Best Practice Guide

PRODA B2B Client

Version 1.2 – 23/11/2020

This document outlines best practice when:

- *Requesting Access Tokens*
- *Refreshing Software Instance Keys*
- *Notifying Users for Software Instance Expiration*

Includes information about PRODA Integration testing performed by PRODA

Table of Contents

1. Summary of Best Practices

This document describes “best practice” for several critical aspects of managing PRODA B2B Software Instances and access token requests. By following the advice in this document, PRODA can continue to be a robust and available authentication platform for clients accessing government services.

PRODA will conduct integration testing to ensure clients are conforming to “best practice” in relation to access token reuse. See section Access Token Integration Testing.

1.1 Access Tokens

- Reuse Access Tokens while still valid
- Do not request a new access token for every call, unless the currently possessed access token has expired.

1.2 Key Expiry

- Track when the device keys are to expire, and ensure new keys are generated and suppled to PRODA before the current key expires.
- Only generate a new key when required.

1.3 Device Expiry

- Keep track of the device’s expiry, and notify the user of impending expiry:
  - 30 days before expiry
  - 7 days before expiry
  - Every day from 7 days till expiry occurs
- Once device has been refreshed and expiry time is reset, messages are no longer generated until next impending expiry event is trigger.

2. Access Token Management

PRODA B2B access tokens have two important attributes:

- can only be used for the Service Provider they were created for
- have a defined lifespan

It is important for the health of the PRODA authorisation server that an access token is reused while it is still valid, and new tokens are not requested for the same Service Provider while an existing token is still valid.

Part of the NOI testing process includes a test to determine if the software is following best practice around access token reuse, so please employ a strategy to reuse valid access tokens.

2.1 Access Token Request Best Practice

2.1.1 Do I Need a New Token?

Before requesting a new access token determine if the device already has a valid token for the relevant Service Provider. This may be achieved by storing access tokens (or a reference to an access token) in a map, relating the Service Provider name (the map "key") to an object (the map "value") containing the expiry time and the access token. It is then a trivial matter to determine if the existing access token is still valid, and then using it if it is valid, or requesting a new one if not valid.

An example of a map might look like this:

```
[
        "Medicare" => {expiry: "2020-03-12 16:05:31", token: "ebyKs6R…….Ug5td"},
        "PBS" => {expiry: "2020-03-01 16:31:10", token: "ebyjU53…….p9iJH1"},
        "CCS" => {expiry: "2020-03-12 16:18:42", token: "eby9iT…….P9ypo1"}
]
```

This structure would make it easy to determine if a new token was required for the Service Provider, i.e. the expiry time recorded against the token is in the past, or about to be in the past. As the absolute time on two computers would never be the same, it would be a good idea to set an expiry threshold of a minute or two (subtract this from the calculated expiry time) to mitigate against using a token that is about to expire.

There are many strategies that may be used to store and relate access tokens to an expiry time, using a map is only a suggestion.

2.1.2 Store New Tokens

When a new access token is acquired, store it in a way that it is easy to determine when it will expire. The previous example of a map would be a simple structure for this purpose, although there are many programmatic strategies that could be used.

The easiest way to calculate the expiry timestamp of a newly acquired access token is to use the current time, and add seconds to it based on the number of seconds contained in the "expires_in" field from the Access Token Request Response (see Raw Response Example). A good practice would be to only add 80% of this value to mitigate against using an about to expire token that may have expired before being received by the Service Provider.

```
expiryTime = currentTime + (0.8 * expires_in)
```

Another more complex, but possibly more accurate, way to obtain the expiry time of the access token is to decode the token itself, and look at the "exp" field (see Payload). This value is a Unix epoch time formatted timestamp. Again, to mitigate against using an about to expire token, it is good practice to subtract a number of minutes from this expiry time.

2.2 Access Token Integration Testing

PRODA will conduct integration testing of your software before a production client identifier is issued to your software. This testing ensures that the PRODA authorisation system stays healthy for all users, and is not affected by software that spams PRODA's authorisation service for access tokens unnecessarily.

2.2.1 Integration Testing Requirements

The following test will need to be passed before PRODA issues a production client id for your software:

- Over a 2 hour period at least 12 successful requests to a Service Provider API must be made by a single running instance of the software, including a valid PRODA access token in the request.
    - A maximum of 10 minutes between these API calls

This test will be judged on the pattern of PRODA access token requests made by the software, evidence of token reuse must be displayed:

- Access tokens have a generic lifespan of 60 minutes, **testing will fail** if access tokens are requested before 50% of this lifespan has expired (30 minutes).
- PRODA will use Service Provider and PRODA access logs to determine the pattern of access token requests and verify successful access to relying party APIs.

3. Device Key Management

Software devices must refresh their signing keys every so often (currently every 11 days, but this may be variable depending on the security requirements of a Service Provider). Software is expected to automatically refresh these signing keys before they expire, else an authorised Organisation user must reactivate the device using the PRODA user interface, possibly disrupting their services while the software is unauthorised.

3.1 Determine When the Device Signing Key Will Expire

The device's signing key's expiry timestamp is obtained from an access token request's response (see Response Field Descriptions) in the key_expiry field. This timestamp is in Australian Eastern Time (Standard or Daylight, depending on what is currently active).

The device key should be refreshed before the expiry timestamp. A good strategy would be to perform the key refresh a day or two before the key expires.

Some software may not be connected to the internet 24/7, so different strategies may be employed to effect key refreshes. On the extreme end of the spectrum, the key could be refreshed once per day, or when an internet connection is detected. This might be suitable for software that is only run once a week.

Software that is connected to the internet 24/7 might refresh its key a day before the key expires.

4. Device Expiry Management

For security reasons B2B Device definitions in PRODA have a defined active period. When this time period has expired, the device becomes inactive and cannot gain authorisation from PRODA. Before this period has expired an Organisation's authorised user must reconfirm the device is still being used via the PRODA web site; this resets the device expiry period.

Access token and key refresh request responses contain the Device expiry timestamp (see Response Field Descriptions), allowing the Device's code to determine when to notify the user when the Device expiry time is approaching.

This timestamp is in Australian Eastern Time (Standard or Daylight, depending on what is currently active).

Currently the lifespan of a Device definition is 180 days, but may be different based on the security requirements of the Service Provider. It is recommended to notify the user of impending expiry at the following time intervals:

- 30 days from expiry
- 14 days from expiry
- Every day from 7 days to the expiry day.

It is recommended to use a method to deliver the notification that the user is forced to acknowledge, such as a popup with OK button.

5. Access Token Request Example Data

An access token request returns several data fields required for proper management of the Software Instance and the access token itself.

5.1 Data Fields in an Access Token Request Response

The response to an access token request is a JSON string.

5.1.1 Raw Response Example


```
{
        "access_token":"eyJraWQiOiJWcms3MkhjWjdMYmFUdWFkdTl3NnZPYlk5bElUTF80TEJ0YjVP
eWdqV3BFIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI5NTQ0NTMxODI1IiwiYXVkIjoiUFJPREEuVU5BVFRFTk
RFRC5CMkIiLCJwcm9kYS5zd2luc3QiOiJ6YXBob2RzLXNwYWNlLWNvbm5lY3RvciIsInByb2RhLnR5cGUi
OiJVTkFUVEVOREVELkIyQiIsInByb2RhLm9yZyI6Ijk1NDQ1MzE4MjUiLCJwcm9kYS5ycCI6IlBCUyIsInByb
2RhLnNwIjpbIlBCUyJdLCJwcm9kYS5hdWQiOiJodHRwczovL21lZGljYXJlYXVzdHJhbGlhLmdvdi5hdS9QQ
lMiLCJpc3MiOiJodHRwczovL3Byb2RhLmh1bWFuc2VydmljZXMuZ292LmF1IiwiaWF0IjoxNTc5MjE1MD
g2LCJleHAiOjE1NzkyMTg2ODZ9.M2JJYzSVsuUBNFSX_mMCk3t9RF73plokfkT1Wt__bjxlh5strvVd52Uk
OBmxWFuXrIFXovBCSlHUKDDGwe3X12b2bMTi0P7cFIHN9kZfwgcYUr-
SjMTlnJqFG2S_f7ImKkyjVM601xOcKjw0QH_-
dYyJDstUwAwpB3fkG430qsYxlrHhFCXityevvJGH3e2vv6ri-
8IxIia_WQDa1p06JsOsWUKTdkqBuk9uzrr5U8f9AFoRA4DAb_KYSVclVsFdZ7a-
zKb9dEEDEY_TdGeFbOwxBdOhag-zFY9NqmckIIfSLImesmJP2gjBnXzP4-
Zk8Uak8yQg0_R9Qrd5GflV5g",
        "scope":"",
        "device_expiry":"2025-01-17 09:49:08",
        "key_expiry":"2025-01-20 09:49:08",
        "token_type":"bearer",
        "expires_in":3600
}
```


5.1.2 Response Field Descriptions

- **access_token** – this field contains the access token. It is a signed JWT, containing encoded information relating to the authorisations the access token possesses.
- **scope** – contains information about the scope of the authorisations the access token possesses. PRODA B2B Unattended currently does not use the scope field.
- **device_expiry** – this is the date and time that the Software Instance (or Device) will expire and no longer be able to request access tokens from PRODA. The Software Instance must be refreshed by an authorised Organisational user via the PRODA web site before this expiry timestamp to prevent deactivation, causing disruption to the Organisation's business.
  Use this value to determine when the device needs to generate a message to the user that the device needs to be refreshed in PRODA by an authorised user.
- **key_expiry** – this is the date and time that the Software Instance's signing key will expire. The Software Instance must refresh the signing key before it expires, otherwise an authorised Organisational user will be required to reactivate the Software Instance via the PRODA web site, causing disruption to the Organisation's business.
  Use this value to determine when the device needs to generate a new key and submit it to PRODA.
- **token_type** – PRODA only issues "bearer" tokens.
- **expires_in** – the number of seconds till the access token expires and becomes inactive.

5.2 Data Fields in an Access Token

A PRODA access token is a signed JWT.

5.2.1 Decoded Access Token

The two main components of a decoded access token are the header and the payload. These are JSON strings.

5.2.1.1 Header


```
{
 "kid": "Vrk72HcZ7LbaTuadu9w6vObY9lITL_4LBtb5OygjWpE",
 "alg": "RS256"
}
```


5.2.1.2 Payload


```
{
 "sub": "9544531825",
 "aud": "PRODA.UNATTENDED.B2B",
 "proda.swinst": "zaphods-space-connector",
 "proda.type": "UNATTENDED.B2B",
 "proda.org": "9544531825",
 "proda.rp": "PBS",
 "proda.sp": [
  "PBS"
 ],
 "proda.aud": "https://medicareaustralia.gov.au/PBS",
 "iss": "https://proda.humanservices.gov.au",
 "iat": 1579215086,
 "exp": 1579218686
}
```


5.2.2 Access Token Field Descriptions

- kid – key identifier; this is the identifier for the key that was used to sign the JWT access token.
- alg – the algorithm used to sign the JWT access token
- sub – the subject of the access token. This is the PRODA Organisation RA of the organisation that owns the Software Instance that requested the access token.
- aud – (depreciated) this is the audience of the access token; the Service Provider that the token will be presented to for access to their services. Will always have a value of PRODA.UNATTENDED.B2B.
- proda.swinst – this is the name of the Software Instance that requested the access token.
- proda.type – this is the type of access token. Will always have a value of UNATTENDED.B2B.
- proda.org - This is the PRODA Organisation RA of the organisation that owns the Software Instance that requested the access token.
- proda.rp – this is the PRODA code that identifies the Service Provider the access token was requested for.
- proda.sp – this is the set of Service Providers the access token was requested for. Some Service Providers are closely related and allow the use of access tokens that are valid for multiple Service Providers.

- proda.aud – The audience of the access token. This value may represent an individual Service Provider, or a group of Service Providers.
- iss – the issuer of the access token. This will always be *https://proda.humanservices.gov.au*
- iat – this is the issued at timestamp of the access token, in Unix epoch time format.
- exp – this is the expiry timestamp of the access token, in UNIX epoch time format.