Dynamic Client Registration API Guide v2.0

Document Version 2.0
Date: 24th March 2021

Table of Contents

1 Overview

The PTSB Dynamic Client Registration API enables a third party provider (TPP) to register their application and receive a Client ID and Shared Secret, in line with the OAuth2.0 framework. The API also enables a TPP to provide business and technical contact details.

Note

- **The current version of the API is backwardly compatible with applications and organisation contacts details that were registered using the previous version of the API**
- **The Sandbox is a separate environment, therefore to register an application on the Sandbox please enrol on the PTSB Developer Portal and refer to the Sandbox Guide.**

1.1 Additional eIDAS Certificate Validation

As the PSD2 eIDAS certificate ecosystem is not fully mature, when an organisation registers their first application, permanent TSB will conduct additional validation of the TPP's eIDAS certificates, during that time the AISP, PISP and CBPII API endpoints will remain inaccessible. The additional validation may take up to one business day after an application has been successfully registered. Permanent TSB will send an e-mail notification to your technical and business contacts to inform them that the validation process is complete and your application can then access the production APIs.

1.2 Terms of Use

Please read our Terms of Use and our Third Party Provider Data Privacy Statement prior to calling our production APIs.

1.3 Application Scope

The Dynamic Client Registration API will define the scope for a registered application based on the PSD2 roles contained within the TPPs QSEAL certificates accompanying the request. For example a certificate containing two roles, ***PSP_AI*** and ***PSP_PI*** – will assign ***openid***, ***accounts*** and ***payments*** scopes to the successfully registered application.

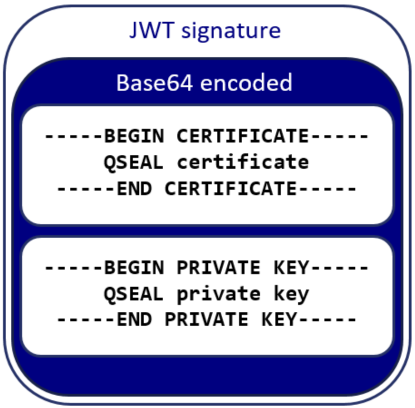
2 Calling Dynamic Registration Endpoints

To register and manage applications programmatically six API calls can be made:

- POST /register – register an application and provide organisation contacts details
- POST /as/token.oauth2 – access token for GET, PUT and DELETE by client ID endpoints
- GET /register?client-id={client ID} – retrieve application and organisation contact details
- PUT /register?client-id={client ID} – update application and/or organisation contact details
- DELETE /register?client-id={client ID} – delete an application
- DELETE /register – delete all application(s) and organisation contact details

Request and response structures of aforementioned APIs exhibit following traits:

- All requests must be accompanied by ***QWAC*** certificate
- Requests without client ID must have a base64 encoded ***QSEAL*** certificate, sent in ***tpp-signature-certificate*** header
- POST and DELETE /register request and response bodies must be a ***signed JWTs*** with encoded ***JSON payload***
- JWTs are signed with ***QSEAL certificate*** and ***private key*** using ***PS256*** algorithm
- QSEAL certificate in JWT signature must be wrapped with ***begin*** and ***end*** tags and ***base64*** encoded as depicted in the figure on the right


JWT signature
Base64 encoded
-----BEGIN CERTIFICATE-----
QSEAL certificate
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
QSEAL private key
-----END PRIVATE KEY-----

API Request	QWAC	QSEAL	Access token	Signed payload
POST /register	✓	✓	×	✓
DELETE /register	✓	✓	×	✓
POST as/token.oauth2	✓	✓	N/A	N/A
GET /register?client-id={client ID}	✓	×	✓	N/A
PUT /register?client-id={client ID}	✓	×	✓	×
DELETE register?client-id={client ID}	✓	×	✓	N/A

- JSON fields and claims are detailed in the table below:

Name	Mandatory	Constraints
Organisation_Name	Yes	String (1-50).
Organisation_Description	No	String (0-200).
Organisation_Website	No	String (0-200).
Application_Name	Yes	String (1-50).
Application_Description	No	String (0-200).
Redirect_URI	Yes	String (4-2048). Must include "www" and "http(s)". Multiple comma separated URIs are allowed.
Business_Contact_Name	Yes	String (1-200).
Business_Contact_Email	Yes	String (5-100).
Business_Contact_Phone	Yes	String (5-50).
Technical_Contact_Name	Yes	String (1-200).
Technical_Contact_Email	Yes	String (5-100).
Technical_Contact_Phone	Yes	String (5-50).
iat	Conditional	NumericDate (seconds since Epoch). Cannot be in the future. Only required for JSON payload in JWTs.
exp	Conditional	NumericDate (seconds since Epoch). Cannot be more than 90 days from current day. Only required for JSON payload in JWTs.
jti	Conditional	NumChar (255). Must be a unique UUIDv4 GUID for each request. Only required for JSON payload in JWTs.

2.1 Register First Application

To register an application and provide organisation details for the first time populate mandatory JSON fields with your application and organisation details, place JSON (see *Example 1*) into JWT body and sign with ***QSEAL certificate*** and ***private key***.

Example 1 – JSON for request

```
{
"iat": numericdate,
"exp": numericdate,
"jti": "numchar",
"Organisation_Name": "string",
"Organisation_Description": "string",
"Organisation_Website": "string ",
"Application_Name": "string",
"Application_Description": "string",
"Redirect_URI": "string",
"Business_Contact_Name": "string",
"Business_Contact_Email": "string",
"Business_Contact_Phone": "string",
"Technical_Contact_Name": "string",
"Technical_Contact_Email": "string",
"Technical_Contact_Phone": "string"
}
```

Example 2 – JSON in response

```
{
"Organisation_Name": "string",
"Organisation_Description": "string",
"Organisation_Website ": "string",
"Redirect_URI": "string",
"Application_Name": "string",
"Application_Description": "string",
"Business_Contact_Name": "string",
"Business_Contact_Phone": "string",
"Business_Contact_Email": "string",
"Technical_Contact_Name": "string",
"Technical_Contact_Phone": "string",
"Technical_Contact_Email": "string",
"Client_Id": "string",
"Client_Secret": "string",
"Scope": "string"
}
```

POST signed JWT to https://api.permanenttsb.ie/register (see *Example 3*).

Example 3 – HTTP request for POST /register API endpoints

```
POST /register HTTP/1.1
Content-Type: application/raw
tpp-signature-certificate: LS0tLS1Host…

eyJhbGciOiJQUzI1NiJ9.eyJpYXQiOjEyMywiZXhwIjo0NTYsImp0aSI6IjFiMmRhNDQ3LTMzNTAtNDgyNi1hYWM5LTE4YmVlOGNlNjRhMiIsIk9yZ2FuaXNhdGlvbl9OYW1l
Ijoic3RyaW5nIiwiT3JnYW5pc2F0aW9uX0Rlc2NyaXB0aW9uIjoic3RyaW5nIiwiT3JnYW5pc2F0aW9uX1dlYnNpdGUiOiJzdHJpbmcgIiwiQXBwbGljYXRpb25fTmFtZSI6I
nN0cmluZyIsIkFwcGxpY2F0aW9uX0Rlc2NyaXB0aW9uIjoic3RyaW5nIiwiUmVkaXJlY3RfVVJJIjoic3RyaW5nIiwiQnVzaW5lc3NfQ29udGFjdF9OYW1lIjoic3RyaW5nIi
wiQnVzaW5lc3NfQ29udGFjdF9FbWFpbCI6InN0cmluZyIsIkJ1c2luZXNzX0NvbnRhY3RfUGhvbmUiOiJzdHJpbmciLCJUZWNobmljYWxfQ29udGFjdF9OYW1lIjoic3RyaW5
nIiwiVGVjaG5pY2FsX0NvbnRhY3RfRW1haWwiOiJzdHJpbmciLCJUZWNobmljYWxfQ29udGFjdF9QaG9uZSI6InN0cmluZyJ9.wExMPOLxA8_AcPg8Cdd5Ezoi_rVzSF1Agnm
N1sNBA9o3H-xZRsgZ9rMupmBo4itfL8YTvPV1d6SRC9FRNgZV1A668r1fVQyc2yb8vZdeLM5ZkfAXrkbJnWuXrIxjcwJ711LbPfxtSceUCQJroOOZQDUsi6ld3jaBTSr-
jiofs9nirIaq_Q5sjjqeWLeh0rMZKeY1Z8XcbRgc24fSrcrAkRNL3Tw5j0NJsJF8PsDtHp0RHgvzV2lSfUp5gsq-
NvzRWUtOO7aMqyvA4L4GLKA8xftIcW7pdmwB1zv55CMpVL2pa1G0xFMUwMr5owEVHbH-pqGkIKAa4T4GBYaWwA0hfQ
```

Capture JWT from response body (see *Example 4*).

Example 4 – HTTP response for POST, GET and PUT /register API endpoints

```
HTTP/1.1 200 OK

eyJhbGciOiJQUzI1NiIsInR5cCI6IkpPU0UiLCJjdHkiOiJhcHBsaWNhdGlvbi9qc29uIiwia2lkIjoicXNSaHNOVUhYMC9UQVg0SDdCS08zQkFBZXVnPSIsImh0dHA6Ly9vc
GVuYmFua2luZy5vcmcudWsvaWF0IjoxNTg0NjE4MjM4LCJodHRwOi8vb3BlbmJhbmtpbmcub3JnLnVrL2lzcyI6Ik9JRC4yLjUuNC45Nz1QU0RJRS1DQkktQzI2OTE0LCBDTj
1QZXJtYW5lbnQgVFNCc2lnbiBxc2VhbCwgT1U9UGVybWFuZW50IFRTQiBwbGMsIE89UEVSTUFORU5UIFRTQiBQVUJMSUMgTElNSVRFRCBDT01QQU5ZLCBDPUlFIiwiaHR0cDo
vL29wZW5iYW5raW5nLm9yZy51ay90YW4iOiJhcGkucGVybWFuZW50dHNiLmllIn0.eyJPcmdhbmlzYXRpb25fTmFtZSI6InN0cmluZyIsIk9yZ2FuaXNhdGlvbl9EZXNjcmlw
dGlvbiI6InN0cmluZyIsIk9yZ2FuaXNhdGlvbl9XZWJzaXRlICI6InN0cmluZyIsIlJlZGlyZWN0X1VSSSI6InN0cmluZyIsIkFwcGxpY2F0aW9uX05hbWUiOiJzdHJpbmciL
CJBcHBsaWNhdGlvbl9EZXNjcmlwdGlvbiI6InN0cmluZyIsIkNsaWVudF9JZCI6InN0cmluZyIsIkNsaWVudF9TZWNyZXQiOiJzdHJpbmciLCJCdXNpbmVzc19Db250YWN0X0
5hbWUiOiJzdHJpbmciLCJCdXNpbmVzc19Db250YWN0X1Bob25lIjoic3RyaW5nIiwiQnVzaW5lc3NfQ29udGFjdF9FbWFpbCI6InN0cmluZyIsIlRlY2huaWNhbF9Db250YWN
0X05hbWUiOiJzdHJpbmciLCJUZWNobmljYWxfQ29udGFjdF9QaG9uZSI6InN0cmluZyIsIlRlY2huaWNhbENvbnRhY3RfRW1haWwiOiJzdHJpbmciLCJTY29wZSI6InN0cmlu
ZyJ9.Gh9NJEuSNlETJguYf4RBEHFq_fQiR4X-
LQYIez3bSO4ldSEyhZcPXyONiDy3TbLV5Fv8keIdzpa006jOpVXiXTk8APMKTIIOAWUXAzS_kQoz7e44k_EfDHfhrxPu92ra9VPD0WqNBWzI3Pk9QJf8bfpz95JGqhTwWVf5R
MCn-wESA99koke6f6P7DTO4sGNK0I-FG9SdEGqYNc-Sqm8-SDyXG473vQAmgI2OcI_YShHhvPqFS1recqLjNzlhJUdhXz30Ahsxt98L-bJLF8f2Tw8BHjYUOSnidCIgRvFV-
dKmm_wKuostVy89I4Z9MGs8lJbL1S4r1ne9O5yRzU-hXg
```

Decode JWT to extract ***Client ID*** and ***Shared Secret*** for newly registered application (see *Example 2*)

Note

- If ***Organisation_Description*** or ***Application_Description*** claims are not populated in the payload for the POST request, the API will automatically populate them with the value in the ***Organisation_Name*** or ***Application_Name*** claims.

2.2 Register Additional Applications

Additional applications can be registered by following the same process as described in section 2.1.

- To register additional applications a POST /register request may be sent with a reduced payload, which needs to include at least an application name and redirect URI.
- The response to the request for an additional application will include the details for the new application and the current values held for the organisation contact details.
- If organisation contact details are included in the POST payload for an additional application, these values will be ignored.

2.3 Request Client Credentials Grant Type Access Token

GET, PUT and DELETE endpoints require a Client Credentials Grant type Access Token, which can be obtained once an application has been successfully registered.

To get an Access Token compile a POST request to https://api.permanenttsb.ie/as/token.oauth2 endpoint:

- Populate “exp” and “jti” JSON claims and place JSON into JWT body and sign with ***QSEAL certificate*** and ***private key***.
- Append to the URL ***client_assertion*** request query parameter with JWT.
- Append to the URL ***scope*** request query parameter with corresponding scope.
- Append to the URL ***grant_type*** request query parameter with ***client_credentials*** value.
- Add Client ID and Shared Secret to ***authorization header***. Use basic authorisation.
- Do not forget to include a base64 encoded QSEAL certificate, sent in ***tpp-signature-certificate*** header.
- Verify if request is similar to *Example 5* below. Note that some ***data is omitted for brevity***.
- Send the request to receive an Access Token, which will be ***valid for 60 minutes***.

Example 5 – HTTP request to obtain an Access Token

```
POST /as/token.oauth2?grant_type=client_credentials&scope=accounts&client_assertion=eyJ0eA… HTTP/1.1
Content-Type: application/x-www-form-urlencoded
tpp-signature-certificate: LS0tLS1Host…
Authorization: Basic bDczZ…4OWFlNg==
```

2.4 Retrieve Application and Organisation Details

GET endpoint can be called to retrieve application and organisation registration details.

To do so compile a **GET** request to https://api.permanenttsb.ie/register endpoint:

- Append to the URL ***client-id*** request query parameter with Client ID value.
- Add the ***Client Credentials Grant*** *type* ***Access Token*** as a bearer token to authorization header.
- Send the request to retrieve current organisation and application information (see Example 2).

2.5 Update Application or Organisation Details

An application and organisation details can be updated by calling the API endpoint with the PUT HTTP method.

To do so compile a **PUT** request to https://api.permanenttsb.ie/register endpoint:

- Append to the URL ***client-id*** request query parameter with Client ID value.
- Add ***Client Credentials Grant type Access Token*** as a bearer token to authorization header.
- Compile ***JSON payload*** (see *Example 1*) and add to PUT request body.
- Send the request to retrieve updated organisation and application information (see Example 2).

Note

- Application scope cannot be updated.
- Since request body is a plain JSON format; “iat”, “exp” and “jti” claims are not required.

2.6 Delete Application and Organisation Information

To delete a single application simply follow the same process as described in section 2.4, but instead of GET method use **DELETE**.

Example 6 – HTTP request to delete an application by Client ID

```
DELETE /register?client-id=0000 HTTP/1.1
Content-Type: application/json
Authorization: Bearer 00001234-0000-0000-0000-000012345678
```

If operation was successful, response body will be in plain JSON format with confirmation: "Message": "The Application {application name} with Client ID: {client ID} was successfully deleted".

2.6.1 Delete all Applications and Organisation details

In order to delete all applications and organisation details follow the same process as described in section 2.1, but instead of POST method use **DELETE** and **add signed *JWT with specific JSON payload*** (see *Example 7*) to DELETE request body.

Example 7 – JWT body JSON for request to delete all applications and organisation information

```
{
"iat": numericdate,
"exp": numericdate,
"jti": "numchar”,
"Delete": "All"
}
```

To verify if operation was successful decode JWT in response body to extract confirmation: "Message": “The Organisation was successfully removed from our records".