Inland Revenue
Te Tari Taake

Business Transformation

Inland Revenue

Build Pack

Identity and Access Services

Date: 04/09/2017
Version: 1.5

IN CONFIDENCE

About this Document

This document is intended to provide Service Providers with the technical detail required to consume the Identity and Access services offered by Inland Revenue.

This is a standalone technical document that supports the on boarding processes of an end to end solution. The associated on-boarding document(s) describe the end-to-end business level solution, of which this build pack is part. This document describes the architecture of the technical solution, the interaction with other build packs, schemas and endpoints. Also included are sample payloads to use in non-production environments.

Inland Revenue
Te Tari Taake

Contents

1 Overview

1.1 This solution

Inland Revenue (IR) is establishing a new set of Identity and Access services. These will provide Service Providers with authentication and authorisation mechanisms for accessing IR's new Gateway Services.

There are two distinct entities for which IR provides mechanisms for authentication and authorisation:

1. Organisations
2. End Users

The mechanisms are as per the diagram below:

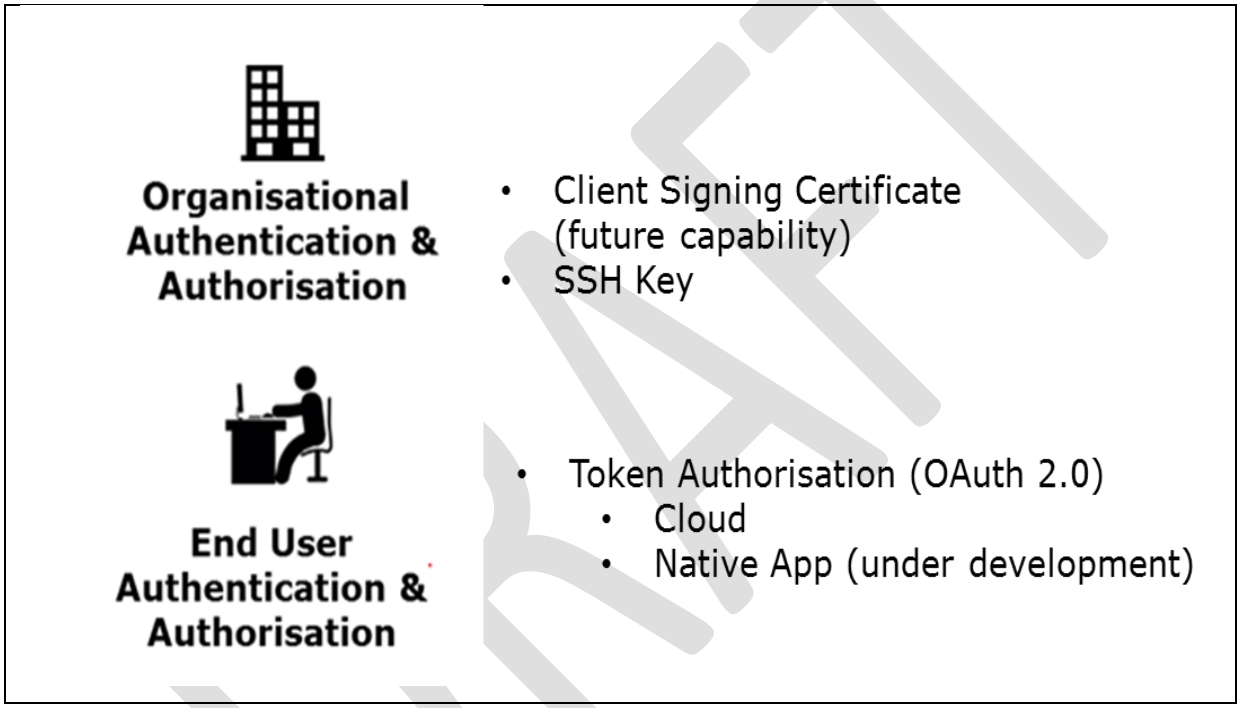


Figure 1 Entities and Authentication & Authorisation mechanisms

1.1.1 Organisational Authentication and Authorisation.

The table below details the current and future mechanisms IR provide to authenticate and authorise an organisation for Machine to Machine (M2M) communication.

Authorisation Mechanism	Uses
SSH keys	This mechanism is used in SFTP file transfers to identify the organisations sending/receiving files. SSH keys need to be exchanged to authenticate both parties.
Client Signing Certificate X.509 certificate based]	Used when the Service Provider server is autonomous. X.509 client certificates are used to sign messages in order to identify the service provider to IR. Service providers will be able to register their certs with IR through a self-help

(Future capability)	portal. This is not currently available for use.

Table 1 Organisational Authentication and Authorisation Methods

1.1.2 End-User Authentication and Authorisation.

The OAuth 2.0 process is used to authenticate end-users using their IR user ID and password and grant 3rd party software consent to access their information.

The OAuth 2.0 mechanism to be used by a service provider is based upon the nature of the client application the end-user will be using.

Authorisation Mechanism	Uses
Cloud application Token Auth OAuth2.0 based	Use when the client application is a web-enabled cloud based application. It requires an online user to enter their myIR user ID and password to grant the application access to their IR information.
Native application Token Auth OAuth2.0 based (Future Capability)	Use when the client application is a desktop or other native application. It also requires an online user to enter their myIR user ID and password to grant the application access to their IR information. See section 2.2 Native Application Token Auth below for details This is not currently available for use.

Table 2: End-User Authentication and Authorisation Methods

Note that currently only Cloud application Token Auth is available for use. This document will be further updated when Native application Token Auth is made available.

1.2 Intended audience

This build pack and the resources to which it refers are primarily focused on the needs of Software Developers' technical teams and development staff.

The reader is assumed to have a suitable level of technical knowledge in order to comprehend the information provided. A range of technical terms and abbreviations are used throughout this document, and while most of these will be understood by the intended readers, a glossary is provided at the end.

This document is not intended for use by managerial staff or those with a purely business focus.

1.3 Information IR will provide Service Providers

1.3.1 Token Auth (Cloud or Native)

1. URLs and parameters for invoking the Authentication services.
2. Client ID (agreed with service consumer)
3. Client secret (used in step 2 in section 2.1.2)

1.3.2 SSH keys

1. SSH keys for SFTP.
2. PGP public keys if used for payload encryption and signing

1.4 Information Service Providers must provide IR

1.4.1 Service Provider Information

1. Full business name
2. Client ID (agreed with IR and used in requests to IR))
3. Key Contact
4. Email of key contact or delegate
5. Mobile Phone number (SMS may be used for some information)
6. IP addresses Service Providers will use for test instances for IR firewall whitelisting.

1.4.2 Token Auth (Cloud or Native)

1. Redirect URI for Authorisation code and Authentication token.

1.4.3 SSH keys

1. SSH public keys if using SFTP.
2. Their own Public Keys if using PGP.

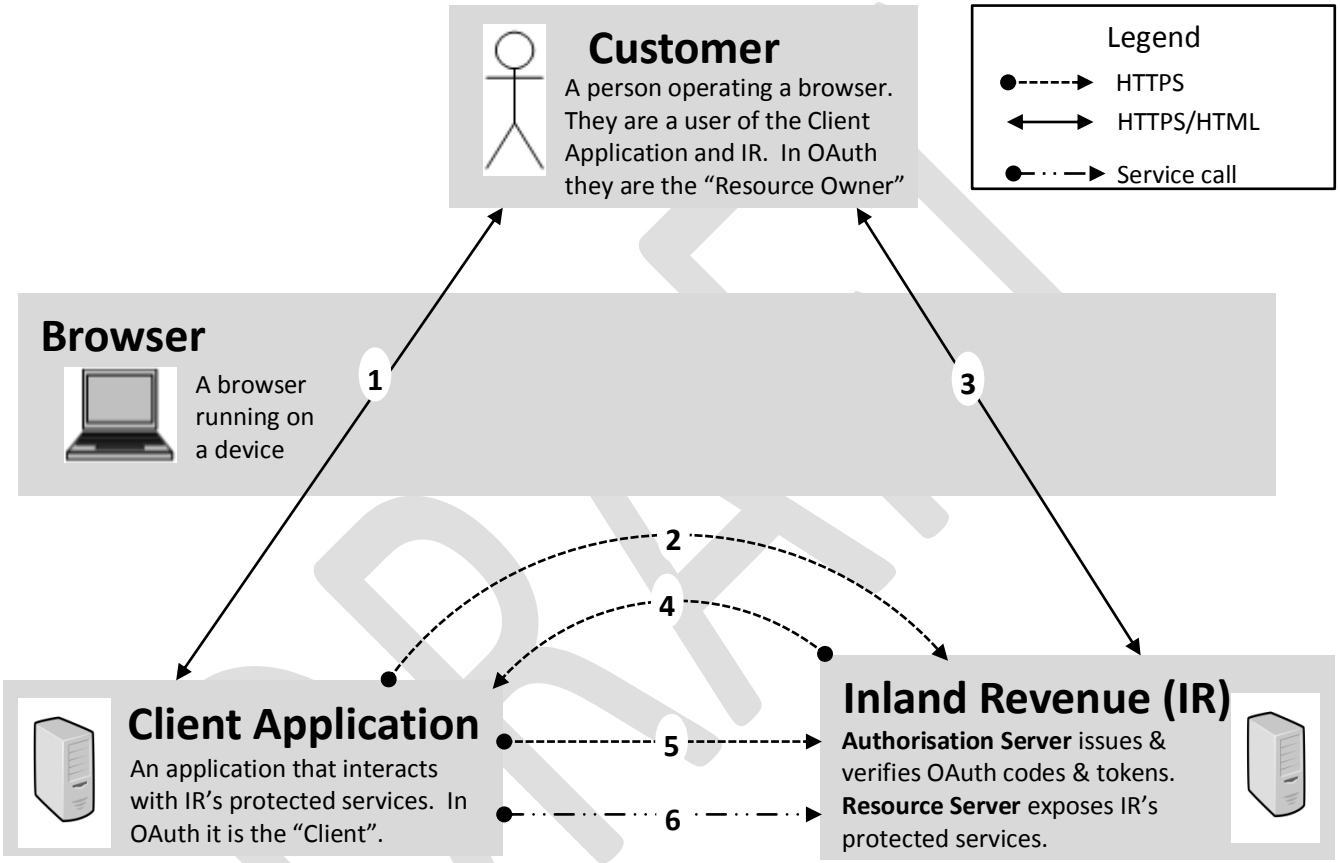
2 Description of the IR Authentication Mechanisms

2.1 IR Token Auth Implementation using OAuth 2.0

This section describes the IR OAuth 2.0 implementation. This high-level description covers both Cloud and Native application usage.

2.1.1 High Level View of OAuth 2.0

For OAuth 2 the following diagram depicts the high level end-to-end view of the components and the interactions between them:



1. The User is interacting with the Client Application. They access a protected service provided by IR (e.g. to file a return, retrieve a balance etc.)
2. The Client Application invokes the Authorisation API to get an authorisation code, the user's browser is redirected to IR's logon page.
3. IR prompts the User to logon, they are authenticated. On first use the User must also supply their consent for the Client Application to access IR on their behalf. IR issues the Authorisation Code.
4. The Authorisation Code is returned to the Client Application.
5. The Client Application invokes IR's Token service to redeem the Authorisation Code for an OAuth Access Token. It has a finite time to live.
6. The Client Application can then invoke IR's protected services (e.g. to file a return etc.) supplying the OAuth Access Token in the header. The OAuth Access Token can be used for multiple calls until it expires.

Inland Revenue's implementation of the OAuth 2 standard conforms to the Authorisation Code Grant flow described in section 4.1 of RFC 6749 (https://tools.ietf.org/html/rfc6749).

2.1.2 Cloud and Native Application OAuth 2.0 Steps

This section describes the steps and service calls required when using the IR implementation of OAuth 2.0. These are the same for both Cloud and Native App usage.

2.1.2.1 Customer accesses the Client Application (Step1)

The Customer accesses the Client application and triggers the need for it to consume one of Inland Revenue's protected services (e.g. to retrieve an account balance, to file a return etc.).

2.1.2.2 Request Authorisation Code (Step 2)

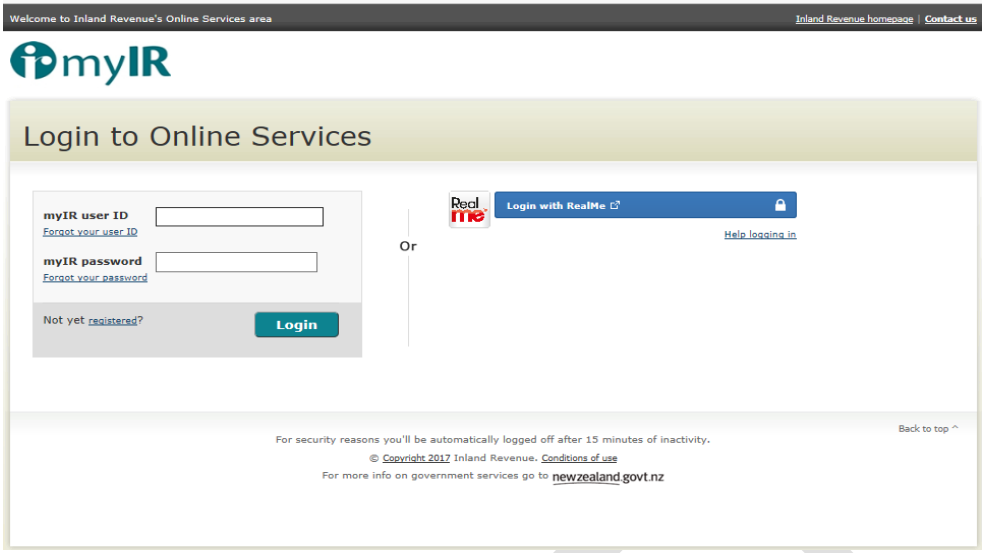
The customer's browser is redirected to the IR Authorisation service to authenticate the user and confirm scope using the GET method described below:

```
https://q.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/authorize?

response_type=code

&client_id=IdOfCompanyUsingTheAPI

&redirect_uri=http://client.example.com/return

&scope=GWS

&state=xyz
```

Name	Description	Required	Valid Values
response_type	Response type requested	Required	"code"
client_id	The agreed Client identifier established at registration. Inland Revenue maintains this list of values.	Required	client_id
redirect_uri	The Client application's redirect URI to which the Authorisation Code is returned.	Required	Business Partner defined
scope	Use space-separated values. Define scope values in the configuration/scope registry.	Required	"GWS"
state	A value used by the Business Partner to maintain state between the request and callback. The parameter should be used to prevent cross-site forgery requests.	Recommended	Business Partner defined

2.1.2.3 Login with myIR Credentials (step 3)

During this step the customer may be required to authenticate, and, if this is required, will be redirected to the following myIR logon screen. For OAuth 2.0 for Native Apps this authorisation request is in an external user-agent (typically the browser).



If the software provider chooses (not generally recommended as this page may change from time to time) to present this page within a frame the minimum recommended size in pixels is 600w x 500h.

Note the customer must already have an IR Online Services credential.

Invalid User ID or password will return a HTTP:200

2.1.2.4 Respond with Authorisation Token (Step 4)

If successful, the authorisation service will respond with the Authorisation Code to the Business Partner redirect_uri as described below:

```
https://client.example.com/return?code=eyJhbG...rWWk8hbs_o6uY&state=xyz
```

Name	Description	Valid Values
Code	Authorisation code value - Includes the following: • Expiry • Client_id • Redirect_uri	Encrypted string ~1000 characters
State	Business Partner defined state	Business Partner defined (as passed)

If not successful an error is sent with a HTTP code and a JSON response containing the error code and description.

Errors are:

HTTP code	Error Type	Description
400	**invalid_redirect_uri**	Redirect URI mismatch with Business Partner app
	Invalid client ID	API Key contains invalid information
	invalid_client	Business partner identifier invalid

	invalid_scope	Requested scope is invalid, unknown, or malformed
	server_error	Authentication - Runtime processing error
	access_denied	End-user denied authorisation
500	**InternalError**	An internal and unexpected error occurred
504	**GatewayError**	Gateway did not receive a timely response from the upstream server

2.1.2.5 Request Authorisation Token (Step 5)

Once an Authorisation Code has been returned to the Client application it must be exchanged for an OAuth Access Token by doing an HTTPS Post to the Token Service as follows:

```
https://q.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/tokens

<form>

    redirect_uri=http%3A%2F%2Fclient.example.com:17001%2Freturn

    &grant_type=authorization_code

    &code=eyJhbG...rWWk8hbs_o6uY

 </form>
```

With the values of:

Name	Description	Required	Valid Values
redirect_uri	The Client application's redirect URI for the Authorisation Token.	Required	Business Partner defined
grant_type	The grant type is authorization_code	Required	authorization_code
code	Authorisation Code as supplied by the authorisation service in step 4	Required	Encrypted string ~1000 characters

The header fields contain the shared secret and content type:

```
Authorization: Basic NTQzMjFpZ...ZWxjb211MQ==

Content-Type: application/x-www-form-urlencoded;charset=UTF-8
```

With the values of:

Name	Description	Required	Valid Values
Authorization	"Basic " + Base64 encoded (ClientID + ":" + Client Secret)	Required	Base64 encoded string
Content-Type	Content type	Required	application/x-www-form-urlencoded;charset=UTF-8

The response contains the OAuth Access Token – this should be passed on subsequent service calls.

If not successful an error is sent with a HTTP code and a JSON response containing the error code and description.

Errors:

HTTP code	Error Type	Description
400	**invalid_redirect_uri**	Redirect URI mismatch with Business Partner app
	Invalid client ID	API Key contains invalid information
	Invalid client_id or client_secret	API Secret contains invalid information
	invalid_client	Business partner identifier invalid
	invalid_scope	Requested scope is invalid, unknown, or malformed
	server_error	Authentication - Runtime processing error
	access_denied	End-user denied authorisation
500	**InternalError**	An internal and unexpected error occurred
504	**GatewayError**	Gateway did not receive a timely response from the upstream server

2.1.2.6 Refresh Token (Standalone step)

A token refresh process is available that allows the client application to request an additional access token with the same scope if the original token has expired.

In this scenario the typical access token (Request Authorisation Token (Step 5) above) response contains an additional parameter:

"refresh_token": "tGzv3JOkF0XG5Qx2TlKWIA"

This token is used to request another access token at a later point via the same request as shown in Step 5, but using a different grant type e.g.

```
https://q.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/tokens

<form>

    redirect_uri=http%3A%2F%2Fclient.example.com:17001%2Freturn

    &grant_type=refresh_token

    &refresh_token=<refresh-token-value>

</form>
```

Where refresh-token-value will be the value returned in original request.

2.1.2.7 Revoke Token (Standalone step)

A token revoke process is not currently supported.

2.1.3 Security Considerations

Protecting the integrity of the Client Secret is an important requirement for providers, the exact implementation is left to the provider but it must not be stored in plain text either in the web, mobile, or desktop application. Our preference is for this to be stored on a back-end server and made available to the Business Partner application.

If a Client Secret is compromised it shall be invalidated and a new secret issued.

The OAuth Authorisation Code has a time to live of 15 minutes.

The OAuth Access Token has a time to live of 30 minutes.

The Refresh token has a time to live of 60 minutes. This refresh capability is a topic for discussion and Service Provider feedback on this point is encouraged.
Inland Revenue is keen to understand how Service Providers currently treat or are intending to treat user inactivity on their applications.

2.1.4 Endpoints

Endpoints for the token based Authentication and Authorisation Service are as follows:

Test Environments:

Code: https://q.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/authorize

Token: https://q.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/tokens

Production Environment

Code: https://services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/authorize

Token: https://services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/tokens

2.2 Native Application Token Auth

The OAuth 2.0 transaction flow described above will not change for Service Providers running Native Applications. A Native app is an application that is installed by the user to their device or a desktop application, as distinct from a web app that runs in the browser context only.

However, changes by the Service Provider and IR will be necessary to implement this standard and is therefore not currently supported.

IR is planning to adopt OAuth2 for Native apps standard for Native applications.

2.3 SSH Keys

This authentication and authorisation mechanism is used in SFTP file transfers to identify the respective organisations sending/receiving files, in this case IR and the Service Provider.

SSH Key authentication and authorisation will be used for file transfers in which SFTP 3.0 is used. This version of SFTP requires the use of SSH version 2.0.

The public key algorithm for SSH authentication keys must be ECDSA with a minimum field/key size of at least 160 bits.

Certain FTP file transfers will also require payload encryption and signing to ensure that once a file is transferred to an endpoint only an authorised party can interpret it. This is optional and the need for this will be identified in the respective On-boarding pack for a file transfer.

The need will be based upon the NZISM Information classification privacy rating based upon the sensitivity of the customer data along with considerations such as the volumes being transferred.

For files from IRD to partners that have PGP the PGP encryption will use Advanced Encryption Standard (AES) with a 256-bit key and the PGP hashing will use Secure Hash Algorithm (SHA) SHA-256.

Currently IR has the ability to push and pull files being exchanged, but the Service Provider always hosts the FTP server.

2.4 Client Signing Certificate

This is Inland Revenue's future M2M authentication mechanism to allow easy on-boarding of clients though a self-help portal.

A X.509 client certificate is used to provide IR with the identity of the Organisation when connecting to the mass market M2M interfaces.

As stated previously in this document, this mechanism is not yet available for use.

3 Appendix A – Sample payloads

3.1 Request Authorisation Code

Service consumer to IR. Refer to step 2 in section 2.1.1.

3.1.1 Request

Service provider will send a HTTP GET request to Auth Server, url format as below:

https://g.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/authorize?client_id=Test30206492&redirect_uri=https://jenkins.kiwisaver.ird.digitalpartner.services/&scope=MYIR.Services&response_type=code

3.2 Authorisation Code response

IR to Service consumer. Refer to step 4 in section 2.1.1.

3.2.1 Success Response – Authorisation Code sent

https://jenkins.kiwisaver.ird.digitalpartner.services/?code=eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCIsIng1dCI6Ino5VFR1R2pHSVJNd01pQjRaUGxjbzhDZ3JlOCIsImtpZCI6Im9yYWtleSJ9.eyJvcmFjbGUub2F1dGgucmVkaXJlY3QtdXJpIjoiaHR0cHM6Ly9qZW5raW5zLmtpd2lzYXZlci5pcmQuZGlnaXRhbHBhcnRuZXIuc2VydmljZXMvIiwic3ViIjpudWxsLCJvcmFjbGUub2F1dGgudXNlcl9vcmlnaW5faWRfdHlwZSI6IkxEQVBfVUlEIiwib3JhY2xlLm9hdXRoLnVzZXJfb3JpZ2luX2lkIjoic2FtbXkzOTAiLCJpc3MiOiJJbmxhbmRSZXZlbnVlIiwib3JhY2xlLm9hdXRoLnN2Y19wX24iOiJPQXV0aFNlcnZpY2VQcm9maWxlIiwiaWF0IjoxNTAyMzIxNDI3LCJvcmFjbGUub2F1dGgudGtfY29udGV4dCI6ImF6YyIsImV4cCI6MTUwMjM1MDIyNywicHJuIjpudWxsLCJqdGkiOiJhNDg2ZTU1Ny0zZTc1LTQ3ZmYtODk0NC1hNTcxZWNlYzgzNmYiLCJvcmFjbGUub2F1dGguc2NvcGUiOiJNWUlSLlNlcnZpY2VzIiwib3JhY2xlLm9hdXRoLmNsaWVudF9vcmlnaW5faWQiOiJUZXN0MzAyMDY0OTIiLCJ1c2VyLnRlbmFudC5uYW1lIjoiRGVmYXVsdERvbWFpbiIsIm9yYWNsZS5vYXV0aC5pZF9kX2lkIjoiMTIzNDU2NzgtMTIzNC0xMjM0LTEyMzQtMTIzNDU2Nzg5MDEyIn0.NTBu3R-JwaaOWfvMdWAHqY7Ji3YI3I-bSTXqx6jauqEUhswLmAG6cbpGaSky50ECbHNv2skU8WVZ0RYv67KPgITGXJz0ZKSjqOgiZ0R4kFCZ7asN8yjIzXgxwWk4mPXL5E02u24-VMbr_hrNZYDZbakOpz4uY6UlSSNECmw0ac8

3.3 Request Authorisation token

Service consumer to IR. Refer to step 5 in section 2.1.1.

3.3.1 Exchange Authorisation Code for oAuth token.

At this step as a service provider you need to send a HTTP POST request to Auth Server with Authorization Code and your credentials.
Url format:

https://g.services.ird.govt.nz/ms_oauth/oauth2/endpoints/oauthservice/tokens

```
Request headers: {Authorization=[Basic
VGVzdDMwMjA2NDkyOk9hdXRoMklSU2VjcmV0dA==], Content-Type=[application/x-www-form-
urlencoded], Content-Length=[1127]}
```

```
Request body:
redirect_uri=https%3A%2F%2Fjenkins.kiwisaver.ird.digitalpartner.services%2F&gran
t_type=authorization_code&code=eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCIsIng1dCI6Ino5V
FR1R2pHSVJNd01pQjRaUGxjbzhDZ3JlOCIsImtpZCI6Im9yYWtleSJ9.eyJvcmFjbGUub2F1dGgucmVk
aXJlY3QtdXJpIjoiaHR0cHM6Ly9qZW5raW5zLmtpd2lzYXZlci5pcmQuZGlnaXRhbHBhcnRuZXIuc2Vy
dmljZXMvIiwic3ViIjpudWxsLCJvcmFjbGUub2F1dGgudXNlcl9vcmlnaW5faWRfdHlwZSI6IkxEQVBf
VUlEIiwib3JhY2xlLm9hdXRoLnVzZXJfb3JpZ2luX2lkIjoic2FtbXkzOTAiLCJpc3MiOiJJbmxhbmRS
ZXZlbnVlIiwib3JhY2xlLm9hdXRoLnN2Y19wX24iOiJPQXV0aFNlcnZpY2VQcm9maWxlIiwiaWF0Ijox
```

NTAyMzIxNDI3LCJvcmFjbGUub2F1dGgudGtfY29udGV4dCI6ImF6YyIsImV4cCI6MTUwMjM1MDIyNywi
cHJuIjpudWxsLCJqdGkiOiJhNDg2ZTU1Ny0zZTc1LTQ3ZmYtODk0NC1hNTcxZWNlYzgzNmYiLCJvcmFj
bGUub2F1dGguc2NvcGUiOiJNWUlSLlNlcnZpY2VzIiwib3JhY2xlLm9hdXRoLmNsaWVudF9vcmlnaW5f
aWQiOiJUZXN0MzAyMDY0OTIiLCJ1c2VyLnRlbmFudC5uYW1lIjoiRGVmYXVsdERvbWFpbiIsIm9yYWNs
ZS5vYXV0aC5pZF9kX2lkIjoiMTIzNDU2NzgtMTIzNC0xMjM0LTEyMzQtMTIzNDU2Nzg5MDEyIn0.NTBu
3R-JwaaOWfvMdWAHqY7Ji3YI3I-
bSTXqx6jauqEUhswLmAG6cbpGaSky50ECbHNv2skU8WVZ0RYv67KPgITGXJz0ZKSjqOgiZ0R4kFCZ7as
N8yjIzXgxwWk4mPXL5E02u24-VMbr_hrNZYDZbakOpz4uY6UlSSNECmw0ac8

3.4 Request Refresh token

Service consumer to IR. Refer to section 2.1.2.6.

3.4.1 Refresh request

Sample payloads will be available when this functionality is available.

3.4.2 Refresh token reply

Sample payloads will be available when this functionality is available.

3.4.3 Error Response.

Sample payloads will be available when this functionality is available.

3.5 Revoke token request

Service consumer to IR. Refer to section 2.1.2.7.

3.5.1 Revoke token request

Sample payloads will be available when this functionality is available.

3.5.2 Revoke token reply

Sample payloads will be available when this functionality is available.

3.5.3 Revoke token Error Response.

Sample payloads will be available when this functionality is available.

4 Appendix B – Glossary

<Terminology used in this document>

Term	Meaning
Abbreviation/Term	Description
Client Application	A Client Application is an operating instance of Software that is deployed in one or more sites. A number of deployment patterns are possible: • A single cloud based instance with multiple tenants and online users, • An on-premise instance (e.g. an organisation's payroll system) • A desktop application with an online user.
Customer	A Customer is the party who is a tax payer or a participant in the social policy products that are operated by Inland Revenue. The Customer might be a person (an "individual") or a non-individual entity such as a company, trust, society etc. Practically all of the service interactions with Inland Revenue are about a Customer (e.g. their returns, accounts, entitlements etc) even though these interactions might be undertaken by an Intermediary on their behalf.
Intermediary	A party who interacts with Inland Revenue on behalf of a Customer. Inland revenue's Customer is a Client of the Intermediary. There are several types of Intermediary including Tax Agents, PTSIs, PAYE Intermediaries etc.
Mutual authentication	refers to two parties authenticating each other at the same time, being a default mode of authentication in some protocols (e.g. SSH) and optional in other (TLS)
OAuth 2.0	OAuth 2.0 is an industry-standard protocol for authorization
Native app	An application that is installed by the user to their device, as distinct from a web app that runs in the browser
Protected Service	A general term for the business related web services that are accessed once authentication has occurred (e.g. the Return Service, the Intermediation Service, the Correspondence Service). This document describes the mechanisms that are used to authenticate access to Protected Services.
SFTP	Secure File Transport Protocol
Software	This is the computer software that contains interfaces to (consume) the services that Inland Revenue exposes. Software is developed and maintained by a Software Developer and subsequently deployed as one or more Client Applications.

Software Developer	The person or people who design, implement and test Software. This build pack and the resources to which it refers are primarily focused on the needs of Software Developers. They might be commercial vendors of software or an in-house developer of software.
TLS 1.2	A cryptographic protocol that provides communications security over a computer network. Version 1.2 is mandated in most cases.
WS-Security	An extension to SOAP to apply security to Web Services. An OASIS Web service specification
X.509 Certificate	A digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.