

Inland Revenue

Build Pack: Contact API

Date: 17/09/2021

Contents

1 Overview.....	3
1.1 This solution	3
1.2 Intended audience.....	3
1.3 Related services	3
1.3.1 Identity and Access Services (required)	3
2 Solution design	4
2.1 Architecture.....	4
2.1.1 Dependencies between the customer service APIs	5
2.2 Messaging	6
2.2.1 Contact operations	6
2.2.1.1 Create	6
2.2.1.1.1 Request payload	6
2.2.1.1.2 Response payload	7
2.2.1.2 Update	7
2.2.1.2.1 Request payload	7
2.2.1.3 Delete	7
2.2.1.3.1 Request payload	7
2.2.2 Phone operations	8
2.2.2.1 Create	8
2.2.2.1.1 Request payload	8
2.2.2.1.2 Response payload	8
2.2.2.2 Update	8
2.2.2.2.1 Request payload	8
2.2.2.3 Delete	9
2.2.2.3.1 Request payload	9
2.3 Security	10
2.3.1 Information classification	10
2.3.2 Transport layer security and certificates	10
2.3.3 Ciphers	11
2.3.4 Authentication options	12
2.3.4.1 OAuth.....	12
2.3.4.2 JWT.....	12
2.3.4.2.1 startLogon.....	13
2.3.4.2.2 sub.....	13
3 Additional development resources	14
3.1 End points.....	14
3.2 OpenAPI specifications	14
5 Change log	15

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 current information about the contact details 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)

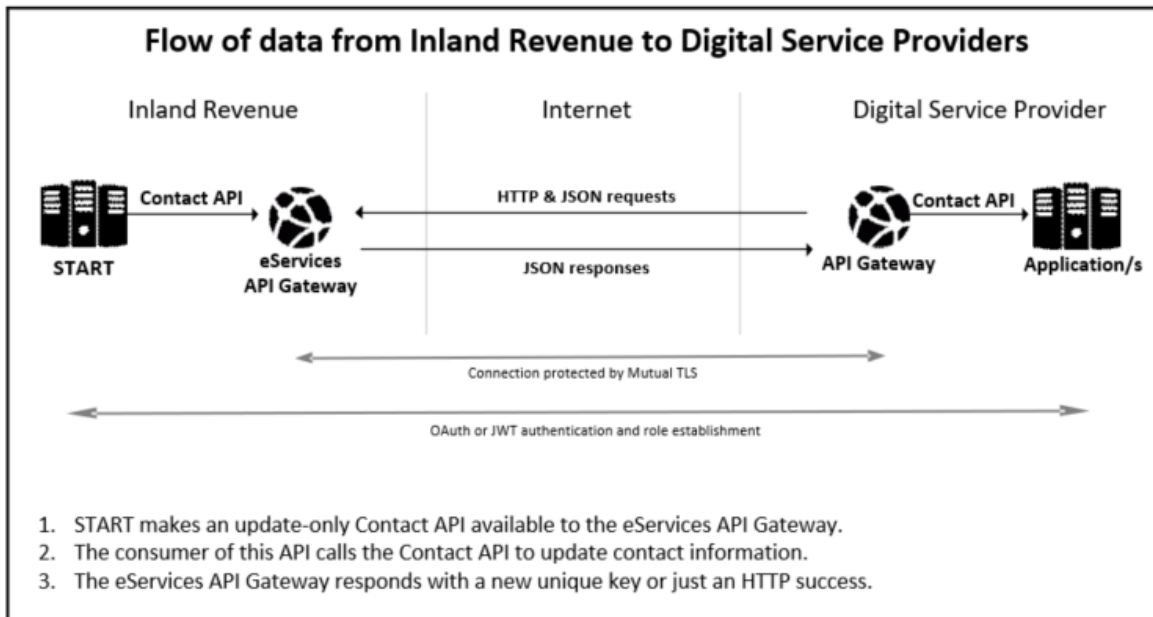
The Identity and Access Services (IAS) are used to authenticate access. Authentication tokens will need to be retrieved via IAS prior to making calls to this API.

2 Solution design

2.1 Architecture

Inland Revenue offers a suite of web applications in order to facilitate interactions via software packages. This API will be used by approved organisations to retrieve contacts from Inland Revenue.

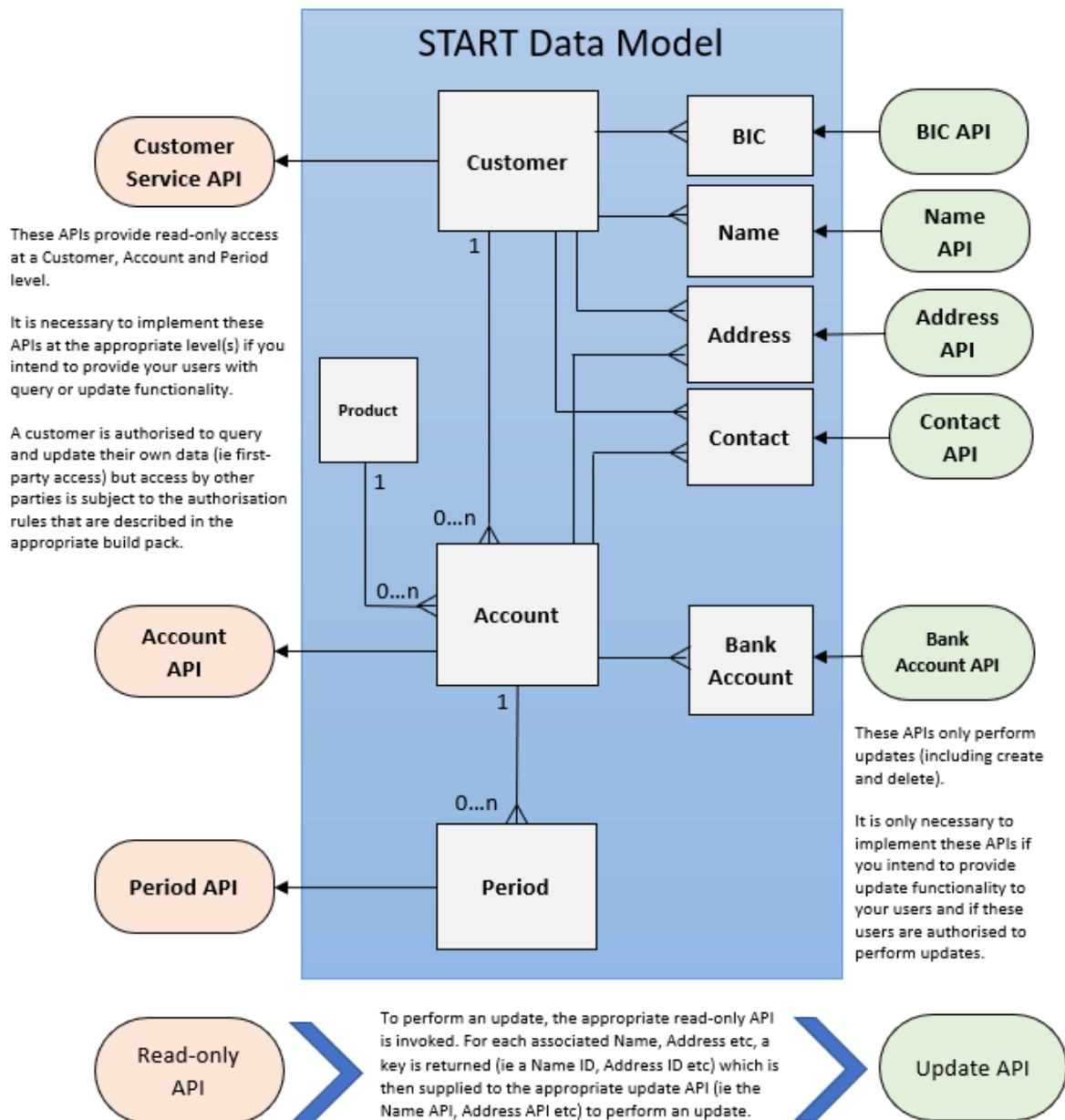
The diagram below illustrates the flow of data from Inland Revenue to the Digital Service Providers.



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:



2.2 Messaging

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

Operation	Methods
Contact	<ul style="list-style-type: none"> • CREATE: POST—Creates a new contact for a given customer • UPDATE: PUT—Updates an existing contact for a given customer • DELETE: DELETE—Deletes an existing contact
Phone	<ul style="list-style-type: none"> • CREATE: POST—Creates a new phone number on an existing contact • UPDATE: PUT—Updates an existing phone number for a given customer • DELETE: DELETE—Deletes an existing phone number

2.2.1 Contact operations

2.2.1.1 Create

This operation is invoked by submitting a POST request to create a new contact. Note that at least one phone number must be provided, but no more than five can be provided for a single contact.

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

If an AccountID and AccountIDType are provided, the contact will only apply to the specified account. If a CustomerID and CustomerIDType are provided, the contact will apply to the entire customer.

2.2.1.1.1. Request payload

Field	Description
CustomerID	ID of the customer for which the contact should be added
CustomerIDType	Type of ID provided
AccountID	ID of the account for which the contact should be added
AccountIDType	Type of ID provided
ContactType	Contact type (ie primary, secondary)
Name	Name of contact
Phone.PhoneType	Mobile, home and/or business phone
Phone.Country	Country for phone—used to determine country code
Phone.AreaCode	Area code portion of phone number
Phone.PhoneNumber	Phone number, without country code
Phone.Extension	Extension number

2.2.1.1.2. *Response payload*

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

Field	Description
ContactID	Unique ID for the contact just created.

2.2.1.2 *Update*

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

2.2.1.2.1. *Request payload*

Field	Description
ContactID	Unique ID for contact
Name	Name of contact

2.2.1.3 *Delete*

This operation is invoked by submitting a DELETE request to remove an existing contact. All successful deletes will return an HTTP 200 status with no response payload.

Once a **ContactID** is removed, the ContactID and all associated PhoneIDs will be ceased and cannot be reactivated. If this is done by mistake, the contact and phone numbers will need to be added again.

2.2.1.3.1. *Request payload*

Field	Description
ContactID	Unique ID for contact

2.2.2 Phone operations

2.2.2.1 Create

This operation is invoked by submitting a POST request to create a new phone number on an existing contact. Note that no more than five phone numbers can be provided for a single contact, so an additional contact must be created if adding a sixth phone number.

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

2.2.2.1.1. Request payload

Field	Description
ContactID	Unique ID for contact
Phone.PhoneType	Mobile, home and/or business phone
Phone.Country	Country for phone—used to determine country code
Phone.AreaCode	Area code portion of phone number
Phone.PhoneNumber	Phone number, without country code
Phone.Extension	Extension number

2.2.2.1.2. Response payload

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

Field	Description
PhoneID	Unique ID for the phone just created.

2.2.2.2 Update

This operation is invoked by submitting a PUT request to update an existing phone number. All successful updates will return an HTTP 200 status with no response payload.

2.2.2.2.1. Request payload

Field	Description
PhoneID	Unique ID for phone number
Phone.PhoneType	Mobile, home and/or business phone
Phone.Country	Country for phone—used to determine country code
Phone.AreaCode	Area code portion of the phone number
Phone.PhoneNumber	Phone number, without country code
Phone.Extension	Extension number

2.2.2.3 *Delete*

This operation is invoked by submitting a DELETE request to remove an existing phone number. All successful deletes will return an HTTP 200 status with no response payload.

Once a **phone number** is removed, the PhoneID will be ceased and cannot be reactivated. If this is done by mistake, the ContactID should be updated and the phone number added again.

2.2.2.3.1. *Request payload*

Field	Description
PhoneID	Unique ID for phone number

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

Inland Revenue currently supports TLS1.2 and TLS1.3, with the latter specifying 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	<ul style="list-style-type: none"> • 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 the future	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 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 contact details can be modified. If the myIR logon is provided, then details will only be shown for customers or accounts 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 contacts that can be modified.

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 contacts can be modified. The subject will always be used to validate the signature of the JWT but will only be used for determining which contacts can be modified when a value for **startLogon** is not provided. The subject can be used for access when the subject is a tax preparer—contacts can be modified for customers or accounts currently linked to the tax preparer.

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.

5 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
	1.2	Additional information added to 'Intended audience' section
	1.3	'Prerequisites' section removed and absorbed into new 'Security' section (2.3)
	1.3	'Related services' section added to build pack
	1.3.1	'Mutual Transport Layer security and certificates' section updated and moved into section 2.3.2
	1.3.2	'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	'Bill item and provisional tax information' heading removed
	2.3	Security section upgraded: <ul style="list-style-type: none"> 'Information classification' section added 'Transport layer security and certificates' updated 'Ciphers' section added 'Authentication options' section modified
	3	'End points and OpenAPI specifications' section renamed 'Additional development resources'
	4	Glossary removed
04/11/20		Updated YAML to correct PhoneNumber to be a required field in Update Phone operation
30/09/20		V1 released