New York ISO
Independent System Operator

UG #xxxx

Metering API
User Guide

Issued: November 2022

Version: 0.4

Effective Date: <mark>MM/DD/YYYY</mark>

Prepared By: Customer Settlements

New York Independent System Operator
10 Krey Boulevard
Rensselaer, NY 12144
(518) 356-6060
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.

©Copyright 1999-2022 New York Independent System Operator

Table of Contents

Revision History

Version	Date	Revisions
0.1	01/21/2022	Initial Release
0.2	03/21/2022	Section 3.1.1 • In the Request Body Parameters, added clarification to the generators, ties, and subzones arrays with regards to how to represent an absence of data for a given entity type. Section 3.1.2.2 • Added a new example Meter Data Submission request to demonstrate two of the methods for representing an absence of data for a given entity type. Section 3.1.3 • Corrected HTTP Headers, which had accidentally been overwritten. Sections 3.2.2, 3.3.2, 3.4.2 • Corrected formatting of the query parameters in the example request URLs to match the specification, replacing "BILLING_MONTH", "START_DATE", and "END_DATE" with "billingMonth", "startDate", and "endDate", respectively.
0.3	05/27/2022	Section 2.2: • Finalized common URL root. Section 3.1.1 • Finalized common URL root. Section 3.2.1 • Finalized common URL root. Section 3.3.1 • Finalized common URL root. Section 3.4.1 • Finalized common URL root. Updated formatting to match current NYISO standards

Version	Date	Revisions
0.4	10/28/2022	Section 3.1.1 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit to reduce confusion, with corresponding updates to the description to flip the logic. Section 3.1.2.1 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit in example. Section 3.1.2.2 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit in example. Section 3.1.3 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit to reduce confusion, with corresponding updates to the description to flip the logic. Section 3.1.4.1 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit in example. Section 3.1.4.2 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit in example. Section 3.1.4.3 • Changed submissionParameters.doNotCommit to submissionParameters.doCommit in example.

1. Introduction

This Guide describes the Metering Application Programming Interface (API) and how to interact with the service, including:

- 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.

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 to 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.

2. Using the Metering API

2.1. Network Protocols

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

2.2. Location of Metering API Service Endpoints

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

```
https://api.nyiso.com/metering/
```

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

2.3. Authentication

Two-factor authentication is required to access the Metering API.

- 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 Metering API endpoints are formatted as JSON.

2.4.1. Formatting Numbers

The Metering API allows for up to four decimals of precision for all MWh values.

2.4.2. Formatting Dates and Times

The Metering API 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

  o "2021-12"

- ISO-8601 Date: yyyy-MM-dd

  o "2021-12-14"

- ISO-8601 Date / Time: yyyy-MM-ddThh:mm:ssX

  o "2021-12-14T07:00:00Z" in GMT, or "2021-12-14T02:00:00-05:00" in EST

  o "2021-07-14T06:00:00Z" in GMT, or "2021-07-14T02:00:00-04:00" in EDT

  o Note: While the Metering API 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.

2.5. Metering Data Availability

Users will be able to download metering data from the Metering API for a three year and ten month period ending with the current month.

2.6. HTTP Status Codes for Responses

The Metering API responses utilize the standard HTTP status codes to communicate either the success or kind 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, as well as validation failures of input data.

- 401 Unauthorized: The provided user credentials failed authentication.

- 403 Forbidden: The user was authenticated but is not authorized for the service requested.

- 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.

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

Parameter *[Required / Optional?]*	Data Type	Description
`submissionParameters.userRequestId` *[Optional]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
`submissionParameters.includeAcceptedDataInResponse` *[Optional]*	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` *[Optional]*	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.
`generators` *[Optional]*	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.
`generators.genPtid` *[Required for each generator meter data record]*	Integer	Unique NYISO-defined point identifier for a given generator
`generators.dateHour` *[Required for each generator meter data record]*	ISO-8601 Date / Time	Service hour for meter data

Parameter *[Required / Optional?]*	Data Type	Description
`generators.meterInjectionEnergyMwh` *[Required for injection-capable generators; prohibited otherwise]*	Number, up to four decimal places 0.0000 <= *x* < 10,000.0000	Metered injections for the given generator and service hour
`generators.meterWithdrawalEnergyMwh` *[Required for withdrawal-capable generators; prohibited otherwise]*	Number, up to four decimal places -10,000.0000 < *x* <= 0.0000	Metered withdrawals for the given generator and service hour
`generators.meterDemandReductionMwh` *[Required for demand reduction-capable generators; prohibited otherwise]*	Number, up to four decimal places 0.0000 <= *x* < 10,000.0000	Metered demand reduction for the given generator and service hour
`ties` *[Optional]*	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` *[Required for each tie meter data record]*	Integer	Unique NYISO-defined point identifier for a given tie
`ties.dateHour` *[Required for each tie meter data record]*	ISO-8601 Date / Time	Service hour for meter data

Parameter *[Required / Optional?]*	Data Type	Description
`ties.meterTieFlowMwh` *[Required for each tie meter data record]*	Number, up to four decimal places -10,000.0000 < *x* < 10,000.0000	Metered tie flow for the given tie
`subzones` *[Optional]*	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` *[Required for each subzone meter data record]*	Integer	Unique NYISO-defined point identifier for a given subzone
`subzones.dateHour` *[Required for each subzone meter data record]*	ISO-8601 Date / Time	Service hour for meter data
`subzones.meterSubzoneLoadMwh` *[Required for each subzone meter data record]*	Number, up to four decimal places 0.0000 <= *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://api.nyiso.com/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

    o   Ties parameter included as empty array

    o   Subzones parameter omitted

    o   Either approach is acceptable, as well as setting the parameter to null, instead of an empty array ([])

- Optional submission parameters included

```
POST https://api.nyiso.com/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 *[Provided...?]*	Data Type	Description
submissionParameters *[Always]*	Submission Parameters object	User-provided parameters for the submission
submissionParameters.userRequestId *[If provided in request]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
submissionParameters.includeAcceptedDataInResponse *[Always]*	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 *[Always]*	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 [Provided...?]	Data Type	Description
nyisoRequestId [Always]	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp [Always]	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
requestSummary [Always]	Summary Object	Processing summary for the request
requestSummary.generators [Always]	Generator Summary object	Processing summary for generators submitted in the request
requestSummary.generators.submitted [Always]	Integer	Number of meter data records submitted for generators in the request
requestSummary.generators.passedValidation [Always]	Integer	Number of meter data records that passed validation for generators in the request
requestSummary.generators.failedValidation [Always]	Integer	Number of meter data records that failed validation for generators in the request
requestSummary.generators.accepted [Always]	Integer	Number of meter data records accepted for generators in the request

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

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

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

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

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

Parameter *[Provided...?]*	Data Type	Description
`accepted.subzones.subzonePtid` *[For each accepted subzone meter data record]*	Integer	Accepted unique NYISO-defined point identifier for a given subzone
`accepted.subzones.dateHour` *[For each accepted subzone meter data record]*	ISO-8601 Date / Time	Accepted service hour for meter data
`accepted.subzones.meterSubzoneLoadMwh` *[For each accepted subzone meter data record]*	Number, up to four decimal places 0.0000 <= *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": {
    "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": {
    "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": [
            "Metering-00001: Tie PTID does not exist: 222222"
          ]
        }
      ]
    }
}
```

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": {
-     "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/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 *[Required / Optional?]*	Data Type	Description
`billingMonth` *[Optional — Either `billingMonth` or `startDate` and `endDate` must be provided, but not both]*	ISO-8601 Month	Service month of requested data.
`startTime` *[Optional — Either `billingMonth` or `startTime` and `endTime` must be provided, but not both]*	ISO-8601 Date / Time	Starting service date / time of requested data.
`endTime` *[Optional — Either `billingMonth` or `startTime` and `endTime` must be provided, but not both]*	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 20th, 2021.

Parameter *[Required / Optional?]*	Data Type	Description
`userRequestId` *[Optional]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
`maUpdateStartTime` *[Optional]*	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.
`maUpdateEndTime` *[Optional — If provided, `maUpdateStartTime` must be provided]*	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 20th, 2021.
`version` *[Optional]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
`genPtid` *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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 *[Required / Optional?]*	Data Type	Description
`tiePtid` *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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` *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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` *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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://api.nyiso.com/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://api.nyiso.com/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 *[Provided...?]*	Data Type	Description
requestParameters *[Always]*	Request Parameters object	User-provided parameters for the request

Parameter *[Provided…?]*	Data Type	Description
`requestParameters.billingMonth` *[If provided in request]*	ISO-8601 Month	Service month of requested data.
`requestParameters.startTime` *[Always]*	ISO-8601 Date / Time	Starting service date / time of requested data.
`requestParameters.endTime` *[Always]*	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 20th, 2021.
`requestParameters.userRequestId` *[If provided in request]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
`requestParameters.maUpdateStartTime` *[If provided in request]*	ISO-8601 Date / Time	Starting meter authority update date / time of requested data.
`requestParameters.maUpdateEndTime` *[If provided in request]*	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 20th, 2021.

Parameter *[Provided...?]*	Data Type	Description
`requestParameters.version` *[If provided in request]*	Number, up to one decimal place 0.0 <= *x* < 10	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
`requestParameters.genPtid` *[If provided in request]*	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` *[If provided in request]*	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.
`requestParameters.subzonePtid` *[If provided in request]*	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.

Parameter *[Provided…?]*	Data Type	Description
`requestParameters.entityType` *[If provided in request]*	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` *[Always]*	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
`requestTimestamp` *[Always]*	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
`generators` *[If any generator metering detail records are returned]*	Array of Generator Metering Detail objects	Metering detail records for generators
`generators.genPtid` *[For each generator metering detail record]*	Integer	Unique NYISO-defined point identifier for a given generator
`generators.generatorName` *[For each generator metering detail record]*	String	Generator name
`generators.dateHour` *[For each generator metering detail record]*	ISO-8601 Date / Time	Service hour for metering detail
`generators.billingDate` *[For each generator metering detail record]*	ISO-8601 Date	Billing / service day for metering detail

Parameter *[Provided...?]*	Data Type	Description
generators.version *[For each generator metering detail record]*	Number, up to one decimal place 0.0 <= *x* < 10	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
generators.billedFlag *[For each generator metering detail record]*	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 *[For each generator metering detail record if generator is authorized for injection revenue-grade metering]*	Number	MA-submitted metered injections for the given generator and service hour
generators.telemetryInjectionEnergyMwh *[For each generator metering detail record if generator is authorized for injection telemetry]*	Number	Hourly integrated injections calculated from telemetry data for the given generator and service hour
generators.meterWithdrawalEnergyMwh *[For each generator metering detail record if generator is authorized for withdrawal revenue-grade metering]*	Number	MA-submitted metered withdrawals for the given generator and service hour
generators.telemetryWithdrawalEnergyMwh *[For each generator metering detail record if generator is authorized for withdrawal telemetry]*	Number	Hourly integrated withdrawals calculated from telemetry data for the given generator and service hour
generators.meterNetEnergyMwh *[For each generator metering detail record if generator is authorized for injection and/or withdrawal revenue-grade metering]*	Number	Calculated MA-submitted metered net energy, combining MA-submitted metered injections and withdrawals for the given generator and service hour

Parameter *[Provided…?]*	Data Type	Description
generators.telemetryNetEnergyMwh *[For each generator metering detail record if generator is authorized for injection and/or withdrawal telemetry]*	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 *[For each generator metering detail record if generator is authorized for demand reduction revenue-grade metering]*	Number	Submitted metered demand reduction for the given generator and service hour
generators.telemetryDemandReductionMwh *[For each generator metering detail record if generator is authorized for demand reduction telemetry]*	Number	Hourly integrated demand reductions calculated from telemetry data for the given generator and service hour
generators.meterAuthority *[For each generator metering detail record]*	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given generator and service hour; null otherwise.
generators.meterAuthorityUpdateTime *[For each generator metering detail record]*	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.

Parameter *[Provided...?]*	Data Type	Description
`generators.meterAuthorityUpdateUser` *[For each generator metering detail record]*	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` *[For each generator metering detail record]*	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Generator Metering Detail for the given generator and service hour.
`ties` *[If any tie metering detail records are returned]*	Array of Tie Metering Detail objects	Metering detail records for ties
`ties.tiePtid` *[For each tie metering detail record]*	Integer	Unique NYISO-defined point identifier for a given tie
`ties.tieName` *[For each generator metering detail record]*	String	Tie name
`ties.dateHour` *[For each generator metering detail record]*	ISO-8601 Date / Time	Service hour for metering detail
`ties.billingDate` *[For each generator metering detail record]*	ISO-8601 Date	Billing / service day for metering detail
`ties.version` *[For each tiemetering detail record]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

Parameter *[Provided...?]*	Data Type	Description
`ties.billedFlag` *[For each tie metering detail record]*	String "Y" or "N"	If "Y", data in metering detail has been used to calculate settlements; if "N", data has not yet been used.
`ties.meterTieFlowMwh` *[For each tie metering detail record]*	Number	Submitted metered tie flow for the given tie
`ties.telemetryTieFlowMwh` *[For each tie metering detail record]*	Number	Hourly integrated tie flow calculated from telemetry data for the given tie and service hour
`ties.meterAuthority` *[For each tie metering detail record]*	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given tie and service hour; null otherwise.
`ties.meterAuthorityUpdateTime` *[For each tie metering detail record]*	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.
`ties.meterAuthorityUpdateUser` *[For each tie metering detail record]*	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.

Parameter *[Provided...?]*	Data Type	Description
`ties.updateTime` *[For each tie metering detail record]*	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Tie Metering Detail for the given tie and service hour.
`subzones` *[If any subzone metering detail records are returned]*	Array of Subzone Metering Detail objects	Metering detail records for subzones
`subzones.subzonePtid` *[For each subzone metering detail record]*	Integer	Unique NYISO-defined point identifier for a given subzone
`subzones.subzoneName` *[For each subzone metering detail record]*	String	Subzone name
`subzones.dateHour` *[For each subzone metering detail record]*	ISO-8601 Date / Time	Submitted service hour for meter data
`subzones.billingDate` *[For each subzone metering detail record]*	ISO-8601 Date	Billing / service day for metering detail
`subzones.version` *[For each subzone metering detail record]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
`subzones.billedFlag` *[For each subzone metering detail record]*	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` *[For each subzone metering detail record]*	Number	Submitted metered subzone load for the given subzone and service hour

Parameter *[Provided...?]*	Data Type	Description
`subzones.meterAuthority` *[For each subzone metering detail record]*	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` *[For each subzone metering detail record]*	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` *[For each subzone metering detail record]*	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` *[For each subzone metering detail record]*	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

- All records passed validation and therefore accepted

- `includeAcceptedDataInResponse` = true

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",
  "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": 345678,
      "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/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 *[Required / Optional?]*	Data Type	Description
`billingMonth` *[Optional — Either* `billingMonth` *or* `startDate` *and* `endDate` *must be provided, but not both]*	ISO-8601 Month	Service month of requested data.
`startTime` *[Optional — Either* `billingMonth` *or* `startTime` *and* `endTime` *must be provided, but not both]*	ISO-8601 Date / Time	Starting service date / time of requested data.
`endTime` *[Optional — Either* `billingMonth` *or* `startTime` *and* `endTime` *must be provided, but not both]*	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 20th, 2021.

Parameter *[Required / Optional?]*	Data Type	Description
userRequestId *[Optional]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
version *[Optional]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtid *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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://api.nyiso.com/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://api.nyiso.com/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 *[Provided...?]*	Data Type	Description
requestParameters *[Always]*	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth *[If provided in request]*	ISO-8601 Month	Service month of requested data.
requestParameters.startTime *[Always]*	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime *[Always]*	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 20th, 2021.
requestParameters.userRequestId *[If provided in request]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.version *[If provided in request]*	Number, up to one decimal place 0.0 <= x < 10	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

Parameter *[Provided...?]*	Data Type	Description
requestParameters.subzonePtid *[If provided in request]*	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 *[Always]*	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp *[Always]*	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
calculatedSubzoneLoads *[If any calculated subzone load summary records are returned]*	Array of Calculated Subzone Load Summary objects	Calculated Subzone Load summary records
calculatedSubzoneLoads.subzonePtid *[For each calculated subzone load summary record]*	Integer	Unique NYISO-defined point identifier for a given subzone
calculatedSubzoneLoads.subzoneName *[For each calculated subzone load summary record]*	String	Subzone name
calculatedSubzoneLoads.dateHour *[For each calculated subzone load summary record]*	ISO-8601 Date / Time	Service hour for calculated subzone load data

Parameter *[Provided…?]*	Data Type	Description
`calculatedSubzoneLoads.billingDate` *[For each calculated subzone load summary record]*	ISO-8601 Date	Billing / service day for calculated subzone load data
`calculatedSubzoneLoads.version` *[For each calculated subzone load summary record]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
`calculatedSubzoneLoads.nyisoCalculatedSubzoneLoadMwh` *[For each calculated subzone load summary record]*	Number	NYISO-calculated subzone load for the given subzone and service hour
`calculatedSubzoneLoads.subzoneLossesMwh` *[For each calculated subzone load summary record]*	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": {
    "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/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 *[Required / Optional?]*	Data Type	Description
`billingMonth` *[Optional — Either* `billingMonth` *or* `startDate` *and* `endDate` *must be provided, but not both]*	ISO-8601 Month	Service month of requested data.
`startTime` *[Optional — Either* `billingMonth` *or* `startTime` *and* `endTime` *must be provided, but not both]*	ISO-8601 Date / Time	Starting service date / time of requested data.
`endTime` *[Optional — Either* `billingMonth` *or* `startTime` *and* `endTime` *must be provided, but not both]*	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 20th, 2021.

Parameter *[Required / Optional?]*	Data Type	Description
userRequestId *[Optional]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
version *[Optional]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
subzonePtid *[Optional — Allows multiple values, either comma-separated or as repeated URL parameters]*	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://api.nyiso.com/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://api.nyiso.com/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:

Parameter [Provided...?]	Data Type	Description
requestParameters *[Always]*	Request Parameters object	User-provided parameters for the request
requestParameters.billingMonth *[If provided in request]*	ISO-8601 Month	Service month of requested data.
requestParameters.startTime *[Always]*	ISO-8601 Date / Time	Starting service date / time of requested data.
requestParameters.endTime *[Always]*	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 20th, 2021.
requestParameters.userRequestId *[If provided in request]*	String Max. 30 characters; allows letters, numbers, hyphens, and underscores	User-provided request ID for tracking purposes.
requestParameters.version *[If provided in request]*	Number, up to one decimal place $0.0 <= x < 10$	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.

Parameter *[Provided…?]*	Data Type	Description
requestParameters.subzonePtid *[If provided in request]*	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 *[Always]*	UUID String 36 characters; includes letters, numbers, and hyphens	NYISO-created UUID to identify request for support purposes.
requestTimestamp *[Always]*	ISO-8601 Date / Time	Timestamp identifying when NYISO received the request.
calculatedSubzoneLoadDetails *[If any calculated subzone load details are returned]*	Array of Calculated Subzone Load Detail objects	Calculated Subzone Load details for each returned subzone
calculatedSubzoneLoadDetails.subzonePtid *[For each calculated subzone load detail record]*	Integer	Unique NYISO-defined point identifier for a given subzone
calculatedSubzoneLoadDetails.subzoneName *[For each calculated subzone load detail record]*	String	Subzone name
calculatedSubzoneLoadDetails.dateHour *[For each calculated subzone load detail record]*	ISO-8601 Date / Time	Service hour for calculated subzone load data

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.billingDate` *[For each calculated subzone load detail record]*	ISO-8601 Date	Billing / service day for calculated subzone load data
`calculatedSubzoneLoadDetails.version` *[For each calculated subzone load detail record]*	Number, up to one decimal place 0.0 <= *x* < 10	Invoice version number of the available data being retrieved. Default to 0, which returns the latest data.
`calculatedSubzoneLoadDetails.totalSubzoneLoadContributionMwh` *[For each calculated subzone load detail record]*	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
`calculatedSubzoneLoadDetails.totalGeneratorSubzoneLoadContributionMwh` *[For each calculated subzone load detail record]*	Number	Total contributions to NYISO-calculated subzone load from generators for the given subzone and service hour
`calculatedSubzoneLoadDetails.totalTieSubzoneLoadContributionMwh` *[For each calculated subzone load detail record]*	Number	Total contributions to NYISO-calculated subzone load from ties for the given subzone and service hour
`calculatedSubzoneLoadDetails.generators` *[If any generator metering detail records are returned]*	Array of Generator Metering Detail objects	Metering detail records for generators
`calculatedSubzoneLoadDetails.generators.genPtid` *[For each generator metering detail record]*	Integer	Unique NYISO-defined point identifier for a given generator

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.generators.generatorName` *[For each generator metering detail record]*	String	Generator name
`calculatedSubzoneLoadDetails.generators.billedFlag` *[For each generator metering detail record]*	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` *[For each generator metering detail record if generator is authorized for injection revenue-grade metering]*	Number	MA-submitted metered injections for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.telemetryInjectionEnergyMwh` *[For each generator metering detail record if generator is authorized for injection telemetry]*	Number	Hourly integrated injections calculated from telemetry data for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.meterWithdrawalEnergyMwh` *[For each generator metering detail record if generator is authorized for withdrawal revenue-grade metering]*	Number	MA-submitted metered withdrawals for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.telemetryWithdrawalEnergyMwh` *[For each generator metering detail record if generator is authorized for withdrawal telemetry]*	Number	Hourly integrated withdrawals calculated from telemetry data for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.meterNetEnergyMwh` *[For each generator metering detail record if generator is authorized for injection and/or withdrawal revenue-grade metering]*	Number	Calculated MA-submitted metered net energy, combining MA-submitted metered injections and withdrawals for the given generator and service hour

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.generators.telemetryNetEnergyMwh` *[For each generator metering detail record if generator is authorized for injection and/or withdrawal telemetry]*	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` *[For each generator metering detail record if generator is authorized for demand reduction revenue-grade metering]*	Number	Submitted metered demand reduction for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.telemetryDemandReductionMwh` *[For each generator metering detail record if generator is authorized for demand reduction telemetry]*	Number	Hourly integrated demand reductions calculated from telemetry data for the given generator and service hour
`calculatedSubzoneLoadDetails.generators.meterAuthority` *[For each generator metering detail record]*	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given generator and service hour; null otherwise.
`calculatedSubzoneLoadDetails.generators.meterAuthorityUpdateTime` *[For each generator metering detail record]*	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.

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.generators.meterAuthorityUp dateUser` *[For each generator metering detail record]*	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.
`calculatedSubzoneLoadDetails.generators.updateTime` *[For each generator metering detail record]*	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Generator Metering Detail for the given generator and service hour.
`calculatedSubzoneLoadDetails.ties` *[If any tie metering detail records are returned]*	Array of Tie Metering Detail objects	Metering detail records for ties
`calculatedSubzoneLoadDetails.ties.tiePtid` *[For each tie metering detail record]*	Integer	Unique NYISO-defined point identifier for a given tie
`calculatedSubzoneLoadDetails.ties.tieName` *[For each generator metering detail record]*	String	Tie name
`calculatedSubzoneLoadDetails.ties.billedFlag` *[For each tie metering detail record]*	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.ties.meterTieFlowMwh` *[For each tie metering detail record]*	Number	Submitted metered tie flow for the given tie
`calculatedSubzoneLoadDetails.ties.telemetryTieFlowMwh` *[For each tie metering detail record]*	Number	Hourly integrated tie flow calculated from telemetry data for the given tie and service hour

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.ties.meterAuthority` *[For each tie metering detail record]*	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given tie and service hour; null otherwise.
`calculatedSubzoneLoadDetails.ties.meterAuthorityUpdateTime` *[For each tie metering detail record]*	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.
`calculatedSubzoneLoadDetails.ties.meterAuthorityUpdateUser` *[For each tie metering detail record]*	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.
`calculatedSubzoneLoadDetails.ties.updateTime` *[For each tie metering detail record]*	ISO-8601 Date / Time	Timestamp of most recent update of any kind to Tie Metering Detail for the given tie and service hour.
`calculatedSubzoneLoadDetails.subzones` *[If any subzone metering detail records are returned]*	Array of Subzone Metering Detail objects	Metering detail records for subzones
`calculatedSubzoneLoadDetails.subzones.subzonePtid` *[For each subzone metering detail record]*	Integer	Unique NYISO-defined point identifier for a given subzone

Parameter *[Provided...?]*	Data Type	Description
`calculatedSubzoneLoadDetails.subzones.subzoneName` *[For each subzone metering detail record]*	String	Subzone name
`calculatedSubzoneLoadDetails.subzones.billedFlag` *[For each subzone metering detail record]*	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.subzones.meterSubzoneLoadMwh` *[For each subzone metering detail record]*	Number	Submitted metered subzone load for the given subzone and service hour
`calculatedSubzoneLoadDetails.subzones.meterAuthority` *[For each subzone metering detail record]*	String	Meter Authority that submitted revenue-grade meter data if any has been submitted for the given subzone and service hour; null otherwise.
`calculatedSubzoneLoadDetails.subzones.meterAuthorityUpdateTime` *[For each subzone metering detail record]*	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.
`calculatedSubzoneLoadDetails.subzones.meterAuthorityUpdateUser` *[For each subzone metering detail record]*	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.

Parameter *[Provided…?]*	Data Type	Description
`calculatedSubzoneLoadDetails.subzones.updateTime` *[For each subzone metering detail record]*	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": {
    "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": 345678,
          "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"
      }
    ]
  }
 ]
}
```