

FACTECH AUTOMATION SOLUTIONS PVT. LTD

Third Party Meter Integration

API

API details for integrating 3rd Meter Services with iSocietyManager

Contents

1. Latest Readings / Balances for Site (All Meters).....	2
2. Latest Readings / Balances for Single Meter.....	3
3. Recharge Meter.....	3
4. Recharge Meter Status.....	4
5. Meter Changed Between Dates.....	5
6. Start Meter.....	6
7. Stop Meter.....	6
8. Check Meter Status.....	7
9. Update Meter Load.....	8
For DG Related Integration.....	8
10. Latest Readings for DG.....	8
11. DG Sensing.....	9
12. DG Status.....	10
13. DG Fuel Status.....	10

Third Party Meter Integration

iSocietyManager System interface with Meter Servers to automate the Meter Reading, Recharge, Cut, Restore Meter and fulfill the customer requirement of unified solution of end to end automation.

Below is the list of APIs that are provided by various Meter Companies.

1. Latest Readings / Balances for Site (All Meters)

Method: GET

Input:

- site_id (String)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter Readings and Balances",
  "count" : "n",
  "data" : [
    {
      "unit_no" : "",
      "meter_no" : "",
      "dg" : "",
      "grid" : "",
      "date_time" : ""
      "balance" ,
      "status": ""
    },...
  ]
}
```

//number of records in data
//array of reading objects
//flat no, C A No
//reading captured date and time in format "YYYY- MM-DD HH:mm:ss"
// Balance in case of Prepaid Meter
// On/ Off

- **Error case**

```
{
  "status" : "error",
  "message" : "Not Found",
  "count" : "0",
  "data" : null
}
```

Remarks:

Fetch latest reading of all the Meters.

<https://isocietymanager.com>

2. Latest Readings / Balances for Single Meter

Method: GET

Input:

- Meter Number (String)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter Readings and Balance",
  "data" : {
    "unit_no" : "", //flat no, C A No
    "meter_no" : "",
    "dg" : "",
    "grid" : "",
    "date_time" : "" //reading captured date and time in format "YYYY- MM-DD HH:mm:ss"
    "balance" : "" // Balance in case of Prepaid Meter
    "status" : "" // On/ Off
  }
}
```

- **Error case**

```
{
  "status" : "Error",
  "message" : "Not Found",
  "count" : "0",
  "data" : null
}
```

Remarks:

Fetch latest reading of the supplied Meter.

3. Recharge Meter

Method: POST

Input:

- site_id (String)
- meter_no (String)
- amount (String)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter Recharged",
  "data" : {
    "unit_no" : "", //flat no, C A No
    "meter_no" : "",
    "amount" : "" // Recharged Amount
  }
}
```

<https://isocietymanager.com>

```

"transaction_id" : "",
"token" : ""
"date_time" : "" //reading captured date and time in format "YYYY- MM-DD HH:mm:ss"
"balance" : "" // Latest Balance in case of Prepaid Meter
"remarks" : "" // Remarks, if any

```

- **Error case**

```

{
  "status" : "Error",
  "message" : "Not Recharged",
  "count" : "0",
  "data" : null
}

```

Remarks:

Recharge the amount in the prepaid Meter.

4. Recharge Meter Status

Method: GET

Input:

- site_id (String)
- transaction_id (String)

Response:

- **Success case**

```

{
  "status" : "success",
  "message" : "Recharge success", //number of records in data
  "data" : {
    "unit_no" : "",
    "meter_no" : "",
    "transaction_id" : "", //unique transaction id or vouture no
    "remarks" : ""
  }
}

```

- **Waiting case //optional case**

```

{
  "status" : "waiting",
  "message" : "Recharge request is pending",
  "data" : {
    "unit_no" : "",
    "meter_no" : "",
    "transaction_id" : "", //unique transaction id or vouture no
    "remarks" : ""
  }
}

```

<https://isocietymanager.com>

- **Error case**

```
{
  "status" : "error",
  "message" : "Recharge was not successful due to ...", //OR no such record found
  "data" : null
}
```

Remarks:

Check the recharge status in case Meter Server maintain queue.

5. Meter Changed Between Dates

Method: GET

Input:

- site_id (String)
- from_date (String format : "YYYY-MM-DD" i.e. "2018-10-10")
- to_date (String format : "YYYY-MM-DD" i.e. "2018-10-10")

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter Changed Details",
  "count" : "n", //number of records in data
  "data" : [ //array of changed meter data
    {
      "unit_no" : "", //unique flat no
      "old_meter_no" : "",
      "old_start_dg" : "",
      "old_end_dg" : "",
      "old_start_grid" : "",
      "old_end_grid" : "",
      "new_meter_no" : "",
      "new_start_dg" : "",
      "new_start_grid" : "",
      "meter_change_date" : "",
      "reading_start_date" : "", //new meter reading start date
    },...
  ]
}
```

- **Error case**

```
{
  "status" : "Error",
  "message" : "Not Details",
  "count" : "0",
  "data" : null
}
```

<https://isocietymanager.com>

Remarks:

Get details of Meter Changed between the periods.

6. Start Meter

Method: POST

Input:

- site_id (String)
- meter_no (Meter to be started)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter Started",
  "data" : {
    "unit_no" : "",
    "meter_no": "",
    "status": "STARTED"
  }
}
```

//unique flat no

- **Error case**

```
{
  "status" : "Error",
  "message" : "Operation Not Successful",
  "Remarks" : " Meter was already STARTED"
}
```

Remarks:

Start the Meter

7. Stop Meter

Method: POST

Input:

- site_id (String)
- meter_no (Meter to be stopped)

Response:

- **Success case**

```
{
  "status" : "success",
```

<https://isocietymanager.com>

```
"message": "Meter Stopped",
"data": {
    "unit_no": "", //unique flat no
    "meter_no": "",
    "status": "STOPPED"
}
```

- **Error case**

```
{
"status": "Error",
"message": "Operation Not Successful",
"Remarks": " Meter not Accessible"
}
```

Remarks:

Stop the Meter

8. Check Meter Status

Method: GET

Input:

- site_id (String)
- meter_no (String)

Response:

- **Success case**

```
{
"status": "success",
"message": "Meter Status success", //number of records in data
"data": {
    "unit_no": "",
    "meter_no": "",
    "status": "", //ON/ Off
    "remarks": ""
}
}
```

- **Error case**

```
{
"status": "error",
"message": "Not able to connect to Meter ...", //OR no such record found
"data": null
}
```


<https://isocietymanager.com>

9. Update Meter Load

Method: POST

Input:

- site_id (String)
- meter_no (Meter for which load to be updated)
- load_dg (New load of DG)
- load_grid (New load of Grid)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "Meter load updated",
  "data" : {
    "unit_no" : "", //unique flat no
    "meter_no": "",
    "status": "Load Updated"
  }
}
```

- **Error case**

```
{
  "status" : "Error",
  "message" : "Operation Not Successful",
  "Remarks" : " Meter Load was not updated"
}
```

Remarks:

Update the Meter load

For DG Related Integration

10. Latest Readings for DG

Method: GET

Input:

- site_id (String)
- dg_number (String)

<https://isocietymanager.com>

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "DG Readings",
  "data" : {
    "dg_no" : "", //dg no, dg id
    "meter_no" : "",
    "reading" : "", //reading captured date and time in format "YYYY- MM-DD HH:mm:ss"
    "date_time" : ""
  }
}
```

- **Error case**

```
{
  "status" : "Error",
  "message" : "Not Found",
  "data" : null
}
```

Remarks:

Fetch latest reading of the specific DG

11. DG Sensing

Method: POST

Input:

- site_id (String)
- dg_number (String)
- url (String) – URL on which call back will happen whenever DG is UP/ Started

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "DG Sensing URL configured",
  "data" : {
    "dg_no" : "", //dg no, dg id
    "url" : ""
  }
}
```

- **Error case**

```
{
  "status" : "Error",
  "message" : "Not Configured",
  "data" : null
}
```

<https://isocietymanager.com>

Remarks:

The API will configure URL which will be called whenever DG supply is ON/ OFF

12. DG Status

Method: GET

Input:

- site_id (String)
- dg_number (String)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "DG Status",
  "data" : {
    "dg_no" : "",
    "status": "ON or OFF",
  }
}
```

//dg no, dg id

- **Error case**

```
{
  "status" : "Error",
  "message" : "Not ",
  "data" : null
}
```

Remarks:

The API will return is DG supply is ON/ OFF

13. DG Fuel Status

Method: GET

Input:

- site_id (String)
- dg_number (String)

Response:

- **Success case**

```
{
  "status" : "success",
  "message" : "DG Status",
  "data" : {
    "dg_no" : "",
    "fuel_balance" : "",
    "datetime": ""
  }
}
```

//dg no, dg id

<https://isocietymanager.com>

- **Error case**

```
{  
  "status" : "Error",  
  "message" : "Not ",  
  "data" : null  
}
```

Remarks:

The API will return fuel balance in the DG as of now