

Inland Revenue

## Build Pack: Transaction Data Services (TDS) APIs

Version 0.5  
Date:

13/01/2023

---

## Contents

<b>1 Overview.....</b>	<b>3</b>
1.1 This solution .....	3
1.2 Intended audience.....	3
1.3 Related services .....	3
1.3.1 Identity and Access Services (required) .....	3
<b>2 Solution design .....</b>	<b>4</b>
2.1 Architecture.....	4
2.2 Messaging .....	4
2.3 Flow of data to Digital Service Providers from Inland Revenue .....	5
2.3.1 TDS Real Time Model .....	5
2.3.2 TDS Reconciliation Model .....	5
2.4 Associated Services .....	5
2.5 TDS Financials .....	6
2.5.1 Request payload .....	6
2.5.2 Response payload .....	7
2.6 TDS Summary .....	9
2.6.1 GET Request.....	9
2.6.2 POST Request payload .....	9
2.6.3 Response payload .....	11
2.7 TDS Transactions (Real Time and Reconciliation).....	12
2.7.1 Request payload .....	13
2.7.2 Response payload .....	13
2.8 Security .....	17
2.8.1 Information classification .....	17
2.8.2 Transport layer security and certificates .....	17
2.9 Authentication options .....	18
2.9.1 OAuth (applicable to Real time model) .....	18
2.9.2 Authentication Access .....	18
<b>3 Error codes .....</b>	<b>20</b>
3.1 Field validation error codes .....	20
3.2 Authentication validation error codes .....	20
3.3 Other error codes .....	20
<b>4 Additional development resources .....</b>	<b>21</b>
4.1 End points .....	21
4.2 OpenAPI specifications .....	21
<b>5 Change log .....</b>	<b>22</b>

---

## 1 Overview

### 1.1 This solution

The Transaction Data Services (TDS) is the 'brand name' for a set of services that supply financial data to accounting software. It is to be used by accounting software for reconciling the customer's financial state with Inland Revenue's (IR's). The objective is to ascertain whether expected actions (e.g. payments, credit transfers etc) have correctly occurred and to identify any unexpected actions that have occurred. These actions might be undertaken (or not) by IR, the customer, the tax agent, or other parties such as banks. These are REST Application Programming Interfaces (APIs) with a pull model, in which IR is not responsible for management of state. The Digital Software Provider is expected to maintain state and take active role in the reconciliation of their data set.

The solution consists of three APIs:

- TDS Summary: this API is used as a change detection mechanism for transaction data change for a particular client list, customer or account.
- TDS Financials: this API is used to provide a high-level financial break down of a particular customer or account.
- TDS Transactions: this API is used to provide transaction details of a particular account's filing period.

---

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

The solution outlined in this document is intended to be used by Digital Service Providers (DSPs).

The reader is assumed to have a suitable level of technical knowledge in order to comprehend the terms and abbreviations that are used throughout this document.

### 1.3 Related services

The following APIs complement this Gateway Service. Instructions on where to find the build packs for these APIs can be found in [section 4](#) of this document.

#### 1.3.1 Identity and Access Services (required)

The START Identity and Access Management (IAM) is used to authenticate access. Authentication tokens will need to be retrieved via START IAM prior to making calls to this API.

---

## 2 Solution design

### 2.1 Architecture

IR is offering a suite of web applications to facilitate interactions via software packages. These APIs will be used by approved organisations to receive transactional data from IR.

### 2.2 Messaging

This service consists of three APIs:

- Financials
- Summary
- Transactions.

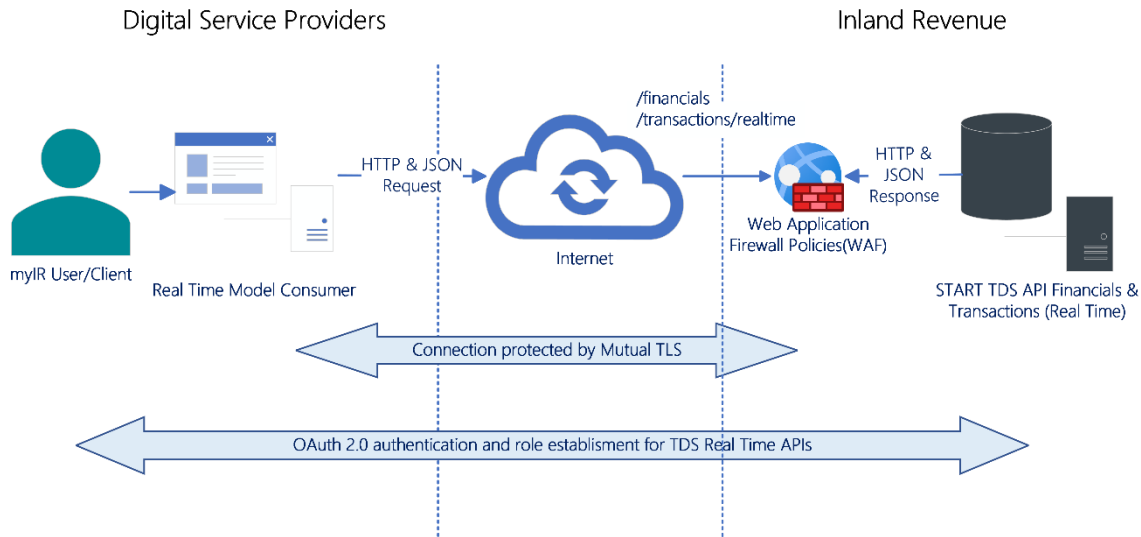
The three APIs are built with a modularized service approach to support two distinct models used by DSPs. Each model supports different use cases and requires its own security model. The two models and their intended usages are as follows:

- **Real Time Model**  
For real time users, utilising this model DSPs will build applications that interacts with gateway services to retrieve data on demand. The end user of the third-party application is the main driver of the request to Gateway Services. Therefore, this model requires a myIR web logon to authenticate with the service. The myIR logon will be supplied via OAuth token. DSP that implements this model are required to implement the two APIs: TDS Financials and TDS Transactions.  
This is the model that IR should encourage all DSP to implement for TDS APIs to retrieve financial and transaction details from START.
- **Data Reconciliation Model**  
For Batch users, utilising this model DSP replicates START financial and transaction data (currently supplied via TDS bulk file) into their own database, which is subsequently consumed by their applications. DSP that implements this model are required to implement the two APIs: TDS Summary and TDS Transactions. The myIR logon is not required to authenticate with this service. For any incumbent DSP who consume the TDS bulk file, this is the model IR recommends replacing their existing process with.

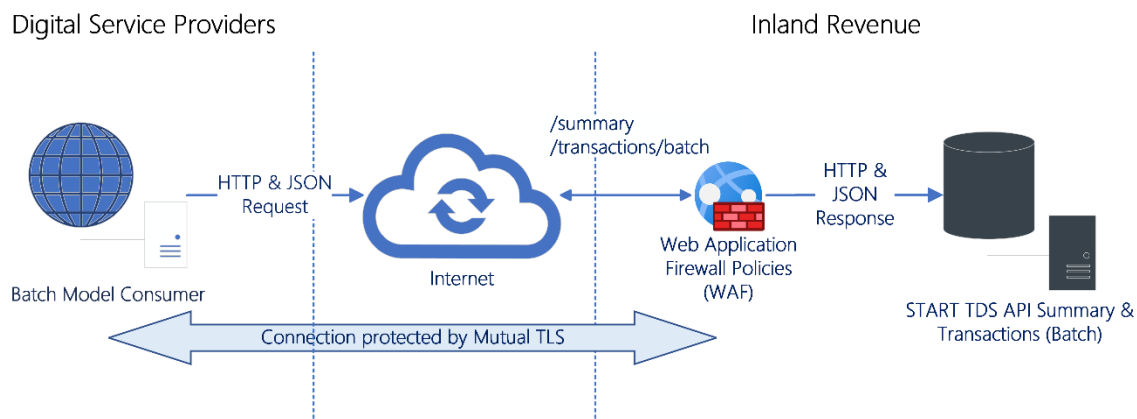
## 2.3 Flow of data to Digital Service Providers from Inland Revenue

The diagrams below illustrate the flow of data to DSPs from IR.

### 2.3.1 TDS Real Time Model



### 2.3.2 TDS Reconciliation Model



## 2.4 Associated Services

The following Associated Services may be used to obtain the required information:

- Software Intermediation SOAP service to obtain the client list IDs
- Customer API to obtain the Customer IDs
- Account API to obtain the Account IDs

## 2.5 TDS Financials

This service only supports retrieving data at the customer level (via the use of `customerId`) and account level (via the use of `accountId`)

- `customerId`: this is the identifier (IRD number) of a particular customer that the software provider is authorized to receive data for. When used, the response returns financial break down of all accounts of the requested customer.
- `accountId`: this is the identifier (ACC number) of a particular account that the software provider is authorized to receive data for. When used, the response returns financial break down of all filing periods of the requested account

Optional parameters:

- `filingPeriod`: only returns data pertained to a specific filing period
- `filingPeriodFrom`: only returns data pertained to filing period beginning from this date
- `filingPeriodTo`: only returns data pertained to filing period ending on this date
- `OutstandingPeriodOnly`: only returns filing period with non-zero balance

### 2.5.1 Request payload

POST /gateway/tds/financials

Content Type: application/json; charset=utf-8

Field	Requirement	Description
<b>customerId</b>	conditional	The unique identifier of a customer. When <code>customerId</code> is supplied, <code>accountId</code> must be omitted.
<b>accountId</b>	conditional	The unique identifier of an account. When <code>accountId</code> is supplied, <code>customerId</code> must be omitted.
<b>filingPeriod</b>	optional	An optional filter to retrieve the data set for inclusive filing periods beginning at a date. Note: INC balance dates are filed 31 March. Only <code>accountId</code> can be used in conjunction with this filter.
<b>filingPeriodFrom</b>	optional	An optional filter to retrieve the data set for inclusive filing periods beginning at a date. Note: INC balance dates are filed 31 March.
<b>filingPeriodTo</b>	optional	An optional filter to retrieve the data set for filing periods ending at a date. Only <code>accountId</code> can be used in conjunction with this filter
<b>outstandingPeriodOnly</b>	optional	An optional filter to include only outstanding periods with non-zero balance. Only <code>accountId</code> can be used in conjunction with this filter.

## 2.5.2 Response payload

HTTP Status: 200 OK

Content Type: application/json; charset=utf-8

Field	Description
<b>customer</b>	Data object representing customer
<b>customer</b> ➤ <b>customerId</b>	Unique identifier of a customer
<b>customer</b> ➤ <b>account</b>	Data object representing an account
<b>Account</b> ➤ <b>accountId</b>	Unique identifier of an account
<b>Account</b> ➤ <b>accountType</b>	An optional filter to retrieve the summary for a specific account type only
<b>Account</b> ➤ <b>accountForecastedBalance</b>	Account forecasted balance
<b>Account</b> ➤ <b>period</b>	Data object representing a filing period
<b>period</b> ➤ <b>filingPeriod</b>	Filing period
<b>period</b> ➤ <b>tax</b>	Balance of a filing period
<b>period</b> ➤ <b>penalty</b>	Penalty
<b>period</b> ➤ <b>penaltyForecasted</b>	Penalty Forecasted
<b>period</b> ➤ <b>interest</b>	Interest
<b>period</b> ➤ <b>interestForecasted</b>	Interest Forecasted
<b>period</b> ➤ <b>other</b>	Other
<b>period</b> ➤ <b>otherForecasted</b>	Other Forecasted
<b>period</b> ➤ <b>credit</b>	Credit
<b>period</b> ➤ <b>creditForecasted</b>	Credit Forecasted

Field	Description
<b>period</b> ➤ <b>balance</b>	Balance
<b>period</b> ➤ <b>balanceForecasted</b>	Balance Forecasted
<b>period</b> ➤ <b>lastPeriodActivity</b>	The most recent date when there is a financial activity occurs for this filing period. When omitted, this means there is no financial activity ever occurred for this filing period. This is omitted when there is no activity.



## 2.6 TDS Summary

TDS Summary API is used as a change detection mechanism for transaction data change for a particular client list, customer or account.

### 2.6.1 GET Request

GET /gateway/tds/summary

Returns a JSON response of when the last known transaction data is ready to be consumed.

There are no query parameter strings required for this GET method.

The **tdsStarted** value is used to identify the night batch job stream.

The expectation is that the DSP will maintain state by recording the **tdsLastUpdated** value for the last batch job. On the next business day, the DSP will compare the **tdsLastUpdated** value with the last known recorded **tdsLastUpdated** value. If there is a difference in the data time value, the DSP will know that transaction data is ready to be consumed.

**Note:** The date time values are presented in NZ local time format.

Field	Description
<b>tdsStarted</b>	When TDS process started
<b>tdsLastUpdated</b>	When TDS data last updated

### 2.6.2 POST Request payload

POST /gateway/tds/summary

Content Type: application/json; charset=utf-8

The request requires at least one input as parameter in order to specify which data set to retrieve.

- **clientId**: this is the identifier of a tax preparer client list that the software provider is authorized to receive data for. When used, the response returns the transaction summary for all active filing periods of client accounts linked to the requested client list.
- **customerId**: this is the identifier (IRD number) of a particular customer that the software provider is authorized to receive data for. When used, the response returns the transaction summary for all active filing periods of supported account types that the requested customer has.
- **accountId**: this is the identifier (ACC number) of a particular account that the software provider is authorized to receive data for. When used, the response returns the transaction summary for all active filing periods of the requested account.

By default, the service only includes filing period that has transactions.

## Optional parameters:

- **accountType**: only returns data pertained to this specific account type
- **filingPeriodFrom**: only returns data pertained to filing period beginning from this date
- **filingPeriodTo**: only returns data pertained to filing period ending on this date

Field	Requirement	Description
<b>clientListId</b>	conditional	Unique identifier of a tax preparer's client list. This can be used to retrieve the summary of all clients linked to a particular client list. When clientListId is supplied, customerId and accountId must be omitted.
<b>customerId</b>	conditional	The unique identifier of a customer. This can be used to retrieve the summary of all accounts belonging to the specified customer. When customerId is supplied, clientListId and accountId must be omitted.
<b>accountId</b>	conditional	The unique identifier of an account. This can be used to retrieve the summary of an account. When accountId is supplied, clientListId and customerId must be omitted.
<b>filingPeriodFrom</b>	optional	An optional filter to retrieve the summary for inclusive filing periods beginning at a date. Note: INC balance dates are filed 31 March.
<b>filingPeriodTo</b>	optional	An optional filter to retrieve the summary for filing periods ending at a date
<b>accountType</b>	optional	An optional filter to retrieve the summary for a specific account type only

**NOTE:** When consuming the Summary API the request needs to contain one of the following fields, clientListId, customerId or accountId.

### 2.6.3 Response payload

HTTP Status: 200 OK

Content Type: application/json; charset=utf-8

Field	Description
<b>periodCount</b>	The number periods included in the response.
<b>customer</b>	Data object representing customer
<b>customer</b> ➤ <b>customerId</b>	Unique identifier of a customer
<b>customer</b> ➤ <b>account</b>	Data object representing an account
<b>Account</b> ➤ <b>accountId</b>	Unique identifier of an account
<b>Account</b> ➤ <b>accountType</b>	An optional filter to retrieve the summary for a specific account type only
<b>Account</b> ➤ <b>accountForecastedBalance</b>	Account forecasted balance
<b>Account</b> ➤ <b>period</b>	Data object representing a filing period
<b>period</b> ➤ <b>filingPeriod</b>	Filing period
<b>period</b> ➤ <b>periodBalance</b>	Balance of a filing period
<b>period</b> ➤ <b>periodForecastedBalance</b>	Forecasted balance of a filing period
<b>period</b> ➤ <b>transactionCount</b>	Number of transactions contained within a filing period

## 2.7 TDS Transactions (Real Time and Reconciliation)

This service supports retrieving data of one single filing period per account per request. This service is accessible via 2 different endpoints:

- POST /gateway/tds/transactions/realtime
- POST /gateway/tds/transactions/batch

These two distinct endpoints will execute the same underlying service. The difference between the two is the security model being enforced. This allows DSP (Digital Service Providers) to be specific with their intent when calling the TDS Transactions API, if they happen to implement both models of the service (real time model vs. data reconciliation model).

The service returns both active and reversed transactions from START. List of transaction types available can be found in the YAML.

The value transferAccountAccountType within the credit transfer data set includes all account types available in START. It is not limited to the list of account types supported. This is due to credit can be transferred to all account types within START. See the YAML file for a list of Account type of the credit transfer destination.

The object bankAccountDetail in the response payload varies based on the bankAccountStandard value. Following is the full list of expected values:

Bank standard description	bankAccountStandard values	bankAccountDetail object structure
<b>New Zealand bank</b>	"NZB"	bank branch suffix account
<b>Australian bank</b>	"AUB"	branch account
<b>New Zealand credit union</b>	"NZCU"	bank branch account suffix referenceNumber
<b>USA/Canada bank</b>	"ABA" or "CAB"	routingNumber accountNumber
<b>SWIFT standard</b>	"BIC"	swiftBICcode basicBankAccountNumber
<b>IBAN standard</b>	"IBAN"	twoCharCountryCode checkDigit basicBankAccountNumber
<b>Any other standard not specified above</b>	Any other enumerated values not specified above	branch account

### 2.7.1 Request payload

POST /gateway/tds/transactions/realtime | batch  
 Content Type: application/json; charset=utf-8

Field	Required	Description
<b>accountId</b>	conditional	The unique identifier of an account. This can be used to retrieve the summary of an account
<b>filingPeriod</b>	conditional	The filing period to retrieve the summary for inclusive filing periods beginning at a date. Note: INC balance dates are filed 31 March.
<b>postedFrom</b>	optional	An optional filter to retrieve transactions from this particular posted date.
<b>postedTo</b>	optional	An optional filter to retrieve transactions up to this particular posted date.

### 2.7.2 Response payload

HTTP Status: 200 OK  
 Content Type: application/json; charset=utf-8

Field	Description
<b>filingPeriod</b>	Filing period
<b>periodBegin</b>	Filing period begin
<b>periodEnd</b>	Filing period end
<b>tax</b>	Tax
<b>penalty</b>	Penalty
<b>penaltyForecasted</b>	Penalty Forecasted
<b>interest</b>	Interest
<b>interestForecasted</b>	Interest Forecasted
<b>other</b>	Other
<b>otherForecasted</b>	Other Forecasted
<b>credit</b>	Credit
<b>creditForecasted</b>	Credit Forecasted
<b>balance</b>	Balance
<b>balanceForecasted</b>	Balance Forecasted
<b>lastPeriodActivity</b>	Last Period Activity. The most recent date when there is a financial activity occurs for this filing period. When omitted, this means there is no financial activity ever occurred for this filing period. This is omitted when there is no activity.
<b>billItems</b>	Data object representing bill items

Field	Description
<b>billItems</b> ➤ <b>id</b>	Bill item Id
<b>billItems</b> ➤ <b>dueDate</b>	Bill item due date
<b>billItems</b> ➤ <b>tax</b>	Balance of a filing period
<b>billItems</b> ➤ <b>penalty</b>	Penalty
<b>billItems</b> ➤ <b>penaltyForecasted</b>	Penalty Forecasted
<b>billItems</b> ➤ <b>interest</b>	Interest
<b>billItems</b> ➤ <b>interestForecasted</b>	Interest Forecasted
<b>billItems</b> <b>other</b>	Other
<b>billItems</b> ➤ <b>otherForecasted</b>	Other Forecasted
<b>billItems</b> ➤ <b>credit</b>	Credit
<b>billItems</b> ➤ <b>creditForecasted</b>	Credit Forecasted
<b>billItems</b> ➤ <b>balance</b>	Balance
<b>billItems</b> ➤ <b>balanceForecasted</b>	Balance Forecasted
<b>billItems</b> ➤ <b>transactions</b>	Data object representing transactions
<b>transactions</b> ➤ <b>transactionId</b>	Transaction Id
<b>transactions</b> ➤ <b>transType</b>	Transaction types (see YAML for a full list of Transaction Types)
<b>transactions</b> ➤ <b>postedDate</b>	Transaction posted date
<b>transactions</b> ➤ <b>effectiveDate</b>	Transaction effective date
<b>transactions</b> ➤ <b>amount</b>	Transaction amount

Field	Description
<b>transactions</b>	Linked transaction Id
➤ <b>linkedTransactionId</b>	
<b>pendingPayments</b>	Data object representing pending payments
<b>pendingPayments</b>	Pending payment Id
➤ <b>paymentId</b>	
<b>pendingPayments</b>	Pending payment amount
➤ <b>paymentAmount</b>	
<b>pendingPayments</b>	Pending payment date
➤ <b>paymentPendingDate</b>	
<b>creditTransfers</b>	Data object representing credit transfer
<b>creditTransfers</b>	Transaction Id of the credit transfer
➤ <b>transactionId</b>	
<b>creditTransfers</b>	Account Id of the credit transfer destination
➤ <b>transferAccountId</b>	
<b>creditTransfers</b>	Account type of the credit transfer destination (see YAML file for a full list of Account Type)
➤ <b>transferAccountAccountType</b>	
<b>creditTransfers</b>	Filing period of the credit transfer destination
➤ <b>transferPeriod</b>	
<b>disbursements</b>	Data object representing disbursement
<b>disbursements</b>	Transaction Id of the credit transfer
➤ <b>transactionID</b>	
<b>disbursements</b>	Bank account standard of the disbursement bank account. See YAML file for a full list of disbursement banks.
➤ <b>bankAccountStandard</b>	
<b>disbursements</b>	Data object representing bank account detail. This object varies for different bank account standards.
➤ <b>bankAccountDetail</b>	
<b>bankAccountDetail</b>	Bank account detail - bank
➤ <b>bank</b>	
<b>bankAccountDetail</b>	Bank account detail - branch
➤ <b>branch</b>	
<b>bankAccountDetail</b>	Bank account detail - account
➤ <b>account</b>	
<b>bankAccountDetail</b>	Bank account detail - suffix
➤ <b>suffix</b>	
<b>bankAccountDetail</b>	Bank account detail - accountNumber
➤ <b>accountNumber</b>	

Field	Description
<b>bankAccountDetail</b> ➤ <b>routingNumber</b>	Bank account detail - routingNumber
<b>bankAccountDetail</b> ➤ <b>referenceNumber</b>	Bank account detail - referenceNumber
<b>bankAccountDetail</b> ➤ <b>basicBankAccountNumber</b>	Bank account detail - basicBankAccountNumber
<b>bankAccountDetail</b> ➤ <b>swiftBICcode</b>	Bank account detail - swiftBICcode
<b>bankAccountDetail</b> ➤ <b>twoCharCountryCode</b>	Bank account detail - twoCharCountryCode
<b>bankAccountDetail</b> ➤ <b>checkDigit</b>	Bank account detail - checkDigit
<b>transactions</b>	Data object representing transactions <b>not</b> associated to a bill item
<b>transactions</b> ➤ <b>transactionId</b>	Transaction Id
<b>transactions</b> ➤ <b>transType</b>	Transaction types (see YAML for a full list of Transaction Types)
<b>transactions</b> ➤ <b>postedDate</b>	Transaction posted date
<b>transactions</b> ➤ <b>effectiveDate</b>	Transaction effective date
<b>transactions</b> ➤ <b>amount</b>	Transaction amount
<b>transactions</b> ➤ <b>linkedTransactionId</b>	Linked transaction Id



## 2.8 Security

### 2.8.1 Information classification

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

### 2.8.2 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 IR does not issue certificates to external vendors for web service security implementations.)

The following requirements are in place 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.

IR 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:

- [Amazon](#)
- [Comodo](#)
- [DigiCert](#)
- [Entrust](#)
- [GeoTrust](#)
- [Let's Encrypt](#)
- [Sectigo](#)
- [Thawte](#).

IR expects DSPs to use their Developer Portal account to create their common name for both Test and Production certificates. Please refer to the [Digital Service Providers](#) pages on the IR website or contact [GatewayServices@ird.govt.nz](mailto:GatewayServices@ird.govt.nz) for further details.

In addition, we impose two different security models depending on which TDS API (and endpoint) being accessed to validate whether the request has sufficient access and authorization to retrieve the data set requested. At a high level, they are:

Available APIs	Security model enforced	OAuth supported	Targeted use case
<b>TDS Summary</b>	Link validation	no	Data reconciliation model
<b>TDS Transaction (batch endpoint)</b>	Link validation	no	Data reconciliation model
<b>TDS Financials</b>	myIR access	yes (myIR logon)	Real time model
<b>TDS Transaction (real time endpoint)</b>	myIR access	yes (myIR logon)	Real time model

## 2.9 Authentication Options

This design will use OAuth2.0 tokens and protocol to establish the calling party's identity. The OAuth2.0 method requires a myIR user to logon.

Self-signed M2M JWT access tokens is not supported for this service.

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.*

### 2.9.1 Authentication using OAuth (including Mutual TLS)

DSP utilising the real time model will use OAuth, the interaction with IR is transacted under the identity of a myIR user, with the OAuth token being used to identify which customers the consumer of service has access to. If the member/investor does not exist in this list, access will be denied. The myIR user 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.

#### **myIR access** (applicable to Real Time model)

myIR logon supplied via OAuth is used to perform the required delegation check.

- when clientListId is used (for the applicable API), the request is considered authorized when the supplied myIR web logon is of the Intermediary logon type and such web logon has client list access to the requested client list.
- when customerId or accountId is used, the request is considered authorized when:
  - the supplied myIR logon has myIR access to the requested customer's account(s); OR
  - the supplied myIR logon is of the Intermediary logon type and such web logon has client list access to the requested customer's account(s)
  - **Note:** In this context, "client list access" refers to myIR client list security. This is an existing security functionality supported in myIR.

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

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

### 2.9.2 Authentication using Mutual TLS

#### **Link validation** (applicable to Data Reconciliation model)

We recommend DSP to set up a separate mutual TLS credential to be used exclusively for the TDS Summary and TDS Transactions (batch endpoint). This credential will be associated to the DSP entity who holds all the applicable software intermediation links.

- when clientListId is used (for the applicable API), the request is considered authorized when the associated software intermediary customer entity holds an active and current "Software intermediary" link to the requested tax preparer's client list. The data set returned includes all accounts with an active and current tax preparer links to the requested tax preparer's client list.

- 
- when customerId or accountId is used, the request is considered authorized when:
    - the associated software intermediary customer entity holds an active and current "Software intermediary" link to the requested customer's account(s), or
    - the associated software intermediary customer entity holds an active and current "Software intermediary" link to a tax preparer client list, and the aforementioned client list holds an active and current tax preparer links to the requested customer's account(s).

### 3 Error codes

#### 3.1 Field validation error codes

Error code	Occurs	TDS Summary	TDS Financials	TDS Transactions	HTTP Status Code
<b>TDS101</b>	Request does not meet defined specifications.	yes	yes	yes	400
<b>TDS102</b>	At least one parameter is required.	yes	yes	yes	400
<b>TDS301</b>	Invalid Id	yes	yes	yes	400
<b>TDS302</b>	Filing period does not exist	no	no	yes	400

#### 3.2 Authentication validation error codes

Error code	Occurs	TDS Summary	TDS Financials	TDS Transactions	HTTP Status Code
<b>TDS200</b>	Insufficient access for the requested client list or customer.	yes	yes	yes	401
<b>TDS201</b>	The requester is not eligible for this service.	yes	yes	yes	401
<b>TDS202</b>	Authentication failure.	no	yes	yes	400
<b>TDS203</b>	Authentication failure due to missing token	no	yes	yes	400
<b>TDS204</b>	Authentication failure due to invalid token	no	yes	yes	400
<b>TDS205</b>	Authentication failure due to missing authentication header	no	yes	yes	400
<b>TDS206</b>	Unknown authentication failure.	no	yes	yes	400

#### 3.3 Other error codes

For other errors, the HttpStatusCode returned will be '500—internal server error'.

---

## 4 Additional development resources

### 4.1 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).

### 4.2 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.

---

## 5 Change log

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

Date of change	Document section	Description
<b>13/01/2023</b>		<ul style="list-style-type: none"><li>Draft version 0.5</li></ul>