



**UG 27**

**Finance APIs  
User's Guide**

**Issued: May 2026**

**Version: 2.0**

**Effective Date: 05/08/2026**

**Recertified: 05/08/2026**

**Prepared By: Customer Settlements**

New York Independent System Operator  
10 Krey Boulevard  
Rensselaer, NY 12144  
(518) 356-6060  
[www.nyiso.com](http://www.nyiso.com)

**Disclaimer: The information contained within this guide, along with other NYISO guides, is intended to be used for information purposes only, and is subject to change. The NYISO is not responsible for the user's reliance on these publications, or for any erroneous or misleading material.**

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>III</b>
<b>REVISION HISTORY</b> .....	<b>VI</b>
<b>1. IMPORTANT USAGE NOTES AND INTRODUCTION</b> .....	<b>1</b>
1.1.Important Usage Notes .....	1
1.2.. Introduction .....	1
<b>2. USING THE FINANCE APIS</b> .....	<b>3</b>
2.1.Network Protocols .....	3
2.2.Location of Finance APIs Service Endpoints .....	3
2.3.Authentication .....	3
2.4.Data Formatting .....	4
2.4.1. <i>Formatting Numbers</i> .....	4
2.4.2. <i>Formatting Dates and Times</i> .....	4
2.5.Data Availability .....	5
2.6.HTTP Status Codes for Responses.....	6
<b>3. USING THE METERING API</b> .....	<b>7</b>
3.1.Meter Data Submission .....	7
3.1.1. <i>Request</i> .....	7
3.1.2. <i>Request Examples</i> .....	11
3.1.3. <i>Response</i> .....	14
3.1.4. <i>Response Examples</i> .....	22
3.2.Retrieval of Submitted Meter Data .....	28
3.2.1. <i>Request</i> .....	28
3.2.2. <i>Request Examples</i> .....	31
3.2.3. <i>Response</i> .....	31
3.2.4. <i>Response Examples</i> .....	41
3.3.Retrieval of Calculated Subzone Load Summary.....	44
3.3.1. <i>Request</i> .....	44
3.3.2. <i>Request Examples</i> .....	46
3.3.3. <i>Response</i> .....	46
3.3.4. <i>Response Examples</i> .....	50
3.4.Retrieval of Calculated Subzone Load Detail .....	51

3.4.1. Request.....	51
3.4.2. Request Examples.....	53
3.4.3. Response.....	53
3.4.4. Response Examples .....	63
3.5.Transmission Owner Load Submission.....	66
3.5.1. Request.....	66
3.5.2. Request Examples.....	68
3.5.3. Response.....	68
3.5.4. Response Examples .....	73
3.6.Retrieval of Submitted Transmission Owner Load.....	77
3.6.1. Request.....	77
3.6.2. Request Examples.....	79
3.6.3. Response.....	80
3.6.4. Response Examples .....	84
3.7.Retrieval of Transmission Owner Load Verification Summary .....	85
3.7.1. Request.....	85
3.7.2. Request Examples.....	86
3.7.3. Response.....	88
3.7.4. Response Examples .....	94
3.8.Retrieval of Transmission Owner Load Verification Detail .....	98
3.8.1. Request.....	98
3.8.2. Request Examples.....	99
3.8.3. Response.....	101
3.8.4. Response Examples .....	107
3.9.Retrieval of Generator Performance Data.....	111
3.9.1. Request.....	111
3.9.2. Request Examples.....	112
3.9.3. Response.....	113
3.9.4. Response Examples .....	118
3.10.Minimum Oil Burn Event Transmission Owner Data Submission.....	120
3.10.1. Request.....	120
3.10.2. Request Examples.....	123
3.10.3. Response.....	124
3.10.4. Response Examples .....	130
3.11.Minimum Oil Burn Event Generator Data Submission .....	138
3.11.1. Request.....	138
3.11.2. Request Examples.....	141
3.11.3. Response.....	143
3.11.4. Response Examples .....	149
3.12.Retrieval of Minimum Oil Burn Event Data .....	159

3.12.1.	<i>Request</i> .....	159
3.12.2.	<i>Request Examples</i> .....	160
3.12.3.	<i>Response</i> .....	161
3.12.4.	<i>Response Examples</i> .....	167
<b>4.</b>	<b>USING THE SETTLEMENTS API</b> .....	<b>169</b>
4.1.	Retrieval of Station Power Detail Data .....	169
4.1.1.	<i>Request</i> .....	169
4.1.2.	<i>Request Examples</i> .....	170
4.1.3.	<i>Response</i> .....	171
4.1.4.	<i>Response Examples</i> .....	176
<b>5.</b>	<b>USING THE INVOICING API</b> .....	<b>178</b>
5.1.	Retrieval of Daily Reconciliation Dollar Data .....	178
5.1.1.	<i>Request</i> .....	178
5.1.2.	<i>Request Examples</i> .....	180
5.1.3.	<i>Response</i> .....	180
5.1.4.	<i>Response Examples</i> .....	185
5.2.	Retrieval of Daily Reconciliation MWh Data .....	189
5.2.1.	<i>Request</i> .....	189
5.2.2.	<i>Request Examples</i> .....	190
5.2.3.	<i>Response</i> .....	191
5.2.4.	<i>Response Examples</i> .....	195

## Revision History

Version	Effective Date	Revisions
1.0	10/18/2024	Initial Release
2.0	05/08/2026	Recertified Section 5.1.2.1 ➤ Updated example Section 5.1.2.2 ➤ Updated example Section 5.2.2.1 ➤ Updated example Section 5.2.2.2 ➤ Updated example

# 1. Important Usage Notes and Introduction

## 1.1. Important Usage Notes

- The Finance API **must be used** by Meter Authorities and Metering Services Entities (MSEs) when submitting hourly revenue grade metering for all Aggregations.
- The Finance API or SDX Upload/Download **may be used** for the submission of the following data until the anticipated retirement of the SDX Upload/Download application:
  - Hourly revenue-grade metering data for traditional generators, storage resources, tie lines, and subzones.
  - TOL (Transmission Owner Load) data.
  - Minimum Oil Burn Event data for transmission owners and qualified generators.

## 1.2. Introduction

This Guide describes the Finance Application Programming Interfaces (APIs) and how to interact with the services, including:

- Metering API
  - Submission and retrieval of hourly revenue-grade meter data for generators, ties, and subzones.
  - Retrieval of calculated subzone load data, in both summary and detailed form.
  - Submission and retrieval of transmission owner load (TOL) data for load buses.
  - Retrieval of load verification data, in both summary and detailed form.
  - Retrieval of generator performance data.
  - Submission of minimum oil burn event data by both transmission owners and generators, and retrieval of minimum oil burn event data.
- Settlements API
  - Retrieval of station power settlement data.
- Invoicing API
  - Retrieval of daily reconciliation data for both settlements and energy.

This Guide assumes prior knowledge in using the NYISO Market Information System (MIS) upload/download batch procedures. For information on the NYISO MIS, and the relevant authorization and Digital Certificate requirements, please refer to the NYISO *Market Participant User's Guide (MPUG)*, available from the NYISO Web site at <https://www.nyiso.com/documents/20142/3625950/mpug.pdf>. For additional information relating to the upload/download process, please refer to Sections 6 and 8 of the *MPUG*.

Further, this Guide also assumes prior knowledge in using the NYISO Marketplace system to manage administrator accounts, user accounts, and privileges associated with metering. For information on the NYISO Marketplace as it pertains to metering and associated privileges, please refer to Section 1 of the NYISO *Settlement Data Applications User's Guide (SDAUG)*, available from the NYISO Web site at [https://www.nyiso.com/documents/20142/3625950/SDA\\_UG.pdf](https://www.nyiso.com/documents/20142/3625950/SDA_UG.pdf).

## 2. Using the Finance APIs

### 2.1. Network Protocols

All Finance API service endpoints shall be accessed via HTTPS 1.1 over TLS 1.2.

### 2.2. Location of Finance APIs Service Endpoints

All Finance API service endpoints share a common root for their Universal Resource Locators (URLs):

<https://api.nyiso.com/finance/>

Metering API service endpoints share a common root for their URLs:

<https://api.nyiso.com/finance/metering/>

Settlements API service endpoints share a common root for their URLs:

<https://api.nyiso.com/finance/settlements/>

Invoicing API service endpoints share a common root for their URLs:

<https://api.nyiso.com/finance/invoicing/>

The full URL for each service is included in the corresponding documentation below.

Note: Examples provided in the documentation below use the API Market Trial environment URLs, rather than the Production URLs, so that users can safely test mimicking the examples. These URLs all start with this common root:

<https://apitest.nyiso.com/finance/>

### 2.3. Authentication

Two-factor authentication is required to access the Finance APIs.

- A valid MIS user account and associated password must be provided using Basic authentication.
- The NAESB certificate associated with the MIS user account must be provided with each request.

## 2.4. Data Formatting

The bodies of all requests and responses for Finance API endpoints are formatted as JSON.

### 2.4.1. Formatting Numbers

The Finance APIs allow for up to four decimals of precision for all MWh values, and two decimals of precision for all dollar (\$) values.

### 2.4.2. Formatting Dates and Times

The Finance APIs utilizes the ISO-8601 standard for representation of dates and times, including using time zone offsets to account for Eastern Standard Time and Eastern Daylight Time. As such, the following formats shall be used, unless otherwise specified:

- ISO-8601 Month: yyyy-MM — Includes entire period from the start of the first day of the month to the end of the last day of the month
  - "2021-12"
- ISO-8601 Date: yyyy-MM-dd — Includes entire period from the start of day to the end of the day
  - "2021-12-14"
- ISO-8601 Date / Time: yyyy-MM-ddThh:mm:ssX — Specifies a moment in time at the finest grain allowed (i.e. seconds)
  - "2021-12-14T07:00:00Z" in GMT, or "2021-12-14T02:00:00-05:00" in EST
  - "2021-07-14T06:00:00Z" in GMT, or "2021-07-14T02:00:00-04:00" in EDT
  - Note #1: While the Finance APIs shall accept any properly ISO-8601 formatted date / time upon submission, all date / times in responses will be represented in Eastern Time, with the offset represented as above.
  - Note #2: To specify the end of an hour, day, or month, the time must be set to 59 minutes and 59 seconds. For example, the end of the 14-Dec-2021 NYISO service day is expressed as either "2021-12-15T04:59:59Z" in GMT, or "2021-12-14T23:59:59-05:00" in EST.

## **2.5. Data Availability**

Users will be able to download metering, settlements, and invoicing data from the Finance APIs for service months included on invoices issued in the prior three years, which typically would include service months from the previous three years and ten months . For example, for requests made in Sep-2024, data would be available for service months Jan-2021 through Sep-2024.

## 2.6. HTTP Status Codes for Responses

Responses from the Finance APIs utilize the standard HTTP status codes to communicate either the success or type of failure that was the result of a given service call. Status codes used include (but may not be limited to):

- **200 OK:** The request was completed successfully, with no failures or errors. For POST requests (such as Meter Data Submission), this denotes that all submitted data has passed validation and been stored (unless otherwise instructed via submission parameters).
- **400 Bad Request:** The request submitted was not valid and could not be processed. This includes improper JSON that cannot be parsed for POST requests, as well as validation failures of input data for GET requests.
- **401 Unauthorized:** The provided user credentials failed authentication.
- **403 Forbidden:** The user was authenticated but is not authorized for the service requested.
- **422 Unprocessable Content:** The request was well-formed JSON, but the content of the request failed validation with one or more errors. Applies to POST requests.
- **500 Internal Server Error:** An unanticipated error has occurred on the server, preventing the successful completion of the service request.

### 3. Using the Metering API

#### 3.1. Meter Data Submission

Meter Authorities (MAs) can use the Meter Data Submission service in the Metering API to provide revenue-grade meter data for generators, ties, and subzones to the NYISO.

Data may be submitted (including both new entries and updates) in accordance with the Lock-Down Schedules posted on the NYISO website. Upon submission, all data is validated, and if any records fail validation, the entire request will be rejected. All validation errors found will be included with the response.

##### 3.1.1. Request

*URL:* <https://api.nyiso.com/finance/metering/v1/powerMetering>

*HTTP Action:* POST

*HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

*Request Body Parameters:*

Parameter <i>[Required / Optional?]</i>	Data Type	Description
submissionParameters <i>[Optional]</i>	Array of Submission Parameters	User-provided parameters for the submission

Parameter <i>[Required / Optional?]</i>	Data Type	Description
<code>submissionParameters.userRequestId</code> <i>[Optional]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Optional]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Optional]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to true if not provided.
<code>generators</code> <i>[Optional]</i>	Array of Generator objects	Meter data records for generators. If no data records for generators are being submitted, this parameter should either be omitted, or included as either null or an empty array.
<code>generators.genPtId</code> <i>[Required for each generator meter data record]</i>	Integer	Unique NYISO-defined point identifier for a given generator
<code>generators.dateHour</code> <i>[Required for each generator meter data record]</i>	ISO-8601 Date / Time	Service hour for meter data

<b>Parameter [Required / Optional?]</b>	<b>Data Type</b>	<b>Description</b>
generators.meterInjectionEnergyMwh <i>[Required for injection-capable generators; prohibited otherwise]</i>	Number, up to four decimal places $0.0000 \leq x < 10,000.0000$	Metered injections for the given generator and service hour
generators.meterWithdrawalEnergyMwh <i>[Required for withdrawal-capable generators; prohibited otherwise]</i>	Number, up to four decimal places $-10,000.0000 < x \leq 0.0000$	Metered withdrawals for the given generator and service hour
generators.meterDemandReductionMwh <i>[Required for demand reduction-capable generators; prohibited otherwise]</i>	Number, up to four decimal places $0.0000 \leq x < 10,000.0000$	Metered demand reduction for the given generator and service hour
ties <i>[Optional]</i>	Array of Tie objects	Meter data records for ties. If no data records for ties are being submitted, this parameter should either be omitted, or included as either null or an empty array.
ties.tiePtId <i>[Required for each tie meter data record]</i>	Integer	Unique NYISO-defined point identifier for a given tie
ties.dateHour <i>[Required for each tie meter data record]</i>	ISO-8601 Date / Time	Service hour for meter data

Parameter <i>[Required / Optional?]</i>	Data Type	Description
ties.meterTieFlowMwh <i>[Required for each tie meter data record]</i>	Number, up to four decimal places $-10,000.0000 < x < 10,000.0000$	Metered tie flow for the given tie
subzones <i>[Optional]</i>	Array of Subzone objects	Meter data records for subzones. If no data records for subzones are being submitted, this parameter should either be omitted, or included as either null or an empty array.
subzones.subzonePtId <i>[Required for each subzone meter data record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
subzones.dateHour <i>[Required for each subzone meter data record]</i>	ISO-8601 Date / Time	Service hour for meter data
subzones.meterSubzoneLoadMwh <i>[Required for each subzone meter data record]</i>	Number, up to four decimal places $0.0000 \leq x < 100,000.0000$	Metered subzone load for the given subzone and service hour

### 3.1.2. Request Examples

#### 3.1.2.1. Meter Data Submission Request Example #1

- One generator (with all channels), one tie, and one subzone submitted
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/powerMetering>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "generators": [
    {
      "genPtid": 345678,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterInjectionEnergyMwh": 75.1234,
      "meterWithdrawalEnergyMwh": -12.3456,
      "meterDemandReductionMwh": 5.6789
    }
  ],
  "ties": [
    {
      "tiePtid": 222222,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterTieFlowMwh": 33.3333
    }
  ],
  "subzones": [
```

```
{  
  "subzonePtid": 24680,  
  "dateHour": "2021-12-14T02:00:00-05:00",  
  "meterSubzoneLoadMwh": 246.7531  
}  
]  
}
```

### 3.1.2.2. Meter Data Submission Request Example #2

- One generator (with all channels), no ties, and no subzones submitted
  - Ties parameter included as empty array
  - Subzones parameter omitted
  - Either approach is acceptable, as well as setting the parameter to null, instead of an empty array ([])
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/powerMetering>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "generators": [
    {
      "genPtid": 345678,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterInjectionEnergyMwh": 75.1234,
      "meterWithdrawalEnergyMwh": -12.3456,
      "meterDemandReductionMwh": 5.6789
    }
  ],
  "ties": []
}
```

### 3.1.3. Response

HTTP Headers:

Content-Type: application/json

Response Body Parameters:

Parameter <i>[Provided...?]</i>	Data Type	Description
submissionParameters <i>[Always]</i>	Submission Parameters object	User-provided parameters for the submission
submissionParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
submissionParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
submissionParameters.includeAcceptedDataInResponse <i>[Always]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
submissionParameters.doCommit <i>[Always]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to false if not provided.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
requestSummary <i>[Always]</i>	Summary Object	Processing summary for the request
requestSummary.generators <i>[Always]</i>	Generator Summary object	Processing summary for generators submitted in the request
requestSummary.generators.submitted <i>[Always]</i>	Integer	Number of meter data records submitted for generators in the request
requestSummary.generators.passedValidation <i>[Always]</i>	Integer	Number of meter data records that passed validation for generators in the request
requestSummary.generators.failedValidation <i>[Always]</i>	Integer	Number of meter data records that failed validation for generators in the request
requestSummary.generators.accepted <i>[Always]</i>	Integer	Number of meter data records accepted for generators in the request

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
requestSummary.generators.rejected <i>[Always]</i>	Integer	Number of meter data records rejected for generators in the request
requestSummary.ties <i>[Always]</i>	Tie Summary object	Processing summary for ties submitted in the request
requestSummary.ties.submitted <i>[Always]</i>	Integer	Number of meter data records submitted for ties in the request
requestSummary.ties.passedValidation <i>[Always]</i>	Integer	Number of meter data records that passed validation for ties in the request
requestSummary.ties.failedValidation <i>[Always]</i>	Integer	Number of meter data records that failed validation for ties in the request
requestSummary.ties.accepted <i>[Always]</i>	Integer	Number of meter data records accepted for ties in the request
requestSummary.ties.rRejected <i>[Always]</i>	Integer	Number of meter data records rejected for ties in the request
requestSummary.subzones <i>[Always]</i>	Subzone Summary object	Processing summary for subzones submitted in the request
requestSummary.subzones.submitted <i>[Always]</i>	Integer	Number of meter data records submitted for subzones in the request
requestSummary.subzones.passedValidation <i>[Always]</i>	Integer	Number of meter data records that passed validation for subzones in the request

Parameter <i>[Provided...?]</i>	Data Type	Description
requestSummary.subzones.failedValidation <i>[Always]</i>	Integer	Number of meter data records that failed validation for subzones in the request
requestSummary.subzones.accepted <i>[Always]</i>	Integer	Number of meter data records accepted for subzones in the request
requestSummary.subzones.rejected <i>[Always]</i>	Integer	Number of meter data records rejected for subzones in the request
failedValidation <i>[If any records have failed validation]</i>	Failed Validation Object	Object containing all meter data records that failed validation, grouped by entity type
failedValidation.generators <i>[If any records have failed validation for generators]</i>	Array of Generator Metering records, including Errors	Meter data records for generators that failed validation
failedValidation.generators.genPtId <i>[For each generator meter data record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given generator
failedValidation.generators.dateHour <i>[For each generator meter data record that failed validation]</i>	ISO-8601 Date / Time	Submitted service hour for meter data
failedValidation.generators.meterInjectionEnergyMwh <i>[When provided as part of generator meter data record that failed validation]</i>	Number	Submitted metered injections for the given generator and service hour

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
failedValidation.generators.meterWithdrawalEnergyMwh <i>[When provided as part of generator meter data record that failed validation]</i>	Number	Submitted metered withdrawals for the given generator and service hour
failedValidation.generators.meterDemandReductionMwh <i>[When provided as part of generator meter data record that failed validation]</i>	Number	Submitted metered demand reduction for the given generator and service hour
failedValidation.generators.errors <i>[For each generator meter data record that failed validation]</i>	Array of Strings	Errors for the meter data record that failed validation
failedValidation.ties <i>[If any records have failed validation for ties]</i>	Array of Tie Metering objects, including Errors	Meter data records for ties that failed validation
failedValidation.ties.tiePtId <i>[For each tie meter data record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given tie
failedValidation.ties.dateHour <i>[For each tie meter data record that failed validation]</i>	ISO-8601 Date / Time	Submitted service hour for meter data
failedValidation.ties.meterTieFlowMwh <i>[For each tie meter data record that failed validation]</i>	Number	Submitted metered tie flow for the given tie
failedValidation.ties.errors <i>[For each generator meter data record that failed validation]</i>	Array of Strings	Errors for the meter data record that failed validation
failedValidation.subzones <i>[If any records have that failed validation for subzones]</i>	Array of Subzone Metering objects, including Errors	Meter data records for subzones that failed validation

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
failedValidation.subzones.subzonePtId <i>[For each meter data record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given subzone
failedValidation.subzones.dateHour <i>[For each subzone meter data record that failed validation]</i>	ISO-8601 Date / Time	Submitted service hour for meter data
failedValidation.subzones.meterSubzoneLoadMwh <i>[For each subzone meter data record that failed validation]</i>	Number	Submitted metered subzone load for the given subzone and service hour
failedValidation.subzones.errors <i>[For each generator meter data record that failed validation]</i>	Array of Strings	Errors for the meter data record that failed validation
accepted <i>[If the includeAcceptedDataInResponse property was true, and any records have been accepted]</i>	Accepted Object	Object containing all accepted meter data records, grouped by entity type
accepted.generators <i>[If any records have been accepted for generators]</i>	Array of Generator Metering objects	Accepted meter data records for generators
accepted.generators.genPtId <i>[For each accepted generator meter data record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given generator
accepted.generators.dateHour <i>[For each accepted generator meter data record]</i>	ISO-8601 Date / Time	Accepted service hour for meter data
accepted.generators.meterInjectionEnergyMwh <i>[When provided as part of accepted generator meter data record]</i>	Number, up to four decimal places  0.0000 ≤ x < 10,000.0000	Accepted metered injections for the given generator and service hour

Parameter <i>[Provided...?]</i>	Data Type	Description
accepted.generators.meterWithdrawalEnergyMwh <i>[When provided as part of accepted generator meter data record]</i>	Number, up to four decimal places $-10,000.0000 < x \leq 0.0000$	Accepted metered withdrawals for the given generator and service hour
accepted.generators.meterDemandReductionMwh <i>[When provided as part of accepted generator meter data record]</i>	Number, up to four decimal places $0.0000 \leq x < 10,000.0000$	Accepted metered demand reduction for the given generator and service hour
accepted.ties <i>[If any records have been accepted for ties]</i>	Array of Tie Metering objects	Accepted meter data records for ties
accepted.ties.tiePtId <i>[For each accepted tie meter data record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given tie
accepted.ties.dateHour <i>[For each accepted tie meter data record]</i>	ISO-8601 Date / Time	Accepted service hour for meter data
accepted.ties.meterTieFlowMwh <i>[For each accepted tie meter data record]</i>	Number, up to four decimal places $0.0000 \leq x < 10,000.0000$	Accepted metered tie flow for the given tie
accepted.subzones <i>[If any records have been accepted for subzones]</i>	Array of Subzone Metering objects	Accepted meter data records for subzones

Parameter <i>[Provided...?]</i>	Data Type	Description
accepted.subzones.subzonePtId <i>[For each accepted subzone meter data record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given subzone
accepted.subzones.dateHour <i>[For each accepted subzone meter data record]</i>	ISO-8601 Date / Time	Accepted service hour for meter data
accepted.subzones.meterSubzoneLoadMwh <i>[For each accepted subzone meter data record]</i>	Number, up to four decimal places  $0.0000 \leq x < 100,000.0000$	Accepted metered subzone load for the given subzone and service hour

### 3.1.4. Response Examples

#### 3.1.4.1. Meter Data Submission Response Example #1

- All records passed validation and therefore accepted
- includeAcceptedDataInResponse = true

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "generators": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    },
    "ties": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    }
  },
  "subzones": {
```

```
    "submitted": 1,
    "passedValidation": 1,
    "failedValidation": 0,
    "accepted": 1,
    "rejected": 0
  }
},
"accepted": {
  "generators": [
    {
      "genPtid": 345678,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterInjectionEnergyMwh": 75.1234,
      "meterWithdrawalEnergyMwh": -12.3456,
      "meterDemandReductionMwh": 5.6789
    }
  ],
  "ties": [
    {
      "tiePtid": 222222,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterTieFlowMwh": 33.3333
    }
  ],
  "subzones": [
    {
      "subzonePtid": 24680,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterSubzoneLoadMwh": 246.7531
    }
  ]
}
```

### 3.1.4.2. Meter Data Submission Response Example #2

- One tie record failed validation, and therefore all records rejected
- `includeAcceptedDataInResponse = true`, but as all records are rejected, no accepted data is included

#### HTTP Status:

400 Bad Request

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "generators": {
      "submitted": 12,
      "passedValidation": 12,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 12
    },
    "ties": {
      "submitted": 3,
      "passedValidation": 2,
      "failedValidation": 1,
      "accepted": 0,
      "rejected": 3
    },
    "subzones": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 1
    }
  }
}
```

```
}
},
"failedValidation": {
  "ties": [
    {
      "tiePtid": 222222,
      "dateHour": "2021-12-14T02:00:00.000-05:00",
      "meterTieFlowMwh": 33.3333,
      "errors": [
        "M00001: This is an example message"
      ]
    }
  ]
}
}
```

### 3.1.4.3. Meter Data Submission Response Example #3

- All records passed validation,
- `doCommit = false`, and therefore no records are accepted, and all rejected
- `includeAcceptedDataInResponse = false` by default

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": false,
    "doCommit": false
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "generators": {
      "submitted": 12,
      "passedValidation": 12,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 0
    },
    "ties": {
      "submitted": 3,
      "passedValidation": 3,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 0
    },
    "subzones": {
      "submitted": 1,
      "passedValidation": 1,
```

```
"failedValidation": 0,  
  "accepted": 0,  
  "rejected": 0  
}  
}  
}
```

## 3.2. Retrieval of Submitted Meter Data

### 3.2.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/powerMetering>

HTTP Action: GET

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
endTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 31 days after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
maUpdateStartTime <i>[Optional]</i>	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.
maUpdateEndTime <i>[Optional — If provided, maUpdateStartTime must be provided]</i>	ISO-8601 Date / Time	Ending meter authority update date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021.
version <i>[Optional]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
genPtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned (along with any data for explicitly specified tie and subzone PTIDs). By default, no PTID filtering is applied.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
tiePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given tie. If defined, only data for specified tie PTIDs shall be returned (along with any data for explicitly specified generator and subzone PTIDs). By default, no PTID filtering is applied.
subzonePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
entityType <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	String  Valid Values: ALL, GENERATOR, TIE, SUBZONE	Entity types to include in results, with all others being excluded (unless specific PTIDs are included as query parameters). Default value is ALL.

### 3.2.2. Request Examples

#### 3.2.2.1. Retrieval of Submitted Meter Data Request Example #1

- Request all data for one month (December 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/powerMetering?billingMonth=2021-12
```

*HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

#### 3.2.2.2. Retrieval of Submitted Meter Data Request Example #2

- Request all data for one day (December 20, 2021)
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/powerMetering?startTime=2021-12-20T00:00:00-05:00&endTime=2021-12-20T23:59:59-05:00&userRequestId=MyRequest_ABCD-001
```

*HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.2.3. Response

*HTTP Headers:*

Content-Type: application/json

*Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters  <i>[Always]</i>	Request Parameters object	User-provided parameters for the request

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.maUpdateStartTime <i>[If provided in request]</i>	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.maUpdateEndTime <i>[If provided in request]</i>	ISO-8601 Date / Time	Ending meter authority update date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021.
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.genPtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned (along with any data for explicitly specified tie and subzone PTIDs). By default, no PTID filtering is applied.
requestParameters.tiePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given tie. If defined, only data for specified tie PTIDs shall be returned (along with any data for explicitly specified generator and subzone PTIDs). By default, no PTID filtering is applied.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
requestParameters.entityType <i>[If provided in request]</i>	String  Valid Values: ALL, GENERATOR, TIE, SUBZONE	Entity types to include in results, with all others being excluded (unless specific PTIDs are included as query parameters). Default value is ALL.
nyisoRequestId <i>[Always]</i>	UUID String  36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
generators <i>[If any generator metering detail records are returned]</i>	Array of Generator Metering Detail objects	Metering detail records for generators
generators.genPtId <i>[For each generator metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator

Parameter <i>[Provided...?]</i>	Data Type	Description
generators.generatorName <i>[For each generator metering detail record]</i>	String	Generator name
generators.dateHour <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Service hour for metering detail
generators.billingDate <i>[For each generator metering detail record]</i>	ISO-8601 Date	Billing / service day for metering detail
generators.version <i>[For each generator metering detail record]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
generators.billedFlag <i>[For each generator metering detail record]</i>	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.
generators.meterInjectionEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection revenue-grade metering]</i>	Number	MA-submitted metered injections for the given generator and service hour
generators.telemetryInjectionEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection telemetry]</i>	Number	Hourly integrated injections calculated from telemetry data for the given generator and service hour
generators.meterWithdrawalEnergyMwh <i>[For each generator metering detail record if generator is authorized for withdrawal revenue-grade metering]</i>	Number	MA-submitted metered withdrawals for the given generator and service hour

Parameter <i>[Provided...?]</i>	Data Type	Description
generators.telemetryWithdrawalEnergyMwh <i>[For each generator metering detail record if generator is authorized for withdrawal telemetry]</i>	Number	Hourly integrated withdrawals calculated from telemetry data for the given generator and service hour
generators.meterNetEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection and/or withdrawal revenue-grade metering]</i>	Number	Calculated MA-submitted metered net energy, combining MA-submitted metered injections and withdrawals for the given generator and service hour
generators.telemetryNetEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection and/or withdrawal telemetry]</i>	Number	Calculated hourly integrated net energy, combining hourly integrated injections and withdrawals calculated from telemetry data for the given generator and service hour
generators.meterDemandReductionMwh <i>[For each generator metering detail record if generator is authorized for demand reduction revenue-grade metering]</i>	Number	Submitted metered demand reduction for the given generator and service hour
generators.telemetryDemandReductionMwh <i>[For each generator metering detail record if generator is authorized for demand reduction telemetry]</i>	Number	Hourly integrated demand reductions calculated from telemetry data for the given generator and service hour
generators.meterAuthority <i>[For each generator metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given generator and service hour; null otherwise.

Parameter <i>[Provided...?]</i>	Data Type	Description
generators.meterAuthorityUpdateTime <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given generator and service hour; null if no data has yet been submitted for the given generator and service hour.
generators.meterAuthorityUpdateUser <i>[For each generator metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given generator and service hour; null if no data has yet been submitted for the given generator and service hour.
generators.updateTime <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Generator Metering Detail for the given generator and service hour.
ties <i>[If any tie metering detail records are returned]</i>	Array of Tie Metering Detail objects	Metering detail records for ties
ties.tiePtId <i>[For each tie metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given tie
ties.tieName <i>[For each generator metering detail record]</i>	String	Tie name
ties.dateHour <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Service hour for metering detail

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<b>ties.billingDate</b> <i>[For each generator metering detail record]</i>	ISO-8601 Date	Billing / service day for metering detail
<b>ties.version</b> <i>[For each tiemetering detail record]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
<b>ties.billedFlag</b> <i>[For each tie metering detail record]</i>	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.
<b>ties.meterTieFlowMwh</b> <i>[For each tie metering detail record]</i>	Number	Submitted metered tie flow for the given tie
<b>ties.telemetryTieFlowMwh</b> <i>[For each tie metering detail record]</i>	Number	Hourly integrated tie flow calculated from telemetry data for the given tie and service hour
<b>ties.meterAuthority</b> <i>[For each tie metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given tie and service hour; null otherwise.
<b>ties.meterAuthorityUpdateTime</b> <i>[For each tie metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given tie and service hour; null if no data has yet been submitted for the given tie and service hour.

Parameter <i>[Provided...?]</i>	Data Type	Description
ties.meterAuthorityUpdateUser <i>[For each tie metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given tie and service hour; null if no data has yet been submitted for the given tie and service hour.
ties.updateTime <i>[For each tie metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Tie Metering Detail for the given tie and service hour.
subzones <i>[If any subzone metering detail records are returned]</i>	Array of Subzone Metering Detail objects	Metering detail records for subzones
subzones.subzonePtId <i>[For each subzone metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
subzones.subzoneName <i>[For each subzone metering detail record]</i>	String	Subzone name
subzones.dateHour <i>[For each subzone metering detail record]</i>	ISO-8601 Date / Time	Submitted service hour for meter data
subzones.billingDate <i>[For each subzone metering detail record]</i>	ISO-8601 Date	Billing / service day for metering detail
subzones.version <i>[For each subzone metering detail record]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

Parameter <i>[Provided...?]</i>	Data Type	Description
subzones.billedFlag <i>[For each subzone metering detail record]</i>	String “Y” or “N”	If “Y”, data in metering detail has been used to calculate settlements; if “N”, data has not yet been used.
subzones.meterSubzoneLoadMwh <i>[For each subzone metering detail record]</i>	Number	Submitted metered subzone load for the given subzone and service hour
subzones.meterAuthority <i>[For each subzone metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given subzone and service hour; null otherwise.
subzones.meterAuthorityUpdateTime <i>[For each subzone metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given subzone and service hour; null if no data has yet been submitted for the given subzone and service hour.
subzones.meterAuthorityUpdateUser <i>[For each subzone metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given subzone and service hour; null if no data has yet been submitted for the given subzone and service hour.
subzones.updateTime <i>[For each subzone metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Subzone Metering Detail for the given subzone and service hour.

### 3.2.4. Response Examples

#### 3.2.4.1. Retrieval of Submitted Meter Data Response Example #1

- One service hour requested

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "generators": [
    {
      "genPtid": 345678,
      "generatorName": "GEN_XYZ_A",
      "dateHour": "2021-12-14T02:00:00-05:00",
      "billingDate": "2021-12-14",
      "version": 0,
      "billedFlag": "Y",
      "meterInjectionEnergyMwh": 75.1234,
      "telemetryInjectionEnergyMwh": 75.1234,
      "meterNetEnergyMwh": 62.7778,
      "telemetryNetEnergyMwh": 62.7778,
      "meterAuthority": "Meter Authority X",
      "meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
      "updateTime": "2021-12-14T05:07:09-05:00"
    },
    {
      "genPtid": 456789,
      "generatorName": "AGG_XYZ_B",
```

```
"dateHour": "2021-12-14T02:00:00-05:00",
"billingDate": "2021-12-14",
"version": 0,
"billedFlag": "Y",
"meterInjectionEnergyMwh": 75.1234,
"telemetryInjectionEnergyMwh": 75.1234,
"meterWithdrawalEnergyMwh": -12.3456,
"telemetryWithdrawalEnergyMwh": -12.3456,
"meterNetEnergyMwh": 62.7778,
"telemetryNetEnergyMwh": 62.7778,
"meterDemandReductionMwh": 5.6789,
"telemetryDemandReductionMwh": 5.6789,
"meterAuthority": "Meter Authority X",
"meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
"updateTime": "2021-12-14T05:07:09-05:00"
}
],
"ties": [
{
"tiePtid": 222222,
"tieName": "TIE_FROM_HERE_TO_THERE",
"dateHour": "2021-12-14T02:00:00-05:00",
"billingDate": "2021-12-14",
"version": 0,
"billedFlag": "Y",
"meterTieFlowMwh": 33.3333,
"telemetryTieFlowMwh": 33.3333,
"meterAuthority": "Meter Authority X",
"meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
"updateTime": "2021-12-14T05:07:09-05:00"
}
],
"subzones": [
{
"subzonePtid": 299999,
"subzoneName": "SUBZONE_S",
"dateHour": "2021-12-14T02:00:00-05:00",
"billingDate": "2021-12-14",
"version": 0,
"meterSubzoneLoadMwh": 246.7531,
"meterAuthority": "Meter Authority X",
"meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
"updateTime": "2021-12-14T05:07:09-05:00"
}
```

```
}  
  ]  
}
```

### 3.3. Retrieval of Calculated Subzone Load Summary

#### 3.3.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/summary>

HTTP Action: GET

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic *encodedCredential*  
 Cache-Control: no-cache

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
endTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 31 days after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
version <i>[Optional]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied, and data for all subzones associated with the requesting user shall be returned.

### 3.3.2. Request Examples

#### 3.3.2.1. Retrieval of Calculated Subzone Load Summary Request Example #1

- Request all data for one month (December 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/summary?billingMonth=2021-12
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

#### 3.3.2.2. Retrieval of Calculated Subzone Load Summary Request Example #2

- Request all data for one day (December 20, 2021)
- Optional user request ID provided
- Optional subzone PTID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/summary?startTime=2021-12-20T00:00:00-05:00&endTime=2021-12-20T23:59:59-05:00&userRequestId=MyRequest_ABCD-001&subzonePtid=299999
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

### 3.3.3. Response

#### HTTP Headers:

```
Content-Type: application/json
```

Response Body Parameters:

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
calculatedSubzoneLoads <i>[If any calculated subzone load summary records are returned]</i>	Array of Calculated Subzone Load Summary objects	Calculated Subzone Load summary records
calculatedSubzoneLoads.subzonePtId <i>[For each calculated subzone load summary record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone

Parameter <i>[Provided...?]</i>	Data Type	Description
calculatedSubzoneLoads.subzoneName <i>[For each calculated subzone load summary record]</i>	String	Subzone name
calculatedSubzoneLoads.dateHour <i>[For each calculated subzone load summary record]</i>	ISO-8601 Date / Time	Service hour for calculated subzone load data
calculatedSubzoneLoads.billingDate <i>[For each calculated subzone load summary record]</i>	ISO-8601 Date	Billing / service day for calculated subzone load data
calculatedSubzoneLoads.version <i>[For each calculated subzone load summary record]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
calculatedSubzoneLoads.nyisoCalculatedSubzoneLoadMwh <i>[For each calculated subzone load summary record]</i>	Number	NYISO-calculated subzone load for the given subzone and service hour
calculatedSubzoneLoads.subzoneLossesMwh <i>[For each calculated subzone load summary record]</i>	Number	Hourly integrated subzone losses calculated from telemetry data for the given subzone and service hour

### 3.3.4. Response Examples

#### 3.3.4.1. Retrieval of Calculated Subzone Load Summary Response Example #1

- One service day requested

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "calculatedSubzoneLoads": [
    {
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "dateHour": "2021-12-14T02:00:00-05:00",
      "billingDate": "2021-12-14",
      "version": 0,
      "nyisoCalculatedSubzoneLoadMwh": 1357.9876,
      "subzoneLossesMwh": 23.2323
    }
  ]
}
```

### 3.4. Retrieval of Calculated Subzone Load Detail

#### 3.4.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/detail>

HTTP Action: GET

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic *encodedCredential*  
 Cache-Control: no-cache

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
endTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 31 days after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
version <i>[Optional]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied, and data for all subzones associated with the requesting user shall be returned.

### 3.4.2. Request Examples

#### 3.4.2.1. Retrieval of Calculated Subzone Load Detail Request Example #1

- Request all data for one month (December 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/detail?billingMonth=2021-12
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

#### 3.4.2.2. Retrieval of Calculated Subzone Load Detail Request Example #2

- Request all data for one day (December 20, 2021)
- Optional user request ID provided
- Optional subzone PTID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/calculatedSubzoneLoad/detail?startTime=2021-12-20T00:00:00-05:00&endTime=2021-12-20T23:59:59-05:00&userRequestId=MyRequest_ABCD-001&subzonePtid=299999
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

### 3.4.3. Response

#### HTTP Headers:

```
Content-Type: application/json
```

*Response Body Parameters:*

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
calculatedSubzoneLoadDetails <i>[If any calculated subzone load details are returned]</i>	Array of Calculated Subzone Load Detail objects	Calculated Subzone Load details for each returned subzone
calculatedSubzoneLoadDetails.subzonePtId <i>[For each calculated subzone load detail record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<code>calculatedSubzoneLoadDetails.subzoneName</code> <i>[For each calculated subzone load detail record]</i>	String	Subzone name
<code>calculatedSubzoneLoadDetails.dateHour</code> <i>[For each calculated subzone load detail record]</i>	ISO-8601 Date / Time	Service hour for calculated subzone load data
<code>calculatedSubzoneLoadDetails.billingDate</code> <i>[For each calculated subzone load detail record]</i>	ISO-8601 Date	Billing / service day for calculated subzone load data
<code>calculatedSubzoneLoadDetails.version</code> <i>[For each calculated subzone load detail record]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
<code>calculatedSubzoneLoadDetails.totalSubzoneLoadContributionMwh</code> <i>[For each calculated subzone load detail record]</i>	Number	Total contributions to NYISO-calculated subzone load for the given subzone and service hour, including contributions from for generators, ties, and submitted subzone load
<code>calculatedSubzoneLoadDetails.totalGeneratorSubzoneLoadContributionMwh</code> <i>[For each calculated subzone load detail record]</i>	Number	Total contributions to NYISO-calculated subzone load from generators for the given subzone and service hour
<code>calculatedSubzoneLoadDetails.totalTieSubzoneLoadContributionMwh</code> <i>[For each calculated subzone load detail record]</i>	Number	Total contributions to NYISO-calculated subzone load from ties for the given subzone and service hour

Parameter <i>[Provided...?]</i>	Data Type	Description
calculatedSubzoneLoadDetails.generators <i>[If any generator metering detail records are returned]</i>	Array of Generator Metering Detail objects	Metering detail records for generators
calculatedSubzoneLoadDetails.generators.genPtId <i>[For each generator metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator
calculatedSubzoneLoadDetails.generators.generatorName <i>[For each generator metering detail record]</i>	String	Generator name
calculatedSubzoneLoadDetails.generators.billedFlag <i>[For each generator metering detail record]</i>	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.
calculatedSubzoneLoadDetails.generators.meterInjectionEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection revenue-grade metering]</i>	Number	MA-submitted metered injections for the given generator and service hour
calculatedSubzoneLoadDetails.generators.telemetryInjectionEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection telemetry]</i>	Number	Hourly integrated injections calculated from telemetry data for the given generator and service hour
calculatedSubzoneLoadDetails.generators.meterWithdrawalEnergyMwh <i>[For each generator metering detail record if generator is authorized for withdrawal revenue-grade metering]</i>	Number	MA-submitted metered withdrawals for the given generator and service hour
calculatedSubzoneLoadDetails.generators.telemetryWithdrawalEnergyMwh <i>[For each generator metering detail record if generator is authorized for withdrawal telemetry]</i>	Number	Hourly integrated withdrawals calculated from telemetry data for the given generator and service hour

Parameter <i>[Provided...?]</i>	Data Type	Description
calculatedSubzoneLoadDetails.generators.meterNetEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection and/or withdrawal revenue-grade metering]</i>	Number	Calculated MA-submitted metered net energy, combining MA-submitted metered injections and withdrawals for the given generator and service hour
calculatedSubzoneLoadDetails.generators.telemetryNetEnergyMwh <i>[For each generator metering detail record if generator is authorized for injection and/or withdrawal telemetry]</i>	Number	Calculated hourly integrated net energy, combining hourly integrated injections and withdrawals calculated from telemetry data for the given generator and service hour
calculatedSubzoneLoadDetails.generators.meterDemandReductionMwh <i>[For each generator metering detail record if generator is authorized for demand reduction revenue-grade metering]</i>	Number	Submitted metered demand reduction for the given generator and service hour
calculatedSubzoneLoadDetails.generators.telemetryDemandReductionMwh <i>[For each generator metering detail record if generator is authorized for demand reduction telemetry]</i>	Number	Hourly integrated demand reductions calculated from telemetry data for the given generator and service hour
calculatedSubzoneLoadDetails.generators.meterAuthority <i>[For each generator metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given generator and service hour; null otherwise.

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>calculatedSubzoneLoadDetails.generators.meterAuthorityUpdateTime</code> <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given generator and service hour; null if no data has yet been submitted for the given generator and service hour.
<code>calculatedSubzoneLoadDetails.generators.meterAuthorityUpdateUser</code> <i>[For each generator metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given generator and service hour; null if no data has yet been submitted for the given generator and service hour.
<code>calculatedSubzoneLoadDetails.generators.updateTime</code> <i>[For each generator metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Generator Metering Detail for the given generator and service hour.
<code>calculatedSubzoneLoadDetails.ties</code> <i>[If any tie metering detail records are returned]</i>	Array of Tie Metering Detail objects	Metering detail records for ties
<code>calculatedSubzoneLoadDetails.ties.tiePtId</code> <i>[For each tie metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given tie
<code>calculatedSubzoneLoadDetails.ties.tieName</code> <i>[For each generator metering detail record]</i>	String	Tie name
<code>calculatedSubzoneLoadDetails.ties.billedFlag</code> <i>[For each tie metering detail record]</i>	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<code>calculatedSubzoneLoadDetails.ties.meterTieFlowMwh</code> <i>[For each tie metering detail record]</i>	Number	Submitted metered tie flow for the given tie
<code>calculatedSubzoneLoadDetails.ties.telemetryTieFlowMwh</code> <i>[For each tie metering detail record]</i>	Number	Hourly integrated tie flow calculated from telemetry data for the given tie and service hour
<code>calculatedSubzoneLoadDetails.ties.meterAuthority</code> <i>[For each tie metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given tie and service hour; null otherwise.
<code>calculatedSubzoneLoadDetails.ties.meterAuthorityUpdateTime</code> <i>[For each tie metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given tie and service hour; null if no data has yet been submitted for the given tie and service hour.
<code>calculatedSubzoneLoadDetails.ties.meterAuthorityUpdateUser</code> <i>[For each tie metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given tie and service hour; null if no data has yet been submitted for the given tie and service hour.
<code>calculatedSubzoneLoadDetails.ties.updateTime</code> <i>[For each tie metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Tie Metering Detail for the given tie and service hour.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<code>calculatedSubzoneLoadDetails.subzones</code> <i>[If any subzone metering detail records are returned]</i>	Array of Subzone Metering Detail objects	Metering detail records for subzones
<code>calculatedSubzoneLoadDetails.subzones.subzonePtId</code> <i>[For each subzone metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
<code>calculatedSubzoneLoadDetails.subzones.subzoneName</code> <i>[For each subzone metering detail record]</i>	String	Subzone name
<code>calculatedSubzoneLoadDetails.subzones.billedFlag</code> <i>[For each subzone metering detail record]</i>	String “Y” or “N”	If “Y”, data in metering detail has been used to calculate settlements; if “N”, data has not yet been used.
<code>calculatedSubzoneLoadDetails.subzones.meterSubzoneLoadMwh</code> <i>[For each subzone metering detail record]</i>	Number	Submitted metered subzone load for the given subzone and service hour
<code>calculatedSubzoneLoadDetails.subzones.meterAuthority</code> <i>[For each subzone metering detail record]</i>	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given subzone and service hour; null otherwise.
<code>calculatedSubzoneLoadDetails.subzones.meterAuthorityUpdateTime</code> <i>[For each subzone metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of revenue-grade meter data by the Meter Authority for the given subzone and service hour; null if no data has yet been submitted for the given subzone and service hour.

Parameter <i>[Provided...?]</i>	Data Type	Description
calculatedSubzoneLoadDetails.subzones.meterAuthorityUpdateUser <i>[For each subzone metering detail record]</i>	String	Meter Authority User who most recently submitted revenue-grade meter data for the given subzone and service hour; null if no data has yet been submitted for the given subzone and service hour.
calculatedSubzoneLoadDetails.subzones.updateTime <i>[For each subzone metering detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Subzone Metering Detail for the given subzone and service hour.

### 3.4.4. Response Examples

#### 3.4.4.1. Retrieval of Calculated Subzone Load Detail Response Example #1

- One service day requested

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "subzoneLoadDetails": [
    {
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "dateHour": "2021-12-14T02:00:00-05:00",
      "billingDate": "2021-12-14",
      "version": 0,
      "totalSubzoneLoadContributionMwh": 351.2810,
      "totalGeneratorSubzoneLoadContributionMwh": 137.9012,
      "totalTieSubzoneLoadContributionMwh": -33.3333,
      "generators": [
        {
          "genPtid": 345678,
          "generatorName": "GEN_XYZ_A",
          "billedFlag": "Y",
          "subzoneLoadContributionMwh": 75.1234,
          "meterInjectionEnergyMwh": 75.1234,
          "telemetryInjectionEnergyMwh": 75.1234,
          "meterNetEnergyMwh": 75.1234,

```

```

    "telemetryNetEnergyMwh": 75.1234,
    "meterAuthority": "Meter Authority X",
    "meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
    "updateTime": "2021-12-14T05:07:09-05:00"
  },
  {
    "genPtid": 456789,
    "generatorName": "AGG_XYZ_B",
    "billedFlag": "Y",
    "subzoneLoadContributionMwh": 62.7778,
    "meterInjectionEnergyMwh": 75.1234,
    "telemetryInjectionEnergyMwh": 75.1234,
    "meterWithdrawalEnergyMwh": -12.3456,
    "telemetryWithdrawalEnergyMwh": -12.3456,
    "meterNetEnergyMwh": 62.7778,
    "telemetryNetEnergyMwh": 62.7778,
    "meterDemandReductionMwh": 5.6789,
    "telemetryDemandReductionMwh": 5.6789,
    "meterAuthority": "Meter Authority X",
    "meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
    "updateTime": "2021-12-14T05:07:09-05:00"
  }
],
"ties": [
  {
    "tiePtid": 222222,
    "tieName": "TIE_FROM_HERE_TO_THERE",
    "billedFlag": "Y",
    "subzoneLoadContributionMwh": -33.3333,
    "meterTieFlowMwh": 33.3333,
    "telemetryTieFlowMwh": 33.3333,
    "meterAuthority": "Meter Authority X",
    "meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
    "updateTime": "2021-12-14T05:07:09-05:00"
  }
],
"subzones": [
  {
    "subzonePtid": 299999,
    "subzoneName": "SUBZONE_S",
    "billedFlag": "Y",
    "subzoneLoadContributionMwh": 246.7531,
    "meterSubzoneLoadMwh": 246.7531,

```

```
"meterAuthority": "Meter Authority X",  
"meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",  
"updateTime": "2021-12-14T05:07:09-05:00"  
}  
]  
}  
]  
}
```

### 3.5. Transmission Owner Load Submission

Meter Authorities (MAs) can use the Transmission Owner Load Submission service in the Metering API to provide load data for load buses to the NYISO.

Data may be submitted (including both new entries and updates) in accordance with the Lock-Down Schedules posted on the NYISO website. Upon submission, all data is validated, and if any records fail validation, the entire request will be rejected. All validation errors found will be included with the response.

#### 3.5.1. Request

*URL:* <https://api.nyiso.com/finance/metering/v1/transmissionOwnerLoad>

*HTTP Action:* POST

*HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

*Request Body Parameters:*

Parameter <i>[Required / Optional?]</i>	Data Type	Description
submissionParameters <i>[Optional]</i>	Array of Submission Parameters	User-provided parameters for the submission

Parameter <i>[Required / Optional?]</i>	Data Type	Description
<code>submissionParameters.userRequestId</code> <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Optional]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Optional]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to true if not provided.
<code>buses</code>	Array of Bus objects	Transmission owner load (TOL) data records for buses
<code>buses.busPtId</code> <i>[Required for each bus TOL data record]</i>	Integer	Unique NYISO-defined point identifier for a given bus
<code>buses.dateHour</code> <i>[Required for each bus TOL data record]</i>	ISO-8601 Date / Time	Service hour for TOL data
<code>buses.meterBusLoadMwh</code> <i>[Required for each bus TOL data record]</i>	Number, up to four decimal places  $0.0000 \leq x < 10,000.0000$	Metered bus load for the given bus and service hour

### 3.5.2. Request Examples

#### 3.5.2.1. Transmission Owner Load Submission Request Example

- One bus submitted
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "buses": [
    {
      "busPtId": 98765,
      "dateHour": "2021-12-14T02:00:00-05:00",
      "meterBusLoadMwh": 246.7531
    }
  ]
}
```

### 3.5.3. Response

#### HTTP Headers:

```
Content-Type: application/json
```

*Response Body Parameters:*

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<code>submissionParameters</code> <i>[Always]</i>	Submission Parameters object	User-provided parameters for the submission
<code>submissionParameters.userName</code> <i>[Always]</i>	String	User name for the user submitting the request
<code>submissionParameters.userRequestId</code> <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Always]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Always]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to false if not provided.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
requestSummary <i>[Always]</i>	Summary Object	Processing summary for the request
requestSummary.buses <i>[Always]</i>	Generator Summary object	Processing summary for buses submitted in the request
requestSummary.buses.submitted <i>[Always]</i>	Integer	Number of TOL data records submitted for buses in the request
requestSummary.buses.passedValidation <i>[Always]</i>	Integer	Number of TOL data records that passed validation for buses in the request
requestSummary.buses.failedValidation <i>[Always]</i>	Integer	Number of TOL data records that failed validation for buses in the request
requestSummary.buses.accepted <i>[Always]</i>	Integer	Number of TOL data records accepted for buses in the request
requestSummary.buses.rejected <i>[Always]</i>	Integer	Number of TOL data records rejected for buses in the request

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
failedValidation <i>[If any records have failed validation]</i>	Failed Validation Object	Object containing all TOL data records that failed validation
failedValidation.buses <i>[If any records have that failed validation for buses]</i>	Array of Bus TOL objects, including Errors	TOL data records for buses that failed validation
failedValidation.buses.busPtId <i>[For each bus TOL data record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given bus
failedValidation.buses.dateHour <i>[For each bus TOL data record that failed validation]</i>	ISO-8601 Date / Time	Submitted service hour for TOL data
failedValidation.buses.meterBusLoadMwh <i>[For each bus TOL data record that failed validation]</i>	Number	Submitted metered bus load for the given bus and service hour
failedValidation.buses.errors <i>[For each bus TOL data record that failed validation]</i>	Array of Strings	Errors for the TOL data record that failed validation
accepted <i>[If the includeAcceptedDataInResponse property was true, and any records have been accepted]</i>	Accepted Object	Object containing all accepted TOL data records
accepted.buses <i>[If any records have been accepted for buses]</i>	Array of Subzone Metering objects	Accepted TOL data records for buses
accepted.buses.busPtId <i>[For each accepted bus TOL data record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given bus
accepted.subzones.dateHour <i>[For each accepted bus TOL data record]</i>	ISO-8601 Date / Time	Accepted service hour for TOL data

Parameter <i>[Provided...?]</i>	Data Type	Description
accepted.subzones.meterBusLoadMwh <i>[For each accepted bus TOL data record]</i>	Number, up to four decimal places  $0.0000 \leq x < 100,000.0000$	Accepted metered bus load for the given bus and service hour

### 3.5.4. Response Examples

#### 3.5.4.1. Meter Data Submission Response Example #1

- All records passed validation and therefore accepted
- includeAcceptedDataInResponse = true

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "buses": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    }
  },
  "accepted": {
    "buses": [
      {
        "busPtid": 98765,
        "dateHour": "2021-12-14T02:00:00-05:00",
        "meterBusLoadMwh": 246.7531
      }
    ]
  }
}
```

```
}  
}  
]
```

### 3.5.4.2. Meter Data Submission Response Example #2

- One bus record failed validation, and therefore all records rejected
- includeAcceptedDataInResponse = true, but as all records are rejected, no accepted data is included

#### HTTP Status:

400 Bad Request

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "buses": {
      "submitted": 1,
      "passedValidation": 0,
      "failedValidation": 1,
      "accepted": 0,
      "rejected": 1
    }
  },
  "failedValidation": {
    "buses": [
      {
        "busPtid": 222222,
        "dateHour": "2021-12-14T02:00:00.000-05:00",
        "meterBusLoadMwh": 33.3333,
        "errors": [
          "M00001: This is an example message"
        ]
      }
    ]
  }
}
```

```
}
```

#### 3.5.4.3. Meter Data Submission Response Example #3

- All records passed validation,
- `doCommit = false`, and therefore no records are accepted, and all rejected
- `includeAcceptedDataInResponse = false` by default

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": false,
    "doCommit": false
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "buses": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 0
    }
  }
}
```

### 3.6. Retrieval of Submitted Transmission Owner Load

#### 3.6.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/transmissionOwnerLoad>

HTTP Action: GET

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
endTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 25 hours after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
maUpdateStartTime <i>[Optional]</i>	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.
maUpdateEndTime <i>[Optional — If provided, maUpdateStartTime must be provided]</i>	ISO-8601 Date / Time	Ending meter authority update date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021.
version <i>[Optional]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
busPtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given bus. If defined, only data for specified bus PTIDs shall be returned. By default, no PTID filtering is applied.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
subzonePtId  <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for buses within the subzones specified by the subzone PTIDs shall be returned. By default, no PTID filtering is applied.

### 3.6.2. Request Examples

#### 3.6.2.1. Retrieval of Submitted Transmission Owner Load Request Example #1

- Request all data for one month (December 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad?billingMonth=2021-12
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

#### 3.6.2.2. Retrieval of Submitted Transmission Owner Load Request Example #2

- Request all data for one day (December 20, 2021)
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad?startTime=2021-12-20T00:00:00-05:00&endTime=2021-12-20T23:59:59-05:00&userRequestId=MyRequest_ABCD-001
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.6.3. Response

HTTP Headers:

Content-Type: application/json

Response Body Parameters:

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
requestParameters.maUpdateStartTime <i>[If provided in request]</i>	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.
requestParameters.maUpdateEndTime <i>[If provided in request]</i>	ISO-8601 Date / Time	Ending meter authority update date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021.
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.busPtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given bus. If defined, only data for specified bus PTIDs shall be returned. By default, no PTID filtering is applied.
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for buses within the subzones specified by the subzone PTIDs shall be returned. By default, no PTID filtering is applied.

Parameter <i>[Provided...?]</i>	Data Type	Description
nyisoRequestId <i>[Always]</i>	UUID String  36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
buses <i>[If any subzone metering detail records are returned]</i>	Array of Bus TOL Detail objects	TOL detail records for buses
buses.busPtId <i>[For each bus TOL detail record]</i>	Integer	Unique NYISO-defined point identifier for a given bus
buses.busName <i>[For each bus TOL detail record]</i>	String	Bus name
buses.dateHour <i>[For each bus TOL detail record]</i>	ISO-8601 Date / Time	Submitted service hour for meter data
buses.billingDate <i>[For each bus TOL detail record]</i>	ISO-8601 Date	Billing / service day for metering detail
buses.version <i>[For each bus TOL detail record]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
buses.billedFlag <i>[For each bus TOL detail record]</i>	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.
buses.meterBusLoadMwh <i>[For each bus TOL detail record]</i>	Number	Submitted metered bus load for the given bus and service hour
buses.meterAuthority <i>[For each bus TOL detail record]</i>	String	Meter Authority that submitted TOL data
buses.meterAuthorityUpdateTime <i>[For each bus TOL detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent submission of TOL data by the Meter Authority for the given bus and service hour
buses.meterAuthorityUpdateUser <i>[For each bus TOL detail record]</i>	String	Meter Authority User who most recently submitted TOL data for the given bus and service hour
buses.updateTime <i>[For each bus TOL detail record]</i>	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Bus TOL Detail for the given bus and service hour

### 3.6.4. Response Examples

#### 3.6.4.1. Retrieval of Submitted Transmission Owner Load Data Response Example #1

- One service hour requested

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "buses": [
    {
      "busPtId": 98765,
      "busName": "BUS_B",
      "dateHour": "2021-12-14T02:00:00-05:00",
      "billingDate": "2021-12-14",
      "version": 0,
      "billedFlag": "N",
      "meterBusLoadMwh": 246.7531,
      "meterAuthority": "Meter Authority X",
      "meterAuthorityUpdateTime": "2021-12-14T05:07:09-05:00",
      "meterAuthorityUpdateUser": "MA_USER",
      "updateTime": "2021-12-14T05:07:09-05:00"
    }
  ]
}
```

### 3.7. Retrieval of Transmission Owner Load Verification Summary

#### 3.7.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/summary>

HTTP Action: GET

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic *encodedCredential*  
 Cache-Control: no-cache

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Starting service date of requested data.
endDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date, and must be in the same service month as the start date.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
version <i>[Optional]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied, and data for all subzones associated with the requesting user shall be returned.
userRequestId <i>[Optional]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

### 3.7.2. Request Examples

#### 3.7.2.1. Retrieval of Transmission Owner Load Verification Summary Request Example #1

- Request all data for one month (December 2021)

GET

<https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/summary?billingMonth=2021>

-12

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

**3.7.2.2. Retrieval of Transmission Owner Load Verification Summary Request Example #2**

- Request all data for one week (December 1–7, 2021)

GET

```
https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/summary?startDate=2021-12-01&endDate=2021-12-07
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

**3.7.2.3. Retrieval of Transmission Owner Load Verification Summary Request Example #3**

- Request all data for one month (December 2021)
- Optional user request ID provided
- Optional subzone PTID provided

GET

```
https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/summary?billingMonth=2021-12&userRequestId=MyRequest_ABCD-001&subzonePtid=299999
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.7.3. Response

HTTP Headers:

Content-Type: application/json

Response Body Parameters:

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startDate <i>[If provided in request]</i>	ISO-8601 Date	Starting service date of requested data.
requestParameters.endDate <i>[If provided in request]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date and must be in the same service month as the start date.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

Parameter <i>[Provided...?]</i>	Data Type	Description
nyisoRequestId <i>[Always]</i>	UUID String  36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
loadVerifications <i>[If any load verification summary records are returned]</i>	Array of Load Verification Summary objects	Load verification summary records
loadVerifications.startTime <i>[For each load verification summary record]</i>	ISO-8601 Date / Time	Starting service date / time of date range.
loadVerifications.endTime <i>[For each load verification summary record]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
loadVerifications.subzonePtId <i>[For each load verification summary record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
loadVerifications.subzoneName <i>[For each load verification summary record]</i>	String	Subzone name

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>LoadVerifications.totalMloadMwh</code> <i>[For each load verification summary record]</i>	Number	Total Mload (i.e., the NYISO-calculated subzone load) for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
<code>LoadVerifications.totalSubzoneMeterBusLoadMwh</code> <i>[For each load verification summary record]</i>	Number	Total submitted metered bus load for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
<code>LoadVerifications.authorizedSubzoneMeterBusLoadMwh</code> <i>[For each load verification summary record]</i>	Number	Total submitted metered bus load for the given subzone and date range, including only those buses for which the user is authorized
<code>LoadVerifications.totalSubzoneLoadAbsoluteDeltaMwh</code> <i>[For each load verification summary record]</i>	Number	Absolute value of the difference between the total Mload (i.e., the NYISO-calculated subzone load) and total submitted metered bus load for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
<code>LoadVerifications.subzoneLoadValidForAllHours</code> <i>[For each load verification summary record]</i>	Boolean (true / false)	True when NYISO-calculated subzone load and total submitted metered bus load match for the given subzone for every hour of the date range; false otherwise

Parameter <i>[Provided...?]</i>	Data Type	Description
LoadVerifications.totalBusCount <i>[For each load verification summary record]</i>	Integer	Total number of buses within the given subzone and having submitted metered load at any point during the date range
LoadVerifications.authorizedBusCount <i>[For each load verification summary record]</i>	Integer	Total number of buses within the given subzone and having submitted metered load at any point during the date range, including only those buses for which the user is authorized
LoadVerifications.mismatchDetails <i>[If any mismatched hours are returned for the load verification summary record]</i>	Array of Load Verification Mismatch Detail Objects	Service hours for which NYISO-calculated subzone load and total submitted metered bus load do not match for the given subzone during the date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.dateHour <i>[For each load verification mismatch detail record]</i>	ISO-8601 Date / Time	Service hour for which NYISO-calculated subzone load and total submitted metered bus load do not match for the given subzone. Not included when the Authorized Bus Count does not equal the Total Bus Count.

Parameter <i>[Provided...?]</i>	Data Type	Description
LoadVerifications.mismatchDetails.mloadMwh <i>[For each load verification mismatch detail record]</i>	Number	Mload (i.e., the NYISO-calculated subzone load) for the given subzone and service hour. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.meterBusLoadMwh <i>[For each load verification mismatch detail record]</i>	Number	Submitted metered bus load for the given subzone and service hour. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.absoluteDeltaMwh <i>[For each load verification mismatch detail record]</i>	Number	Absolute value of the difference between the Mload (i.e., the NYISO-calculated subzone load) and the submitted metered bus load for the given subzone and service hour

### 3.7.4. Response Examples

#### 3.7.4.1. Retrieval of Transmission Owner Load Verification Summary Response Example #1

- One service month requested
- User only authorized to see one subzone
- No mismatched hours

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "billingMonth": "2021-12",
    "startTime": "2021-12-01T00:00:00-05:00",
    "endTime": "2022-01-01T04:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "loadVerifications": [
    {
      "startTime": "2021-12-01T00:00:00-05:00",
      "endTime": "2022-01-01T04:59:59-05:00",
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "billingMonth": "2021-12-14",
      "totalMloadMwh": 1357.9876,
      "totalSubzoneMeterBusLoadMwh": 1357.9876,
      "authorizedSubzoneMeterBusLoadMwh": 1357.9876,
      "totalSubzoneLoadAbsoluteDeltaMwh": 0.0000,
      "subzoneLoadValidForAllHours": true,
      "totalBusCount": 77,
```

```
    "authorizedBusCount": 77  
  }  
]  
}
```

### 3.7.4.2. Retrieval of Transmission Owner Load Verification Summary Response Example #2

- One service month requested
- User only authorized to see one subzone
- Three mismatched hours

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "billingMonth": "2021-12",
    "startTime": "2021-12-01T00:00:00-05:00",
    "endTime": "2022-01-01T04:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "loadVerifications": [
    {
      "startTime": "2021-12-01T00:00:00-05:00",
      "endTime": "2022-01-01T04:59:59-05:00",
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "billingMonth": "2021-12-14",
      "totalMloadMwh": 1357.9876,
      "totalSubzoneMeterBusLoadMwh": 1327.9876,
      "authorizedSubzoneMeterBusLoadMwh": 1327.9876,
      "totalSubzoneLoadAbsoluteDeltaMwh": 30.0000,
      "subzoneLoadValidForAllHours": false,
      "totalBusCount": 77,
      "authorizedBusCount": 77,
      "mismatchDetails": [
```

```
{
  "dateHour": "2021-12-01T07:00:00-05:00",
  "mloadMwh": 45.5555,
  "meterBusLoadMwh": 55.5555,
  "absoluteDeltaMwh": 10.0000
},
{
  "dateHour": "2021-12-12T15:00:00-05:00",
  "mloadMwh": 45.5555,
  "meterBusLoadMwh": 55.5555,
  "absoluteDeltaMwh": 10.0000
},
{
  "dateHour": "2021-12-23T23:00:00-05:00",
  "mloadMwh": 45.5555,
  "meterBusLoadMwh": 55.5555,
  "absoluteDeltaMwh": 10.0000
}
]
}
```

### 3.8. Retrieval of Transmission Owner Load Verification Detail

#### 3.8.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/detail>

HTTP Action: GET

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Starting service date of requested data.
endDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date, and must be in the same service month as the start date.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
version <i>[Optional]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied, and data for all subzones associated with the requesting user shall be returned.
userRequestId <i>[Optional]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

### 3.8.2. Request Examples

#### 3.8.2.1. Retrieval of Transmission Owner Load Verification Detail Request Example #1

- Request all data for one month (December 2021)

GET

<https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/detail?billingMonth=2021->

12

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

**3.8.2.2. Retrieval of Transmission Owner Load Verification Detail Request Example #2**

- Request all data for one week (December 1–7, 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/detail?startDate=2021-12-01&endDate=2021-12-07
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

**3.8.2.3. Retrieval of Transmission Owner Load Verification Detail Request Example #3**

- Request all data for one day (December 20, 2021)
- Optional user request ID provided
- Optional subzone PTID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad/verification/detail?billingMonth=2021-12&userRequestId=MyRequest_ABCD-001&subzonePtId=299999
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.8.3. Response

HTTP Headers:

Content-Type: application/json

Response Body Parameters:

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startDate <i>[If provided in request]</i>	ISO-8601 Date	Starting service date of requested data.
requestParameters.endDate <i>[If provided in request]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date, and must be in the same service month as the start date.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.subzonePtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given subzone. If defined, only data for specified subzone PTIDs shall be returned (along with any data for explicitly specified generator and tie PTIDs). By default, no PTID filtering is applied.
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

Parameter <i>[Provided...?]</i>	Data Type	Description
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
loadVerifications <i>[If any load verification summary records are returned]</i>	Array of Load Verification Summary objects	Load verification summary records
loadVerifications.startTime <i>[For each load verification summary record]</i>	ISO-8601 Date / Time	Starting service date / time of date range.
loadVerifications.endTime <i>[For each load verification summary record]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
loadVerifications.subzonePtId <i>[For each load verification summary record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
loadVerifications.subzoneName <i>[For each load verification summary record]</i>	String	Subzone name

Parameter <i>[Provided...?]</i>	Data Type	Description
LoadVerifications.totalMloadMwh <i>[For each load verification summary record]</i>	Number	Total Mload (i.e., the NYISO-calculated subzone load) for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.totalSubzoneMeterBusLoadMwh <i>[For each load verification summary record]</i>	Number	Total submitted metered bus load for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.authorizedSubzoneMeterBusLoadMwh <i>[For each load verification summary record]</i>	Number	Total submitted metered bus load for the given subzone and date range, including only those buses for which the user is authorized
LoadVerifications.totalSubzoneLoadAbsoluteDeltaMwh <i>[For each load verification summary record]</i>	Number	Absolute value of the difference between the total Mload (i.e., the NYISO-calculated subzone load) and total submitted metered bus load for the given subzone and date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.subzoneLoadValidForAllHours <i>[For each load verification summary record]</i>	Boolean (true / false)	True when NYISO-calculated subzone load and total submitted metered bus load match for the given subzone for every hour of the date range; false otherwise

Parameter <i>[Provided...?]</i>	Data Type	Description
LoadVerifications.totalBusCount <i>[For each load verification summary record]</i>	Integer	Total number of buses within the given subzone and having submitted metered load at any point during the date range
LoadVerifications.authorizedBusCount <i>[For each load verification summary record]</i>	Integer	Total number of buses within the given subzone and having submitted metered load at any point during the date range, including only those buses for which the user is authorized
LoadVerifications.mismatchDetails <i>[If any mismatched hours are returned for the load verification summary record]</i>	Array of Load Verification Mismatch Detail Objects	Service hours for which NYISO-calculated subzone load and total submitted metered bus load do not match for the given subzone during the date range. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.dateHour <i>[For each load verification mismatch detail record]</i>	ISO-8601 Date / Time	Service hour for which NYISO-calculated subzone load and total submitted metered bus load do not match for the given subzone. Not included when the Authorized Bus Count does not equal the Total Bus Count.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
LoadVerifications.mismatchDetails.mloadMwh <i>[For each load verification mismatch detail record]</i>	Number	Mload; i.e., the NYISO-calculated subzone load for the given subzone and service hour. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.meterBusLoadMwh <i>[For each load verification mismatch detail record]</i>	Number	Submitted metered bus load for the given subzone and service hour. Not included when the Authorized Bus Count does not equal the Total Bus Count.
LoadVerifications.mismatchDetails.absoluteDeltaMwh <i>[For each load verification mismatch detail record]</i>	Number	Absolute value of the difference between the Mload (i.e., the NYISO-calculated subzone load) and the submitted metered bus load for the given subzone and service hour
LoadVerifications.buses <i>[If any load verification bus metering detail records are returned]</i>	Array of Load Verification Bus Metering Detail objects	Metering detail records for buses for purposes of load verification
LoadVerifications.buses.busPtId <i>[For each load verification bus metering detail record]</i>	Integer	Unique NYISO-defined point identifier for a given subzone
LoadVerifications.buses.busName <i>[For each load verification bus metering detail record]</i>	String	Subzone name
LoadVerifications.buses.meterBusLoadMwh <i>[For each load verification bus metering detail record]</i>	Number	Total submitted metered bus load for the given bus and date range

### 3.8.4. Response Examples

#### 3.8.4.1. Retrieval of Transmission Owner Load Verification Detail Response Example #1

- One service month requested
- User only authorized to see one subzone
- No mismatched hours

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "billingMonth": "2021-12",
    "startTime": "2021-12-01T00:00:00-05:00",
    "endTime": "2022-01-01T04:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "loadVerifications": [
    {
      "startTime": "2021-12-01T00:00:00-05:00",
      "endTime": "2022-01-01T04:59:59-05:00",
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "billingMonth": "2021-12-14",
      "totalMloadMwh": 1357.9876,
      "totalSubzoneMeterBusLoadMwh": 1357.9876,
      "authorizedSubzoneMeterBusLoadMwh": 1357.9876,
      "totalSubzoneLoadAbsoluteDeltaMwh": 0.0000,
      "subzoneLoadValidForAllHours": true,
      "totalBusCount": 77,
    }
  ]
}
```

```
"authorizedBusCount": 77,  
"buses": [  
  {  
    "busPtid": 98765,  
    "busName": "BUS_B",  
    "meterBusLoadMwh": 22246.7531  
  },  
  ...  
]  
]  
}
```

### 3.8.4.2. Retrieval of Transmission Owner Load Verification Detail Response Example #2

- One service month requested
- User only authorized to see one subzone
- Three mismatched hours

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "billingMonth": "2021-12",
    "startTime": "2021-12-01T00:00:00-05:00",
    "endTime": "2022-01-01T04:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "loadVerifications": [
    {
      "startTime": "2021-12-01T00:00:00-05:00",
      "endTime": "2022-01-01T04:59:59-05:00",
      "subzonePtid": 299999,
      "subzoneName": "SUBZONE_S",
      "billingMonth": "2021-12-14",
      "totalMLoadMwh": 1357.9876,
      "totalSubzoneMeterBusLoadMwh": 1327.9876,
      "authorizedSubzoneMeterBusLoadMwh": 1327.9876,
      "totalSubzoneLoadAbsoluteDeltaMwh": 30.0000,
      "subzoneLoadValidForAllHours": false,
      "totalBusCount": 77,
      "authorizedBusCount": 77
    },
    {
      "mismatchDetails": [
        {

```

```
    "dateHour": "2021-12-01T07:00:00-05:00",
    "mloadMwh": 45.5555,
    "meterBusLoadMwh": 55.5555,
    "absoluteDeltaMwh": 10.0000
  },
  {
    "dateHour": "2021-12-12T15:00:00-05:00",
    "mloadMwh": 45.5555,
    "meterBusLoadMwh": 55.5555,
    "absoluteDeltaMwh": 10.0000
  },
  {
    "dateHour": "2021-12-23T23:00:00-05:00",
    "mloadMwh": 45.5555,
    "meterBusLoadMwh": 55.5555,
    "absoluteDeltaMwh": 10.0000
  }
],
"buses": [
  {
    "busPtid": 98765,
    "busName": "BUS_B",
    "meterBusLoadMwh": 22246.7531
  },
  ...
]
}
]
```

### 3.9. Retrieval of Generator Performance Data

#### 3.9.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/generatorPerformance>

HTTP Action: GET

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
<b>billingDate</b> <i>[Optional — Either billingDate or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date	Service date of requested data.
<b>startTime</b> <i>[Optional — Either billingDate or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
<b>endTime</b> <i>[Optional — Either billingDate or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 25 hours after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
genPtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

### 3.9.2. Request Examples

#### 3.9.2.1. Retrieval of Generator Performance Data Request Example #1

- Request all data for one day (December 20, 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/generatorPerformance?billingDate=2021-12-20
```

#### *HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

#### 3.9.2.2. Retrieval of Generator Performance Data Request Example #2

- Request all data for one hour (December 20, 2021 08:00:00–08:59:59 EST)

```
GET https://apitest.nyiso.com/finance/metering/v1/generatorPerformance?startTime=2021-12-20T08:00:00-
```

05:00&endTime=2021-12-20T08:59:59-05:00

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic encodedCredential  
 Cache-Control: no-cache

**3.9.2.3. Retrieval of Generator Performance Data Request Example #3**

- Request all data for one day (December 20, 2021)
- Optional generator PTID provided
- Optional user request ID provided

GET https://apitest.nyiso.com/finance/metering/v1/transmissionOwnerLoad?billingDate=2021-12-20&genPtid=345678&userRequestId=MyRequest\_ABCD-001

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic encodedCredential  
 Cache-Control: no-cache

**3.9.3. Response**

*HTTP Headers:*

Content-Type: application/json

*Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingDate <i>[Always]</i>	ISO-8601 Date	Service date of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.genPtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
generatorPerformanceDetails <i>[If any generator performance detail records are returned]</i>	Array of Generator Performance Detail objects	Performance detail records for generators
generatorPerformanceDetails.genPtId <i>[For each generator performance detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator
generatorPerformanceDetails.generatorName <i>[For each generator performance detail record]</i>	String	Generator name
generatorPerformanceDetails.timestamp <i>[For each generator performance detail record]</i>	ISO-8601 Date / Time	RTD Interval end timestamp
generatorPerformanceDetails.avgActualMw <i>[For each generator performance detail record]</i>	Number	Average actual power for the given generator and RTD interval
generatorPerformanceDetails.avgAgcDesiredMw <i>[For each generator performance detail record]</i>	Number	Average Automatic Generation Control (AGC) desired basepoint for the given generator and RTD interval
generatorPerformanceDetails.avgRtdRampedMw <i>[For each generator performance detail record]</i>	Number	Average ramped basepoint for the given generator and RTD interval

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
generatorPerformanceDetails.regulationPerformanceIndex <i>[For each generator performance detail record]</i>	Number	Regulation Performance Index for the given generator and RTD interval
generatorPerformanceDetails.avgPositiveControlErrorMw <i>[For each generator performance detail record]</i>	Number	Average Positive Control Error for the given generator and RTD interval
generatorPerformanceDetails.avgNegativeControlErrorMw <i>[For each generator performance detail record]</i>	Number	Average Negative Control Error for the given generator and RTD interval
generatorPerformanceDetails.inServiceFlag <i>[For each generator performance detail record]</i>	String "Y" or "N"	In Service Flag for the given generator and RTD interval. If "Y", the given generator was in service during the RTD interval; otherwise, "N".
generatorPerformanceDetails.onControlFlag <i>[For each generator performance detail record]</i>	String "Y" or "N"	On Control Flag for the given generator and RTD interval. If "Y", the given generator was on control (i.e. regulating) during the RTD interval; otherwise, "N".
generatorPerformanceDetails.iprOutputLimitFlag <i>[For each generator performance detail record]</i>	String "Y" or "N"	Intermittent Power Resource (IPR) Output Limit Flag for the given generator and RTD interval. If "Y", the given generator was instructed to limit output during the RTD interval; otherwise, "N".
generatorPerformanceDetails.avgActualInjectionMw <i>[For each generator performance detail record]</i>	Number	Average actual injection for the given generator and RTD interval

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
generatorPerformanceDetails.avgActualWithdrawalMw <i>[For each generator performance detail record]</i>	Number	Average actual withdrawal for the given generator and RTD interval
generatorPerformanceDetails.avgActualDemandReductionMw <i>[For each generator performance detail record]</i>	Number	Average actual demand reduction for the given generator and RTD interval
generatorPerformanceDetails.reservePerformanceIndex <i>[For each generator performance detail record]</i>	Number	Reserve Performance Index for the given generator and RTD interval
generatorPerformanceDetails.totalRegulationMovementMw <i>[For each generator performance detail record]</i>	Number	Total regulation movement for the given generator and RTD interval

### 3.9.4. Response Examples

#### 3.9.4.1. Retrieval of Generator Performance Data Response Example #1

- One service hour requested

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "generatorPerformanceDetails": [
    {
      "genPtid": 345678,
      "generatorName": "GEN_XYZ_A",
      "timestamp": "2021-12-14T02:35:00-05:00",
      "avgActualMw": 246.7531,
      "avgAgcDesiredMw": 247.0,
      "avgRtdRampedMw": 246.8750,
      "regulationPerformanceIndex": 0.9999,
      "avgPositiveControlError": 0.1234,
      "avgNegativeControlError": -0.0246,
      "inServiceFlag": "Y",
      "onControlFlag": "Y",
      "iprOutputLimitFlag": "Y",
      "avgActualInjectionMw": 246.7531,
      "avgActualWithdrawalMw": 0.0000,
      "avgActualDemandReductionMw": 0.0000,
      "reservePerformanceIndex": 0.9987,
      "totalRegulationMovementMw": 5.4321
    }
  ]
}
```

```
  },  
  ...  
]  
}
```

### 3.10. Minimum Oil Burn Event Transmission Owner Data Submission

Transmission Owners in the Minimum Oil Burn program can use the Minimum Oil Burn Event Transmission Owner Data Submission service in the Finance API to provide Minimum Oil Burn Event data to the NYISO.

Data may be submitted (including both new entries and updates) in accordance with the Lock-Down Schedules posted on the NYISO website. Upon submission, all data is validated, and if any records fail validation, the entire request will be rejected. All validation errors found will be included with the response.

#### 3.10.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/minOilBurnEvent/transmissionOwner>

HTTP Action: POST

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

Request Body Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
submissionParameters <i>[Optional]</i>	Array of Submission Parameters	User-provided parameters for the submission

Parameter <i>[Required / Optional?]</i>	Data Type	Description
<code>submissionParameters.userRequestId</code> <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Optional]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Optional]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to true if not provided.
<code>eventTransmissionOwnerDetails</code>	Array of Minimum Oil Burn Event Transmission Owner Detail objects	Minimum Oil Burn event data records for transmission owners
<code>eventTransmissionOwnerDetails.genPtId</code> <i>[Required for each minimum oil burn event transmission owner detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator

Parameter <i>[Required / Optional?]</i>	Data Type	Description
eventTransmissionOwnerDetails.transmissionOwnerStartTime <i>[Required for each minimum oil burn event transmission owner detail record]</i>	ISO-8601 Date / Time	Transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event
eventTransmissionOwnerDetails.transmissionOwnerEndTime <i>[Required for each minimum oil burn event transmission owner detail record]</i>	ISO-8601 Date / Time	Transmission owner-requested end time for the given generator to respond to a Minimum Oil Burn event. The time period (between the transmission owner start and end time) may extend over multiple service days; however, in that case, the API will automatically store the data as multiple separate records split by service day in order to facilitate the daily settlement cycle. (See Section 3.10.2.2 for an example request, and Section 3.10.4.5 for an example response.)

### 3.10.2. Request Examples

#### 3.10.2.1. Minimum Oil Burn Event Transmission Owner Data Submission Request Example #1

- Transmission owner data for one Minimum Oil Burn event submitted
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent/generator>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "eventTransmissionOwnerDetails": [
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "transmissionOwnerEndTime": "2021-12-14T06:00:00-05:00"
    }
  ]
}
```

### 3.10.2.2. Minimum Oil Burn Event Transmission Owner Data Submission Request Example #2

- Transmission owner data for one Minimum Oil Burn event submitted
- Minimum Oil Burn Event spans multiple service days
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent/generator>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "eventTransmissionOwnerDetails": [
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "transmissionOwnerEndTime": "2021-12-16T06:00:00-05:00"
    }
  ]
}
```

### 3.10.3. Response

#### HTTP Headers:

```
Content-Type: application/json
```

*Response Body Parameters:*

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
<code>submissionParameters</code> <i>[Always]</i>	Submission Parameters object	User-provided parameters for the submission
<code>submissionParameters.userName</code> <i>[Always]</i>	String	User name for the user submitting the request
<code>submissionParameters.userRequestId</code> <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Always]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Always]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to false if not provided.

Parameter <i>[Provided...?]</i>	Data Type	Description
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
requestSummary <i>[Always]</i>	Summary Object	Processing summary for the request
requestSummary.eventTransmissionOwnerDetails <i>[Always]</i>	Minimum Oil Burn Event Transmission Owner Summary object	Processing summary for Minimum Oil Burn event transmission owner data submitted in the request
requestSummary.eventTransmissionOwnerDetails.submitted <i>[Always]</i>	Integer	Number of Minimum Oil Burn event transmission owner data records submitted in the request
requestSummary.eventTransmissionOwnerDetails.passedValidation <i>[Always]</i>	Integer	Number of Minimum Oil Burn event transmission owner data records that passed validation in the request
requestSummary.eventTransmissionOwnerDetails.failedValidation <i>[Always]</i>	Integer	Number of Minimum Oil Burn event transmission owner data records that failed validation in the request

Parameter <i>[Provided...?]</i>	Data Type	Description
requestSummary.eventTransmissionOwnerDetails.accepted <i>[Always]</i>	Integer	Number of Minimum Oil Burn event transmission owner data records accepted in the request
requestSummary.eventTransmissionOwnerDetails.rejected <i>[Always]</i>	Integer	Number of Minimum Oil Burn event transmission owner data records rejected in the request
failedValidation <i>[If any records have failed validation]</i>	Failed Validation Object	Object containing all Minimum Oil Burn event generator data records that failed validation
failedValidation.eventTransmissionOwnerDetails <i>[If any records have that failed validation for minimum oil burn event transmission owner details]</i>	Array of Minimum Oil Burn Event Generator Detail objects, including Errors	Minimum Oil Burn event generator data records that failed validation
failedValidation.eventTransmissionOwnerDetails.genPtId <i>[For each minimum oil burn event transmission owner detail record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given generator
failedValidation.eventTransmissionOwnerDetails.transmissionOwnerStartTime <i>[For each minimum oil burn event transmission owner detail record that failed validation]</i>	ISO-8601 Date / Time	Submitted transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event
failedValidation.eventTransmissionOwnerDetails.transmissionOwnerEndTime <i>[For each minimum oil burn event transmission owner detail record that failed validation]</i>	ISO-8601 Date / Time	Submitted transmission owner-requested end time for the given generator to respond to a Minimum Oil Burn event

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>failedValidation.eventTransmissionOwnerDetails.errors</code> <i>[For each minimum oil burn event transmission owner detail record that failed validation]</i>	Array of Strings	Errors for the Minimum Oil Burn event generator data record that failed validation
<code>accepted</code> <i>[If the includeAcceptedDataInResponse property was true, and any records have been accepted]</i>	Accepted Object	Object containing all accepted Minimum Oil Burn event generator data records
<code>accepted.eventTransmissionOwnerDetails</code> <i>[If any records have been accepted for minimum oil burn event transmission owner details]</i>	Array of Minimum Oil Burn Event Generator Detail objects	Accepted Minimum Oil Burn event generator data records
<code>accepted.eventTransmissionOwnerDetails.genPtId</code> <i>[For each accepted minimum oil burn event transmission owner detail record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given generator
<code>accepted.eventTransmissionOwnerDetails.transmissionOwnerStartTime</code> <i>[For each accepted minimum oil burn event transmission owner detail record]</i>	ISO-8601 Date / Time	Accepted transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event

Parameter <i>[Provided...?]</i>	Data Type	Description
<p>accepted.eventTransmissionOwnerDetails.transmissionOwnerEndTime</p> <p><i>[For each accepted minimum oil burn event transmission owner detail record]</i></p>	<p>ISO-8601 Date / Time</p>	<p>Accepted transmission owner-requested end time for the given generator to respond to a Minimum Oil Burn event. The time period (between the transmission owner start and end time) may extend over multiple service days; however, in that case, the API will automatically store the data as multiple separate records split by service day in order to facilitate the daily settlement cycle. (See Section 3.10.2.2 for an example request, and Section 3.10.4.5 for an example response.)</p>
<p>accepted.eventTransmissionOwnerDetails.transmissionOwnerEventValidationStatus</p> <p><i>[For each accepted minimum oil burn event transmission owner detail record]</i></p>	<p>String</p>	<p>“Not Validated”: The generator has not yet submitted event data for the same transmission owner start time.</p> <p>“Pass Validation”: The generator has submitted event data for the same transmission owner start time</p>

### 3.10.4. Response Examples

#### 3.10.4.1. Minimum Oil Burn Event Transmission Owner Data Submission Response Example #1

- All records passed validation and therefore accepted
- includeAcceptedDataInResponse = true
- The Transmission Owner data is being submitted before the Generator data for the same event
  - As a result, the Transmission Owner Event Validation Status is set to “Not Validated”

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventTransmissionOwnerDetails": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    }
  },
  "accepted": {
    "eventTransmissionOwnerDetails": [
      {
```

```
"genPtid": 345678,  
  "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",  
  "transmissionOwnerEndTime": "2021-12-14T06:00:00-05:00",  
  "transmissionOwnerEventValidationStatus": "Not Validated"  
}  
]  
}  
}
```

#### 3.10.4.2. Minimum Oil Burn Event Transmission Owner Data Submission Response Example #2

- All records passed validation and therefore accepted
- includeAcceptedDataInResponse = true
- The Transmission Owner data is being submitted after the Generator data for the same Transmission Owner Start Time
  - As a result, the Transmission Owner Event Validation Status is set to “Pass Validation”

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventTransmissionOwnerDetails": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    }
  },
  "accepted": {
    "eventTransmissionOwnerDetails": [
      {
        "genPtid": 345678,
        "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
```

```
    "transmissionOwnerEndTime": "2021-12-14T06:00:00-05:00",  
    "transmissionOwnerEventValidationStatus": "Pass Validation"  
  }  
]  
}  
}
```

### 3.10.4.3. Minimum Oil Burn Event Transmission Owner Data Submission Response Example #3

- One record failed validation, and therefore all records rejected
- includeAcceptedDataInResponse = true, but as all records are rejected, no accepted data is included

#### HTTP Status:

400 Bad Request

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventTransmissionOwnerDetails": {
      "submitted": 1,
      "passedValidation": 0,
      "failedValidation": 1,
      "accepted": 0,
      "rejected": 1
    }
  },
  "failedValidation": {
    "eventTransmissionOwnerDetails": {
      {
        "genPtid": 222222,
        "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
        "transmissionOwnerEndTime": "2021-12-14T06:00:00-05:00",
        "errors": [
          "M00001: This is an example message"
        ]
      }
    }
  ]
}
```

```
}

```

#### 3.10.4.4. Minimum Oil Burn Event Transmission Owner Data Submission Response Example #4

- All records passed validation,
- doCommit = false, and therefore no records are accepted, and all rejected
- includeAcceptedDataInResponse = false by default

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": false,
    "doCommit": false
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventTransmissionOwnerDetails": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 0
    }
  }
}
```

#### 3.10.4.5. Minimum Oil Burn Event Transmission Owner Data Submission Response Example #5

- All records passed validation and therefore accepted
- Minimum Oil Burn Event spans multiple service days
- includeAcceptedDataInResponse = true
- The Transmission Owner data is being submitted before the Generator data for the same event
  - As a result, the Transmission Owner Event Validation Status is set to “Not Validated”

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventTransmissionOwnerDetails": {
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    }
  },
  "accepted": {
    "eventTransmissionOwnerDetails": [
      {
```

```
"genPtid": 345678,  
  "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",  
  "transmissionOwnerEndTime": "2021-12-14T23:59:59-05:00",  
  "transmissionOwnerEventValidationStatus": "Not Validated"  
},  
{  
  "genPtid": 345678,  
  "transmissionOwnerStartTime": "2021-12-15T00:00:00-05:00",  
  "transmissionOwnerEndTime": "2021-12-15T06:00:00-05:00",  
  "transmissionOwnerEventValidationStatus": "Not Validated"  
},  
{  
  "genPtid": 345678,  
  "transmissionOwnerStartTime": "2021-12-16T00:00:00-05:00",  
  "transmissionOwnerEndTime": "2021-12-16T06:00:00-05:00",  
  "transmissionOwnerEventValidationStatus": "Not Validated"  
}  
]  
}  
}
```

### 3.11. Minimum Oil Burn Event Generator Data Submission

Market Participants with Generators in the Minimum Oil Burn program can use the Minimum Oil Burn Event Generator Data Submission service in the Metering API to provide Minimum Oil Burn Event data to the NYISO.

Data may be submitted (including both new entries and updates) in accordance with the Lock-Down Schedules posted on the NYISO website. Upon submission, all data is validated, and if any records fail validation, the entire request will be rejected. All validation errors found will be included with the response.

Note: In the event that a Generator trips and restarts later during the same day during an G.2/G.3 call, the Generator Owner is requested to enter the entire time period (including the hours the unit was offline) and the fuel consumption for the entire time period, rather than submitting multiple records for the same service day.

#### 3.11.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/minOilBurnEvent/generator>

HTTP Action: POST

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

Request Body Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
submissionParameters <i>[Optional]</i>	Array of Submission Parameters	User-provided parameters for the submission

Parameter <i>[Required / Optional?]</i>	Data Type	Description
<code>submissionParameters.userRequestId</code> <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Optional]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Optional]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to true if not provided.
<code>eventGeneratorDetails</code>	Array of Minimum Oil Burn Event Generator Detail objects	Minimum Oil Burn event data records for generators
<code>eventGeneratorDetails.genPtId</code> <i>[Required for each minimum oil burn event generator detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator
<code>eventGeneratorDetails.transmissionOwnerStartTime</code> <i>[Required for each minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	Transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event

Parameter <i>[Required / Optional?]</i>	Data Type	Description
eventGeneratorDetails.generatorStartTime <i>[Required for each minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	Start time for the given generator's engagement for the given Minimum Oil Burn event
eventGeneratorDetails.generatorEndTime <i>[Required for each minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	End time for the given generator's engagement for the given Minimum Oil Burn event. The time period (between the generator start and end time) must be contained within a single service day. If the original event called by the transmission owner spanned multiple service days, the generator needs to enter records for each service day of the event separately in order to facilitate the daily settlement cycle. (See Section 3.11.2.2 for an example request, and Section 3.11.4.6 for an example response.)
eventGeneratorDetails.fuelConsumptionBarrels <i>[Required for each minimum oil burn event generator detail record]</i>	Number, up to two decimal places  $0.00 \leq x < 1,000.00$	Total fuel consumption (in barrels) for the given generator during the given Minimum Oil Burn event

### 3.11.2. Request Examples

#### 3.11.2.1. Minimum Oil Burn Event Generator Data Submission Request Example #1

- Generator data for one Minimum Oil Burn event submitted
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent/generator>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "eventGeneratorDetails": [
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "generatorStartTime": "2021-12-14T02:15:00-05:00",
      "generatorEndTime": "2021-12-14T05:45:00-05:00",
      "fuelConsumptionBarrels": 246.75
    }
  ]
}
```

### 3.11.2.2. Minimum Oil Burn Event Generator Data Submission Request Example #2

- Generator data for one Minimum Oil Burn event submitted
- Event called by Transmission Owner spans multiple service days, from 2021-12-14 02:00:00 EST to 2021-12-16 06:00:00 EST
  - Finance API splits event into Transmission Owner Minimum Oil Burn Event records for each service day
  - Generator submits separate records for each service day
- Optional submission parameters included

POST <https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent/generator>

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
Content-Type: application/json
```

#### Request Body:

```
{
  "submissionParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": false
  },
  "eventGeneratorDetails": [
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "generatorStartTime": "2021-12-14T02:15:00-05:00",
      "generatorEndTime": "2021-12-14T23:59:59-05:00",
      "fuelConsumptionBarrels": 246.75
    },
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-15T00:00:00-05:00",
      "generatorStartTime": "2021-12-15T00:00:00-05:00",
      "generatorEndTime": "2021-12-15T23:59:59-05:00",
    }
  ]
}
```

```

    "fuelConsumptionBarrels": 357.86
  },
  {
    "genPtid": 345678,
    "transmissionOwnerStartTime": "2021-12-16T00:00:00-05:00",
    "generatorStartTime": "2021-12-16T00:00:00-05:00",
    "generatorEndTime": "2021-12-16T05:45:00-05:00",
    "fuelConsumptionBarrels": 135.64
  }
]
}

```

### 3.11.3. Response

*HTTP Headers:*

Content-Type: application/json

*Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
submissionParameters <i>[Always]</i>	Submission Parameters object	User-provided parameters for the submission
submissionParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
submissionParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>submissionParameters.includeAcceptedDataInResponse</code> <i>[Always]</i>	Boolean (true / false)	If true, include accepted data in the response; otherwise, only summary data and errors (if any) are provided in the response. Defaults to false.
<code>submissionParameters.doCommit</code> <i>[Always]</i>	Boolean (true / false)	If false, only validates data, but does not store any records; designed primarily for use in testing. Defaults to false if not provided.
<code>nyisoRequestId</code> <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
<code>requestTimestamp</code> <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
<code>requestSummary</code> <i>[Always]</i>	Summary Object	Processing summary for the request
<code>requestSummary.eventGeneratorDetails</code> <i>[Always]</i>	Minimum Oil Burn Event Generator Summary object	Processing summary for Minimum Oil Burn event generator data submitted in the request
<code>requestSummary.eventGeneratorDetails.submitted</code> <i>[Always]</i>	Integer	Number of Minimum Oil Burn event generator data records submitted in the request

<b>Parameter <i>[Provided...?]</i></b>	<b>Data Type</b>	<b>Description</b>
requestSummary.eventGeneratorDetails.passedValidation <i>[Always]</i>	Integer	Number of Minimum Oil Burn event generator data records that passed validation in the request
requestSummary.eventGeneratorDetails.failedValidation <i>[Always]</i>	Integer	Number of Minimum Oil Burn event generator data records that failed validation in the request
requestSummary.eventGeneratorDetails.accepted <i>[Always]</i>	Integer	Number of Minimum Oil Burn event generator data records accepted in the request
requestSummary.eventGeneratorDetails.rejected <i>[Always]</i>	Integer	Number of Minimum Oil Burn event generator data records rejected in the request
failedValidation <i>[If any records have failed validation]</i>	Failed Validation Object	Object containing all Minimum Oil Burn event generator data records that failed validation
failedValidation.eventGeneratorDetails <i>[If any records have that failed validation for minimum oil burn event generator details]</i>	Array of Minimum Oil Burn Event Generator Detail objects, including Errors	Minimum Oil Burn event generator data records that failed validation
failedValidation.eventGeneratorDetails.genPtId <i>[For each minimum oil burn event generator detail record that failed validation]</i>	Integer	Submitted unique NYISO-defined point identifier for a given generator

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>failedValidation.eventGeneratorDetails.transmissionOwnerStartTime</code> <i>[For each minimum oil burn event generator detail record that failed validation]</i>	ISO-8601 Date / Time	Submitted transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event
<code>failedValidation.eventGeneratorDetails.generatorStartTime</code> <i>[For each minimum oil burn event generator detail record that failed validation]</i>	ISO-8601 Date / Time	Submitted start time for the given generator's engagement for the given Minimum Oil Burn event
<code>failedValidation.eventGeneratorDetails.generatorEndTime</code> <i>[For each minimum oil burn event generator detail record that failed validation]</i>	ISO-8601 Date / Time	Submitted end time for the given generator's engagement for the given Minimum Oil Burn event
<code>failedValidation.eventGeneratorDetails.fuelConsumptionBarrels</code> <i>[For each minimum oil burn event generator detail record that failed validation]</i>	Number	Submitted total fuel consumption (in barrels) for the given generator during the given Minimum Oil Burn event
<code>failedValidation.eventGeneratorDetails.errors</code> <i>[For each minimum oil burn event generator detail record that failed validation]</i>	Array of Strings	Errors for the Minimum Oil Burn event generator data record that failed validation
<code>accepted</code> <i>[If the includeAcceptedDataInResponse property was true, and any records have been accepted]</i>	Accepted Object	Object containing all accepted Minimum Oil Burn event generator data records
<code>accepted.eventGeneratorDetails</code> <i>[If any records have been accepted for minimum oil burn event generator details]</i>	Array of Minimum Oil Burn Event Generator Detail objects	Accepted Minimum Oil Burn event generator data records

Parameter <i>[Provided...?]</i>	Data Type	Description
accepted.eventGeneratorDetails.genPtId <i>[For each accepted minimum oil burn event generator detail record]</i>	Integer	Accepted unique NYISO-defined point identifier for a given generator
accepted.eventGeneratorDetails.transmissionOwnerStartTime <i>[For each accepted minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	Accepted transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event
accepted.eventGeneratorDetails.generatorStartTime <i>[For each accepted minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	Accepted start time for the given generator's engagement for the given Minimum Oil Burn event
accepted.eventGeneratorDetails.generatorEndTime <i>[For each accepted minimum oil burn event generator detail record]</i>	ISO-8601 Date / Time	Accepted end time for the given generator's engagement for the given Minimum Oil Burn event. The time period (between the generator start and end time) must be contained within a single service day. If the original event called by the transmission owner spanned multiple service days, the generator needs to enter records for each service day of the event separately in order to facilitate the daily settlement cycle. (See Section 3.11.2.2 for an example request, and Section 3.11.4.6 for an example response.)
accepted.eventGeneratorDetails.fuelConsumptionBarrels <i>[For each accepted minimum oil burn event generator detail record]</i>	Number	Accepted total fuel consumption (in barrels) for the given generator during the given Minimum Oil Burn event

Parameter <i>[Provided...?]</i>	Data Type	Description
<p>accepted.eventGeneratorDetails.generatorEventValidationStatus</p> <p><i>[For each accepted minimum oil burn event generator detail record]</i></p>	String	<p>“Not Validated”: The transmission owner has not yet submitted event data for the same transmission owner start time and service day.</p> <p>“Pass Validation”: The transmission owner has submitted event data for the same transmission owner start time, and the time period submitted by the generator is within the time bounds established by the generator’s engagement time and disengagement time.</p> <p>“Fail Validation”: The transmission owner has submitted event data for the same transmission owner start time; however, the time period submitted by the generator is not within the time bounds established by the generator’s engagement time and disengagement time.</p>

### 3.11.4. Response Examples

#### 3.11.4.1. Minimum Oil Burn Event Generator Data Submission Response Example #1

- All records passed validation and therefore accepted
- includeAcceptedDataInResponse = true
- The Generator data is being submitted before the Transmission Owner data for the same event
  - As a result, the Transmission Owner Event Validation Status is set to “Not Validated”

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      {
        "submitted": 1,
        "passedValidation": 1,
        "failedValidation": 0,
        "accepted": 1,
        "rejected": 0
      }
    ]
  },
  "accepted": {
    "eventGeneratorDetails": [
      {
```

```
"genPtid": 345678,  
  "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",  
  "generatorStartTime": "2021-12-14T02:30:00-05:00",  
  "generatorEndTime": "2021-12-14T05:45:00-05:00",  
  "fuelConsumptionBarrels": 246.75,  
  "generatorEventValidationStatus": "Not Validated"  
}  
]  
}
```

#### 3.11.4.2. Minimum Oil Burn Event Generator Data Submission Response Example #2

- All records passed validation and therefore accepted
- `includeAcceptedDataInResponse = true`
- The Generator data is being submitted after the Transmission Owner data for the same Transmission Owner Start Time
- Further, the Generator Start Time and Generator End Time are within valid bounds
  - Bounds are established by the generator's engagement time and disengagement time
  - As a result, the Transmission Owner Event Validation Status is set to "Pass Validation"

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    ]
  },
  "accepted": {
```

```
"eventGeneratorDetails": [  
  {  
    "genPtid": 345678,  
    "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",  
    "generatorStartTime": "2021-12-14T02:30:00-05:00",  
    "generatorEndTime": "2021-12-14T05:45:00-05:00",  
    "fuelConsumptionBarrels": 246.75,  
    "generatorEventValidationStatus": "Pass validation"  
  }  
]  
}
```

#### 3.11.4.3. Minimum Oil Burn Event Generator Data Submission Response Example #3

- All records passed validation and therefore accepted
- `includeAcceptedDataInResponse = true`
- The Generator data is being submitted after the Transmission Owner data for the same Transmission Owner Start Time
- Further, the Generator Start Time and Generator End Time are not within valid bounds
  - Bounds are established by the generator's engagement time and disengagement time
  - As a result, the Transmission Owner Event Validation Status is set to "Fail Validation"

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 1,
      "rejected": 0
    ]
  },
  "accepted": {
```

```
"eventGeneratorDetails": [  
  {  
    "genPtid": 345678,  
    "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",  
    "generatorStartTime": "2021-12-14T02:30:00-05:00",  
    "generatorEndTime": "2021-12-14T08:45:00-05:00",  
    "fuelConsumptionBarrels": 246.75,  
    "generatorEventValidationStatus": "Fail validation"  
  }  
]  
}
```

#### 3.11.4.4. Minimum Oil Burn Event Generator Data Submission Response Example #4

- One record failed validation, and therefore all records rejected
- includeAcceptedDataInResponse = true, but as all records are rejected, no accepted data is included

##### HTTP Status:

400 Bad Request

##### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      {
        "submitted": 1,
        "passedValidation": 0,
        "failedValidation": 1,
        "accepted": 0,
        "rejected": 1
      }
    ]
  },
  "eventGeneratorDetails": [
    {
      "genPtid": 222222,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "generatorStartTime": "2021-12-14T02:30:00-05:00",
      "generatorEndTime": "2021-12-14T05:45:00-05:00",
      "fuelConsumptionBarrels": 246.75,
      "errors": [
        "M00001: This is an example message"
      ]
    }
  ]
}
```

```
}
}
```

#### 3.11.4.5. Minimum Oil Burn Event Generator Data Submission Response Example #5

- All records passed validation,
- doCommit = false, and therefore no records are accepted, and all rejected
- includeAcceptedDataInResponse = false by default

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": false,
    "doCommit": false
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      "submitted": 1,
      "passedValidation": 1,
      "failedValidation": 0,
      "accepted": 0,
      "rejected": 0
    ]
  }
}
```

#### 3.11.4.6. Minimum Oil Burn Event Generator Data Submission Response Example #6

- All records passed validation and therefore accepted
- `includeAcceptedDataInResponse = true`
- The Generator data is being submitted after the Transmission Owner data for the same Transmission Owner Start Time
- Event called by Transmission Owner spans multiple service days, from 2021-12-14 02:00:00 EST to 2021-12-16 06:00:00 EST
  - Finance API splits event into Transmission Owner Minimum Oil Burn Event records for each service day
  - Generator submits separate records for each service day
- Further, the Generator Start Time and Generator End Time are within valid bounds
  - Bounds are established by the generator's engagement time and disengagement time
  - As a result, the Transmission Owner Event Validation Status is set to "Pass Validation"

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "submissionParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "includeAcceptedDataInResponse": true,
    "doCommit": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "requestSummary": {
    "eventGeneratorDetails": [
      "submitted": 1,
```

```
    "passedValidation": 1,
    "failedValidation": 0,
    "accepted": 1,
    "rejected": 0
  }
},
"accepted": {
  "eventGeneratorDetails": [
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "generatorStartTime": "2021-12-14T02:30:00-05:00",
      "generatorEndTime": "2021-12-14T23:59:59-05:00",
      "fuelConsumptionBarrels": 246.75,
      "generatorEventValidationStatus": "Pass validation"
    },
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-15T00:00:00-05:00",
      "generatorStartTime": "2021-12-15T00:00:00-05:00",
      "generatorEndTime": "2021-12-15T23:59:59-05:00",
      "fuelConsumptionBarrels": 357.86,
      "generatorEventValidationStatus": "Pass validation"
    },
    {
      "genPtid": 345678,
      "transmissionOwnerStartTime": "2021-12-16T00:00:00-05:00",
      "generatorStartTime": "2021-12-16T00:00:00-05:00",
      "generatorEndTime": "2021-12-16T05:45:00-05:00",
      "fuelConsumptionBarrels": 135.64,
      "generatorEventValidationStatus": "Pass validation"
    }
  ]
}
}
```

### 3.12. Retrieval of Minimum Oil Burn Event Data

#### 3.12.1. Request

URL: <https://api.nyiso.com/finance/metering/v1/minOilBurnEvent>

HTTP Action: GET

HTTP Headers:

Accept: application/json  
 Accept-Encoding: gzip, deflate  
 Authorization: Basic *encodedCredential*  
 Cache-Control: no-cache

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
endTime <i>[Optional — Either billingMonth or startTime and endTime must be provided, but not both]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021). The end time must be no earlier than the start time, and no later than 31 days after the start time.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
version <i>[Optional]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
genPtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

### 3.12.2. Request Examples

#### 3.12.2.1. Retrieval of Minimum Oil Burn Event Data Request Example #1

- Request all data for one month (December 2021)

```
GET https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent?billingMonth=2021-12
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.12.2.2. Retrieval of Minimum Oil Burn Event Data Request Example #2

- Request all data for one month (December 2021)
- Optional generator PTID provided
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/metering/v1/minOilBurnEvent?billingMonth=2021-12&genPtid=345678&userRequestId=MyRequest_ABCD-001
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 3.12.3. Response

*HTTP Headers:*

```
Content-Type: application/json
```

*Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.version <i>[If provided in request]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.genPtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

Parameter <i>[Provided...?]</i>	Data Type	Description
nyisoRequestId <i>[Always]</i>	UUID String  36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
minOilBurnEvents <i>[If any minimum oil burn event detail records are returned]</i>	Array of Minimum Oil Burn Event Detail objects	Minimum Oil Burn event detail records for generators
minOilBurnEvents.genPtId <i>[For each minimum oil burn event detail record]</i>	Integer	Unique NYISO-defined point identifier for a given generator
minOilBurnEvents.generatorName <i>[For each minimum oil burn event detail record]</i>	String	Generator name
minOilBurnEvents.billingDate <i>[For each minimum oil burn event detail record]</i>	ISO-8601 Date	Billing / service day for metering detail
minOilBurnEvents.version <i>[For each minimum oil burn event detail record]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
minOilBurnEvents.transmissionOwnerStartTime <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	Transmission owner-requested start time for the given generator to respond to a Minimum Oil Burn event

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>minOilBurnEvents.transmissionOwnerEndTime</code> <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	Transmission owner-requested end time for the given generator to respond to a Minimum Oil Burn event
<code>minOilBurnEvents.generatorStartTime</code> <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	Start time for the given generator's engagement for the given Minimum Oil Burn event
<code>minOilBurnEvents.generatorEndTime</code> <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	End time for the given generator's engagement for the given Minimum Oil Burn event
<code>minOilBurnEvents.fuelConsumptionBarrels</code> <i>[For each minimum oil burn event record]</i>	Number	Total fuel consumption (in barrels) for the given generator during the given Minimum Oil Burn event
<code>minOilBurnEvents.transmissionOwnerEventValidationStatus</code> <i>[For each minimum oil burn event record]</i>	String	<p>"Not Validated": The generator has not yet submitted event data for the same transmission owner start time.</p> <p>"Pass Validation": The generator has submitted event data for the same transmission owner start time</p>

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>minOilBurnEvents.generatorEventValidationStatus</code> <i>[For each minimum oil burn event record]</i>	String	<p>“Not Validated”: The transmission owner has not yet submitted event data for the same transmission owner start time and service day.</p> <p>“Pass Validation”: The transmission owner has submitted event data for the same transmission owner start time, and the time period submitted by the generator is within the time bounds established by the generator’s engagement time and disengagement time.</p> <p>“Fail Validation”: The transmission owner has submitted event data for the same transmission owner start time; however, the time period submitted by the generator is not within the time bounds established by the generator’s engagement time and disengagement time.</p>
<code>minOilBurnEvents.transmissionOwnerUpdateTime</code> <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	Update time of transmission owner data for the given Minimum Oil Burn event

Parameter <i>[Provided...?]</i>	Data Type	Description
<code>minOilBurnEvents.transmissionOwnerUpdateUser</code> <i>[For each minimum oil burn event record]</i>	String	Transmission owner user who most recently data for the given Minimum Oil Burn event; null if no data has yet been submitted by the transmission owner for the given Minimum Oil Burn event.
<code>minOilBurnEvents.generatorUpdateTime</code> <i>[For each minimum oil burn event record]</i>	ISO-8601 Date / Time	Update time of generator data for the given Minimum Oil Burn event
<code>minOilBurnEvents.generatorUpdateUser</code> <i>[For each minimum oil burn event record]</i>	String	Generator user who most recently data for the given Minimum Oil Burn event; null if no data has yet been submitted by the generator for the given Minimum Oil Burn event.
<code>minOilBurnEvents.billedFlag</code> <i>[For each minimum oil burn event record]</i>	String "Y" or "N"	If "Y", data has been used to calculate settlements; if "N", data has not yet been used.

### 3.12.4. Response Examples

#### 3.12.4.1. Retrieval of Generator Performance Data Response Example #1

- One service hour requested (retrieval is based on Transmission Owner Start Time)

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-14T02:00:00-05:00",
    "endTime": "2021-12-14T02:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "minOilBurnEvents": [
    {
      "billingDate": "2021-12-14",
      "genPtid": 345678,
      "generatorName": "GEN_XYZ_A",
      "version": 0,
      "transmissionOwnerStartTime": "2021-12-14T02:00:00-05:00",
      "transmissionOwnerEndTime": "2021-12-14T06:00:00-05:00",
      "generatorStartTime": "2021-12-14T02:30:00-05:00",
      "generatorEndTime": "2021-12-14T05:45:00-05:00",
      "fuelConsumptionBarrels": 123.4567,
      "transmissionOwnerEventValidationStatus": " Pass Validation ",
      "generatorEventValidationStatus": "Pass Validation",
      "transmissionOwnerUpdateTime": "2021-12-19T12:34:00-05:00",
      "transmissionOwnerUpdateUser": "TO_USER",
      "generatorUpdateTime": "2021-12-20T13:57:00-05:00",
      "generatorUpdateUser": "GEN_USER",
      "billedFlag": "N"
    }
  ]
}
```

}

## 4. Using the Settlements API

### 4.1. Retrieval of Station Power Detail Data

#### 4.1.1. Request

*URL:* `https://api.nyiso.com/finance/settlements/v1/stationPower`

*HTTP Action:* GET

*HTTP Headers:*

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

*URL Parameters:*

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Required]</i>	ISO-8601 Month	Service month of requested data.
version <i>[Required]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
genPtId <i>[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

#### 4.1.2. Request Examples

##### 4.1.2.1. Retrieval of Station Power Detail Data Request Example #1

- Request all data for one month (December 2021) and version (v2)

```
GET https://apitest.nyiso.com/finance/settlements/v1/stationPower?billingMonth=2021-12&version=2
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

#### 4.1.2.2. Retrieval of Station Power Detail Data Request Example #2

- Request all data for one month (December 2021) and version (v2)
- Optional generator PTID provided
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/settlements/v1/stationPower? billingMonth=2021-12&version=2&genPtid=567890&userRequestId=MyRequest_ABCD-001
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

#### 4.1.3. Response

#### *HTTP Headers:*

```
Content-Type: application/json
```

#### *Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[Always]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.version <i>[Always]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.genPtId <i>[If provided in request]</i>	Integer	Unique NYISO-defined point identifier for a given generator. If defined, only data for specified generator PTIDs shall be returned. By default, no PTID filtering is applied.

<b>Parameter <i>[Provided...?]</i></b>	<b>Data Type</b>	<b>Description</b>
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
stationPowerDetails <i>[If any station power detail records are returned]</i>	Array of Station Power Detail objects	Station power settlements detail records for generators
stationPowerDetails.dateHour <i>[For each station power detail record]</i>	ISO-8601 Date / Time	Service hour for station power settlements data  Note: Records for the last day of the month with the time portion of the dateHour equaling "23:59:59" represent the monthly total of all Station Power usage for the given station power generator.
stationPowerDetails.billingDate <i>[For each station power detail record]</i>	ISO-8601 Date	Billing / service day for station power settlements data
stationPowerDetails.genPtId <i>[For each station power detail record]</i>	Integer	Unique NYISO-defined point identifier for a given station power generator
stationPowerDetails.generatorName <i>[For each station power detail record]</i>	String	Station power generator name

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
stationPowerDetails.stationPowerLoadBidMw <i>[For each station power detail record]</i>	Number	Station power load bid for the given station power generator and service hour
stationPowerDetails.stationPowerLoadForecastMw <i>[For each station power detail record]</i>	Number	Station power load forecast for the given station power generator and service hour
stationPowerDetails.meterStationPowerBusLoadMwh <i>[For each station power detail record]</i>	Number	Submitted metered station power bus load for the given station power generator and service hour
stationPowerDetails.netGenerationMwh <i>[For each station power detail record]</i>	Number	Net generation for the given station power generator and service hour
stationPowerDetails.thirdPartyLsePtId <i>[For each station power detail record]</i>	Integer	Unique NYISO-defined point identifier for the third-party load serving entity (LSE) associated with a given station power generator
stationPowerDetails.thirdPartyLseName <i>[For each station power detail record]</i>	String	LSE name for the third-party LSE associated with a given station power generator
stationPowerDetails.thirdPartyLoadAllocationMwh <i>[For each station power detail record]</i>	Number	Third-party load allocation for the third-party LSE associated with a given station power generator and service hour
stationPowerDetails.thirdPartyStationPowerSettlementDollars <i>[For each station power detail record]</i>	Number	Third-party station power settlement for the third-party LSE associated with a given station power generator and service hour

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
stationPowerDetails.ancillaryServiceChargeDollars <i>[For each station power detail record]</i>	Number	Ancillary service charge (except Annual Budget and FERC Fees) for the given station power generator and service hour
stationPowerDetails.remoteSelfSupplyEnergyMwh <i>[For each station power detail record]</i>	Number	Remote self-supply energy for the given station power generator and service hour
stationPowerDetails.ntacChargeDollars <i>[For each station power detail record]</i>	Number	NTAC charge for the given station power generator and service hour
stationPowerDetails.rateScheduleOneNetAnnualBudgetChargeDollars <i>[For each station power detail record]</i>	Number	Rate Schedule One net annual budget charge for the given station power generator and service hour
stationPowerDetails.rateScheduleOneFercFeeChargeDollars <i>[For each station power detail record]</i>	Number	Rate Schedule One FERC fee charge for the given station power generator and service hour

#### 4.1.4. Response Examples

##### 4.1.4.1. Retrieval of Station Power Detail Data Response Example #1

- One service month requested
- Includes sample summary record with end-of-month/end-of-day dateHour ("23:59:59")

##### HTTP Status:

200 OK

##### HTTP Headers:

Content-Type: application/json

##### Request Body:

```
{
  "requestParameters": {
    "billingMonth": "2021-12",
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startTime": "2021-12-01T00:00:00-05:00",
    "endTime": "2021-12-31T23:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "stationPowerDetails": [
    {
      "dateHour": "2021-12-01T00:00:00-05:00",
      "billingDate": "2021-12-01",
      "genPtid": 345678,
      "generatorName": "GEN_XYZ_A",
      "stationPowerLoadBidMw": 5.5,
      "stationPowerLoadForecastMw": 5.3,
      "meterStationPowerBusLoadMwh": 5.3432,
      "netGenerationMwh": 1.1111,
      "thirdPartyLsePtid": 7777,
      "thirdPartyLseName": "LSE L",
      "thirdPartyLoadAllocationMwh": 1.0001,
      "thirdPartyStationPowerSettlementDollars": 11.22,
      "ancillaryServiceChargeDollars": 4.44,
    }
  ]
}
```

```
"remoteSelfSupplyEnergyMwh": 3.5753,  
"ntacChargeDollars": 1.23,  
"rateScheduleOneNetAnnualBudgetChargeDollars": 0.17,  
"rateScheduleOneFercFeeChargeDollars": 0.07  
},  
...  
{  
  "dateHour": "2021-12-31T23:59:59-05:00",  
  "billingDate": "2021-12-31",  
  "genPtid": 345678,  
  "generatorName": "GEN_XYZ_A",  
  "stationPowerLoadBidMw": 165.5,  
  "stationPowerLoadForecastMw": 159.3,  
  "meterStationPowerBusLoadMwh": 159.3456,  
  "netGenerationMwh": 31.3131,  
  "thirdPartyLsePtid": 7777,  
  "thirdPartyLseName": "LSE L",  
  "thirdPartyLoadAllocationMwh": 31.1001,  
  "thirdPartyStationPowerSettlementDollars": 226.26,  
  "ancillaryServiceChargeDollars": 133.44,  
  "remoteSelfSupplyEnergyMwh": 108.7553,  
  "ntacChargeDollars": 31.23,  
  "rateScheduleOneNetAnnualBudgetChargeDollars": 3.17,  
  "rateScheduleOneFercFeeChargeDollars": 3.07  
},  
...  
]  
}
```

## 5. Using the Invoicing API

### 5.1. Retrieval of Daily Reconciliation Dollar Data

#### 5.1.1. Request

**URL:** `https://api.nyiso.com/finance/invoicing/v1/dailyReconciliation/dollar`

**HTTP Action:** GET

**HTTP Headers:**

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

**URL Parameters:**

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Starting service date of requested data.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
endDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date, and no later than 31 days after the start date.
nyisowide <i>[Optional]</i>	Boolean	If true, response includes NYISO-wide aggregate sums of all Daily Reconciliation values; if false, only includes Daily Reconciliation values for the organization to which the requesting user belongs. Default to false.
version <i>[Required]</i>	Number, up to one decimal place  $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
userRequestId <i>[Optional]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

## 5.1.2. Request Examples

### 5.1.2.1. Retrieval of Daily Reconciliation Dollar Data Request Example #1

- Request all data for one day (December 20, 2021) and version (v2)

```
GET https://apitest.nyiso.com/finance/invoicing/v1/dailyReconciliation/dollar?startDate=2021-12-20&endDate=2021-12-20&version=2
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

### 5.1.2.2. Retrieval of Daily Reconciliation Dollar Data Request Example #2

- Request all data for one day (December 20, 2021) and version (v2)
- Optional NYISO-wide parameter provided
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/invoicing/v1/dailyReconciliation/dollar?startDate=2021-12-20&endDate=2021-12-20&version=2&nyisowide=true&userRequestId=MyRequest_ABCD-001
```

#### HTTP Headers:

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: Basic encodedCredential  
Cache-Control: no-cache
```

## 5.1.3. Response

#### HTTP Headers:

```
Content-Type: application/json
```

*Response Body Parameters:*

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startDate <i>[If provided in request]</i>	ISO-8601 Date	Starting service date of requested data.
requestParameters.endDate <i>[If provided in request]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021).
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.nyisowide <i>[If provided in request]</i>	Boolean	If true, response includes NYISO-wide aggregate sums of all Daily Reconciliation values; if false, only includes Daily Reconciliation values for the organization to which the requesting user belongs. Default to false.
requestParameters.version <i>[Always]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request
requestParameters.userRequestId <i>[If provided in request]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
nyisoRequestId <i>[Always]</i>	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.

Parameter <i>[Provided...?]</i>	Data Type	Description
organizationName <i>[Always]</i>	String	If nyisoWide is true, or the requesting user is a NYISO user, “NYISO-Wide”; otherwise, the organization name for the organization to which the user belongs.
revenues <i>[Always]</i>	Array of Daily Reconciliation Dollar Detail objects	Revenue-side Daily Reconciliation Dollar detail records for generators
revenues.dailyReconciliationCode <i>[For each daily reconciliation dollar detail record]</i>	Integer	Unique NYISO-defined identifier for a given Daily Reconciliation entry
revenues.description <i>[For each daily reconciliation dollar detail record]</i>	String	Daily Reconciliation entry description
revenues.itemValueDollars <i>[For each daily reconciliation dollar detail record if the record is for an individual entry]</i>	Number	Daily Reconciliation entry value for an individual entry
revenues.totalValueDollars <i>[For each daily reconciliation dollar detail record if the record is for a total or subtotal entry]</i>	Number	Daily Reconciliation entry value for a total or subtotal entry
expenditures <i>[Always]</i>	Array of Daily Reconciliation Dollar Detail objects	Revenue-side Daily Reconciliation Dollar detail records for generators
expenditures.dailyReconciliationCode <i>[For each daily reconciliation dollar detail record]</i>	Integer	Unique NYISO-defined identifier for a given Daily Reconciliation entry

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
expenditures.description <i>[For each daily reconciliation dollar detail record]</i>	String	Daily Reconciliation entry description
expenditures.itemValueDollars <i>[For each daily reconciliation dollar detail record if the record is for an individual entry]</i>	Number	Daily Reconciliation entry value for an individual entry
expenditures.totalValueDollars <i>[For each daily reconciliation dollar detail record if the record is for a total or subtotal entry]</i>	Number	Daily Reconciliation entry value for a total or subtotal entry

#### 5.1.4. Response Examples

##### 5.1.4.1. Retrieval of Daily Reconciliation Dollar Data Response Example #1

- One service day requested
- Select data for user's organization (by default)

HTTP Status:

200 OK

HTTP Headers:

Content-Type: application/json

Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startDate": "2021-12-14",
    "endDate": "2021-12-14",
    "startTime": "2021-12-14T00:00:00-05:00",
    "endTime": "2021-12-14T23:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "organizationName": "Billing Org O",
  "revenues": [
    {
      "dailyReconciliationCode": 70101,
      "description": "DAM LSE Internal LBMP Energy",
      "itemValueDollars": 1000000.00
    },
    ...
    {
      "dailyReconciliationCode": 1,
      "description": "DAM LBMP Revenue",
      "totalValueDollars": 1234567.89
    },
    ...
  ]
}
```

```
],  
"expenditures": [  
  {  
    "dailyReconciliationCode": 30101,  
    "description": "DAM Internal PS LBMP Energy Purchases Expenditure",  
    "itemValueDollars": 123456.78  
  },  
  ...  
  {  
    "dailyReconciliationCode": 13,  
    "description": "DAM LBMP Expenditure",  
    "totalValueDollars": 200000.00  
  },  
  ...  
]  
}
```

#### 5.1.4.2. Retrieval of Daily Reconciliation Dollar Data Response Example #2

- One service day requested
- NYISO-wide data requested

##### HTTP Status:

200 OK

##### HTTP Headers:

Content-Type: application/json

##### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startDate": "2021-12-14",
    "endDate": "2021-12-14",
    "startTime": "2021-12-14T00:00:00-05:00",
    "endTime": "2021-12-14T23:59:59-05:00",
    "nyisowide": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "organizationName": "NYISO-wide",
  "revenues": [
    {
      "dailyReconciliationCode": 70101,
      "description": "DAM LSE Internal LBMP Energy",
      "itemValueDollars": 1000000.00
    },
    ...
    {
      "dailyReconciliationCode": 1,
      "description": "DAM LBMP Revenue",
      "totalValueDollars": 1234567.89
    },
    ...
  ],
  "expenditures": [
```

```
{
  "dailyReconciliationCode": 30101,
  "description": "DAM Internal PS LBMP Energy Purchases Expenditure",
  "itemValueDollars": 123456.78
},
...
{
  "dailyReconciliationCode": 13,
  "description": "DAM LBMP Expenditure",
  "totalValueDollars": 200000.00
},
...
]
}
```

## 5.2. Retrieval of Daily Reconciliation MWh Data

### 5.2.1. Request

URL: <https://api.nyiso.com/finance/invoicing/v1/dailyReconciliation/mwh>

HTTP Action: GET

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

URL Parameters:

Parameter <i>[Required / Optional?]</i>	Data Type	Description
billingMonth <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Month	Service month of requested data.
startDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Starting service date of requested data.
endDate <i>[Optional — Either billingMonth or startDate and endDate must be provided, but not both]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021). The end date must be no earlier than the start date, and no later than 31 days after the start date.

Parameter <i>[Required / Optional?]</i>	Data Type	Description
nyisowide <i>[Optional]</i>	Boolean	If true, response includes NYISO-wide aggregate sums of all Daily Reconciliation values; if false, only includes Daily Reconciliation values for the organization to which the requesting user belongs. Default to false.
version <i>[Required]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
userRequestId <i>[Optional]</i>	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.

## 5.2.2. Request Examples

### 5.2.2.1. Retrieval of Daily Reconciliation MWh Data Request Example #1

- Request all data for one day (December 20, 2021) and version (v2)

```
GET https://apitest.nyiso.com/finance/invoicing/v1/dailyReconciliation/mwh?startDate=2021-12-20&endDate=2021-12-20&version=2
```

#### HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 5.2.2.2. Retrieval of Daily Reconciliation MWh Data Request Example #2

- Request all data for one day (December 20, 2021) and version (v2)
- Optional NYISO-wide parameter provided
- Optional user request ID provided

```
GET https://apitest.nyiso.com/finance/invoicing/v1/dailyReconciliation/mwh?startDate=2021-12-20&
endDate=2021-12-20&version=2&nyisowide=true&userRequestId=MyRequest_ABCD-001
```

HTTP Headers:

```
Accept: application/json
Accept-Encoding: gzip, deflate
Authorization: Basic encodedCredential
Cache-Control: no-cache
```

### 5.2.3. Response

*HTTP Headers:*

```
Content-Type: application/json
```

*Response Body Parameters:*

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters <i>[Always]</i>	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth <i>[If provided in request]</i>	ISO-8601 Month	Service month of requested data.
requestParameters.startDate <i>[If provided in request]</i>	ISO-8601 Date	Starting service date of requested data.

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.endDate <i>[If provided in request]</i>	ISO-8601 Date	Ending service date of requested data, inclusive (e.g., an end time of “2021-12-20” includes everything through the service date December 20 <sup>th</sup> , 2021).
requestParameters.startTime <i>[Always]</i>	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime <i>[Always]</i>	ISO-8601 Date / Time	Ending service date / time of requested data, inclusive (e.g., an end time of “2021-12-20T23:59:59-05:00” includes everything through the end of service date December 20 <sup>th</sup> , 2021).
requestParameters.nyisowide <i>[If provided in request]</i>	Boolean	If true, response includes NYISO-wide aggregate sums of all Daily Reconciliation values; if false, only includes Daily Reconciliation values for the organization to which the requesting user belongs. Default to false.
requestParameters.version <i>[Always]</i>	Number, up to one decimal place $0.0 \leq x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
requestParameters.userName <i>[Always]</i>	String	User name for the user submitting the request

Parameter <i>[Provided...?]</i>	Data Type	Description
requestParameters.userRequestId <i>[If provided in request]</i>	String  Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
nyisoRequestId <i>[Always]</i>	UUID String  36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp <i>[Always]</i>	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
organizationName <i>[Always]</i>	String	If nyisoWide is true, or the requesting user is a NYISO user, "NYISO-Wide"; otherwise, the organization name for the organization to which the user belongs.
mwhDetails <i>[Always]</i>	Array of Daily Reconciliation MWh Detail objects	Revenue-side Daily Reconciliation MWh detail records for generators
mwhDetails.dailyReconciliationCode <i>[For each daily reconciliation MWh detail record]</i>	Integer	Unique NYISO-defined identifier for a given Daily Reconciliation entry

<b>Parameter [Provided...?]</b>	<b>Data Type</b>	<b>Description</b>
mwhDetails.description <i>[For each daily reconciliation MWh detail record]</i>	String	Daily Reconciliation entry description
mwhDetails.itemValueMwh <i>[For each daily reconciliation MWh detail record if the record is for an individual entry]</i>	Number	Daily Reconciliation entry value for an individual entry
mwhDetails.totalValueMwh <i>[For each daily reconciliation MWh detail record if the record is for a total or subtotal entry]</i>	Number	Daily Reconciliation entry value for a total or subtotal entry

## 5.2.4. Response Examples

### 5.2.4.1. Retrieval of Daily Reconciliation Dollar MWh Response Example #1

- One service day requested
- Select data for user's organization (by default)

#### HTTP Status:

200 OK

#### HTTP Headers:

Content-Type: application/json

#### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startDate": "2021-12-14",
    "endDate": "2021-12-14",
    "startTime": "2021-12-14T00:00:00-05:00",
    "endTime": "2021-12-14T23:59:59-05:00"
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "organizationName": "Billing Org O",
  "mwhDetails": [
    {
      "dailyReconciliationCode": 70001,
      "description": "DAM LSE Internal LBMP Energy Sales",
      "itemValueMwh": 1000.0000
    },
    ...
    {
      "dailyReconciliationCode": 34,
      "description": "DAM LBMP Energy Market Mwh",
      "totalValueMwh": 1500.00
    },
    ...
  ]
}
```

```
]
}
```

#### 5.2.4.2. Retrieval of Daily Reconciliation Dollar MWh Response Example #2

- One service day requested
- NYISO-wide data requested

##### HTTP Status:

200 OK

##### HTTP Headers:

Content-Type: application/json

##### Request Body:

```
{
  "requestParameters": {
    "userName": "USER_U",
    "userRequestId": "MyRequest-20211215_123456",
    "startDate": "2021-12-14",
    "endDate": "2021-12-14",
    "startTime": "2021-12-14T00:00:00-05:00",
    "endTime": "2021-12-14T23:59:59-05:00",
    "nyisowide": true
  },
  "nyisoRequestId": "18bf5352-def6-48cd-8b25-dc841b8879b6",
  "requestTimestamp": "2021-12-21T11:30:00-05:00",
  "organizationName": "NYISO-wide",
  "mwhDetails": [
    {
      "dailyReconciliationCode": 70001,
      "description": "DAM LSE Internal LBMP Energy Sales",
      "itemValueMwh": 1000.0000
    },
    ...
    {
      "dailyReconciliationCode": 34,
      "description": "DAM LBMP Energy Market Mwh",
      "totalValueMwh": 1500.00
    },
    ...
  ]
}
```