APX MarketSuite® REST API Technical Interface Specification

Version 1.0



Table of Contents

[1. Revision History 3](#_Toc26320406)

[2. Introduction 4](#_Toc26320407)

[2.1 Regional Variations 4](#_Toc26320408)

[2.2 General API Structure and Overview 4](#_Toc26320409)

[2.2.1 Submission of pre-market ISO Scheduling data 4](#_Toc26320410)

[*2.2.2 Upload File* 4](#_Toc26320411)

[*2.2.3 Get Status* 5](#_Toc26320412)

[*2.2.4 Get Schedule Data* 5](#_Toc26320413)

[2.2.5 Data Access 5](#_Toc26320414)

[*File List Since* 5](#_Toc26320415)

[*File Download* 5](#_Toc26320416)

[*Report Download* 5](#_Toc26320417)

[2.2.6 Working Document 5](#_Toc26320418)

[3. APX REST API Integration 6](#_Toc26320419)

[3.1 Creating an API Login 6](#_Toc26320420)

[3.2 Swagger Technical API Specification 6](#_Toc26320421)

[3.3 View MarketSuite® APIs on Swagger 6](#_Toc26320422)

[3.4 Generate Client for MarketSuite® APIs 6](#_Toc26320423)

[3.5 Generate Server for MarketSuite® APIs 7](#_Toc26320424)

[3.6 Authentication Protocol 7](#_Toc26320425)

[3.6.1 Headers 7](#_Toc26320426)

[3.6.1 Parameters 7](#_Toc26320427)

[3.6.2 Results 8](#_Toc26320428)

[3.6.3 Status Codes 8](#_Toc26320429)

[3.7 Get / Post Security 8](#_Toc26320430)

[3.7.1 Headers 8](#_Toc26320431)

[4. REST API Endpoints 9](#_Toc26320432)

[4.1 Upload File 9](#_Toc26320433)

[4.1.1 Parameters 9](#_Toc26320434)

[4.1.2 Return Value 10](#_Toc26320435)

[4.2 GetStatus 10](#_Toc26320436)

[4.2.1 Parameters 10](#_Toc26320437)

[4.2.2 Return Value 10](#_Toc26320438)

[4.3 GetScheduleData 10](#_Toc26320439)

[4.3.1 Parameters 11](#_Toc26320440)

[4.3.2 Return Value 11](#_Toc26320441)

[4.4 File List 12](#_Toc26320442)

[4.4.1 Parameters 12](#_Toc26320443)

[4.4.2 Return Value 12](#_Toc26320444)

[4.5 File List Since 13](#_Toc26320445)

[4.5.1 Parameters 13](#_Toc26320446)

[4.5.2 Return Value 13](#_Toc26320447)

[4.6 File Download 14](#_Toc26320448)

[4.6.1 Parameters 14](#_Toc26320449)

[4.6.2 Return Value 14](#_Toc26320450)

[4.7 Report Download 14](#_Toc26320451)

[4.7.1 Parameters 15](#_Toc26320452)

[4.7.2 Return Value 15](#_Toc26320453)

[5. Appendix C – Report Download Paths and Keys 16](#_Toc26320454)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Description | Updated By |
| 0.1 | 12/3/2019 | * Initial Version
 | Adam Barrett |
| 1.0 | 6/4/2020 | * FileUpload and GetScheduleData requests may contain seven days of data for PJM, ISONE, MISO, SPP, NYISO.
* API username and password are lowercase.
* Detail the valid fileTypes for UploadFile method
* GetScheduleData lastIntervalEnd required for xMarket regions.
 | Adam Barrett |

# Introduction

The purpose of this document is to describe the REST APIs APX provides to support our client’s integration with the APX MarketSuite®.

APX recognizes the need for external entities to automate data integration with the APX MarketSuite®. To help satisfy the needs of integration and interoperability, APX has created several REST interfaces—including the File Upload, GetStatus (for upload exceptions), GetScheduleData, Download File, and Reports—which enable the automation of data uploading and downloading with the APX MarketSuite®.

This document will explain how to create an API Login for your account, the rules governing the use of the API, the performance expectations, and validations that occur on incoming data and request parameters.

This document also contains basic instructions on how to use [Swagger](https://swagger.io/) to get the technical details for each API, including all input parameters and output data set structures.

## Regional Variations

The APX MarketSuite® operates on two schemas for ISO Scheduling: one schema supports CAISO and ERCOT, and the other supports all other regions (PJM, ISONE, MISO, SPP, NYISO). For Settlements, all regions are on the same schema, except ERCOT.

APX is working towards unifying these schemas, but currently with these different schemas comes some minor differences in how these interfaces work. These differences will be explained where they exist under each method. The primary difference is that non-CAISO/ERCOT regions must be scheduled one participant/day at a time, whereas MarketSuite® offers support for multi-day/multi-participant submissions.

## General API Structure and Overview

The MarketSuite® REST APIs are designed to support two primary processes: ISO scheduling submission and data access.

### Submission of pre-market ISO Scheduling data

This includes Bids, Trades, and Resource Parameters and the following methods:

### *Upload File*

Submit pre-market ISO scheduling data to MarketSuite®, with the option to immediately send all valid data to the ISO.

* 1. **CAISO/ERCOT**: All files must contain data for only one ISO participant, for no more than a span of 25 hours, including Bids/Offers, Bilateral Schedules, and Resource Parameters.
	2. **PJM, ISONE, MISO, SPP, NYISO:** Submitted files may contain up to seven days of data for multiple participants within a region.

### *Get Status*

This method returns data associated with a specific upload file:

* 1. **CAISO/ERCOT**: Gets all scheduling data for the specific file submitted, including exception data.
	2. **PJM, ISONE, MISO, SPP, NYISO:** Gets all Bid/Offer exception data for the specific file submitted. (Trade and Resource Parameters exception data not currently available)

### *Get Schedule Data*

This method retrieves all scheduling data in the APX MarketSuite®, per the supplied parameters.

* 1. **CAISO/ERCOT**: Requests must be for one trade data and one participant at a time.
	2. **PJM, ISONE, MISO, SPP, NYISO:** Data may be requested for multiple participants within a region, for a span of up to seven days.

### Data Access

This includes Bids, Trades, and Resource Parameters and the following methods:

### *File List Since*

This method retrieves a list of files available to the account holder.

### *File Download*

This downloads specific file available to the account holder.

### *Report Download*

This method runs MarketSuite® reports and downloads the result set.

### Working Document

This is a working document that includes specifications for functionality that may not yet be implemented. As of this version the following is a summary of the items that are currently not implemented:

1. For f. PJM, ISONE, MISO, SPP, NYISO, GetStatus method will not return exception data for BilateralSchedules or ResourceParameters.

# APX REST API Integration

## Creating an API Login

MarketSuite® Accounts will have dedicated logins for accessing the REST APIs. Regular user logins will not be able to authenticate through the API. Contact your APX Account Manager if you need a login.

## Swagger Technical API Specification

MarketSuite® uses the Swagger specification to describe the integration API endpoints available to consumers. Provided is a swagger.json file (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 page available through the registry website (see links below)

Swagger.json file: Prod | [UAT](https://pmfileapiuat.apx.com/swagger-ui.html)

Swagger UI Page: Prod | [UAT](https://pmfileapiuat.apx.com/swagger-ui.html)

## View MarketSuite® 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 MarketSuite® Swagger API json 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 GIS 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 MarketSuite® APIs

1. Perform the “View MarketSuite® 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 MarketSuite® APIs.

## Generate Server for MarketSuite® APIs

1. Perform the “View MarketSuite® 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 APIs stub.

## Authentication Protocol

Client API consumers will authenticate against an OAuth2 endpoint exposed by the client 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>

### Headers

|  |  |  |
| --- | --- | --- |
| Key | Value | Description |
| Authorization | Basic {client:secret} | The client:secret for the token request. This is Base64-encoded as standard for HTTP Basic Authentication. This value authorizes your software to access APX resources. |
| Content-Type | application/x-www-form-urlencoded | Indicates that the POST-ed content is URL-encoded |

### Parameters

These parameters must be POST-ed to the endpoint above as URL-encoded form data.

|  |  |
| --- | --- |
| Field | Description |
| username | Client API Service User Name |
| password | Client API Service Password |
| grant\_type | Value: password[This is associated with the OAuth2 password credentials scenario](https://tools.ietf.org/html/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 in subsequent requests. |
| token\_type | The type of the token to be used in the API Requests. The value returned will be “bearer” |
| expires\_in | Duration (in seconds) in which the token will expire and a subsequent authentication request will need to be made if time expires. |
| scope | The scope of the granted access.The value returned will be "access" |

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

### Status Codes

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Status | Status Message |
| 200 | SUCCESS | Successfully authenticated |
| 401 | ERROR | Bad RequestInvalid Login ID or Password |

## Get / Post Security

In order to successfully call into the APIs the below authorization header will need to be presented.

### 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. |

# REST API Endpoints

## Host Environment URLs

Here are the host environment URLs for the MarketSuite® REST APIs:

UAT: <https://pmfileapiuat.apx.com>

Production: <https://pmfileapi.apx.com>

The authentication tokens for the above environments must come from the corresponding UAT/Prod authentication server.

## Upload File

The Upload File Request is designed to support uploading files to the APX MarketSuite®, primarily files for ISO scheduling data. The response is a string representing the file unique identifier (GUID). This identifier can be used to retrieve the file using the Get File method. This identifier can also be used to retrieve the status of all data submitted (for CAISO/ERCOT submissions) or the exception data (other regions) using the GetStatus method.

If the file uploaded is the ISO Scheduling Data xml file, the method does XSD validation of the file and will reject any XML file submissions which are not valid. The ISO Scheduling Data CSV file is also validated for format. The returned error message will include the validation details.

The maximum request size for files uploaded through this Web service is 10MB.

**Endpoint (POST):** /fileRegistry/file

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| location | ISO code. Values include PJM, ISONE, MISO, SPP, NYISO, MRTU (used for CAISO), TX (used for ERCOT). |
| participant | MarketSuite® participant API code. Often, this is the ISO Scheduling code, preceded by some relevant text to make it unique. Please confirm the correct code with your APX Account manager. |
| intervalBegin | DateTime value marking the start of the interval for the file. The value is in XML DateTime format. This is the date that will be associated with the file when it is registered. The time component should always be 00:00, e.g. 2019-11-18T00:00:00 |
| intervalEnd | DateTime value marking the end of the interval for the file. This should be the interval\_begin + one day. Note that the data in the file may extend past this date.  |
| fileType | Type of file being uploaded. Valid fileTypes for the regions **PJM, ISONE, MISO, SPP, NYISO** are:* APX\_SchedulingData\_CSV (See CSV Spec for format details)
* APX\_SchedulingData\_XML

Valid fileTypes for the regions **TX (ERCOT) and MRTU (CAISO)** are:* APX\_NonXmkrt\_SchedulingData\_CSV (See CSV Spec for format details)
* APX\_Bids\_and\_Offers (See XML Spec for format details)
 |
| source | Name of the source. For ISO Scheduling data submissions, use ‘Client’ |
| rendition | Some file types have more than one rendition. Those are named sets that look identical but represent different sets. For example: “DA” and ”RT” or “Initial”, ”recalc”, and “final”. For ISO Scheduling data submissions, use ‘DAM’ or ‘RTM’. |
| file | Byte array representing the compressed data file. |

### Return Value

The return value is a string representing the file unique identifier. This identifier can be used to retrieve the file using the GetFile method and to retrieve the ISO scheduling data exception details of the file using the GetStatus method.

## GetStatus

The GetStatus Request allows the user to retrieve the exception details of scheduling data uploaded to the APX MarketSuite®. A unique identifier called the File handle is returned from the Upload File Request. This unique identifier is used in the GetStatus Request to retrieve the exception details of the specified file data.

Each exception is returned in the results. Some individual rows may generate more than one exception, in which case one row per exception is returned.

The response of the GetStatus Request is in the form of a compressed XML file.

**For CAISO/ERCOT,** the returned XML complies with the schema definition SchedulingData.xsd. The full specification appears in the documents “APX ISO Scheduling API specification.”

**For PJM/ISONE/SPP/MISO/NYISO,** the returned XML complies with the schema definition ScheduleData.xsd. The full specification appears in the documents “APX ISO Scheduling API specification - xMarket”. Note that in the future other schemas will be returned, depending on the type of file uploaded.

**Endpoint (GET):** /fileRegistry/file/{fileHandle}/status

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| fileHandle | Unique identifier of the file for which status is being requested. |

### Return Value

A byte array containing the compressed data. The format of the file will be the same as the file that was uploaded: requesting the status of an xml file upload will return results in xml, and a CSV upload will have results in CSV.

## GetScheduleData

The GetScheduleData request retrieves schedule data from the APX MarketSuite®. From many aspects it can be thought of as the API version of the Checkout application. The requested data sets may include trades, bid-offer or resource parameters, pre-market (APX image or ISO image) or post-market (awards and final schedules and trades).

The data may be requested in either XML or CSV format, and in either case will be in the same format as for uploads, with some additional info for status of the data.

The response of the GetScheduleData Request is in the form of a compressed XML file.

**For CAISO/ERCOT,** the returned XML complies with the schema definition SchedulingData.xsd. The full specification appears in the documents “APX ISO Scheduling API specification.”

**For PJM/ISONE/SPP/MISO/NYISO,** the returned XML complies with the schema definition ScheduleData.xsd. The full specification appears in the documents “APX ISO Scheduling API specification - xMarket”. Note that in the future other schemas will be returned, depending on the type of file uploaded.

**Endpoint (PUT):** /fileRegistry/getScheduleData/{fileName}

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| fileName (in endpoint) | The name of the file that will be created in the zip archive that is sent back as the return parameter. Clients can specify the file name and then use it to extract the contents of the zip archive. |
| region | ISO code. Values include PJM, ISONE, MISO, SPP, NYISO, MRTU (used for CAISO), TX (used for ERCOT). |
| marketParticipants | List of MarketSuite® participant codes for which the method should return data. Often, this is the ISO Scheduling code, preceded by some relevant text to make it unique. Please confirm the correct code with your APX Account manager. |
| marketStage | Market stage of the requested data |
| marketStatus | ‘Pre’ or ‘Post’—pre-market data with its status, or post-market awards. **NOTE: Post-market awards only available for CAISO(MRTU) and ERCOT (TX). Post-market awards for other regions coming soon.** |
| firstIntervalBegin | Required parameter that specifies the starting date for which data is being requested. |
| lastIntervalEnd | For PJM, ISONE, SPP, MISO, and NYISO required parameter that specifies the ending date for which data is being requested. If not given, default is tradeDate+1 day. **Note: Not allowed for CAISO(MRTU) or ERCOT(TX), where each request may only be for one day.** |
| applications | ‘BidsOffers’, ‘BilateralSchedules’, and/or ‘ResourceParameters’—applications for which data is being requested. **Note: For PJM, ISONE, SPP, MISO, and NYISO, request is limited to ‘BilateralSchedule’ OR BidsOffers/ResourceParameters.** |
| format | CSV or XML. XML will comply with the schema definition SchedulingData.xsd. CSV will be same format used for upload. |

### Return Value

A byte array containing the compressed data file.

## File List

The File List Request is designed to retrieve a list files which can be downloaded from the APX MarketSuite®. The response returns an array of file\_list\_items.

**Endpoint (PUT):** /fileRegistry/listFiles

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| region | ISO code. Values include PJM, ISONE, MISO, SPP, NYISO, MRTU (used for CAISO), TX (used for ERCOT). |
| participant | MarketSuite® participant API code. Often, this is the ISO Scheduling code, preceded by some relevant text to make it unique. Please confirm the correct code with your APX Account manager. |
| fileType | Type of file included in the list. |
| source | Data source of the files to search for. |
| rendition | Some file types have more than one rendition. Those are named sets that look identical but represent different sets. For example: “DA” and ”RT” or “Initial”, ”recalc”, and “final”. |
| latestVersionOnly | Flag indicating whether or not to list the most recent version of a file. True means list the most recent version only. False means list all versions. The default is true. |

### Return Value

A json array of file\_list\_items. file\_list\_items include the following properties:

|  |  |
| --- | --- |
| Field | Description |
| location | The name of the ISO or more granular location designation if one was specified when the file was registered. |
| participant | The name of the participant associated with this file |
| intervalBegin | DateTime value of interval\_begin that was specified when the file was uploaded or otherwise registered. This is not necessarily true for the content of the file. |
| intervalEnd | Date time value of interval\_end that was specified when the file was uploaded or otherwise registered. This is not necessarily true for the content of the file. |
| source | Data source of the files to search for. |
| rendition | Some file types have more than one rendition. Those are named sets that look identical but represent different sets. For example: “DA” and ”RT” or “Initial”, ”recalc”, and “final”. |
| fileType | Type of file included in the list. |
| fileVersion | The version of the file. |
| fileStructureVersion | The version of the structure of the file |
| fileFormat | The format of the file (CSV, XML, MDEF, etc.) |
| fileHandle | The handle of the file used to retrieve the file in the GetFile method |
| createDate | The create date of the file |

## File List Since

The File List Since Request is designed to retrieve a list files which can be downloaded from the APX MarketSuite® which have changed since the specified date. The response returns an array of file\_list\_items.

**Endpoint (PUT):** /fileRegistry/listFilesSince

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| region | ISO code. Values include PJM, ISONE, MISO, SPP, NYISO, MRTU (used for CAISO), TX (used for ERCOT). |
| participant | MarketSuite® participant API code. Often, this is the ISO Scheduling code, preceded by some relevant text to make it unique. Please confirm the correct code with your APX Account manager. |
| fileType | Type of file included in the list. |
| source | Data source of the files to search for. |
| rendition | Some file types have more than one rendition. Those are named sets that look identical but represent different sets. For example: “DA” and ”RT” or “Initial”, ”recalc”, and “final”. |
| latestVersionOnly | Flag indicating whether or not to list the most recent version of a file. True means list the most recent version only. False means list all versions. The default is true. |
| watermarkDate | DateTime value indicating the last date/time that files were fetched, results in the new list including only files that have changed since that date/time (the creation date is greater than and not equal to the watermark date). The value is in XML DateTime format which includes the GMT offset. |
| newWatermarkDate | Output DateTime value to use as the watermark\_date in the next GetListSince call. The value is the maximum create\_date of the files returned in the list. The value is in XML DateTime format which includes the GMT offset. |

### Return Value

A json array of file\_list\_items. file\_list\_items include the following properties:

|  |  |
| --- | --- |
| Field | Description |
| location | The name of the ISO or more granular location designation if one was specified when the file was registered. |
| participant | The name of the participant associated with this file |
| intervalBegin | DateTime value of interval\_begin that was specified when the file was uploaded or otherwise registered. This is not necessarily true for the content of the file. |
| intervalEnd | Date time value of interval\_end that was specified when the file was uploaded or otherwise registered. This is not necessarily true for the content of the file. |
| source | Data source of the files to search for. |
| rendition | Some file types have more than one rendition. Those are named sets that look identical but represent different sets. For example: “DA” and ”RT” or “Initial”, ”recalc”, and “final”. |
| fileType | Type of file included in the list. |
| fileVersion | The version of the file. |
| fileStructureVersion | The version of the structure of the file |
| fileFormat | The format of the file (CSV, XML, MDEF, etc.) |
| fileHandle | The handle of the file used to retrieve the file in the GetFile method |
| createDate | The create date of the file |

## File Download

The File Download Request is designed to retrieve a specific file from the APX MarketSuite®. The file is returned in a zip file byte array.

**Endpoint (GET):** /fileRegistry/file/{fileHandle}

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| fileHandle (in endpoint) | Unique string used to identify the file in the APX MarketSuite®. |
| file\_name | Output parameter, the name of the file inside the compressed byte array. |

### Return Value

A byte array containing the compressed data file.

## Report Download

The Report Download Request is designed to retrieve data from the APX MarketSuite® in the specified format. The data file is returned in a zip file byte array.

**Endpoint (PUT):** /reporting/getReport

### Parameters

|  |  |
| --- | --- |
| Field | Description |
| reportPath | The report-specific path for the report to run. See Appendix C for reports and paths. |
| [key/value pairs] | Parameter name/value pairs required to run the report. These are the report-specific parameters that will be unique for each report. See Appendix C for report-specific parameters (keys). |
| reportFormat | Output file format – CSV, Excel, and XML. |

### Return Value

A byte array containing the compressed data file.

# Appendix C – Report Download Paths and Keys

TBD.