



Dynamic Client Registration API v1.0 Guide

Version 1.0

28th June 2019

Table of Contents

1.1 Overview	3
1.1.1 API Swagger File	3
1.1.2 eIDAS Certificates Production Verification	3
1.2 Terms of Use	3
1.3 Application Scope	3
1.4 Register Initial Application	3
1.5 Register Additional Applications	5
1.6 Retrieve Application Information	6
1.7 OAuth 2.0 Grant Type	7
1.8 Message Signing	7
1.8.1 TPP API Requests	7
1.8.2 PTSB API Responses	7
1.9 Endpoints	7
1.10 Update / Delete Application	7
1.11 Payload Details	8
1.12 Error Responses	9

1.1 Overview

The PTSB Dynamic Client Registration API enables a 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. Requests to the API must be signed by a TPP's QSealC. Currently there are two client registration endpoints:

- POST /register – for application registration
- GET /register/client_id – for application information retrieval

1.1.1 API Swagger File

To view and download the API Swagger definitions please enrol on our [Developer Portal](#).

1.1.2 eIDAS Certificates Production Verification

Please note because the PSD2 eIDAS ecosystem is not yet fully mature, there is a requirement for an interim period for PTSB to conduct additional validation on eIDAS certificates. Your registered application will not be able to access our production APIs until PTSB completes this process.

- The process will be completed no later than close of business, one business day after you have successfully registered your application
- We will send an e-mail to your technical and business contacts to inform them the process is complete and your application is ready to use with our production APIs

1.2 Terms of Use

Please read our [Terms of Use](#) and our [Data Protection Notice](#) before accessing 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 a TPPs QSealC, for example:

PSD2 Role	Scope
PSP_AI	openid accounts
PSP_AI, PSP_PI	openid accounts payments

1.4 Register Initial Application

This sub-section of the guide describes how to register your first application.

Prerequisites:

- eIDAS certificates
- /register API URL - <https://api.permanentsb.ie/register>

To register an application, populate the API [Payload](#) with your application and organisation contact details. All fields are mandatory, apart from Organisation Description, Application Description and Organisation Website.

- If Organisation Description or Application Description are not populated in the POST request, the API will automatically populate them with the value in the Organisation Name or Application Name field

The response to the POST will include a unique Client ID and Shared Secret for the application.

POST /register

- The API allows TPP to register an application
- The TPP submits a JWT payload with data describing the characteristics of the application to be created and also provides contacts details for the TPP organisation
- TPP submits their QSealC in the JWT header (tpp-signature-certificate) and a signed JWT payload in the body
- The response will be a signed JWT payload replaying the application details, organisation contact details, scope based on PSD2 roles and a Client ID and Shared Secret

Examples of the POST /register API payload & response are outlined below.

Request

Example – Request Header

tpp-signature-certificate: "jwt encoded QSealC"

Example – Decoded Request Payload JWT

```
{
  "Organisation_Name":"string",
  "Organisation_Description":"optional",
  "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"
}
```

Note

- The body of the payload of the request should be JWT format

Response

Example – Response Header

x-jws-signature: A detached JWS signature of the body of the payload

Example - Response Body

```
{
  "Organisation_Name": "string",
  "Organisation_Description": "string",
  "Organisation_Website": "string",
  "Redirect_URI": "string",
  "Application_Name": "string",
  "Application_Description": "string",
  "Client_Id": "/78852967ec693453bb11c4ef2ee5140cd",
  "Client_Secret": "84c2c46fcb6044e88ga8cf265d5a4ff0",
  "Business_Contact_Name": "string",
  "Business_Contact_Phone": "string",
  "Business_Contact_Email": "string",
  "Technical_Contact_Name": "string",
  "Technical_Contact_Phone": "string",
  "TechnicalContact_Email": "string",
}
```

```
"Scope": "openid accounts"
}
```

Note

- The payload of the response will also be returned as a signed JWT
- As noted above PTSB will conduct additional eIDAS validation and will provide confirmation by e-mail when this is complete and your application is ready for use with the PTSB APIs

1.5 Register Additional Applications

This section of the guide describes how a TPP can register additional applications

Prerequisites:

- eIDAS certificates
- /register API URL - <https://api.permanenttsb.ie/register>

To register additional applications a POST /register request can be submitted with a reduced payload, which only needs to include the following fields:

- Application Name
- Application Description (optional)
- Redirect URI

Note

The API does not currently support the ability to update or delete a registered application. There will be a subsequent release of the API which will include this functionality. Therefore:

- if contact information is populated in the POST for an additional application, these values will be ignored
- the response to the POST will include the details for the additional application and the organisation contact details that were provided when the first application was registered

Examples of the POST /register API payload & response are outlined below.

Request

Example – Request Header

```
tpp-signature-certificate: "jwt encoded QSealC"
```

Example – Decoded Request Body JWT

```
{
  "Application_Name":"string",
  "Application_Description":"string",
  "Redirect_URI":"string"
}
```

Response

Example – Response Header

```
x-jws-signature: A detached JWS signature of the body of the payload
```

Example - Response Body

```
{
  "Organisation_Name": "string",
  "Organisation_Description": "string",
  "Organisation_Website": "string",
  "Redirect_URI": "string",
  "Application_Name": "string",
  "Application_Description": "string",
  "Client_Id": "178852967ec693453bb11c4ef2ee5140cd",
  "Client_Secret": "84c2c46fcb6044e88ga8cf265d5a4ff0",
  "Business_Contact_Name": "string",
  "Business_Contact_Phone": "string",
  "Business_Contact_Email": "string",
  "Technical_Contact_Name": "string",
  "Technical_Contact_Phone": "string",
  "TechnicalContact_Email": "string",
  "Scope": "openid account"
}
```

1.6 Retrieve Application Information

This section describes how to retrieve details of a registered application

Prerequisites:

- OAuth token
- client_id=xxxx
- /register/{ClientId} API URL - https://api.permanenttsb.ie/register?client_id={client_id}

To retrieve details compile a request for the GET /register/{ClientId} API endpoint

Get /register/{ClientId}

- The API enables a TPP to retrieve the details for a registered application
- The response will be a JSON payload, encoded into a signed JWT, with data that describes the application and organisation contact details
- If the request is unsuccessful an error code will be returned

An example of a GET /register/{ClientId} API payload and response are outlined below. All parameters are mandatory.

Request

Example – Query Parameters

```
client_id = 17f8cbfa29258b4236b3987753c0ec3b00
```

Example – Authorisation

```
Bearer Token = "xxxxxxxxxxxxxxxxxxxxxx"
```

Response

Example – Response Header

```
x-jws-signature: A detached JWS signature of the body of the payload
```

Example - Response Body

```
{
  "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",
  "TechnicalContact_Email": "string",
  "Scope": " openid account "
}
```

1.7 OAuth 2.0 Grant Type

PTSB are implementing OAuth2.0 “Hybrid flow”, as a result all registered TPP application(s) will default to “**Confidential**” grant “**Type**”. “Public” grant Type only applies for the “Implicit Flow”, which is not been implemented by PTSB.

1.8 Message Signing

1.8.1 TPP API Requests

All requests a TPP submits to the Dynamic Client Registration API endpoints must be signed. The API requests must be accompanied by the TPP’s QSealC certificate. The TPP’s QSealC certificate will be transmitted in the “**tpp-signature-certificate**” header.

1.8.2 PTSB API Responses

All Dynamic Client Registration API response messages sent to a TPP will be signed by PTSB’s QSealC certificate. The API response messages sent to the TPP will be accompanied by the “**x-jws-signature**” in the header. TPPs should use our JWKS discovery endpoint to get the cert for signature validation.

1.9 Endpoints

The following end points are available

Request	Method	Endpoint
Application Registration	POST	/register
Review Application Details	GET	/register/{ClientId}

1.10 Update / Delete Application

The Dynamic Client Registration API v1.0 does not currently support the ability to update or delete a registered application or the organisation contact details. There will be a subsequent release of the API which will include this functionality.

- To request an update or delete your application or organisation contact details please contact our [support](#)

1.11 Payload Details

Field	Mandatory/ Optional	Type / Max Character	Restriction
Organisation_Name	Mandatory	String(50)	Full ASCII less the following characters:- \$+;=?@£# <>^*()%!
Organisation_Description	Optional	String(200)	Full ASCII less the following characters:- \$+;=?@£# <>^*()%! <ul style="list-style-type: none"> If not populated will automatically be populated with the value in the Organisation_Name field
Organisation_Website	Optional	String(200)	Accepts Full ASCII set. Must include www and/or http(s)
Application_Name	Mandatory	String(50)	Full ASCII less the following characters: \$+;=?@£# <>^*()%!
Application_Description	Optional	String(200)	Full ASCII less the following characters: \$+;=?@£# <>^*()%! <ul style="list-style-type: none"> If not populated will automatically be populated with the value in the Application_Name field
Redirect_URI	Mandatory	String (2048)	Accepts ASCII. Must include www and/or http(s). Can be multiple URIs separated by commas
Business_Contact_Name	Mandatory	String(200)	Full ASCII less the following characters: \$+;=?@£# <>^*()%!
Business_Contact_Email	Mandatory	String(100)	Accepts: [a-z0-9!#\$%&'*/+=?^_`{ }~-]@[a-zA-Z0-9].[a-zA-Z0-9]
Business_Contact_Phone	Mandatory	String(30)	Accepts: [a-zA-Z0-9_] and [+..]
Technical_Contact_Name	Mandatory	String(200)	Full ASCII less the following characters: \$+;=?@£# <>^*()%!
Technical_Contact_Email	Mandatory	String(100)	Accepts: [a-z0-9!#\$%&'*/+=?^_`{ }~-]@[a-zA-Z0-9].[a-zA-Z0-9]
Technical_Contact_Phone	Mandatory	String(30)	Accepts [a-zA-Z0-9_] and [+..]

Note

The Sandbox and Production environments are completely separate. Therefore the contact information for your organisation for the Sandbox and Production environments are held separately

1.12 Error Responses

HTTP Status	Simplified Message PTSB is providing	Cause
400 Bad Request	Request has malformed, missing or non-compliant JSON body, URL parameters or header fields.	The API swagger documentation contains a set of rules which must be followed when sending a request. If a request is submitted with malformed, missing or non-compliant JSON body, URL parameters or header fields then this error will be generated: <ul style="list-style-type: none"> Please note where a request includes an unsafe character (as per RFC 3986) that is not percent-encoded a 400 Bad Request will be returned but the message text may be blank.
	The certificate provided in the HTTP header is invalid.	Not a valid certificate due to: <ul style="list-style-type: none"> root cert validation failure, cert chain check failure, revocation list check failure , QCStatement failure.
	The certificate provided in the HTTP header has expired.	Certificate is expired.
	The certificate provided in the HTTP header has invalid role information.	Invalid role information: <ul style="list-style-type: none"> Cert does not contain any role(s), naming convention of the role(s) are incorrect , role(s) does not match scope requested.
	The certificate provided in the HTTP header has an invalid NCA ID.	Invalid National Competent Authority: <ul style="list-style-type: none"> NCA ID incorrect.
	The certificate provided in the HTTP header has an invalid Organisation ID.	Invalid organisation ID provided.
	A required HTTP header has not been provided.	The JWS "tpp-signature-certificate" header is not provided.
	The signature provided is invalid.	The signature header x-jws-signature was parsed and has a valid JOSE header that complies with the specification. However, the signature itself could not be verified.
	The signature contains an invalid claim.	The JOSE header in the x-jws-signature has one or more claims with an invalid value. (e.g. a kid that does not resolve to a valid certificate). The name of the missing claim should be specified in the path field of the error response.
	A required claim has not been provided in the signature.	The JOSE header in the x-jws-signature has one or more mandatory claim(s) that are not specified. The name of the missing claim(s) should be specified in the path field of the error response.
	The signature provided is malformed.	The x-jws-signature in the request header was malformed and could not be parsed as a valid JWS.
	A required HTTP header has not been provided.	The API request expected an x-jws-signature in the header, but it was missing.
	A header-value is provided in the API request that is not expected.	The API request was not expecting to receive an x-jws-signature in the header, but the TPP made a request that included an x-jws-signature.
	Dynamic Client Registration: /Get Client Key or Secret or Org ID is invalid	Organisation or Client ID does not exist
	Dynamic Client Registration: Invalid JWT provided in POST request	Invalid JWT
	Dynamic Client Registration: Internal Server Problem (failure to create org in Portal etc.)	The method \${method} was not successful
Dynamic Client Registration: /POST {same application name on existing Org}	Application Name already in use	
Dynamic Client Registration: /POST registering new	Organisation Name already in use	

	Organisation but name already exists and is mapped to another Organisation in the Portal	
Dynamic Client Registration:	Organisation Name is empty	Organisation Name is Required
Dynamic Client Registration:	Organisation Name includes illegal Characters	Organisation Name Includes Illegal Characters
Dynamic Client Registration:	Organisation Name character count is invalid	Organisation Name Character Size Invalid. Must be between(1-50) Characters
Dynamic Client Registration:	Application Name is empty	Application Name is Required
Dynamic Client Registration:	Application Name includes illegal Characters	Application Name Includes Illegal Characters
Dynamic Client Registration:	Application Name character count is invalid	Application Name Character Size Invalid. Must be between(1-50) Characters
Dynamic Client Registration:	Redirect URI is empty	Redirect URI is Required
Dynamic Client Registration:	Redirect URI includes illegal characters	Redirect URI is Invalid
Dynamic Client Registration:	Redirect URI character count is invalid	Redirect URI Character Size Invalid. Must be between(5-2048) Characters
Dynamic Client Registration:	Organisation Description includes illegal characters	Organisation Description includes Illegal Characters
Dynamic Client Registration:	Organisation Description character count is invalid	Organisation Description Character Size Invalid. Must be between(1-200) Characters
Dynamic Client Registration:	Application Description includes illegal characters	Application Description includes Illegal Characters
Dynamic Client Registration:	Application Description character count is invalid	Application Description Character Size Invalid. Must be between(1-200) Characters
Dynamic Client Registration:	Organisation Website includes illegal characters	Organisation Website is Invalid
Dynamic Client Registration:	Organisation Website character count is invalid	Organisation Website Character Size Invalid. Must be between(5-200) Characters
Dynamic Client Registration:	Business Contact Name is empty	Business Contact Name is Required
Dynamic Client Registration:	Business Contact Name character count is invalid	Business Contact Name Character Size Invalid. Must be between(2-200)Characters
Dynamic Client Registration:	Business Contact Name includes illegal characters	Business Contact Name is Invalid
Dynamic Client Registration:	Business Contact Email is empty	Business Contact Email is Required
Dynamic Client Registration:		Business Contact Email Character Size Invalid. Must be

	Business Contact Email character count is invalid	between(5-100)Characters
	Dynamic Client Registration: Business Contact Email includes illegal characters	Business Contact Email is Invalid
	Dynamic Client Registration: Business Contact Phone is empty	Business Contact Phone is Required
	Dynamic Client Registration: Business Contact Phone character count is invalid.	Business Contact Phone Character Size Invalid. Must be between(5-30)Characters
	Dynamic Client Registration: Business Contact Phone includes illegal characters	Business Contact Phone is Invalid
	Dynamic Client Registration: Technical Contact Phone is empty	Technical Contact Phone is Required
	Dynamic Client Registration: Technical Contact Phone includes illegal characters	Technical Contact Phone is Invalid
	Dynamic Client Registration: Technical Contact Phone character count is invalid.	Technical Contact Phone Character Size Invalid. Must be between (5-30) Characters.
	Dynamic Client Registration: Technical Contact Name is empty	Technical Contact Name is Required
	Dynamic Client Registration: Technical Contact Name includes illegal characters	Technical Contact Name is Invalid
	Dynamic Client Registration: Technical Contact Name character count is invalid	Technical Contact Name Character Size Invalid. Must be between(2-200)Characters
	Dynamic Client Registration: Technical Contact Email is empty	Technical Contact Email is Require
	Dynamic Client Registration: Technical Contact Email includes illegal characters	Technical Contact Email is Invalid
	Dynamic Client Registration: Technical Contact Email character count is invalid.	Technical Contact Email Character Size Invalid. Must be between(5-100)Characters
	Dynamic Client Registration: Client ID is not provided	Client ID is required
	Dynamic Client Registration: Client Secret is not provided	Client Secret is required
	Dynamic Client Registration: Client Credentials is not provided	Client Credentials are required
401 Unauthorized	Authorization header missing or invalid token. The given client credentials were not valid.	The Header which allows authorisation is missing from the request or contains an invalid token. Invalid API key provided. Invalid Shared Secret provided.
403 Forbidden	Token has incorrect scope or a security policy was violated	User has given incorrect Account Id and Consent Id.
405 Method Not Allowed	The TPP tried to access the resource with a method that is not supported.	User has performed a request with an unsupported method e.g. using POST instead of GET.
406 Not Acceptable	The request contained an Accept header other than permitted media types and a character set other than UTF-8.	User has attempted to send a request in an incorrect character set, required set UTF-8.
415 Unsupported Media Type	The operation was refused because the payload is in a format not supported by this method on the	User tries to send request in body other than specified JSON i.e. XML.

	target resource.	
429 Too Many Request	The operation was refused as too many requests have been made within a certain timeframe.	A user tries to access a resource too frequently.
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request. Something went wrong on the API gateway or micro-service.	User had an unhandled error within their code.
	Dynamic Client Registration Code	User had an unhandled error within their code
	Dynamic Client Registration API version number	User has entered incorrect API version number