

Inland Revenue

## Build Pack: Customer API

**Date:** 30/09/2020  
**Version:** v1

## Contents

<b>1 Overview.....</b>	<b>3</b>
1.1 This solution .....	3
1.2 Intended audience.....	3
1.3 Prerequisites.....	3
1.3.1 Mutual Transport Layer Security and certificates .....	3
1.3.2 Authentication options .....	4
1.3.2.1 OAuth.....	4
1.3.2.2 JWT.....	4
<b>2 Solution design .....</b>	<b>5</b>
2.1 Architecture.....	5
2.2 Supported HTTP methods .....	5
2.3 Dependencies between the customer service APIs .....	5
2.4 Customer API .....	7
2.4.1 Request payload fields .....	7
2.4.2 Response payload fields .....	7
2.4.2.1 Address notes.....	8
2.5 Security .....	9
2.5.1 OAuth.....	10
2.5.2 M2M JWT .....	11
2.5.2.1 Header .....	11
2.5.2.2 Payload.....	12
2.5.2.3 startLogon .....	12
2.5.2.4 sub .....	12
<b>3 End points and OpenAPI specifications .....</b>	<b>13</b>
3.1 End points.....	13
3.2 OpenAPI specifications .....	13
<b>4 Glossary .....</b>	<b>14</b>
<b>5 Change log .....</b>	<b>16</b>

## 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 customer information as held by Inland Revenue.

---

Before continuing, please consult [www.ird.govt.nz/digital-service-providers/services-catalogue](http://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 customer 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.

### 1.3 Prerequisites

Party	Requirement	Description
<b>Digital Service Provider</b>	Acquire a X.509 certificate from a competent authority for the Test and Production environments.	This is required when using mutual TLS with cloud-based service providers or financial institutions.  Note that the same certificate cannot be used for the Test and Production environments.

#### 1.3.1 Mutual Transport Layer Security and certificates

Mutual Transport Layer Security (TLS) is implemented for this API. This requires the use of a publicly-issued X509 certificate from one of the trusted certificate authorities. Inland Revenue does not issue certificates to external vendors for web service security implementations.

Inland Revenue has the following minimum requirements for accepting public X509 keys:

- Minimum Key Length: 2048
- Signature Algorithm: SHA256[RSA]
- Self-signed certificates are not accepted
- Certificates issued by a private/internal certificate authority are not accepted.

In general, shorter-lived certificates offer a better security posture since the impact of key compromise is less severe but there is no minimum requirement for certificate expiry periods.

Below is a list for examples of certificate authority providers with no recommendations or rankings incorporated. It is recommended that a business researches which certificate authority meets their requirements:

- [Comodo](#)
- [GeoTrust](#)
- [DigiCert](#)
- [GlobalSign](#)
- [Symantec](#)
- [Thawte](#)
- [IdenTrust](#)
- [Entrust](#)
- [Network Solutions](#)
- [RapidSSL](#)
- [Entrust Datacard](#)
- [GoDaddy](#).

### 1.3.2 Authentication options

#### 1.3.2.1 OAuth

When using OAuth the interaction with IR 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.

#### 1.3.2.2 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.

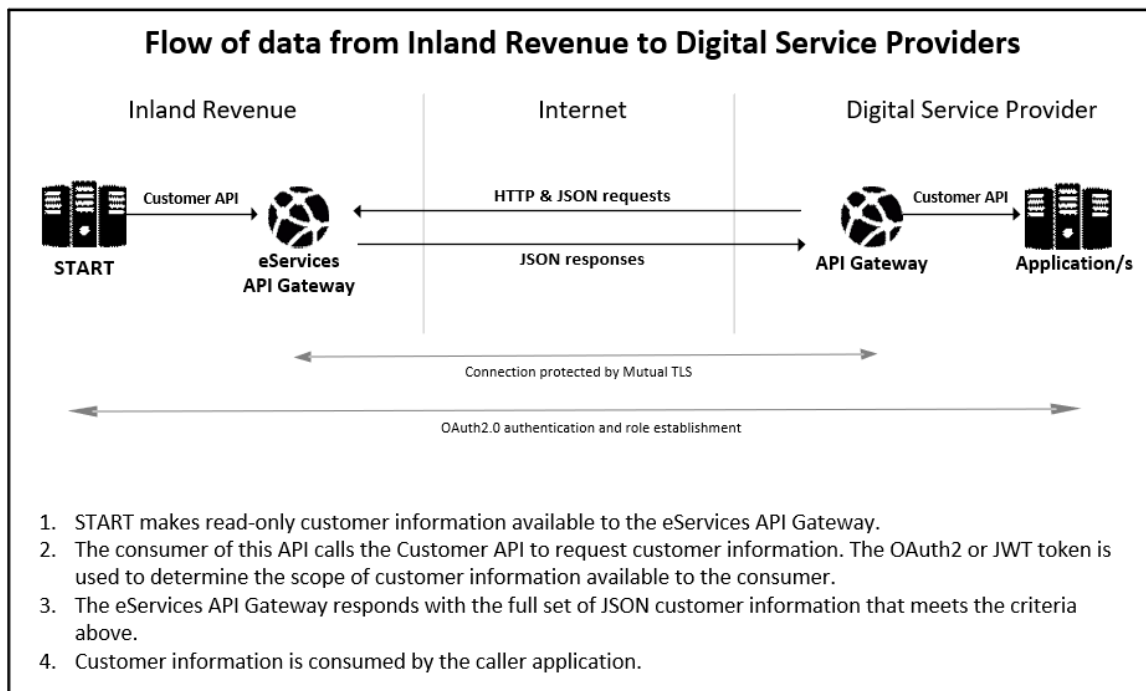
These factors tend to limit the use JWT to larger corporations and public sector organisations. It is not suitable for cloud-based applications as it requires all application users to have their own digital certificates—this is administratively burdensome and requires these users to lodge their private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

## 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 retrieve customer information from Inland Revenue.

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



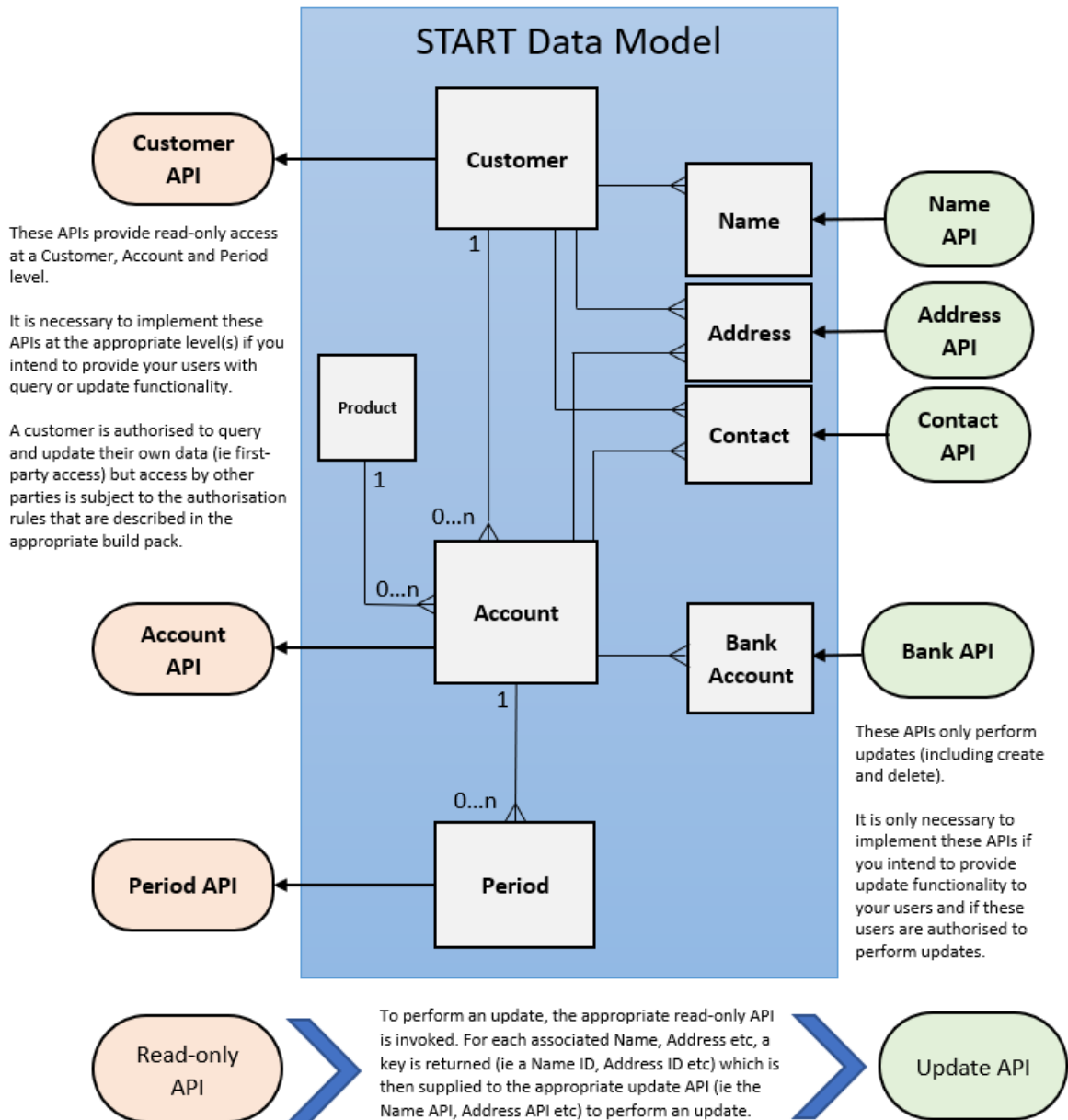
### 2.2 Supported HTTP methods

This service only supports the GET HTTP method and will not allow any updates via methods like POST or PUT.

### 2.3 Dependencies between the customer service APIs

This API is one of seven 'customer service' APIs designed to be used together—Account, Address, Bank, 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 on the following page.



## 2.4 Customer API

### 2.4.1 Request payload fields

Field	Requirement	Description
<b>CustomerID</b>	Mandatory	Unique ID to identify customer
<b>CustomerIDType</b>	Mandatory	Type of ID submitted in Customer ID field

### 2.4.2 Response payload fields

Field	Description
<b>ID</b>	The ID of the customer
<b>IDType</b>	The type of ID submitted in the ID field
<b>EntityType</b>	Customer subtype for non-individuals, INDVDL for Individuals
<b>EntitySubType</b>	Operating structure for non-individual and Customer segment for Individual
<b>Commence</b>	Commencement date of customer
<b>Cease</b>	Cessation date of customer
<b>NZBN</b>	New Zealand Business Number of the requested customer
<b>BIC.BICCode</b>	The Business Identification Code of a non-individual
<b>Indicator.Indicator</b>	A field that indicates additional specific information about this customer
<b>Address.AddressID</b>	Unique ID for address
<b>Address.Type</b>	Type of address (mailing or physical)
<b>Address.Formatted</b>	Formatted, single-line address
<b>Address.Street</b>	Street address line 1
<b>Address.Street2</b>	Street address line 2
<b>Address.Unit</b>	Unit identifier
<b>Address.UnitType</b>	Unit type
<b>Address.City</b>	City name
<b>Address.County</b>	County name
<b>Address.State</b>	State name
<b>Address.PostCode</b>	Postal code
<b>Address.Country</b>	ISO 2 digit standard (New Zealand is NZ)
<b>Address.Attention</b>	The person to whom the correspondence is addressed
<b>Address.Urbanisation</b>	Urbanisation (See address notes below)
<b>Address.District</b>	District type (See address notes below)

Field	Description
<b>Address.SubDistrict</b>	District identifier (See address notes below)
<b>Address.SubProvince</b>	Sub-province name (See address notes below)
<b>Address.Updated</b>	Date on which address was last updated
<b>Name.NameID</b>	Unique ID for name
<b>Name.Type</b>	Name type (legal, preferred, trade)
<b>Name.Formatted</b>	Formatted name
<b>Name.LastName</b>	Family name value
<b>Name.MiddleName</b>	Middle name value
<b>Name.FirstName</b>	Given name value
<b>Name.Title</b>	Title name
<b>Name.Suffix</b>	Name suffix
<b>Name.Updated</b>	Date on which name was last updated
<b>Contact.ContactID</b>	Unique ID for contact
<b>Contact.ContactType</b>	Contact type (ie primary, secondary)
<b>Contact.Name</b>	Name of contact
<b>Contact.Updated</b>	Date on which contact was last updated
<b>Contact.Phone.PhoneID</b>	Unique ID for phone
<b>Contact.Phone.PhoneType</b>	Mobile, home and/or business phone
<b>Contact.Phone.Country</b>	Country for phone—used to determine country code
<b>Contact.Phone.AreaCode</b>	Area code portion of phone number
<b>Contact.Phone.PhoneNumber</b>	Phone number, without country code
<b>Contact.Phone.Extension</b>	Extension number

Note: The BIC, Indicator, address, name, contact and phone objects can be repeated depending on what other customer information exists.

#### 2.4.2.1 *Address notes*

The following fields contain different data depending on the country of the address:

Field	Region	Data
<b>Urbanisation</b>	New Zealand	Suburb/Rural
	Australia	Suburb/Place
	Europe	Distribution
<b>District</b>	New Zealand	Floor type
	Australia	Floor type
	Finland	Entrance



Field	Region	Data
	Poland	Post office
<b>SubDistrict</b>	New Zealand	Floor number
	Australia	Floor number
<b>SubProvince</b>	New Zealand	Building
	Australia	Building
<b>Unit</b>	Caribbean	PO Box

## 2.5 Security

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

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

Each HTTPS header contains the authorisation attribute JWT/OAuth:

1. A signed JSON Web Token (JWT) token. This will establish a registered digital services provider identity via the asymmetric public key held in the key store established during onboarding.
2. An OAuth2.0 token that is a customer- or intermediary-level XIAMS user account recognised by START.

This API uses an HTTPS transport layer, with HTTP1.1 transport protocol supported.

Regarding transport layer security (TLS), note that while TLS1.3 is now an industry standard, it is not yet widely adopted, as doing so requires upgrades to perimeter security devices and software. Inland Revenue will upgrade to TLS1.3 once it is adopted widely enough, and where practical, external software partners should also anticipate upgrading to this version. TLS1.0 and TLS1.1 are not supported by myIR or Gateway Services.

Asymmetric keys of approved strength must be used. Inland Revenue requires the following ciphers and key strengths to be used:

<b>Encryption:</b>	Advanced Encryption Standard (AES)	FIPS 197	256-bit key
<b>Hashing:</b>	Elliptic Curve Digital Signature Algorithm (ECDSA) using P-256 or Secure Hash Algorithm (SHA-2) NOTE: ECDSA is preferred but RSA will be supported.	FIPS 180-3	SHA-256 (or greater)

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:

`"Authorization: {JWTAccessToken}"`

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

	End point for connections
<b>Purpose</b>	<ul style="list-style-type: none"> <li>End point to which digital service providers will connect</li> </ul>
<b>Client application type</b>	<ul style="list-style-type: none"> <li>Cloud applications or in-house servers</li> </ul>
<b>Constraints</b>	<ul style="list-style-type: none"> <li>Only for source locations with client-side TLS certificates</li> <li>On the cloud end point Inland Revenue has controls to shield service providers from issues caused by heavy usage from other providers</li> </ul>
<b>Mutual TLS</b>	<ul style="list-style-type: none"> <li>Inland Revenue explicitly trusts the certificate the service provider associates with the TLS connection as client for Mutual TLS connections and uses it to identify the web service's sending party</li> </ul>
<b>Minimum TLS version</b>	<ul style="list-style-type: none"> <li>1.2</li> </ul>
<b>URL</b>	<ul style="list-style-type: none"> <li>Contains .../gateway/..</li> </ul>
<b>Port</b>	<ul style="list-style-type: none"> <li>4046</li> </ul>
<b>Web service consumer identification</b>	<ul style="list-style-type: none"> <li>Machine-to-machine authentication using client-signed JSON web tokens (JWT)</li> <li>OAuth2 authorisation using tokens generated by XIAMS</li> </ul>
<b>Firewalling in production</b>	<ul style="list-style-type: none"> <li>No IP address restrictions</li> <li>Access limited by certificate enrolment</li> </ul>
<b>Firewalling in non-production environments</b>	<ul style="list-style-type: none"> <li>No IP address restrictions</li> <li>Access limited by certificate enrolment</li> </ul>

**Delegated permissions:** The service will allow for the retrieval of customer data for a user (as represented by the JWT or OAuth2 token) who has delegated access. If the user does not have access to the customer in the request parameters, an error will be returned.

### 2.5.1 OAuth

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

HTTP header	Example value
<b>Authorization</b>	Bearer {JWTAccessToken}

*Refer to the Identity and Access Services build pack for more information on authorisation flows.*

## 2.5.2 M2M JWT

Authorisation intended for M2M (machine-to-machine) communication will not use "Bearer " flag on the HTTP header and only contain the JWT. The JWT will contain a field "startLogon" which can resolve to a myIR logon. The M2M JWT will be identified by a value of "M2M" in the Key ID ("kid"). The M2M JWT will be signed with a self-signed certificate, for which the public key was provided during onboarding.

HTTP header	Example value
<b>Authorization</b>	{JWTAccessToken}

Example data structure used for M2M authorisation:

```

Base64Url encoded {
  "alg": <algorithm value>,
  "typ": "JWT",
  "kid": "M2M"
}
.
Base64Url encoded {
  "sub": <token subject>,
  "iss": <issuer value>,
  "startLogon": <myIR_user>,
  "iat": <epoch issued value>,
  "exp": <epoch expired value>
}
.
JWS Signature (
  base64UrlEncode(header) + "." + base64UrlEncode(payload)
)
  
```

### 2.5.2.1 Header

Field	Requirement	Description	Valid values
<b>alg</b>	Required	Signature or encryption algorithm	RS256, RS384, RS512 ES256, ES384, ES512
<b>typ</b>	Required	Type of token	JWT
<b>kid</b>	Required	Key ID	M2M

### 2.5.2.2 *Payload*

Field	Requirement	Description	Valid values
<b>sub</b>	Required	Subject (to whom the token refers)	SHA-1 Thumbprint/fingerprint of signing certificate
<b>iss</b>	Required	Issuer who created this token	eg CompanyNameA
<b>startLogon</b>	Required	The myIR logon of a representative of the token subject. The subject must be the data owner.	Valid myIR logon, or null
<b>iat</b>	Required	Issued at. The number of seconds since Unix epoch 1 Jan 1970, UTC.	Must not precede the signing certificate issue date Example: 1560144847
<b>exp</b>	Required	Expiration time. The number of seconds since Unix epoch 1 Jan 1970, UTC.	Must not exceed 8 hours from the <b>iat</b> (issued at) time value Example: 1574323940

### 2.5.2.3 *startLogon*

A myIR logon can be provided in order to use the myIR delegation model for identifying customers for whom customer information should be retrieved. If the myIR logon is provided, then information will only be shown for customers the logon can access.

### 2.5.2.4 *sub*

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

---

## 3 End points and OpenAPI specifications

### 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](https://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 Glossary

Acronym/term	Definition
<b>API</b>	Application Programming Interface—set of functions and procedures that allow applications to access the data or features of another application, operating system or other service.
<b>Authentication</b>	The process that verifies the identity of the party attempting to access Inland Revenue
<b>Authorisation</b>	The process of determining whether a party is entitled to perform the function or access a resource
<b>End points</b>	A term used to describe a web service that has been implemented
<b>FIPS</b>	Federal Information Processing Standard—a suite of IT standards from the US Federal Government
<b>Gateway</b>	Inland Revenue’s web services gateway
<b>HTTP, HTTPS</b>	Hyper Text Transmission Protocol (Secure)—the protocol by which web browsers and servers interact with each other. When implemented over TLS1.2 HTTP becomes HTTPS.
<b>IAMS</b>	Identity and Access Management—a logical component that performs authentication and authorisation. Physically it is a set of discrete hardware and software products, plug-ins and protocols. Usually implemented as separate External IAMS (XIAMS) and Internal IAMS.
<b>IAS</b>	Identity and Access Service
<b>IP</b>	Internet Protocol—the principal communication protocol in the Internet protocol suite for relaying datagrams across networks
<b>IRD</b>	Inland Revenue Department (ie IRD number)
<b>OAuth</b>	An HTTPS based protocol for authorising access to a resource, currently at version 2
<b>OpenAPI specifications</b>	Formerly known as Swagger specifications—a specification for machine-readable interface files for describing, producing, consuming and visualising RESTful web services.
<b>Payloads</b>	This refers to the data contained within the messages that are exchanged when a web service is invoked. Messages consist of a header and a payload.
<b>Schemas</b>	An XML schema defines the syntax of an XML document, in particular of a payload. The schema specifies what a valid payload must or can contain, as well as validating the payload.
<b>SHA</b>	Secure Hashing Algorithm. There is a family of them that provide different strengths. SHA-2 is currently favoured over SHA-1, which has been compromised.
<b>SOAP</b>	Simple Object Access Protocol—a set of standards for specifying web services. GWS uses SOAP version 1.2

Acronym/term	Definition
<b>SSL</b>	Secure Sockets Layer certificates—used to establish an encrypted connection between a browser or user’s computer and a service or website
<b>START</b>	Simplified Taxation and Revenue Technology—IR’s new core tax processing application. It is an implementation of the GenTax product from FAST Enterprises.
<b>TLS1.2</b>	Transport Layer Security version 1.2—the protocol that is observed between adjacent servers for encrypting the data that they exchange. Prior versions of TLS and all versions of SSL have been compromised and are superseded by TLS1.2.
<b>URL</b>	Universal Resource Locator—also known as a web address
<b>X.509 certificate</b>	An international standard for encoding and describing a digital certificate. In isolation a public key is just a very large number, the X.509 certificate to which it is bound identifies whose key it is, who issued it, when it expires etc. When a counterparty’s X.509 digital certificate is received, the recipient takes their public key out of it and store the key in their own keystore. The recipient can then use this key to encrypt and sign the messages that they exchange with this counterparty.
<b>XIAMS</b>	External IAMS—an instance of IAMS that authenticates and authorises access by external parties, for example customers, trading partners etc, as opposed to internal parties such as staff
<b>YAML</b>	"YAML Ain't Markup Language"—a human-readable data-serialisation language commonly used for configuration files and in applications where data is stored or transmitted.

---

## 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
30/09/20		V1 released