

MTS Actions API Specification

Introduction

This document describes the MTS Actions API which can be used by applications to perform various meter actions via MTS programmatically.

Basics

Authentication

All HTTP calls must contain an MTS Actions API Access Token contained in an Authorization header with the Bearer authentication scheme. An API Access token can be generated within the MTS application by an administrator.

To generate a token go to the Extras - MTS API page. Click **Create new token** and choose a valid duration (after which time the token will expire) and set the scope to **MTS Actions API**. Once the token has been created it must be copied and saved. It is not possible to view the token in MTS once it has been generated. The token is not stored in MTS and it is the responsibility of the MTS administrator to store the token securely. If the administrator believes that a token is compromised they can delete it in MTS at any time.

Note that the MTS Actions API is only available over HTTPS.

Actions

The actions that can be performed are as follows:

TimeUpdate

Used to set the meter time to the current GMT time. This action will only attempt to adjust the time if it deviates more that a fixed amount as determined by the MTS configuration.

MeterConfigure

Used to upload a configuration file to a meter. The contents of the file must be included in one of 2 formats: text or binary-base64.

GprsSetup

Used to set the GPRS values.

Limitations

The MTS Actions API is a work in progress and support for the actions does not exist for all meters. The table below shows the current action support:

| Meter type/Action | TimeUpdate | MeterConfigure | GprsSetup |
|-------------------|------------|----------------|-----------|
|-------------------|------------|----------------|-----------|

| Meter type/Action | TimeUpdate | MeterConfigure | GprsSetup |
|-------------------|------------|----------------|-----------|
| CEWEPRO | | | |
| CEWEPRO100 | supported | | |
| EDMIATLAS | supported | | supported |
| ELSTERA1700 | supported | | |
| ELSTERAS230 | supported | | |
| ELSTERA1140 | supported | | |
| EMLITECOP10 | supported | supported | |
| ISKRA_MX37X | supported | | |
| LG_DLMS | supported | | |
| PREMIERPRI | | | |

Multiple actions

It is possible to combine the TimeUpdate action with one other action in a single request provided the meter type supports both. It is not possible to combine other actions even if the meter type supports both.

API Methods

The API is implemented as an HTTP based set of methods that use JSON as the data representation format. The following methods are supported:

| Method | Type | HTTP Verb | Purpose |
|--------------------------------------|---------|-----------|---|
| action-request | Command | POST | Request an action to be performed on a meter. |
| action-status | Query | GET | Read the current status of a previously requested action. |
| action-cancel | Command | DELETE | Cancel a previously requested action. |
| time-update DEPRECATED | Command | POST | Set time for a meter to the current time. |
| meter-configure DEPRECATED | Command | POST | Upload a configuration file to a meter. |

action-request command

The action-request command is used to request for a new meter action to be performed.

A unique Request ID will be returned which can be used query the action status.

JSON Command Parameters

| Name | Type | Value | Mandatory |
|-------------------|--------------------------------|--|-----------|
| requestReference | String | Optional reference that the client may include in the request for their own use, e.g. MPAN. This parameter will be returned in the result but has no other significance. | NO |
| immediate | Bool | A boolean value. true => action will be run immediately, false => action will be run overnight. If omitted a value of false is assumed. | NO. |
| meterType | String | Specifies the type of meter as specified in the MTS user guide for batch requests. | YES |
| remoteAddress | String | Specifies the remote address used to connect to the meter. As specified in the MTS user guide for batch requests. | YES |
| comsSettings | String | Normally this field should be omitted but for cases where meters are configured in a non standard way this field can be used to override the default coms settings. This is only applicable for modem connections and can be used to specify the data bits, parity and stop bits in the form DPS, e.g. 7E1 to specify 7 stop bits, even parity and 1 stop bit. | NO |
| outstationAddress | String | Specifies the outstation address/device id of the meter. | NO |
| serialNumber | String | The meter serial number. A check will be made to determine if the meter returns this serial number and an error will be reported if there is a mismatch and the action will NOT proceed. | YES |
| password | String | The meter password. | NO |
| timeUpdate | Bool | Perform the TimeUpdate action. | NO |
| meterConfigure | Meter Configuration Parameters | Perform the MeterConfigure action. | NO |
| gprsSetup | GPRS Parameters | Perform the GprsSetup action. | NO |

Meter Configuration Parameters

| Name | Type | Value | Mandatory |
|----------|--------|---|-----------|
| contents | String | The content of the configuration file as an Unicode string. See comments above. | YES. |

| Name | Type | Value | Mandatory |
|--------------|--------|---|------------------------------|
| contentsType | String | This can have the values 'text' or 'binary-base64'. | NO. Default value is 'text'. |
| fileName | String | File name of the configuration file. | YES. |

For meters that expect a configuration file to be in a text format the contentsType should be set to **text**. All control characters in the text **must** be escaped using the \ character and sent as a two-character sequences, e.g. a new line must appear as \n and a carriage return as \r. A \ character must also be escaped and appear as \\. For more information see chapter 2.5 of [RFC 4627](#).

For meters that expect a configuration file to be in a binary format the contentsType should be set to **binary-base64** and the contents encoded as a Base 64 string. For more information see [RFC 4648](#)

GPRS Parameters

At least of these parameters must be included with a value. If a parameter is not included the register value will not be changed.

| Name | Type | Value | Mandatory |
|--------------|---------|--|-----------|
| apn | String | The new APN register value. | NO. |
| meterUdpPort | Integer | The new meter UDP port register value. | NO. |

JSON Response parameters

| Name | Type | Value | Mandatory |
|-----------|---------|---------------------|-----------|
| requestId | Integer | The MTS request ID. | YES |

Sample - TimeUpdate request

```
POST https://www.coherent-research.co.uk/MTS/action-request
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN
```

```
{
  "requestReference": "ABC",
  "meterType": "ELSTERA1700",
  "remoteAddress": "07777000000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "timeUpdate": true
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "requestId": 1234
}
```

Sample - MeterConfigure request

```
POST https://www.coherent-research.co.uk/MTS/action-request
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "requestReference": "ABC",
  "meterType": "EMLITECOP10",
  "remoteAddress": "12.34.56.78:1000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "meterConfigure": {
    "contents": "[Backlight]\r\n255.255.16 - 0x00\r\n[Decimal
Places]\r\n255.255.12 - 0x02"
  }
}

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "requestId": 1234
}
```

Sample - GprsSetup request

```
POST https://www.coherent-research.co.uk/MTS/action-request
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "requestReference": "ABC",
  "meterType": "EMLITECOP10",
  "remoteAddress": "12.34.56.78:1000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "gprsSetup": {
    "apn": "hello.world",
    "meterUdpPort": 123
  }
}
```

```
    }
  }

  HTTP/1.1 200 OK
  Content-Type: application/json; charset=utf-8

  {
    "requestId": 1234
  }
```

Sample - error case

```
POST https://www.coherent-research.co.uk/MTS/action-request
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "requestReference": "ABC",
  "meterType": "ELSTERA1700",
  "remoteAddress": "abc",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000"
}

HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8

{
  "details": "No actions specified. "
}
```

action-cancel command

The client can request the cancellation of any previously requested action. MTS will remove the matching request from the queue. If the request has already been processed (or is being processed) MTS will respond positively.

JSON Request Parameters

| Name | Type | Value | Mandatory |
|-----------|---------|---|-----------|
| requestId | Integer | The request ID returned by the command that requested the action. | YES |

JSON Response parameters

| Name | Type | Value | Mandatory |
|------|------|-------|-----------|
|------|------|-------|-----------|

| Name | Type | Value | Mandatory |
|-----------|---------|---|-----------|
| requestId | Integer | The request ID returned by the command that requested the action. | YES |

Sample - successful case

```
DELETE https://www.coherent-research.co.uk/MTS/action-cancel
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN
```

```
{
  "requestId": 1234
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "requestId": 1234
}
```

action-status query

The client can request the status of any previously requested action using the request ID.

URL Request Parameters

| Name | Type | Value | Mandatory |
|-----------|---------|--------------------------------------|-----------|
| requestId | Integer | ID returned by the relevant command. | YES |

JSON Response parameters

| Name | Type | Value | Mandatory |
|-------------------|---------|---|-----------|
| requestId | Integer | The MTS request ID. | YES |
| requestReference | String | Note 1 | NO |
| meterType | String | Note 1 | YES |
| remoteAddress | String | Note 1 | YES |
| comsSettings | String | Note 1 | NO |
| outstationAddress | String | Note 1 | NO |
| password | String | Note 1 | YES |
| action | String | Indicates the action the was requested. Values are "TimeUpdate", "MeterConfigure" | YES |

| Name | Type | Value | Mandatory |
|---------------------------|--------|---|------------|
| result | String | Indicates the overall result of the command. Values are "PENDING" (i.e. the command has not been performed yet), "SUCCESS" or "ERROR: details" where details describe the problem that caused the command to fail. | YES |
| actionRequestTime | String | The time the action command was received | NO |
| actionStartTime | String | The time the action started. In the case of multiple retries this will be the start of the first attempt. Note 1 | NO |
| actionEndTime | String | The time the action ended. In the case of multiple retries this will be the end of the last attempt. Note 1 | NO. |
| connectionStartTime | String | The time that the communication channel was opened. In the case of multiple retries this will be the start of the last attempt. Note 1 | NO |
| connectionEndTime | String | The time that the communication channel was closed. In the case of multiple retries this will be the of the last attempt. Note 1 | NO |
| serialNumber | String | The serial number received from the meter, or "ERROR: details" if it was not possible to fetch the serial number from the meter or if the serial number from the request is included and does not match the serial number read from the meter. | NO |
| meterTimeOffset | String | The amount of time the meter deviates from MTS server time (BEFORE any TimeUpdate action is performed). This is in the format +/-Ns, e.g. -203s (indicates that the time was 203 seconds behind the server time when it was read). If no time was read this will contain "ERROR: details" where details describe the problem. | NO. Note 3 |
| meterTimeOffsetPostUpdate | String | The amount of time the meter deviates from MTS server time AFTER a TimeUpdate action has been performed. The format is the same as for meterTimeOffset. | NO. Note 3 |
| NO | | | |

Notes:

1. The parameters from the original command are repeated here.
2. All times are in UTC and in the format YYYY-MM-DDTHH:mm:ssZ

3. These properties are only applicable for the TimeUpdate action.

Sample - pending action

```
GET https://www.coherent-research.co.uk/MTS/action-status?requestId=1234
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "requestId": 1234,
  "resultSummary": "PENDING",
}
```

Sample - successfully completed action

```
GET https://www.coherent-research.co.uk/MTS/action-status?requestId=1234
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "requestId": 1234,
  "requestReference": "0001",
  "meterType": "ELSTERA1700",
  "remoteAddress": "07777000000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "actionType": "TIME-UPDATE",
  "resultSummary": "SUCCESS",
  "actionStartTime": "2019-01-02T04:00:00",
  "actionEndTime": "2019-01-02T04:01:30",
  "connectionStartTime": "2019-01-02T04:02:00",
  "connectionEndTime": "2019-01-02T04:01:28",
  "serialNumber": "12345678",
  "meterTime": "2019-01-02T03T04:02:15 +10s"
}
```