Inland Revenue

Build Pack: Name API

Date: 17/09/2021

Contents

1 Overview

1.1 This solution

Inland Revenue has a suite of digital services available for consumption by our service providers that supports efficient, electronic business interactions with Inland Revenue.

The application programming interface (API) described in this build pack document provides the ability to update the names held by Inland Revenue.

Before continuing, please consult
www.ird.govt.nz/digital-service-providers/services-catalogue
for business-level context, use cases and links to relevant policy.
The information available here explains how to integrate with Inland Revenue's services.

1.2 Intended audience

Access to the API end point is open to any software provider that has been on-boarded to the API (referred to throughout the remainder of this document as 'Digital Service Providers'). Access to the account data is open to any logon that currently has access to these resources on eServices. This includes tax intermediaries (such as tax agents and bookkeepers) and to customers using software on their own behalf.

The solution outlined in this document is intended to be used by technical teams and development staff, as it describes the technical interactions provided by this service. The reader is assumed to have a suitable level of technical knowledge to comprehend the information provided.

1.3 Related services

The following application programming interfaces (APIs) complement this Gateway Service. Instructions on where to find the build packs for these APIs can be found in section 3 of this document.

1.3.1 Identity and Access Services (required)

The Identity and Access Services (IAS) are used to authenticate access. Authentication tokens will need to be retrieved via IAS prior to making calls to this API.

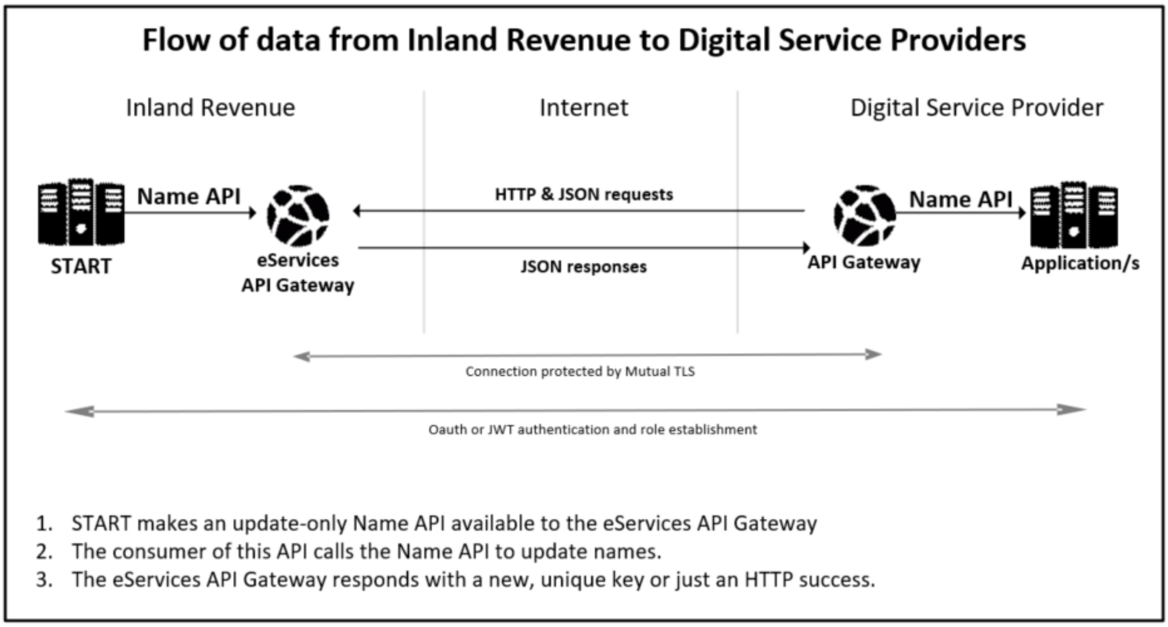
2 Solution design

2.1 Architecture

Inland Revenue is offering a suite of web applications in order to facilitate interactions via software packages.

This API will be used by approved organisations to create or update non-legal names (for example 'Preferred' or 'Doing Business As') from Inland Revenue.

The diagram below illustrates the flow of data from Inland Revenue to the Digital Service Providers.



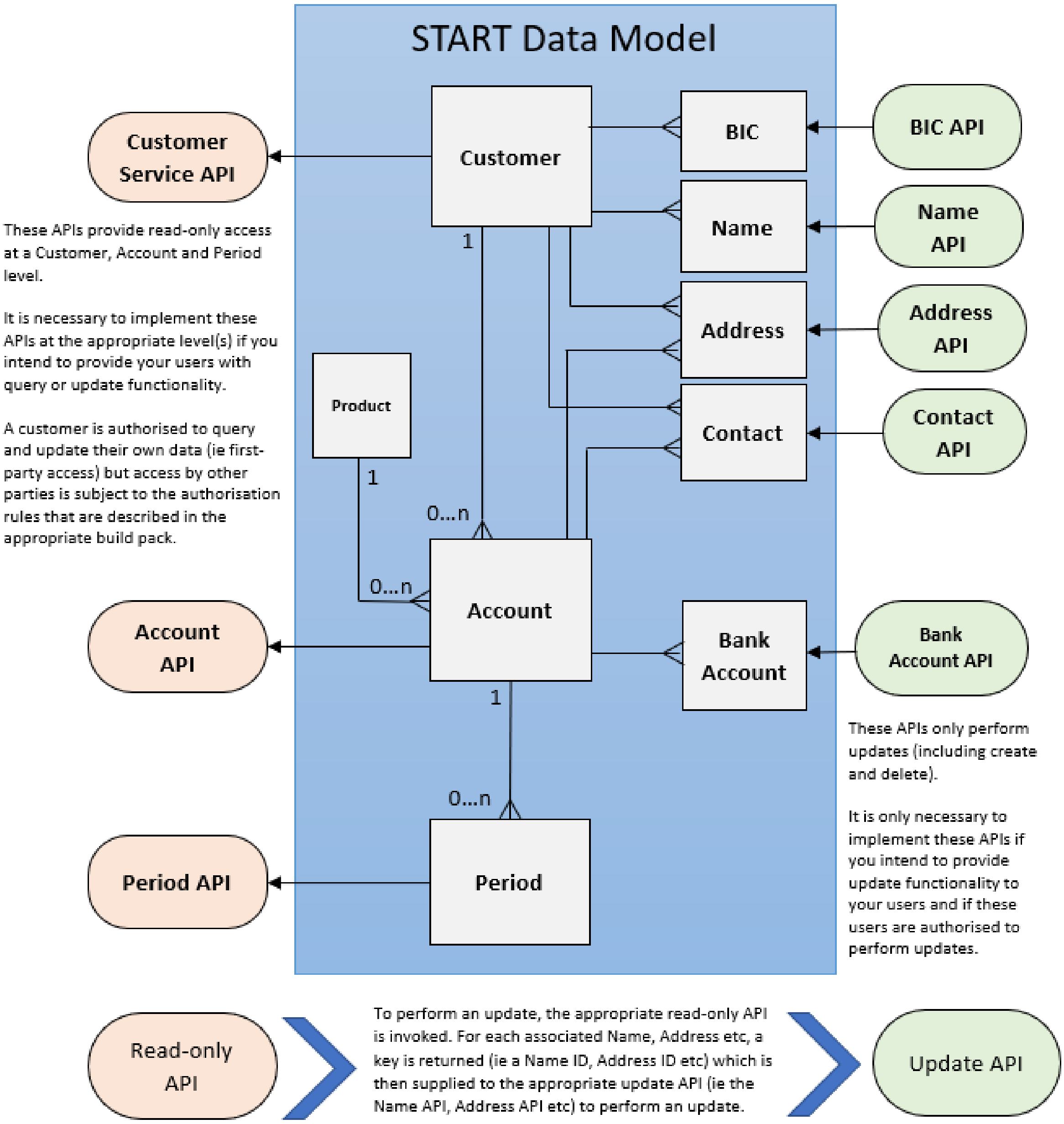
Flow of data from Inland Revenue to Digital Service Providers

1. START makes an update-only Name API available to the eServices API Gateway
2. The consumer of this API calls the Name API to update names.
3. The eServices API Gateway responds with a new, unique key or just an HTTP success.

2.1.1 Dependencies between the customer service APIs

This API is one of eight 'customer service' APIs designed to be used together—Account, Address, Bank, BIC, Contact, Customer, Name and Period. It is important to understand the dependencies between these when deciding which ones to implement, how to correctly sequence their adoption, how authorisation rules impact access, and how to use them in general.

These APIs align to START's data model as depicted below:



These APIs provide read-only access at a Customer, Account and Period level.

It is necessary to implement these APIs at the appropriate level(s) if you intend to provide your users with query or update functionality.

A customer is authorised to query and update their own data (ie first-party access) but access by other parties is subject to the authorisation rules that are described in the appropriate build pack.

These APIs only perform updates (including create and delete).

It is only necessary to implement these APIs if you intend to provide update functionality to your users and if these users are authorised to perform updates.

To perform an update, the appropriate read-only API is invoked. For each associated Name, Address etc, a key is returned (ie a Name ID, Address ID etc) which is then supplied to the appropriate update API (ie the Name API, Address API etc) to perform an update.

2.2 Messaging

This service supports the POST, PUT and DELETE HTTP methods. This service has three operations that use these methods:

- **CREATE:** POST—This operation is used to create a new name.
- **UPDATE:** PUT—This operation is used to update an existing name.
- **DELETE:** DELETE—This operation is used to cease an existing name.

2.2.1 Create

This operation is invoked by submitting a POST request to create a new name. These names can only be non-legal names, such as a trade name or preferred name. Trade names can be submitted for non-individuals or individuals, while preferred names can only be submitted for individuals.

Note that the requirement status of each field is provided in the accompanying YAML file.

2.2.1.1 Request payload

Field	Description
CustomerID	ID of the customer whose name should be added
CustomerIDType	Type of ID provided
Type	Type of name (preferred, trade). See name types below.
Unformatted	This is a 'Doing Business As' (also known as trade name) for individuals or non-individuals.
LastName	This is an individual's family name.
MiddleName	This is an individual's middle name.
FirstName	This is an individual's given first name.
Title	This is an individual's title.
Suffix	This is an individual's name suffix.

A list of the valid values for **CustomerIDtype** is as follows:

Type	Description
IRD	The IRD number for the customer.
CST	The customer identifier for the customer.

A list of valid values for **Type** is as follows:

Type	Description
PREFER	This is for a 'Preferred' name which is a name by which an individual prefers to be called.
DBACST	This is for a 'Doing Business As' name (also known as trade name) which is an alternate name under which the customer conducts business. This customer can be an individual or non-individual.

2.2.2 Update

This operation is invoked by submitting a PUT request to update an existing, non-legal name. All successful updates will return an HTTP 200 status with no response payload.

2.2.2.1 Request payload

Field	Description
NameID	ID of the customer whose name is to be updated.
Unformatted	This is a trade name for individuals or non-individuals.
LastName	This is an individual's family name.
MiddleName	This is an individual's middle name.
FirstName	This is an individual's given first name.
Title	This is an individual's title.
Suffix	This is an individual's name suffix.

2.2.3 Delete

This operation is invoked by submitting a DELETE request to remove an existing, non-legal name. All successful updates will return an HTTP 200 status with no response payload. Once a name is removed, the NameID will be ceased and cannot be reactivated. If this is done by mistake, a new name can be added which will generate a new NameID.

2.2.3.1 Request payload

Field	Description
NameID	This is the unique identifier created for the new name.

2.3 Security

2.3.1 Information classification

The information exchanged via this API has an information classification of “**IN CONFIDENCE**”. The following security standards therefore apply.

2.3.2 Transport layer security and certificates

Mutual Transport Layer Security (TLS) is implemented for this service. This requires the use of a publicly-issued X.509 certificate from one of the trusted certificate authorities listed further below in this section. (Note that Inland Revenue does not issue certificates to external vendors for web service security implementations.)

Inland Revenue has the following requirements for accepting public X.509 keys:

- ECDSA (preferred) key length: 384 bits (or RSA key length: 2048 bits)
- Self-signed certificates are not accepted
- Certificates issued by private/internal certificate authorities are not accepted
- The same certificate cannot be used for the Test and Production environments.

Inland Revenue has adopted a trust-based authentication model and will only accept certificates that contain a pre-approved subject common name and have been issued by one of the following root certificate authorities, trusted and approved by Inland Revenue:

- Amazon
- Comodo
- DigiCert
- Entrust
- GeoTrust
- Let’s Encrypt
- Sectigo
- Thawte.

Inland Revenue expects Digital Service Providers to use their Inland Revenue Developer Portal account to create their common name for both Test and Production certificates.

Please refer to the Digital Service Providers pages on the Inland Revenue website or contact your Inland Revenue onboarding representative at GatewayServices@ird.govt.nz for further details.

2.3.3 Ciphers

While Inland Revenue currently supports TSL1.2 and TLS1.3 which specifies a much smaller and more prescriptive suite of ciphers. As Inland Revenue's security gateways do not currently support the CCM mode (*counter with cipher block chaining message authentication code*) of operation, only the following ciphers will be supported over TLS1.3:

Status	TLS1.3 ciphers
Supported now and in the future	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256

The following TLS1.2 ciphers are currently supported but some will be deprecated as below:

Status	TLS1.2 ciphers
Supported now and in future	• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
Supported now but will be deprecated on 31 March 2022	• TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
Supported now but will be deprecated on 31 December 2022	• TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384

2.3.4 Authentication options

This design will use JSON Web Tokens (JWT) or OAuth2.0 tokens and protocol to establish the calling party's identity. The OAuth2.0 method requires a myIR user to logon, while JWT is a machine-to-machine credential.

This API will require a unique identifier in order to establish the calling party's identity and to allow the access model to authenticate.

Refer to the Identity and Access Services build pack for more information.

2.3.4.1 OAuth

When using OAuth, the interaction with Inland Revenue is transacted under the identity of a myIR user. OAuth requires the presence of a myIR user, as this person must be available to supply their user ID, password and consent at run-time in order to be authenticated. OAuth is especially suited to cloud-based applications where the transacting parties are application users rather than providers.

HTTP headers intended for OAuth access services will be have the JWT prefixed with "Bearer ":

HTTP header	Example value
Authorization	Bearer {JWTAccessToken}

2.3.4.2 JWT

The alternative to OAuth is JWT, which does not require the presence of a myIR user. Authentication is based on the verification of a digital signature that (provably) belongs to a customer. In order to digitally sign their messages, the customer must acquire a digital certificate from a trusted certificate authority, or generate a self-signed certificate, and supply it to Inland Revenue during the on-boarding process. JWT is therefore appropriate when the following conditions apply:

- The interaction with Inland Revenue is conducted under the identity of an organisation, as opposed to a person AND
- The organisation has the technical and operational capability to securely obtain and manage digital certificates AND
- The organisation's interactions with Inland Revenue can occur in the absence of specific people due to staffing issues such as out-of-hours non-availability, staff turnover and absence from work.

These factors tend to limit the use JWT to larger corporations and public sector organisations. It is not suitable for cloud-based applications as it requires all application users to have their own digital certificates—this is administratively burdensome and requires these users to lodge their private keys with their application provider, which is insecure.

Gateway Services will use this token in the HTTP header of a message in the same manner that an OAuth token has been used, namely:

HTTP header	Example value
Authorization	{JWTAccessToken}

2.3.4.2.1. startLogon

A myIR logon can be provided in order to use the myIR delegation model for identifying which customer names should be retrieved. If the myIR logon is provided, then names will only be shown for the customer the logon can access. If a myIR logon is not used, the field should be included with a value of null, and the subject will determine the names shown.

2.3.4.2.2. sub

A subject must be provided, which is the thumbprint of the signing certificate, and can be used to determine which names should be retrieved. The subject will always be used to validate the signature of the JWT but will only be used for determining which names to retrieve when a value for **startLogon** is not provided. The subject can be used for access when the subject is a tax preparer—names will be returned for customers currently linked to the tax preparer.

3 Additional development resources

Current environment information for this service—including the end points for each environment—is available within the relevant Software Development Kit (SDK).

To access the SDK, do one of the following:

- Go to https://github.com/InlandRevenue and select this service
- Go to https://developerportal.ird.govt.nz and click the link to the SDK within the Gateway Service documentation (please register first).

3.1 OpenAPI specifications

An OpenAPI file allows for the description of the entire API, end points, operations on each end point, and operation parameters. The included .yaml file can be used along with an OpenAPI editor such as editor.swagger.io to view technical specifications for this operation and generate example client code.

To access the latest OpenAPI definition for this service, please do the following:

- Login to the developer portal at https://developerportal.ird.govt.nz (register first)
- Download and view the OpenAPI definition within the Gateway Service documentation.

4 Change log

This table lists all material changes that have been made to this build pack document since the release of V1 (most recent changes listed first). It does not encompass non-material changes, such as to formatting etc.

Date of change	Document section	Description
17/09/21		October 2021 release changes
		New YAML file issued
	2.1.1	List and diagram of customer service APIs expanded to include new BIC API
	4	Glossary removed
	1.3	'Prerequisites' section removed and absorbed into new 'Security' section (2.3)
	1.3	'Related services' section added
	1	'Mutual Transport Layer security and certificates' section updated and moved into section 2.3.2
		'Authentication options' section modified and moved into section 2.3.4
	2.1	Diagram updated to include JWT
	2.1.1	'Dependencies between the customer services APIs' section moved here
	2.2	Heading changed from 'Supported HTTP methods' to 'Messaging'
	2.3	Security section upgraded: • 'Information classification' section added • 'Transport layer security and certificates' updated • 'Ciphers' section added • 'Authentication options' section modified
	3	Heading changed from 'End points and OpenAPI specifications' to 'Additional development resources'
30/09/20		V1 released