



# Corporation for Deposit Insurance

---

## Data Submission SFTP and API Integration Channel Technical Guide

November 2024

© South African Reserve Bank

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without fully acknowledging the South African Reserve Bank as the source. While every precaution is taken to ensure the accuracy of information, the South African Reserve Bank shall not be liable to any person for inaccurate information or opinions contained in this document.

## Contents

<b>1.</b>	<b>Introduction.....</b>	<b>4</b>
<b>2.</b>	<b>SFTP integration.....</b>	<b>4</b>
<b>2.1</b>	<b>When to use SFTP as the integration channel for data submissions .....</b>	<b>5</b>
<b>2.2</b>	<b>Data submission prerequisites.....</b>	<b>5</b>
2.2.1	Data file preparation.....	5
2.2.2	Data file formats.....	6
2.2.3	Data file naming conventions .....	6
<b>2.3</b>	<b>Setting up the SFTP client .....</b>	<b>7</b>
2.3.1	SFTP client installation.....	7
2.3.2	Whitelisted and static IP addresses.....	8
2.3.3	Connection and user details.....	8
<b>2.4</b>	<b>Submitting data to CODI via the SFTP client .....</b>	<b>8</b>
2.4.1	SFTP client interface data transfer .....	8
2.4.2	SFTP command line data transfer.....	9
2.4.3	Shell scripting SFTP data transfers .....	10
<b>2.5</b>	<b>Submission processing and workflow communication.....</b>	<b>10</b>
<b>2.6</b>	<b>Replacement file submission .....</b>	<b>11</b>
<b>2.7</b>	<b>SFTP configuration support.....</b>	<b>12</b>
<b>3.</b>	<b>API integration .....</b>	<b>12</b>
<b>3.1</b>	<b>When to use the APIs as the integration channel for data submissions .....</b>	<b>13</b>
<b>3.2</b>	<b>Data submission prerequisites.....</b>	<b>13</b>
3.2.1	Data file preparation.....	13
3.2.2	Connection and user details.....	14
<b>3.3</b>	<b>Authentication API.....</b>	<b>14</b>
3.3.1	Authentication API header.....	15
3.3.2	Authentication API success response .....	15
3.3.3	Authentication API error response and parameter validations .....	16
<b>3.4</b>	<b>Data submission API .....</b>	<b>17</b>
3.4.1	Data submission API header .....	18
3.4.2	Data submission API body .....	19
3.4.3	Data submission API success response.....	20
3.4.4	Data submission API error response and parameter validations .....	20

3.5	Data submission workflow notifications.....	22
3.6	Replacement file submission .....	23
3.7	API technical configuration (YAML file) .....	24
3.8	API Integration support.....	24

## Tables

Table 1: Key SFTP features .....	4
Table 2: Summary of member bank mandatory data submissions .....	5
Table 3: Summary of common SFTP commands .....	9
Table 4: Summary of member bank mandatory data submissions .....	13
Table 5: Authentication API URL .....	14
Table 6: Authentication API processing steps .....	14
Table 7: Authentication API parameter definition .....	15
Table 8: Authentication API success response.....	15
Table 9: Authentication API parameter validations .....	16
Table 10: Authentication API header error response.....	17
Table 11: Data submission API URL .....	17
Table 12: Data submission API processing steps.....	18
Table 13: Data submission API header parameters .....	18
Table 14: Data submission API body parameters .....	19
Table 15: Data submission API success response .....	20
Table 16: Data submission API parameter validations (header and body) .....	20
Table 17: Data submission API error response.....	22

## Figures

Figure 1: Member bank data submission file naming convention.....	7
Figure 2 Member bank declaration file naming convention .....	7
Figure 3: Standard workflow for the data submission .....	11
Figure 4: File naming convention – sequence number increment for replacement files ..	12
Figure 5: Standard workflow for data submission and approval.....	23

## ABBREVIATIONS

Abbreviation	Full form
API	Application programming interface
CODI	Corporation for Deposit Insurance
CSV	Comma-Separated Values
IT	Information Technology
JSON	Java Script Object Notation
PDF	Portable Document Format
SCV	Single Customer View
SFTP	Secure File Transfer Protocol
YAML	A human-readable data serialisation language that is often used for writing configuration files

## 1. Introduction

As part of the next phase of its rollout to member banks, the Corporation for Deposit Insurance (CODI) will be providing additional data submission channels – a secure file transfer protocol (SFTP) and an application programming interface (API) – to supplement the existing portal-based file upload channel. These additional channels are Secure File Transfer Protocol (SFTP) integration and direct Application Programming Interface (API) based integration.

## 2. SFTP integration

SFTP is a network protocol designed to provide secure and reliable file transfers. SFTP is specifically designed for transmitting large numbers of files or large files very efficiently and securely. The SFTP protocol encrypts data while in motion, protecting data against malicious attacks. Table 1 lists key features of SFTP.

**Table 1: Key SFTP features**

No.	Feature
1	Part of the Secure Shell protocol group, encrypts both data and commands
2	Offers a variety of authentication mechanisms (e.g. password, public key, and host-based authentication)
3	Supports a variety of file transfer operations, such as uploading, downloading, renaming, and deleting files, and allows for directory listings as well as the creation of directories on the remote server
4	Supports file segmentation and reassembly, allowing for efficient transfer of large files without the risk of data corruption
5	Being a connection-oriented protocol, SFTP can resume the file transfer where it left off in the event of an interruption
6	Compatible with most operating systems, making it a versatile tool for file transfers
7	Provides logging and auditing capabilities that enable regulatory compliance for both CODI and banks while strengthening the security of the data
8	Uses a wide variety of encryption algorithms (e.g. 3DES, Blowfish, AES, etc.), making it extremely difficult for hackers to decipher the contents of the files
9	Data files can be protected from 'packet sniffing', 'man-in-the-middle' and other attacks
10	Utilises hashing to ensure the integrity of the data

No.	Feature
	Hashing is a one-way mathematical function that turns data into a string of nondescript text that cannot be reversed or decoded. In the context of cybersecurity, hashing is a way to keep sensitive information and data, including passwords, messages and documents, secure.

## 2.1 When to use SFTP as the integration channel for data submissions

Member bank single customer view (SCV) data submissions to CODI may consist of large files, containing sensitive and personally identifiable depositor and financial data. SFTP integration offers a secure and efficient way for member banks to share large volumes of data with CODI. It ensures the confidentiality, integrity, and availability of financial information while complying with industry-specific regulations such as the Sarbanes-Oxley Act of 2002. It is recommended that member banks opt to utilise the SFTP data submission channel when:

- data submission files are larger than 500 megabytes in size;
- internet connectivity is slow or unstable; and
- the member bank's data preparation team or integration prefers tighter integration to CODI as opposed to the portal-based manual upload processes.

## 2.2 Data submission prerequisites

### 2.2.1 Data file preparation

Member banks are required to prepare their data submission files as per the specifications provided in the CODI Member Bank Data Handbook as well as CODI's SCV calculation guide. Standard data submissions, their respective data submission codes, and submission frequencies are outlined in Table 2 below.

**Table 2: Summary of member bank mandatory data submissions**

Data submission	Data submission code	Frequency
Total qualifying deposits and total covered deposits	R005	Monthly
Quarterly SCV calculations	R006	Quarterly

Data submission	Data submission code	Frequency
Ad hoc SCV calculations	R007	Ad hoc
Member bank declaration	A1, A2 and A3	With every monthly, ad hoc, and resolution data submission
Bank in resolution submission	R008	As per resolution guidelines

### 2.2.2 Data file formats

Only comma-separated values (CSV) and ZIP file formats are permitted for data files. Declarations must be in portable document format (PDF). ZIP files must not be password protected.

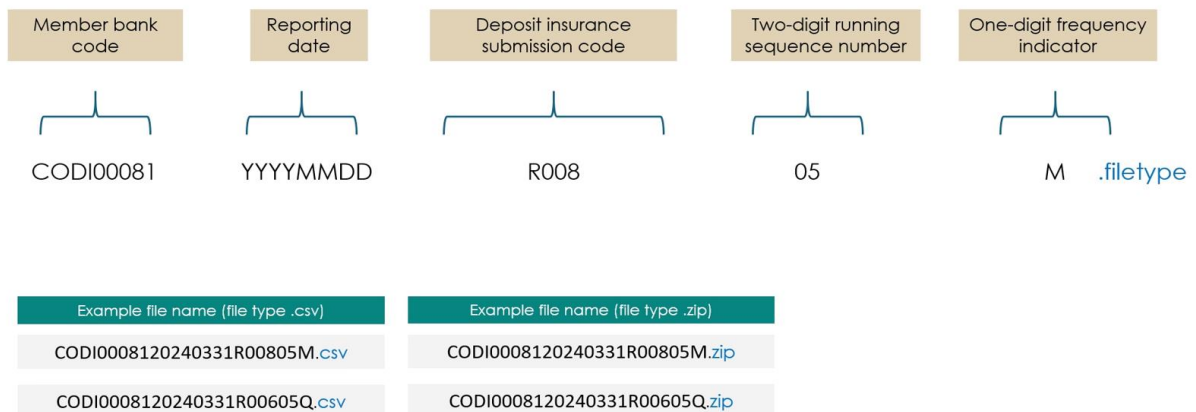
### 2.2.3 Data file naming conventions

Each uploaded file must adhere to a specific file name format to ensure consistency and traceability. File-based metadata is embedded within the respective file names to advise the data collection platform of the submission entity, the reporting period, the type of data submission, the version of the file and the submission frequency. The file name format must contain the:

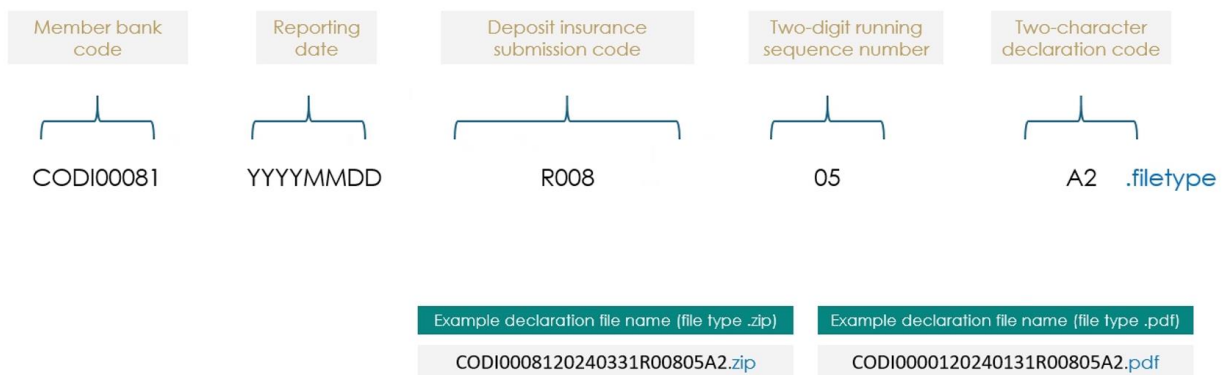
- **member bank code** – the nine-digit unique code that CODI allocates to a member bank to identify a member bank;
- **reporting date** – the date that the information for the preparation of SCV calculations is based on, formatted as YYMMDD;
- **deposit insurance submission code** – the four-digit code representing the type of collection report;
- **running sequence** – a two-digit sequence number to ensure file uniqueness, especially for version control; and
- **frequency indicator** – a one-digit character indicating the frequency of the report (monthly, quarterly, etc.).

Figure 1 below provides a summary of the file naming convention segments and for each data submission and declaration file.

**Figure 1: Member bank data submission file naming convention**



**Figure 2: Member bank declaration file naming convention**



## 2.3 Setting up the SFTP client

### 2.3.1 SFTP client installation

The member bank's technical team will be required to install/configure a suitable SFTP client on the member bank machine(s) that will be submitting data to CODI. Should the member bank not have a dedicated standard application that provides SFTP services, any of the freely available SFTP clients can be downloaded from:

- WINSCP (<https://winscp.net/>); or
- FileZilla (<https://filezilla-project.org/>).



### 2.3.2 Whitelisted and static IP addresses

To maintain CODI's data submission security protocols, only whitelisted member bank machines will be allowed to submit data to CODI via the SFTP channel. Member banks will be required to set up these machines with **static IP addresses**. These IP addresses must be provided to the CODI security team for whitelisting. Member banks can check the public IP addresses of their client machines by visiting the following URL: <https://whatismyipaddress.com/>.

### 2.3.3 Connection and user details

Before member banks can submit data utilising their SFTP client, the CODI technical team will provide the following connection and user details:

- the SFTP end point/host name;
- the secure port number;
- the member bank SFTP username;
- the member bank SFTP password; and
- confirmation of whitelisted IP addresses per member bank.

Once the SFTP client has been set up, member banks will be able to upload member bank data, using either the respective SFTP client's user interface, command line execution or scripted shell transfer.

## 2.4 Submitting data to CODI via the SFTP client

Once the SFTP client has been installed and configured, the member bank will have various options to submit data files to CODI. Each member bank's submitted data files will land in a dedicated secure landing zone.

### 2.4.1 SFTP client interface data transfer

Each SFTP client will have its unique user interface. Once the member bank has successfully connected to the target CODI SFTP remote server, most interfaces will provide a drag and drop file transfer interface to allow the member bank to transfer data files from the host machine location to the CODI remote SFTP server. In the event of an interruption during the file transfer, the user interface should also provide an option to resume the upload.

## 2.4.2 SFTP command line data transfer

SFTP services can also be accessed via a command line. Typical SFTP commands to upload and resume data transfers are outlined in Table 3 below.

**Table 3: Summary of common SFTP commands**

Step	Description	SFTP commands
Connecting to the remote server	<p>To initiate an SFTP connection to a remote server, the <b>host name</b> or <b>IP address</b>, <b>username</b> and, optionally, the <b>port number</b> will be required.</p> <p>CODI will provide each member bank with a:</p> <ul style="list-style-type: none"><li>• member bank SFTP username;</li><li>• host name; and</li><li>• secure port number.</li></ul> <p>The interface will prompt the user to provide a password. Once connected, the member bank will land on a dedicated target remote server directory at CODI.</p>	<code>sftp -P 1234 username@hostname</code>
Navigating local server directories	<p>Once connected, the following SFTP commands to navigate to the local server directories to locate upload files can be used:</p> <ul style="list-style-type: none"><li>• <code>lpwd</code>; and</li><li>• <code>lcd</code>.</li></ul>	<ul style="list-style-type: none"><li>• <b>lpwd</b> displays the current local directory</li><li>• <b>lcd</b> changes the current directory on the local machine</li></ul> <p><i>Example:</i> <code>lcd/path/to/local/directory</code></p>
Uploading data submission files (single file transfer)	<p>Once the member bank has navigated to its local directory, the '<b>put</b>' function can be used to transfer a single data submission file from the local directory to the remote server.</p> <p>This process must be repeated if multiple files must be submitted.</p>	<ul style="list-style-type: none"><li>• <b>put</b> uploads a single file from the local machine to the remote server</li></ul> <p><i>Example:</i> <code>put/local/path/to/file.txt</code></p>

Step	Description	SFTP commands
Uploading data submission files (multi-file transfer with wildcard)	Once the member bank has navigated to its local directory, the ' <b>mput</b> ' function can be used to transfer multiple data submission files from the local directory to the remote server.	<ul style="list-style-type: none"> <li>• <b>mput</b> uploads multiple files from the local machine to the remote server. A wildcard character (*) must be used as a placeholder for multiple file names for files with a common extension.</li> </ul> <p><i>Example:</i>  <code>mput /local/path/*.txt</code></p>
Verifying uploads	Utilise the ' <b>ls</b> ' and ' <b>lls</b> ' commands to confirm that all files have successfully been transferred from the local client directory to the remote server target directory.	<ul style="list-style-type: none"> <li>• <b>ls</b> lists files in the current directory on the remote server</li> <li>• <b>lls</b> lists files in the current directory on the local machine</li> </ul>
Resuming an interrupted upload	If a file upload has been interrupted, the ' <b>reput</b> ' command can be used, which resumes uploading a partially uploaded file from the local system to the remote server.	<ul style="list-style-type: none"> <li>• <b>reput</b> resumes uploading a partially uploaded file from the local system to the remote server</li> </ul>

### 2.4.3 Shell scripting SFTP data transfers

Member bank technical teams will also have to use batch or shell scripts to execute and schedule file uploads. The following sample shell scripts are available for download from the CODI Information Technology (IT) Solution under the download menu:

- SFTP – sample Windows batch script
- SFTP – sample Linux shell script

**Note:** Parameters should be replaced with actual values of the member bank's production environment.

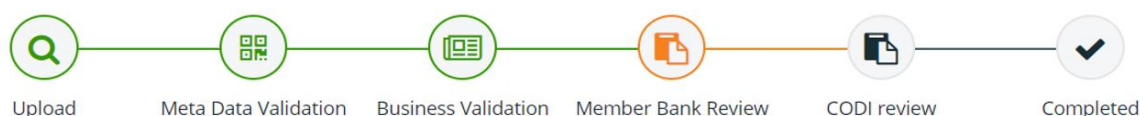
## 2.5 Submission processing and workflow communication

- A dedicated landing zone is provided for each member bank with restricted access.
- Once uploaded onto the dedicated directory on the SFTP server, the member bank's data submission will be encrypted.

- Data submission files will be automatically queued for upload into the CODI IT Solution.
- The member bank 'maker users' will be informed via email that their data submission file is being processed and that the respective data submission file's metadata and business data validations are being performed.
- Once the metadata and business data validations are complete, the member bank 'maker users' will receive an email confirming completion and notifying them if any errors were encountered.
- The standard data submission approval processes will apply from that point onwards.

Figure 3 below illustrates the standard data submission and approval workflow.

**Figure 3: Standard workflow for the data submission**



## 2.6 Replacement file submission

Member banks can submit replacement files if a data file is incorrect. Replacement file submissions are allowed when:

- no data submission has been *previously concluded* for that reporting period (i.e. the previous member bank data submission has been approved by both the member bank approver ('checker user') and the CODI data manager) (if this has occurred, an official resubmission request must be logged); and
- the data submission is still within an approval workflow for that reporting period.

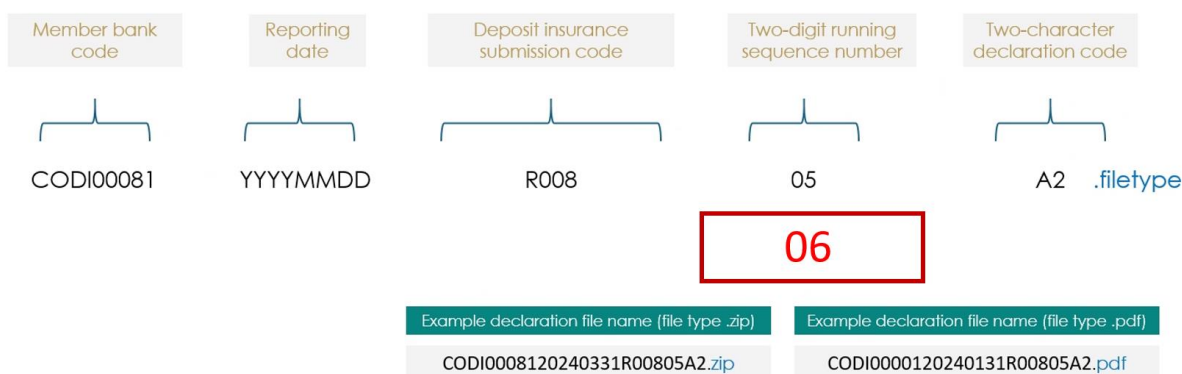
If there is a data submission still within an approval workflow, and it is in the workflow submission and approval step where it still requires member bank approval, the member bank 'checker user' will be required to reject with the data submission before the member bank will be allowed to submit a replacement file.

If the data submission is in the approval step where it still requires CODI data approval, then the member bank is required to log a support request at [CODISupport@resbank.co.za](mailto:CODISupport@resbank.co.za) requesting that the data submission be rejected.

Once the data submission has been rejected, this will effectively reset the data submission process and member banks can upload the corrected replacement file.

When preparing replacement files, the member bank data preparation or data integration teams must increase the sequence number in the file name. This is so that the CODI data collection platform utilises the correct file version and the submission does not fail to upload based on a duplicate file name validation. Figure 3 below illustrates the version sequence number that must be incremented when submitting a replacement data submission file.

**Figure 4: File naming convention – sequence number increment for replacement files**



**Increase the sequence number to indicate newer version**

## 2.7 SFTP configuration support

Should any assistance be required in setting up and using an appropriate SFTP client, member banks can log a support query at [CODISupport@resbank.co.za](mailto:CODISupport@resbank.co.za).

## 3. API integration

CODI will provide an API to allow member banks to upload their data submissions directly to the CODI IT Solution, which utilises a RESTful API architecture style.

- **Service type:** RESTful web services
- **Request object type:** JavaScript Object Notation (JSON)
- **Response object type:** JSON

### 3.1 When to use the APIs as the integration channel for data submissions

It is recommended that member banks opt to utilise the API data submission channel when:

- data submission files are smaller than 500 megabytes in size;
- internet connectivity is fast and stable (API-based integration methods are stateless and do not have a resume capability and thus, if a failure or interruption occurs, the API-based data submission must be restarted); and
- the member bank data preparation team or integration prefers tighter integration to CODI as opposed to the portal-based manual upload.

### 3.2 Data submission prerequisites

#### 3.2.1 Data file preparation

Member banks are required to prepare their data submission files as per the specifications provided in the CODI Member Bank Data Handbook as well as the CODI SCV calculation guide. Standard data submissions, their respective data submission codes and respective frequencies are outlined in Table 4 below. Files submitted using the API data submission channel are not required to have any specific naming conventions as they will be converted into file objects and incorporated as part of the API JSON payload. The JSON payload will include relevant metadata that advises the data collection platform of the:

- member bank making the data submission;
- reporting period; and
- type of data submission being submitted.

**Table 4: Summary of member bank mandatory data submissions**

Data Submission	Data Submission Code	Frequency
Total qualifying deposits and total covered deposits	R005	Monthly
Quarterly SCV calculations	R006	Quarterly
Ad hoc SCV calculations	R007	Ad hoc
Member Bank Declaration	A1, A2, A3	With every monthly, ad hoc, and resolution data submission
Bank in resolution submission	R008	As per resolution guidelines

### 3.2.2 Connection and user details

Before member banks can utilise the CODI data submission API, the CODI technical support team will provide the following connection and user details:

- the API URL/endpoints for:
  - authentication; and
  - data submission;
- the member bank entity token;
- the member bank API username; and
- the member bank API password.

### 3.3 Authentication API

API-based data submission to CODI is executed through a two-stage integration process. During the first stage, member bank clients making the submission are required to be authenticated, after which they will receive an authentication token. This authentication token is used in the next stage so that data files can be submitted to CODI via the data submission API. The authentication API may be accessed using the URL/endpoint in Table 5.

**Table 5: Authentication API URL**

API URL	Method
APICollect/authenticate/1.0.0/getAuthToken	POST

Table 6 outlines the steps that are executed as part of the authentication API workflow.

**Table 6: Authentication API processing steps**

Integration step	Description
1	The client inputs the <b>entity token</b> , <b>entity code</b> , <b>username</b> and <b>password</b> in the header to authenticate the connection
2	After successful authentication, the API server provides an <b>authentication token</b> to the user to make the next call while passing the actual data to be submitted (this token has a defined validity/expiry time)

### 3.3.1 Authentication API header

Member banks are required to provide relevant data to the authentication API. Table 7 below provides an example of the key value pairs that must be input in the header section of the authentication API.

**Table 7: Authentication API parameter definition**

Parameters	Parameter type	Data type	Sample input	Comments
entityToken*	header	string	a8bf3264f58259e26334 2e91522c7a07647014f 68da3ad7466a7500110 a7d91b	A unique token will be generated and provided to each member bank.
entityCode*	header	string	CODI000630	Member bank entity code
userName*	header	string	test	Member bank API username
userPswd*	header	string	test@1234	Member bank API password

### 3.3.2 Authentication API success response

Upon successful authentication of the member bank details, the authentication API will issue a JSON response with an authentication token. An example is provided in Table 8 below.

**Table 8: Authentication API success response**

Model schema	Sample output	Comments
{ "response": {}, "status": boolean, "statusCode": "string", "statusMessage": "string" }	{ "status": true, "response": { "authToken": "Mp/AO/0jZxfGaZCxAlPjS+xWaiLSbLVPsm qoJ7Bz2BM=" } }	The default validity of the authentication token is 120 seconds. This can be configured upon agreement with member banks.



### 3.3.3 Authentication API error response and parameter validations

To ensure integrity of the integration payload, various validations are applied to the respective authentication API header payload parameters. If a parameter fails a validation, the API will return a JSON response indicating the nature of the failure. The following two tables provide a list of the validations that apply to the header payload as well as a sample JSON failure response if any data validation error(s) are returned.

**Table 9: Authentication API parameter validations**

Parameters	Validation	Status	Status code	Status message
entityToken*	Has the token been provided?	false	API_COLLECT_ERR_001	entityToken must be provided
	Does the entity token match the one issued to the member bank?	false	API_COLLECT_ERR_002	Invalid entityToken provided
entityCode*	Has the entity code been provided?	false	API_COLLECT_ERR_003	entityCode must be provided
	Does the entity code match the one issued to the member bank?	false	API_COLLECT_ERR_004	entityCode does not match with the one present in the application
userName*	Has the username been provided?	false	API_COLLECT_ERR_005	userName must be provided
userPswd*	Has the user password been provided?	false	API_COLLECT_ERR_006	userPswd must be provided
userName* and userPswd*	Is the username or password valid?	false	API_COLLECT_ERR_007	Invalid credentials

Table 10 provides a sample error response in the event that a validation error is triggered based on the respective data authentication API header parameter validations for supportingDocType.

**Table 10: Authentication API header error response**

Authentication API validation	Sample response
Authentication API error response  The API can return multiple responses if the input payload has multiple validation errors	{ "status": false, "statusCode": "API_COLLECT_ERR_001", "statusMessage": "entityToken is mandatory to be provided" }

### 3.4 Data submission API

In the second stage of the data submission, the API utilises the authentication token provided in the previous stage to confirm the identity of the client machine. Once confirmed, the API will allow the submitted data files to be uploaded with the file metadata. Once the member bank's data files have been uploaded, the data collection platform will perform the business data and metadata validations in the same manner as if a user had uploaded the files manually through the data submission portal. Member banks can call the CODI authentication API from the URL/endpoint in Table 11.

**Table 11: Data submission API URL**

API URL	Method
APICollect/collect/1.0.0/submitFilingDataFile	POST

Table 12 outlines the process steps that are executed as part of the data submission API workflow.

**Table 12: Data submission API processing steps**

Integration step	Description
1	<ul style="list-style-type: none"> <li>The client is required to pass the <b>entity token</b>, <b>entity code</b>, <b>return code</b> and <b>authentication token</b> in the API payload header.</li> <li>The client is required to pass the <b>reporting period end date</b>, <b>reporting type</b> and binary file objects for the <b>data submission data file</b> and <b>supporting declaration</b>.</li> </ul>
2	The data submission API checks if the authentication token received is valid and active. Upon confirmation of valid credentials, the API will process the API data submission payload body and upload the data submission file objects into the data collection platform.

### 3.4.1 Data submission API header

Member banks are required to provide relevant data to the data submission API. Table 13 below provides an example of the key value pairs that must be passed in the header section to the data submission API.

**Table 13: Data submission API header parameters**

Parameters	Sample input	Data type	Comments
entityToken*	entityToken: a8bf3264f58259e263342e91522c7a07647014f68da3 ad7466a7500110a7d91b	string	A unique token is issued by CODI to each member bank.
authToken*	authToken: zb2f5672f58259e263342e91522c7a07647014f68da3 ad7466a7500110a7a82g	string	A unique authentication token is received after the first step of authentication.

Parameters	Sample input	Data type	Comments
entityCode*	entityCode: CODI00001	string	The entity working code
returnCode*	returnCode: R0001	string	The return code for which filing is being done

### 3.4.2 Data submission API body

Member banks are required to provide relevant data to the Data Submission API in JSON format. The table below provides an example of the key value pairs that must be passed in the body section to the Data Submission API.

**Table 14: Data submission API body parameters**

Parameter	Value	Sample input	Data type	Content type	Comments
reportingPeriodEndDate supportingDocType filingDoc supportingDoc	{ "reportingPeriodEndDate": number,  "supportingDocType": string,  "filingDocObj ect": File }  File	{ "reportingPeriodEndDate": "30031990",  "supportingDocType": "A", "filingDoc": {FILE_OBJECT}, "supportingDoc": {FILE_OBJECT} }  }	Model schema	application/json	<ul style="list-style-type: none"> <li>• <b>reportingPeriodEndDate</b> – the reporting period in <b>ddmmyyyy</b> format</li> <li>• <b>supportingDocType</b> – This field is mandatory if the <b>supportingDoc</b> is reported. The valid values are A2 and A3.</li> <li>• <b>filingDoc</b> – the actual data file in the multipart file object format</li> <li>• <b>supportingDoc</b> – the attachment file in the multipart file object format</li> </ul>

### 3.4.3 Data submission API success response

Upon successful execution of the data upload via the data submission API, the API will provide the following response JSON with a confirmed 'filingId', indicating that the data files have been successfully submitted.

**Table 15: Data submission API success response**

Response	Model schema	Sample output
Success response	<pre>{   "response": {},   "status": true,   "statusCode": "string",   "statusMessage": "string" }</pre>	<pre>{   "status": true,   "statusCode": null,   "statusMessage": null,   "response": {"filingId":123} }</pre>

### 3.4.4 Data submission API error response and parameter validations

To ensure the integrity of the integration payload, various validations are applied to the respective header parameters. If a parameter fails validation, the API will return a JSON response indicating the nature of the failure. The following two tables provide a list of the validations applied to the header and body payload as well as a sample JSON failure response if any data validation error(s) are returned.

**Table 16: Data submission API parameter validations (header and body)**

Parameters	Validation	Status	Status code	Status message
entityToken*	Has the entity token been provided?	false	API_COLLECT_ERR_001	entityToken must be provided

Parameters	Validation	Status	Status code	Status message
	Does the entity token match the one issued to the member bank?	false	API_COLLECT_ERR_002	Invalid entityToken provided
entityCode*	Has the entity code been provided?	false	API_COLLECT_ERR_003	entityCode must be provided
	Does the entity code match the one issued to the member bank?	false	API_COLLECT_ERR_004	entityCode does not match with the one present in the application
authToken*	Has the authentication token been provided?	false	API_COLLECT_ERR_010	authToken must be provided
	Does the authentication token match the one issued during the authentication step?	false	API_COLLECT_ERR_011	authToken does not match with the one generated for this entity from the application (or) the token is not valid/active
returnCode*	Has the return code been provided?	false	API_COLLECT_ERR_012	returnCode must be provided
	Does the return code match a valid return code in the data collection application?	false	API_COLLECT_ERR_013	returnCode does not match with the one present in the application
	Is this return code allowed to be submitted by the submission user?	false	API_COLLECT_ERR_014	returnCode is not assigned to the entity/entity-user for submission via API channel
reportingPeriodEndDate*	Has the reporting period end date been provided?	false	API_COLLECT_ERR_015	reportingPeriodEndDate not a valid date reported in ddmmyyyy format

Parameters	Validation	Status	Status code	Status message
supportingDocType	Optional to be reported	false	API_COLLECT_ERR_016	The supporting document type. The valid options are A2 and A3. This is optional and has to be reported if the <b>supportingDoc</b> is being reported
filingDoc*	Has the filing document object been provided?	false	API_COLLECT_ERR_017	Not a valid file object.
supportingDoc	Optional to be reported	false	API_COLLECT_ERR_017	Not a valid file object.

The following table provides a sample error response if a validation error is triggered based on the respective data submission API body parameter validations.

**Table 17: Data submission API error response**

Data submission API validation	Sample response
Data submission API error response: <i>The API can return multiple responses if the input payload has multiple validation errors</i>	{ "status": false, "statusCode": "API_COLLECT_ERR_015", "statusMessage": "reportingPeriodEndDate not a valid date reported in ddmmyyyy format" }

### 3.5 Data submission workflow notifications

- Once member banks have successfully completed their submission via the API and receive a data submission ID, the CODI IT Solution will begin processing the data submission files.

- Member bank 'maker users' will be informed via email that their data submission file is being processed and that the respective data submission file's metadata and business data validations are being performed.
- The member bank 'maker users' will receive email confirmation once the metadata and business data validations are complete and a notification if any errors were encountered.
- The standard data submission approval processes will apply from this point forward.

Figure 5 illustrates the standard data submission and approval workflow.

**Figure 5: Standard workflow for data submission and approval**



### 3.6 Replacement file submission

Member banks can submit replacement files if a data file is incorrect. Replacement file submissions are allowed when:

- no data submission has been *previously concluded* for that reporting period (i.e. the previous member bank data submission has been approved by **both** the member bank approver ('checker user') and CODI data manager) (if this has occurred an official resubmission request must be logged); and
- the data submission is still within an approval workflow for that reporting period.

If there is a data submission still within an approval workflow, and it is in the workflow submission and approval step where it still



requires member bank approval, the member bank 'checker user' will be required to reject the data submission before the member bank will be allowed to submit a replacement file.

If the data submission is in the approval step where it still requires CODI data approval, then the member bank is required to log a support request at [CODISupport@resbank.co.za](mailto:CODISupport@resbank.co.za) requesting that the data submission be rejected.

Once the data submission has been rejected, this will effectively reset the data submission process and member banks can upload the corrected replacement file.

### **3.7 API technical configuration (YAML file)**

To assist the member bank's development teams with their API configuration, a **.YML** file has been provided that defines the technical configuration details for accessing and submitting data using the following API's:

- Authentication API
- Data submission API

The file API-Channel-Submission-Specification-V2.0.yaml can be downloaded from the CODI IT Solution (downloads menu).

### **3.8 API Integration support**

Should there be any queries regarding configuring or testing the API-based integration to CODI, member banks can log a support query at [CODISupport@resbank.co.za](mailto:CODISupport@resbank.co.za).