TIGRs Transactional API Specification

Version 2.0



Table of Contents

[1. Revision History 1](#_Toc115858438)

[2. Summary 1](#_Toc115858439)

[3. Creating API Access 1](#_Toc115858440)

[4. General API Information 1](#_Toc115858441)

4.1. [Authentication 1](#_Toc115858442)

4.1.1. [Parameters 1](#_Toc115858443)

4.1.2. [Detailed information on how to pass parameters to an OAuth2 authentication endpoint are available at this link: OAuth2 Authentication 1](#_Toc115858444)

4.1.3. [Results 1](#_Toc115858445)

4.1.4 [Status Codes 1](#_Toc115858446)

4.2. [Get / Post Security 1](#_Toc115858447)

4.2.1. [Headers 1](#_Toc115858448)

4.3. [Reference Data 1](#_Toc115858449)

4.3.1. [User Acceptance Testing (UAT) and Production Base URLs 1](#_Toc115858450)

[5. Swagger Technical API Specification 1](#_Toc115858451)

5.1. [View TIGRs APIs on Swagger 1](#_Toc115858452)

5.2. [Generate Client for TIGRs APIs 1](#_Toc115858453)

5.3. [Generate Server for TIGRs APIs 1](#_Toc115858454)

[6. Get APIs 1](#_Toc115858455)

6.1. [General GET API Behavior 1](#_Toc115858456)

6.1.1. [Response Codes 1](#_Toc115858457)

6.1.2. [Error Results 1](#_Toc115858458)

6.2. [Get certificate holdings (ledger/holding) 1](#_Toc115858459)

6.2.1. [Parameters 1](#_Toc115858460)

6.2.2. [Success Results 1](#_Toc115858461)

6.3. [Get Corporate Entity 1](#_Toc115858462)

6.3.1. [Parameters 1](#_Toc115858463)

6.3.2. [Success Results 1](#_Toc115858464)

6.4. [Get System Ping (system/ping) 1](#_Toc115858465)

6.4.1. [Parameters 1](#_Toc115858466)

6.4.2. [Success Results 1](#_Toc115858467)

[7. Post APIs 1](#_Toc115858468)

7.1. [General POST API Behavior 1](#_Toc115858469)

7.1.1. [Response Codes 1](#_Toc115858470)

7.1.2. [Error Results 1](#_Toc115858471)

7.2. [Post Create New Sub-account (subaccount/subaccountType) 1](#_Toc115858472)

7.2.1. [Parameters 1](#_Toc115858473)

7.2.2. [Post Contents 1](#_Toc115858474)

7.2.3. [Success Results 1](#_Toc115858475)

7.3. [Post Certificate Retirement (ledger/retire) 1](#_Toc115858476)

7.3.1. [Parameters 1](#_Toc115858477)

7.3.2. [Post Contents 1](#_Toc115858478)

7.3.3. [Success Results 1](#_Toc115858479)

# Revision History

|  |  |  |
| --- | --- | --- |
| Revision | Date | Description |
| 1.0 | April 11, 2022 | * Initial Version
 |
| 2.0 | August 24, 2022 | * Draft 2
 |

# Summary

This document describes the Application Programming Interface (API) available to TIGRs users, including functionality of the individual endpoints comprising the API. This document also explains how to gain access and integrate with the API.

# Creating API Access

Each account holder must submit to TIGRs Administrator the IP addresses for the machines from which the API will be accessed. The IP addresses must be reviewed, approved, and configured within the API by the TIGRs Administrator before the machines are able to connect the API. IP addresses need to be submitted to TIGRs Administrator via email: tigrs@apx.com

# General API Information

## Authentication

Client API consumers will authenticate against an OAuth2 endpoint exposed by the API (see endpoint URLs below). The OAuth2 endpoint acts as the authorization server for your client and will provide the granted credentials for access to the API Endpoints. This Authorization API POST request will return a short-lived JSON Web Token that will be provided in calls to the application endpoints exposed by the API.

 **Production Authentication Endpoint:** https://apxjwtauthprod.apx.com/oauth/token

**UAT Authentication Endpoint:** <https://apxjwtauthuat.apx.com/oauth/token>

In order to authenticate and obtain the JSON Web Token, API consumer needs to present two sets of credentials:

* Client ID and Client Secret
	+ To be obtained from TIGRs Administrator
* Username and Password
	+ Account holder to request TIGRs Administrator to create username and password for accessing the API. Note, multiple username/password combinations can be created for one account holder, if so desired.
* Please see section 4.1.1 below for technical details of how the authentication is to be performed

### Parameters

|  |  |
| --- | --- |
| **Field** | **Description** |
| Client ID and Secret | To be passed to the authentication endpoint via HTTP basic access authentication |
| Username and Password | To be passed to the authentication endpoint as part of the HTTP POST request content |
| grant\_type | To be hardcoded to ‘password’ (omitting the single quotes). To be passed to the authentication endpoint as part of the HTTP POST request content. |

### Detailed information on how to pass parameters to an OAuth2 authentication endpoint are available at this link: [OAuth2 Authentication](https://www.rfc-editor.org/rfc/rfc6749#section-4.3.2)

### Results

|  |  |
| --- | --- |
| **Field** | **Description** |
| access\_token | This is the token to be used in the “bearer” value of the HTTP Authorization header |
| token\_type | The type of the token to be used in the API Requests. This value returned will be “Bearer” |
| expires\_in | Duration in which the token will expire and a subsequent authentication request will need to be made if time expires. |

### Status Codes

Attempts to call application endpoints without a valid token will result in an HTTP error message being returned.

|  |  |  |
| --- | --- | --- |
| **HTTP Status Code** | **Status** | **Status Message** |
| 200 | SUCCESS | Successfully authenticated |
| 401 | ERROR | Bad RequestInvalid Client ID, and/or Client Secret, and/or Username, and/or Password |

## Get / Post Security

Present the authorization header below to call into the APIs.

### Headers

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| Authorization | Bearer {access\_token} | The access\_token that is returned from the authentication request will be inserted into the value field. |

## Reference Data

Some of the data sent and received through the API uses codes to represent certain field values. See the set of codes below; to be added.

|  |  |  |
| --- | --- | --- |
| **API Call** | **Reference for:** | **Retirement Reason Code** |
| **Retirement Reason Description** | **Retirement Reason Type Code** |
| GET Certificate Holdings & POST Certificate Retirement | Meet Corporate Renewable Energy Goals | GRN | 1 |
| Meet Carbon Neutrality Goals | GRN | 2 |
| Meet Overall Sustainability Goals | GRN | 3 |
| Meet Corporate Renewable Energy Goals | BBO | 4 |
| Meet Carbon Neutrality Goals | BBO | 5 |
| Meet Overall Sustainability Goals | BBO | 6 |
| Meet Corporate Renewable Energy Goals | OTH | 7 |
| Meet Carbon Neutrality Goals | OTH | 8 |
| Meet Overall Sustainability Goals | OTH | 9 |

|  |  |  |
| --- | --- | --- |
| **API Call** | **Retirement Type Description** | **Retirement Reason Type Code** |
| GET Certificate Holdings & POST Certificate Retirement | Primary Account Retirement | GRN |
| On Behalf of Retail Retirement | BBO |
| On Behalf of Corporate Retirement | OTH |

|  |  |  |
| --- | --- | --- |
| **API Call** | **Reference for Retirement Type Description** | **Fuel Type Code** |
| GET Certificate Holdings | Biogas - Agricultural Methane | BAM |
| Biomass Combustion - Agricultural Products | BAP |
| Biomass Combustion - Agricultural Waste | BAW |
| Biomass Combustion - Animal Waste | BCA |
| Biogas - Animal Waste | BGA |
| Biogas - Wastewater Methane | BGW |
| Biogas - Liquid Biofuels | BLB |
| Biogas - Landfill Methane | BLF |
| Biomass Combustion - Liquid Biofuels | BML |
| Biomass - Animal Waste - Swine Waste, Solid or Gas | BA1 |
| Biomass - Animal Waste - Poultry Waste, Solid or Gas | BA2 |
| Biomass Combustion - Wood Products | BWP |
| Biomass Combustion - Wood Waste | BWW |
| Coal (used for Multi-Fuel Generators; will not create TIGRs) | CO1 |
| Diesel (used for Multi-Fuel Generators; will not create TIGRs) | DI1 |
| Fuel Cells using Hydrogen derived from fossil fuels (used for Multi-Fuel Generators; will not create TIGRs) | FC1 |
| Geothermal Energy | GE1 |
| Hydroelectric Water - Dam/Impoundment | H2O |
| Hydroelectric - Run-of-River | HRR |
| Hydroelectric - Tidal | HTI |
| Hydroelectric - Wave power | HWA |
| Fuel cells using hydrogen derived from renewables | HYD |
| Jet Fuel (used for Multi-Fuel Generators; will not create TIGRs) | JET |
| Municipal solid waste | MS1 |
| Natural Gas (used for Multi-Fuel Generators; will not create TIGRs) | NG1 |
| Oil (used for Multi-Fuel Generators; will not create TIGRs) | OIL |
| Concentrated Solar Power | SCP |
| Solar Photovoltaics | SO1 |
| Solar Serving On-Site Load | SOL |
| Wind | WND |

|  |  |  |
| --- | --- | --- |
| **API Call** | **Reference for Retirement Certificate Type** | **Retirement Certificate Type** |
| GET Certificate Holdings & POST Certificate Retirement | Individual Detailed | SINGLE |

|  |  |  |
| --- | --- | --- |
| **API Call** | **Reference for Sub Account Type** | **Sub Account Type**  |
| POST Create New Sub-Account | Active | ACT |
| Retirement | RET |

## User Acceptance Testing (UAT) and Production Base URLs

To call a specific API, append “api/*method name*” to the following base URLs

User Acceptance Testing (UAT) -> <https://TIGRs-api-uat.apx.com/clientapi/>

Production -> <https://tigrsregistry.apx.com/clientapi/>

# Swagger Technical API Specification

TIGRs uses the Swagger specification to describe the integration API endpoints available to consumers. A swagger.json file is provided (available on registry website; see links below) that contains the definitions of the endpoints that will allow you to become familiar with the requests and responses provided. You can follow the steps outlined below to get started. There is also a Swagger UI page available through the registry website (see links below)

**Swagger UAT page:** [**https://tigrs-app-uat01.apx.com/clientapi/swagger-ui.html**](https://tigrs-app-uat01.apx.com/clientapi/swagger-ui.html)

## View TIGRs APIs on Swagger

1. Go to [Swagger website](http://swagger.io/). This site is the community site that describes the swagger specification and has demonstrations and downloads available
2. Go to the [“demo” area](http://editor.swagger.io/#/). This will take you to a hosted solution where you can view the definitions and generate servers and clients in many mainstream languages. You will need to create the client code for consumption of the endpoints. The server generated code can be used to create stubs to simulate interactions with the live endpoint.
3. Upload the TIGRs swagger file (see links, above) to the swagger editor
	1. Go to “File” menu item
	2. Go to the “Import File” menu item
	3. Navigate to the file location
	4. Upload the file
4. You can now view the TIGRs API definitions in the right-hand pane. (The “Warnings” can be ignored as they are alerting you to a non-standard description field that is generated)

## Generate Client for TIGRs APIs

1. Perform the “View TIGRs APIs” as described above.
2. Select the “Generate Client” menu item.
3. Select your language of choice and download the SDK
	1. This will download an SDK in your language of choice. Please note that you may need to make modifications to the toolkit (e.g., Username / Passwords, Endpoint URL changes, et al).
	2. This can serve as a starting point to setting up your code to consume the APIs.

## Generate Server for TIGRs APIs

1. Perform the “View TIGRs APIs” as described above.
2. Select the “Generate Server” menu item.
3. Select your language of choice.
	1. This will download an SDK in your language of choice. Please note that you may need to make modifications to the toolkit (e.g., Username / Passwords, Endpoint URL changes, et al).
	2. This can serve as a starting point to setting up the server API stubs.

# Get APIs

## General GET API Behavior

* If no data is found for request, empty dataset is returned AND a Success code (HTTP 200).
* Result sets/Responses limited to 9,999 lines. If over 9,999, an error will be returned (HTTP 400).

### Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP Status Code** | **Status** | **Status Message** |
| 200 | SUCCESS |   |
| 400 | ERROR | Invalid parameter(s) |
| 401 | ERROR | Unauthorized access |
| 500 | ERROR | An unexpected error has occurred |

### Error Results

If the request fails validation, then the following structure will be returned. This will correspond to an HTTP Status Code = 400, 401, or 500. That return set includes the following:

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Data Type** |
| errors | Container for errors | N/A |
| errors {corelationId} | NULL for Get methods | String |
| errors {message} | String corresponding to the specific error encountered for the corresponding Row ID.  | String |
| errors {parameterName} | The field name of the parameter that failed | String |
| operationId | GUID associated with system logging. Provide when requiring APX support. | String |

## Get certificate holdings (ledger/holding)

A request to retrieve the list of all certificates held by the API user. This will return certificates belonging to the API user that are in active and retired states.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| accountIds | Account ids: if specified, limits results to the items related to the accounts whose ids match the provided set; otherwise, all results related to any authorized account will be returned | array[string] |
| vintageStart | Vintage start: results will be those that intersect with the start and end range. Example: If the start is 2021-01-01 and the end is 2021-12-31, all holdings with a vintage period within the range inclusive of the start and end, will be returned.  | string($date-time) |
| vintageEnd | Vintage end: results will be those that intersect with the start and end range. Example: If the start is 2021-01-01 and the end is 2021-12-31, all holdings with a vintage period within the range inclusive of the start and end, will be returned. | string($date-time)  |

### Success Results

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Data Type** |
| accountId | The account identifier | String |
| beneficialOwner | The beneficial owner or who the certificate is retired on behalf of | String |
| certificateSerialNumberRange | The certificate serial number range | String |
| certificateStatus | The certificate status | String |
| countryIso2Code | The country to which resource belongs as an ISO-2 code | String |
| fuelTypeCode | See enumerated list. Example: BAM | String |
| fuelTypeDescription | See enumerated list. Example: Biogas - Agricultural Methane | String |
| quantity | The holding quantity | Number |
| resourceName | The asset name | String |
| resourceProgramAssignedIdentifier | The asset identifier | String |
| retirementDate | Date of retirement | String |
| retirementDetails | Information regarding retirement activity | String |
| retirementReasonCode | See enumerated list. Example: 1 | String |
| retirementReasonDescription | See enumerated list. Example: Meet Corporate Renewable Energy Goals | String |
| retirementTypeCode | See enumerated list. Example: GRN | String |
| retirementTypeDescription | See enumerated list. Example: Primary Account Retirement | String |
| subaccountId | The subaccount identifier | String |
| subaccountName | The subaccount name | String |
| vintageEnd | The vintage end | String($date-time) |
| vintageStart | The vintage start | String($date-time) |

## Get Corporate Entity

A request to retrieve the list of beneficial owners or corporate entities. This will return corporate entity name identifier needed for the POST Retire endpoint.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| accountIds | Account ids: if specified, limits results to the items related to the accounts whose ids match the provided set; otherwise, all results related to any authorized account will be returned. | array[string] |

### Success Results

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Data Type** |
| accountId | The account identifier | String |
| corporateEntityId | The identifier of the beneficial owner or corporate entity for whom the certificate is retired on behalf of. Pass in the POST Retire, onBehalfOfCorporateEntityId variable. | String |
| corporateEntityName | The beneficial owner or corporate entity name for whom the certificate is retired on behalf of. | String |
| corporateIdType | The corporate identifier type  | String |
| corporateId | The corporate identifier  | String |

##

## Get System Ping (system/ping)

A call to ping the API connection.

### Parameters

None

### Success Results

|  |  |  |
| --- | --- | --- |
| **Field**  | **Description**  | **Data Type**  |
| pong\* | The current time on the server | string($date-time) |

# Post APIs

## General POST API Behavior

* Posts limited to 250 lines; post over that threshold will be rejected.

### Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP Status Code** | **Status** | **Status Message** |
| 200 | SUCCESS |   |
| 400 | ERROR | Invalid parameter(s) |
| 401 | ERROR | Unauthorized access |
| 413 | ERROR | Request entity too large |
| 500 | ERROR | An unexpected error has occurred |

### Error Results

If the request fails request validation, then the following structure will be returned. This will correspond to an HTTP Status Code = 400, 401, 413, or 500. That return set includes the following:

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Data Type** |
| errors | Container for errors | N/A |
| errors {corelationId} | The row number corresponding to the error message  | String |
| errors {message} | String corresponding to the specific error encountered for the corresponding correlation ID.  | String |
| errors {parameterName} | The field name of the parameter that failed | String |
| operationId | GUID associated with system logging. Provide when requiring APX support. | String |

## Post Create New Sub-account (subaccount/subaccountType)

This API is for creating new active and retirement sub-accounts within a given account.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| onBehalfOfAccountId | Account id for which the request is being made; defaults to the primary account id of the requester unless specified | String |
| subAccountType\* | Subaccount type is active (ACT) or retirement (RET). Pass either ACT or RET. | String |

### Post Contents

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| additionalDetails | Information on the subaccount | String |
| alias | The subaccount alias | String  |
| correlationId\* | Correlation identifier: an opaque value that will be returned in the responses and error messages to correlate the response or error with the corresponding request | String |
| name\* | The subaccount name | String |

### Success Results

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| additionalDetails | Information on the subaccount | String |
| alias | The subaccount alias | String  |
| correlationId | Correlation identifier: an opaque value that will be returned in the responses and error messages to correlate the response or error with the corresponding request | String |
| name | The subaccount name | String |
| subaccountId | The subaccount identifier | String |

## Post Certificate Retirement (ledger/retire)

This API is for retiring certificates within an account.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter**  | **Description**  | **Data Type**  |
| onBehalfOfAccountId | Account id for which the request is being made; defaults to the primary account id of the requester unless specified | String |

### Post Contents

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| certificateSerialNumberRange\* | The certificate serial number range: certificate ranges as returned by the holdings call, may not be combined with other ranges, even if contiguous | String  |
| correlationId\* | Correlation identifier: an opaque value that will be returned in the responses and error messages to correlate the response or error with the corresponding request | String |
| details | Information regarding retirement activity | String |
| emailNotification\* | The e-mail address for notification | String  |
| onBehalfOfCorporateEntityId\* | The identifier associated to the corporate entity or beneficial owner for whom the certificate is retired on behalf of; this is for “on behalf of Corporate Retirement”.  | String  |
| quantity\* | The quantity to transfer | Number |
| retirementCertificateType\* | Only option would be SINGLE, which represents "Individual Detailed" | String |
| reasonCode\* | See enumerated list. Example: Meet Corporate Renewable Energy Goals | String |
| retirementTypeCode\* | See enumerated list. Example: GRN | String |
| subaccountId\* | The subaccount identifier | String |

### Success Results

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Data Type** |
| certificateSerialNumberRange | The certificate serial number range: certificate ranges as returned by the holdings call, may not be combined with other ranges, even if contiguous | String  |
| correlationId | Correlation identifier: an opaque value that will be returned in the responses and error messages to correlate the response or error with the corresponding request | String |
| details | Information regarding the retirement activity | String |
| emailNotification | The e-mail address for notification | String  |
| onBehalfOfCorporateEntityId | The identifier associated to the corporate entity or beneficial owner for whom the certificate is retired on behalf of | String  |
| quantity | The quantity to transfer | Number |
| retirementCertificateType | Only option would be SINGLE, which represents "Individual Detailed" | String |
| reasonCode | See enumerated list. Example: Meet Corporate Renewable Energy Goals | String |
| retirementTypeCode | See enumerated list. Example: GRN | String |
| subaccountId | The subaccount identifier | String |
| transferIdentifier | Transaction or batch identifier | String  |