Inland Revenue
Te Tari Taake

Inland Revenue

Build Pack: BIC API

Date: 17/09/2021

Inland Revenue
Te Tari Taake

Contents

Inland Revenue
Te Tari Taake

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.

This service is an application programming interface (API) that external applications can call in real-time to retrieve or modify a customer's Business Industry Classification (BIC) codes. Business Industry Classification codes are used by Inland Revenue to identify a customer as participating in a particular business activity, such as dairy farming or clothes retailing. Inland Revenue passes the customer's BIC to the Accident Compensation Corporation (ACC), who use it to derive the customer's ACC levy.

The objective of this API is to allow software providers to add new BIC codes as well as change or cease existing ones for a particular IRD number in Inland Revenue's database.

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

The solution outlined in this document is intended to be used by software providers. The reader is assumed to have a suitable level of technical knowledge to understand 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)

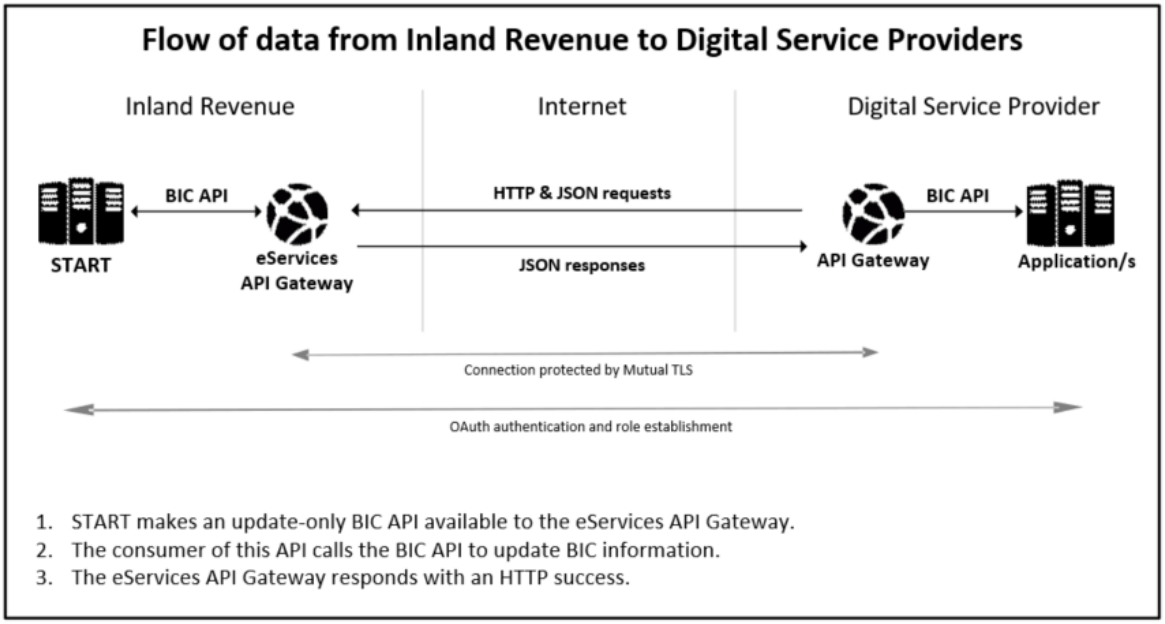
The Identity and Access Services (IAS) are used to authenticate access. Authentication tokens will need to be retrieved via IAS prior to making calls to this API.

2 Solution design

2.1 Architecture

Inland Revenue offers a suite of web applications to facilitate interactions via software packages. This API using JSON messaging, will be used by approved organisations to add, update, and cease BIC codes of Inland Revenue customer accounts identified by either their IRD number or customer identifiers.

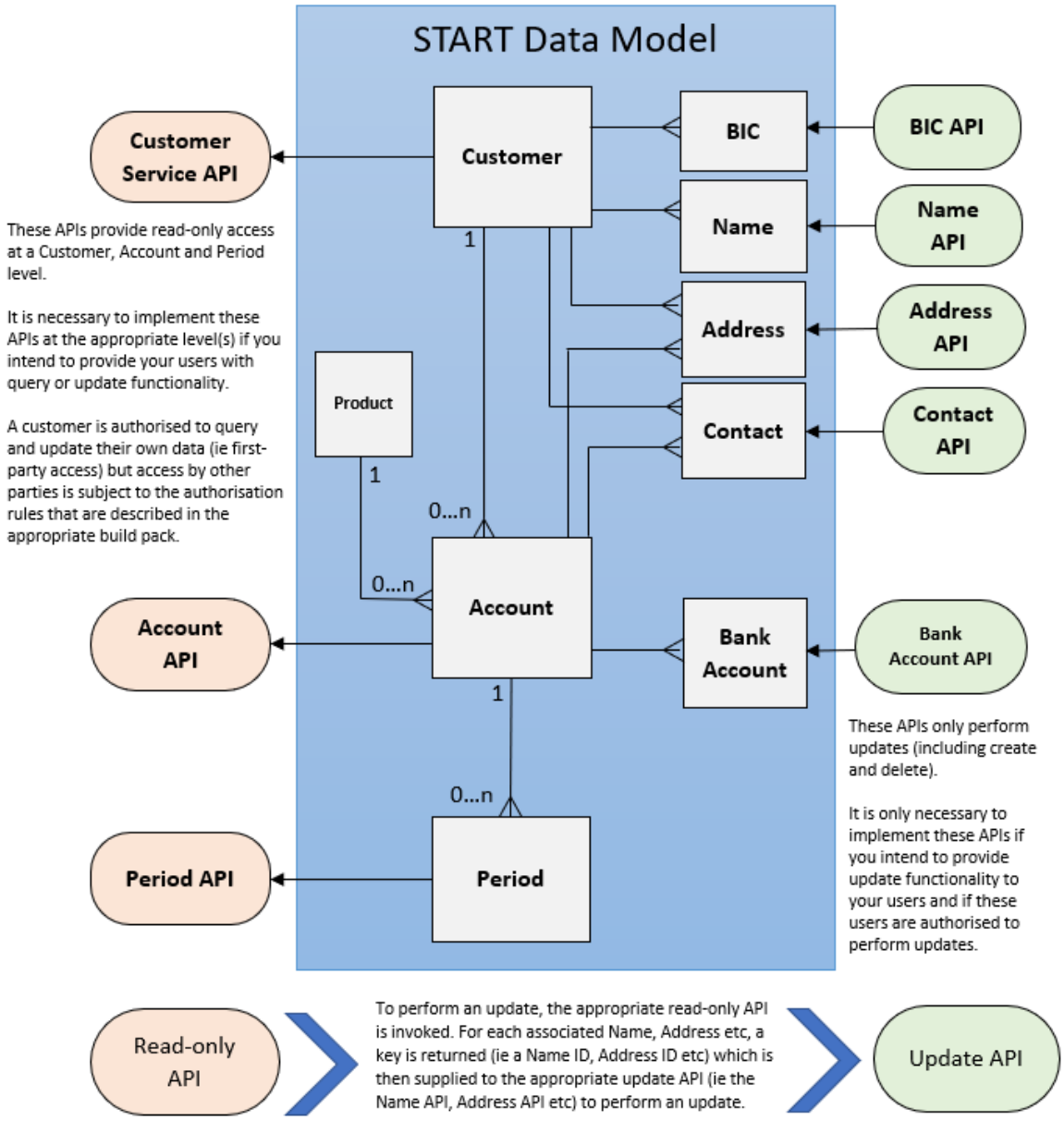
The diagram below illustrates the flow of data from Inland Revenue to the Digital Service Providers.



2.2 Dependencies between the customer services APIs

This API is one of eight 'customer services' 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:



2.3 Messaging

This service supports the following message types:

- **Add / update:** Associates a new BIC code to a specific IRD number. If the IRD number provided belongs to an account with an existing BIC code, that code will be ceased and the new BIC code added. This message type uses the resource "/bic" and the "POST" HTTP method.

- **Cease:** Ceases an existing BIC code indicating that the business activity associated with this BIC code ended. This message type uses the resource "/bic" and the "DELETE" HTTP method.

2.3.1 Add / update

2.3.1.1 Request payload

Field	Description
CustomerID	IRD number of customer identifier of the account being updated.
CustomerIDType	Accepts "IRD" (IRD number) and "CST" (customer identifier).
BICCode	A well-formed BIC Code (for examples and detailed description, see https://www.businessdescription.co.nz/)
Commence	Commence date—indicates when the activity associated with the BIC code provided began. Value can be back-dated, but not future-dated. If not provided, this value will be defaulted to today's date. If there is already a BIC code on the account, the existing code will be ceased with commence date less one day. Values are expected to be provided in the format "YYYY-MM-DD".

A consequence of the above commencement date rules is that a customer can only have one active BIC at a time. Under exceptional circumstances ACC does recognise that a customer might have more than one concurrently active BIC and therefore require a bespoke ACC levy. In this case ACC advises the customer to contact them directly.

The other customer update APIs (Address, Name etc) require an identifier that is obtained by calling the read-only Customer or Account APIs. For the BIC API this identifier is the BIC (the above BICCode parameter), which means that it is possible to call this API without having to make a preceding call to the Customer or Account APIs.

Inland Revenue
Te Tari Taake

2.3.2 Cease

2.3.2.1 Request payload

Field	Description
CustomerID	IRD number of customer identifier of the account being updated.
CustomerIDType	Accepts "IRD" (IRD number) and "CST" (customer identifier).
BICCode	A well-formed BIC Code (for examples and detailed description, see https://www.businessdescription.co.nz/)
Cease	Cease date—indicates when the activity associated with the BIC code provided ended. Value can be back-dated, but not future-dated. If not provided, this value will be defaulted to today's date. Values are expected to be provided in the format "YYYY-MM-DD".

2.3.2.2 Response payloads

For both **add/update** as well as **cease** message types, if the request is successful, no payload will be provided in response. The consumer should expect to receive HTTP response code 200 (success).

If an error occurs, one of the following JSON array "errors" will be returned:

errors[]	Description
errors[] ➢ **code**	Error code returned by the API (see Responses section for descriptions).
errors[] ➢ **type**	Type of error received. Possible values are "result", "security", "server", and "validation".
errors[] ➢ **message**	Description of the error code received.

2.4 Security

2.4.1 Information classification

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

2.4.2 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.

Inland Revenue
Te Tari Taake

2.4.3 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

2.4.4 Authentication options

This design uses JSON Web Tokens (JWT) or OAuth2.0 tokens and protocol to establish the calling party's identity. For this API, both JWT and OAuth2.0 methods require a myIR user to logon.

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

2.4.4.1 OAuth

When using OAuth2.0 the API consumer is authenticated using their myIR credential and authorised using their myIR access rights. Having been authenticated, the API consumer must either be:

- The customer who is updating their own BIC, or
- Have access in myIR to customer's BIC, or
- Be an intermediary who is linked to the customer whose BIC is being updated, where the customer is identified by the CustomerIDType and CustomerID request parameters.

If the API consumer does not have access to the customer identified by the IRD number in the request parameters, an error will be returned (see Responses section).

HTTP headers intended for OAuth access services will have the token prefixed with "Bearer ".

HTTP header	Example value
Authorization	Bearer {JWTAccessToken}

2.4.4.2 JWT

The alternative to OAuth for authenticating the API consumer is to use signed JWT. The authenticated consumer of the API is the signatory whose public key was successfully used to verify the digital signature of the JWT. The public key is bound to the API consumer's IRD number.

Signed JWT is applicable in scenarios where the software platform operates autonomously in the absence of online users, a typical use case being some large payroll systems and banking platforms.

If the JWT contain a STARTLogon then the access rights that are associated with that logon are applied (ie it is used for authorisation, not authentication). Use of the STARTLogon is discouraged because it requires the lodging of myIR user IDs in the software platform, which is poor practice. Furthermore, if the software platform is able to provide a logon then it should use OAuth.

Gateway Services will use the JWT 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}

Inland Revenue
Te Tari Taake

3 Additional development resources

3.1 End points

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).

3.2 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.

Inland Revenue
Te Tari Taake

4 Responses

The response from the BIC API will always include an HTTP status code and if an error code is returned, a message with the error status code and its description. The following table describes the possible error codes that the BIC API will respond with.

Code	Error	Error description
BIC000	Unknown error occurred	Processing of request failed despite request being valid, try again later
BIC101	No BIC found	Unable to find activity code provided on the customer
BIC102	Commence date is in the future	Activity commence date may not be in the future
BIC103	Cease date is in the future	Activity cease date may not be in the future
CST404	Customer record could not be located	A record could not be located for the given identifier
EU6001	Unexpected error	Unexpected error occurred
EV1020	Authentication failure	Authentication failure means the token (JWT or OAuth) provided is not valid
EV1021	Missing authentication token	No OAuth or JWT token is present as an HTTP header
EV1022	Unauthorised access	Access is not permitted for the request to perform this operation for the submitted identifier
EV1100	Invalid parameters	Invalid input parameters. Please check documentation
EV2234	IRD number failed check digit	IRD number failed check digit

Inland Revenue
Te Tari Taake

5 Change log

This table lists all material changes that have been made to this build pack document since the release of v1.0. It does not encompass non-material changes, such as to formatting etc.

Date of change	Document section	Description
29/11/2021	4	Added new error code EV1022 (Unauthorised access)
17/09/21		First draft released