

## **Production Tracking System (PTS) Monthly Automated Reporting (AR)**

The PTS Monthly Automated Reporting (AR) API is designed to facilitate the automatic monthly loading of energy production data. The AR API accepts data in JSON format only. In addition to the well-formed PTS AR data, header information must be added for user authentication. The PTS AR API provides a method to easily submit current meter readings and obtain information of the Organizations current systems.

There is a Reporting Period window that opens each month. All meter readings will need to be uploaded during this window. Additionally, only readings with a valid date within the window will be accepted. A meter reading cannot be uploaded unless the reading date is within the current reporting period. If a reading contains multiple months of data, the data will be extrapolated for prior months.

On the first upload of the Reporting Period the submission will be saved to an interim table to assist in the tracking of submissions and each system that was submitted.

There will be a maximum limit of 2 valid submissions per system per Reporting Window. There will also be a maximum limit of 3 failures per system. These can be overridden by MassCEC Help Desk on special requests.

Validation will occur on the system's meter reading to ensure the meter is producing energy that is consistent with its listed estimated annual production and other factors that could affect the performance. As an example, there is less sunlight during the winter months, therefore the generated energy from the system should be less in those months than in summer months.

## **Developer Platform**

Developers need to sign-up for an account to obtain an API Key, which provides access to a specific set of API calls. MassCEC Helpdesk will then email the developer, notifying that the API Key is linked to the corresponding Organization in the PTS database.

- URL: <https://api.masscec-pts.com>
- Sign up for an account, One developer account per organization.
- Email [pts@masscec.com](mailto:pts@masscec.com), with your developer account email.
- Helpdesk will setup the appropriate permissions on the Staging API.
- Demonstrate upload of data into Staging API.
- Helpdesk will then verify data in Staging is acceptable.
- If acceptable, access to production APIs will be given.
- Staging APIs will stay available for testing.

The API has been developed using Restful API patterns, using GET and POST Https requests. Some notable standards/rules are listed below:

- JSON objects
- All API calls will need two Header key/value pairs. Substitute with your API Key
  - Ocp-Apim-Subscription-Key:yourAPIKey
  - Ocp-Apim-Trace:true

- fk\_System is the Key identifier for all systems moving forward.
- All Date/Times are in Eastern Daylight Time.
- Will accommodate batching of systems or all systems at one time, depending on the size of the JSON object.
- Submitting a System for validation more than once will invalidate the previous submittal. Last one in wins.
- Maximum limit of 2 valid submissions per system will be enforced
- Maximum limit of 3 failed validation submissions per system will be enforced
- API Key may be invalidated for abuse of the system

## API Endpoints

### 1. GetSystems

#### a. GET request

- i. Accepts a query parameter

```
?fk_system=1000,2000,3000
```

#### b. POST request

- i. Accepts a Body of systems object to filter by fk\_system

```
{
  "systems": [
    {
      "fk_system": 1000
    }, {
      "fk_system": 2000
    }
  ]
}
```

- c. Produces a JSON object array containing a list of Active Systems.

```
[{
  "fk_system": 1000,
  "SystemID": "NFG-NN100-01000",
  "SystemName": "Test",
  "capacity": 10.586,
  "Error": null
},...]
```

- d. Error column is reserved for any errors that are trapped in the API Call. Trap for anything not null.

- e. Produces the following Http Results:

- i. 200 : Object is Returned
- ii. 400 : Bad API Key
- iii. 500 : Service Unavailable

## 2. GetLastReading

- a. GET request
  - i. Accepts a query parameter
  - ii. `?fk_system=1000,2000,3000`
- b. Post request
  - i. Accepts a Body of systems object to filter by fk\_system

```
{
  "systems": [
    {
      "fk_system": 1000
    }, {
      "fk_system": 2000
    }
  ]
}
```

- c. Produces a JSON object array containing the last known Meter Reading for each Active System

```
[{
  "fk_system": 1000,
  "SystemID": " NFG-NN100-01000",
  "SystemName": "Test",
  "MeterValue": 5448.0,
  "MeterDate": "2019-12-01T00:00:00",
  "DateTimeReported": "2020-01-01T00:00:00",
  "PeriodEndDate": "2019-12-31T00:00:00",
  "EnergyProduced": 20,
  "IsEstimated": false,
  "Notes": "",
  "Error": null
},...]
```

- d. Error column is reserved for any errors that are trapped in the API Call. Trap for anything not null.
- e. Produces the following Http Results:
  - i. 200 : Object is Returned
  - ii. 400 : Bad API Key
  - iii. 500 : Service Unavailable

## 3. ARPostTest

- a. POST request
- b. Data will be run through Validation, just not saved into Production
- c. Accepts a JSON object array containing the current Meter Reading for each Active System

```
{
```

```
"APIKey": "your api key",
"Notes": "test1",
"systems": [
  {
    "fk_system": 11,
    "MeterValue": 2835.3,
    "MeterDate": "2019-11-30T23:59:42",
    "Notes": null
  },
  {
    "fk_system": 298,
    "MeterValue": 10453.9,
    "MeterDate": "2019-11-30T23:59:54",
    "Notes": null
  },...
]
```

- d. Produces the following Http Results:
- 200 : Received Data
  - 400 : Bad Request
  - 500 : Service Unavailable
4. ARPost
- POST request
  - Data will be run through Validation, at the end of the Reporting Window will be added into Production
  - Accepts a JSON object array containing the current Meter Reading for each Active System

```
{
  "APIKey": "your api key",
  "Notes": "test1",
  "systems": [
    {
      "fk_system": 11,
      "MeterValue": 2835.3,
      "MeterDate": "2019-11-30T23:59:42",
      "Notes": null
    },
    {
      "fk_system": 298,
      "MeterValue": 10453.9,
      "MeterDate": "2019-11-30T23:59:54",
      "Notes": null
    },...
  ]
}
```

- d. Produces the following Http Results:
  - i. 200 : Received Data
  - ii. 400 : Bad API Key
  - iii. 500 : Service Unavailable
- 5. GetSubmissionStatus
  - a. GET request
    - i. Accepts a Body of systems object to filter by fk\_system

```
{
  "systems": [
    {
      "fk_system": 1000
    }, {
      "fk_system": 2000
    }
  ]
}
```

- b. Produces a JSON object array containing the status of each system that was submitted during the current Reporting window.

```
[{
  "fk_system": 1000,
  "SystemID": " NFG-NN100-01000",
  "SystemName": "Test",
  "MeterValue": 5448.0,
  "MeterDate": "2019-12-01T00:00:00",
  "DateTimeReported": "2020-01-01T00:00:00",
  "PeriodEndDate": "2019-12-31T00:00:00",
  "EnergyProduced": 20,
  "IsEstimated": false,
  "Notes": "",
  "ValidationNotes": "",
  "ValidationStatus": 0
},...]
```

- c. Validation Status
  - i. 0 : Validating
  - ii. -1 : Failed
  - iii. 1 : Passed
- d. Produces the following Http Results:
  - i. 200 : Object is Returned
  - ii. 400 : Bad API Key
  - iii. 500 : Service Unavailable