。

機關單位登入管理平臺後，點選「資料提供者管理」功能項目，若尚未同意「資料提供者合作契約」內容，需先同意契約內容成為資料提供者身分後，即可開通資料提供者管理相關功能使用權限（資料集新增、編輯與管理功能）。

二、資料集註冊

步驟項次	流程內容
1	資料提供者使用 MyData 註冊管理後台「新增資料集」功能。
2	完成新增資料集頁面中相關欄位內容，並提交審核。
3	經維運管理團隊確認相關內容符合 MyData 規範後，通知機關單位（申請人）審核通過並於註冊管理後台的資料集清單中上架資訊；若審核未通過，將回覆機關單位（申請人）修正建議，機關單位依修正建議調整後可重新提交審核申請。
4	服務提供者於管理後台中的「可運用資料集」功能中，閱覽審核通過上架之資料集。
5	待服務提供者依資料提供者說明之介接方式完成資料介接作業。



新增資料集	
基本資料欄位	機關單位名稱、申請人、聯絡電話、聯絡 Email、資料集類別、資料集名稱、資料集說明、授權取用資料安全等級、資料更新時間、資料集欄位、資料集存取範圍、resource_id、介接 API 文件。
選填欄位	無
功能動作	上傳 API 文件、取消新增、提交審核。
系統產生欄位	申請日期、resource_id、resource_secret(審核通過後才會顯示)。

欄位介面示意：

三、資料集管理

顯示已註冊、申請中之資料資源項目清單，並提供關鍵字查詢與新增、修改、刪除功能。刪除功能僅限資料集服務狀態為停用者，服務狀態為啟用之資料集應先完成停用申請流程。

修改資料集	
基本資料欄位	機關單位名稱、申請人、聯絡電話、聯絡 Email、資料集類別、資料集名稱、資料集說明、授權取用資料安全等級、資料更新時間、資料集欄位、資料集存取範圍、resource_id、resource_secret(審核通過後才會顯示)、介接 API 文件。
選填欄位	介接 API 文件。
功能動作	啟用 / 停用資料集、上傳介接 API 文件、取消修改、提交審核。

欄位介面示意：

四、資料集運用情形

資料提供者需檢視各啟用之資料集現行運用之情形時，可使用「被運用資料集」功能項目，將以清單顯示介接資料集之服務提供者與相對應服務名稱，並提供依資料集名稱篩選及服務提供者、註冊服務關鍵字搜尋功能。

五、查詢所有服務列表

提供查詢 MyData 專區已註冊服務提供者與服務清單，並提供服務連結（服務提供者服務申辦頁面）。

六、異常管理

顯示 MyData 系統輪巡偵測 DP 服務異常記錄，內容如下：

服務異常清單	
基本資料欄位	項次、建立日期、資料集類別、資料集名稱、狀態。
功能動作	異常資料集頁面連結。

陸、MyData 整合協作流程說明

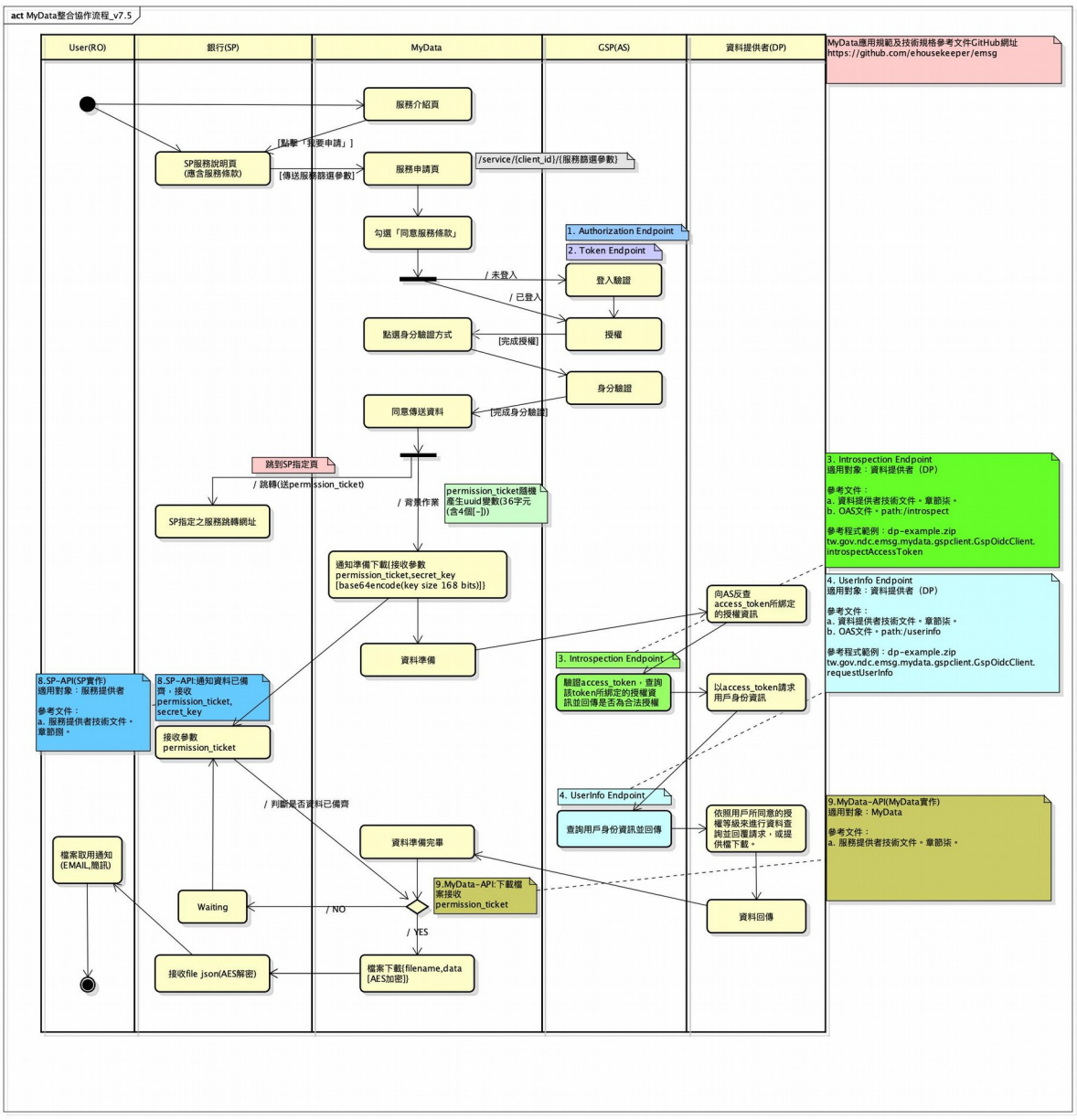
MyData 平臺的身分驗證及授權機制，採整合介接 GSP 所提供的 OAuth2 身分驗證及授權主機，民眾用戶使用我的 E 政府帳號進行登入驗證，且需同意授權 MyData 平臺向資料提供者取得用戶自己的個人資料。MyData 平臺則會將代表該用戶已同意授權的憑據 access_token 傳遞給資料提供者，讓資料提供者可依此向 GSP OAuth2 授權主機發出請求，以檢核用戶是否已同意授權及取得用戶資訊。

GSP OAuth2 授權主機之身分驗證及授權機制符合 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，以可互操作和 REST 的方式獲取有關最終用戶的基本配置文件信息。

本文件主要對象為提供資料提供者參考，為避免混淆，僅著重描述資料提供者何時請求呼叫以下 API：

- Introspection Endpoint
- UserInfo Endpoint

一、流程示意圖



註：流程說明圖檔案可至下述 Github 連結下載、瀏覽。

https://github.com/ehousekeeper/emsg/blob/master/MyData 應用規範與技術文件/MyData 整合協作流程_v7.jpg

二、應用範圍

（一）檢核用戶同意授權的有效性

當 MyData 平臺向資料提供者發出資源存取請求時，會一併傳遞民眾用戶同意授權的憑據 access_token 給資料提供者，資料提供者需檢核這個 access_token 是否有效，若有效，才接續以 access_token 取得用戶資訊。

資料提供者檢核 access_token 是否有效的方法，是呼叫 GSP OAuth2 授權主機提供的 API, Introspection Endpoint，請參閱章節「柒、三、Introspection Endpoint」。

（二）取得用戶資訊

當資料提供者已檢核確認來自於 MyData 平臺請求的 access_token 為合法有效的 token 後，則接下來可以呼叫 GSP OAuth2 授權主機提供的 API, UserInfo Endpoint 來取得用戶資訊，取得用戶資訊的主要目的是為了識別用戶身分，通常是以用戶的身分證字號做為識別的主要依據。除非資料提供者系統原本即使用我的 E 政府帳號做為系統會員登入驗證機制，就可以我的 E 政府帳號識別鍵值，做為用戶身分識別的依據。

資料提供者取得用戶資訊的方法，是呼叫 GSP OAuth2 授權主機提供的 API, UserInfo Endpoint，請參閱章節「柒、四、UserInfo Endpoint」。

柒、授權主機 API Endpoint 規格說明

一、系統環境與條件授權主機 API Endpoint

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

GSP 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。設定資訊檔網址，如下：

https://login.cp.gov.tw/v1/.well-known/openid-configuration

三、Introspection Endpoint

Introspection Endpoint 主要令「資料提供者」向 GSP 授權主機檢核「MyData 平臺」送來的 access_token 的有效性。

（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 resource_id 及 resource_secret 以冒號（:）合併為一個字串後（如：resource_id:resource_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 /v1/connect/introspect HTTP/1.1
Host: login.cp.gov.tw
Content-Type: application/x-www-form-urlencoded
Authorization:Basic {credential}

token={access_token}
```

參數/欄位說明：

參數/欄位	說明
credential	resource_id 及 resource_secret 以冒號:合併為一個字串後再以 Base64 編碼後的字符串。
access_token	代表用戶同意授權之 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(resourceId,resourceSecret)
```

```
);

IntrospectEntity introspectEntity = null;
HttpResponse response = httpClient.execute(post);

private String basicAuthenticationSchema(
     String resourceId,
     String resourceSecret) {
     StringBuilder sb = new StringBuilder();
     sb.append(resourceId)
       .append(":")
       .append(resourceSecret);
     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：回應欄位數目未來可能因需求而增加，接收時請考量擴充性。

參數/欄位	說明
nbf	非必要。 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。 整數時間戳，以 1970 年 1 月 1 日 UTC 之後的秒數測量，表明此 access_token 何時過期。
iss	非必要。 表示該 access_token 的發行者的字符串。
aud	非必要。 特定於服務的字符串標識符或表示此 access_token 的預期受眾的字符串標識符列表。
client_id	非必要。 服務提供者於註冊作業中取得。
sub	非必要。 access_token 的主題，通常是授權此 access_token 的資源所有者的機器可讀標識符。
auth_time	非必要。 授權此 access_token 的時間（unix timestamp）。
scope	非必要。 OAuth2 的 scope 參數。 關聯的資料集存取範圍，多筆以空白字串分隔。
active	必要。 指示是否所呈現的 access_token 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。 請參考 RFC6749 section 5.2 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。

四、UserInfo Endpoint

（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 /v1/connect/userinfo HTTP/1.1
     Host: login.cp.gov.tw
     Authorization: Bearer {access_token}
```

http header：

參數/欄位	說明
access_token	用戶同意授權 token。

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");
 if(StringUtils.isNotEmpty(responseString)) {
   ObjectMapper om = new ObjectMapper();
   userInfo = om.readValue(
     responseString,UserInfoEntity.class);
 }
}else {
 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。

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：回應欄位數目未來可能因需求而增加，接收時請考量擴充性。

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。 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。 invalid_request： 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。 invalid_token： 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。 insufficient_scope： 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。

捌、DP-API Endpoint 規格準則

一、系統環境與條件

API endpoint 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、DP-API 請求及回覆規格說明

（一）MyData 發出請求

```
GET /mydata-dp/{resource} HTTP/1.1
Host: xxx.xxx.xxx.xx
Content-Type: {content_type}
Authorization: Bearer {access_token}
```

參數說明：

參數	說明
resource	用來識別資料集的字符串，DP 自行決定即可。
content_type	application/json 代表請求回覆「json 檔」 application/pdf 代表請求回覆「pdf 檔」 application/zip 代表請求回覆「資料打包 zip 檔」
access_token	代表用戶同意授權的 token

（二）DP 回覆請求成功 – 即時回應

```
HTTP/1.1 200 OK
Content-Type: {content_type}
Content-Disposition: attachment; filename={filename}
Content-Transfer-Encoding: binary
Accept-Ranges: bytes
```

依據請求的 content_type 回應該格式的檔案。filename 中註明檔案名稱。

參數說明：

參數	說明
content_type	application/json 代表回覆「json 檔」 application/pdf 代表回覆「pdf 檔」 application/zip 代表回覆「資料打包 zip 檔」
filename	檔案名稱

（三）DP 回覆請求成功 – 等候處理

```
HTTP/1.1 429 Too Many Requests
Content-Type: {content_type}
Retry-After: {delay_seconds}
```

若 DP-API 不能即時回應請求，則以 HTTP 429 回應。

參數說明：

參數	說明
content_type	application/json 代表請求回覆「json 檔」 application/pdf 代表請求回覆「pdf 檔」 application/zip 代表請求回覆「資料打包 zip 檔」
delay_seconds	下次發動請求前需等待的秒數。

（四）DP 回覆請求失敗

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
```

DP 以 HTTP 狀態碼來表示回覆請求失敗的狀況。

HTTP 狀態碼	說明
401	access_token 檢核失敗。
403	拒絕存取。
504	無法完成傳送個人資料檔案。

三、DP-API Heartbeat 機制說明

DP-API Heartbeat 機制的設計目的，是為了令 MyData 平台確認 DP-API 仍有效存在。目前 MyData 平台預設以每一小時呼叫一次的頻率來確認 DP-API Endpoint 的狀態。

（一）MyData 發出 heartbeat 請求

```
GET /mydata-dp/{resource}?heartbeat=true HTTP/1.1
Host: xxx.xxx.xxx.xx
Content-Type: application/json
```

（二）DP 回覆 heartbeat 請求成功

```
HTTP/1.1 200 OK
```

DP-API 回應 HTTP 200 即代表 API Endpoint 仍有效存在。

（三）DP 回覆 heartbeat 請求失敗

DP-API 回應 HTTP 狀態碼為非 200 時，MyData 平台即視為 heartbeat 回應失敗。

當 DP-API 發生連線逾時，MyData 平台即視為 heartbeat 回應失敗。

玖、DP 資料打包檔案規格準則

一、DP 資料打包檔案規格說明

為了使 MyData 的應用更加廣泛，同一份資料可能提供多種格式，包括機器可讀的格式，如：json, csv, xml 等，以及易於人讀的格式，如：pdf 等。

此外也包括了保證資料未經竄改的數位簽章檔，及為了方便服務提供者進行驗簽的憑證檔。

為了簡化 MyData 平臺與資料提供者之間傳遞資料的機制，MyData 平臺希望資料提供者可以將同一份資料的多個檔案打包成為一個壓縮 zip 檔案後再傳遞給 MyData 平臺，因此 MyData 平臺規範了用戶個人資料檔案的打包規則，同時也使服務提供者在處理來自不同資料提供者的檔案時，可以有一致性的處理規則及做法。

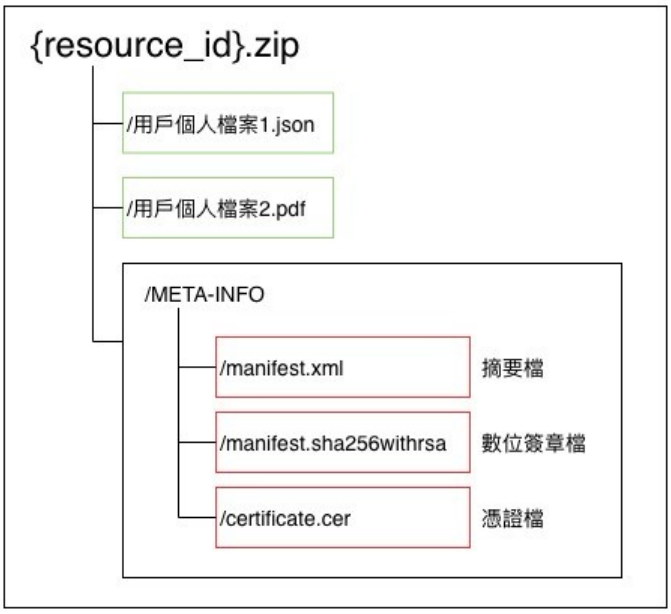
（一）DP 資料打包檔案規格要點如下：

1. DP 資料打包檔格式為壓縮 zip 檔。
2. zip 檔的檔名非限定但建議為 {resource_id}.zip。resource_id 是變數，代表資料集的識別代碼。
3. zip 檔中可以包含多個個人資料檔案，依資料提供者規範需包含的檔案用途、數量、檔名、副檔名等。
4. 如提供數位簽章，則包含於 META-INFO 子目錄。

承上，META-INFO 子目錄中包含檔案：

- manifest.xml 摘要檔
- manifest.sha256withrsa 數位簽章檔
- certificate.cer 憑證檔

（二）DP 資料打包檔案目錄結構示意如下：



（三）META-INFO 目錄及內含檔案說明：

META-INFO 目錄下放置摘要檔、數位簽章檔及憑證檔。若 DP 不產製數位簽章，則不必產生 META-INFO 子目錄。

manifest.xml：

針對各別的資料檔案，以 SHA256 演算法，演算出的數位指紋（摘要值）後載明於 manifest.xml 檔案中。

內容格式示意如下：

```
<?xml version="1.0" encoding="UTF-8">
<files>
  <file>
    <filename>用戶個人資料檔案1.json</filename>
    <digest>{digest value}</digest>
  </file>
  <file>
    <filename>用戶個人資料檔案2.pdf</filename>
    <digest>{digest value}</digest>
  </file>
</files>
</xml>
```

manifest.sha256withrsa：

以 SHA256 演算出 manifest.xml 的數位指紋後，以 DP 的 RSA 私鑰進行加密演算後所得的二進位內容，以副檔名 sha256withrsa 來示意所使用的演算機制為 SHA256withRSA。

建議 DP 使用長度至少 2048bits 的私鑰，向 CA 申請數位簽章憑證時，須申請支援 SHA256 的憑證。

certificate.cer：

憑證檔。PEM 格式的憑證資訊。PEM 格式的檔案是 ASCII (base64) 檔案，內容包含前置及後置文字，如下示意：

```
-----BEGIN CERTIFICATE-----
MIID/
zCCAuegAwIBAgIJAMhtYm3fde9AMA0GCSqGSIb3DQEBCwUAMIGVMQswCQ
YD
-----END CERTIFICATE-----
```

二、SP 驗證 DP 資料打包沒有被竄改的方法說明

（一）驗證憑證檔的有效性

服務提供者 SP 向簽發憑證的 CA 驗證憑證有效性。建議 DP 向 GCA 政府憑證管理中心來申請數位簽章用的憑證。

GCA 支援兩種驗證憑證有效性的方法，包括：CRL 及 OCSP。

（二）憑證檔中取出 DP 公鑰

DP 夾帶的憑證檔為 PEM 格式，SP 須從憑證檔中取出 DP 公鑰，用於後續驗證數位簽章檔案 manifest.sha256withrsa。

（三）驗證 manifest.xml 沒有被竄改

manifest.xml 檔案中載明了各別資料檔案的數位指紋（摘要值）。因此先驗證 manifest.xml 沒有被竄改，即代表 manifest.xml 檔中所載明的摘要值沒有被竄改。

manifest.sha256withrsa 是針對 manifest.xml 所做出來的數位簽章檔，SP 須以 DP 的公鑰對 manifest.sha256withrsa 進行解密後，可得到正確的 manifest.xml 的摘要值。

SP 對 manifest.xml 進行 SHA256 演算後，比較前後兩者摘要值是否相符，若相符則代表 manifest.xml 沒有被竄改。

（四）驗證各別的資料檔案沒有被竄改

SP 讀取 manifest.xml 內容後，得到各別資料檔案的正確的摘要值，再針對各別資料檔進行 SHA256 演算後，比較前後兩者摘要值是否相符，若相符則代表該資料檔案沒有被竄改。