MTS API Specification

Introduction

This document describes the MTS API which can be used by applications to perform MTS meter tests programmatically.

Terminology

- Server: refers to the MTS instance that receives the API calls.
- Client: refers to the application that makes the API calls.
- Command: an API call that may cause an action on the server, e.g. making a request to start a test.
- Query: an API call that does not cause an action on the server but may return data, e.g. requesting the current status of a test.

Basics

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**
authentication-sign-in	Command	POST	Send credentials to the server and get an access token. Note 1.
test-request	Command	POST	Request a new meter test
test-cancel	Command	DELETE	Cancel a previously requested test
test-status	Query	GET	Read the current status of a previously requested test
service-status	Query	GET	Check the current status and version of the MTS server. Note 1.

Notes

1. These methods do not require an authentication token.

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

Command Requests

All commands are sent as an HTTP request message and the command parameters are sent as a JSON object in the HTTP body:

```
VERB URL
Accept: application/json
Content-type: application/json

{
  "param1": "value1",
  "param2": "value2"
}
```

where VERB is either POST or DELETE depending on the command.

Query Requests

All queries are sent as an HTTP GET message and the query parameters are sent as part of the URL as a query string in the form:

```
GET URL?param1=value1&param2=value2
Accept: application/json
```

Response codes

Standard HTTP response codes are used in the response to all commands and queries. The following codes are used

Response	Meaning	Use
200	OK	Indicates that the command/query was successful and the response body contains a JSON object with the resulting data.
400	Bad Request	Indicates the request is not valid or understood by the server. The body of the response will provide more details as to why the request is considered bad.
401	Unauthorized	Indicates that the caller has not provided a valid authentication token (if required). See above
403	Forbidden	Indicates that the caller does not have the appropriate authority to perform the request (even though the authentication token is valid). The body of the response will provide more details.
429	Too Many Requests	Indicates that the caller has sent too many requests in a given amount of time. See the chapter Rate Limiting for more details.
500	Internal Server Error	Indicates a fault on the server. This should be considered an MTS error and the issue raised with Coherent Research

Response data

A response from the server (either to a command or a query) may contain data as a JSON object in the body of the HTTP response.

In the case of an error (i.e. any response code > 200) the response data will contains the following parameter:

Name	Type	Value	Mandatory
details	String	Reason the request failed as a human readable string	NO

In certain cases extra fields may be added to the response object (e.g. see the chapter on Rate limiting below).

Future compatibility

It is envisaged that the response payloads may gain extra properties over time. Therefore to assist in maintaining future compatibility the consumers of the service should silently ignore any properties in the response payloads that are not recognised.

authenticate-sign-in command

The authenticate-sign-in command is used to obtain an access token from the server. The access token is returned as a cookie in the response header and must be returned by the client in subsequent calls.

> The authentication cookie will have an expiration time after which it will no longer be accepted by the server and the server will respond to all requests with response code 401. In this case the authenticate-sign-in command must be repeated to obtain a new authentication token.

JSON Parameters

Name	Type	Value	Mandatory
username	String	MTS username	NO
password	String	MTS password	NO

Response

If the credentials are correct the server will respond with response code 200 and a cookie containing an authentication token. The cookie name will start with COHERENT-DCS-API.

Sample

```
POST https://www.coherent-research.co.uk/MTS/authenticate-sign-in
Accepts: application/json
Content-type: application/json

{"username": "user", "password": "mypassword" }

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Set-Cookie: COHERENT-MTS-API ...
```

```
{
  "serviceVersion": "3.0.0"
}
```

The cookie should be sent by the client to the server in all subsequent HTTP messages. The information returned in the response body is for information only.

Once authenticated the client must include the access code as a cookie in the header, e.g.:

```
GET https://www.coherent-research.co.uk/MTS/xxx
Accept: application/json
Cookie: COHERENT-MTS-API...
```

test-request command

The test-request command is used to request a new meter test.

The test-request command will use an HTTP POST message and the command parameters will be sent in the body of the message as a JSON object. If the request is accepted the HTTP response will have a status code 200 and the response parameters will be sent in the body of the message as a JSON object.

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
priority	Integer	An integer value. 1 => test will be run immediately, 2 => test will be run overnight. If omitted a value is 2 is assumed.	NO.
meterType	String	Specifies the type of meter to be tested. 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

Name	Type	Value	Mandatory
serialNumber	String	The meter serial number. If included 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	NO
password	String	The meter password.	NO
surveyDays	Number	Specifies the number of days of survey data to read. If this field is missing or zero no survey data will be collected	NO
surveyDate	String	Specifies the start date for reading survey data in the form yyyy-MM-dd. If this field is empty and surveyDays is > 0 then SURVEY_DATE will be assumed to be SURVEY_DAYS before the date this request is received.	NO

JSON Response parameters

Name	Type	Value	Mandatory
testId	Integer	The MTS test ID.	YES

Sample - successful case

```
POST https://www.coherent-research.co.uk/MTS/test-request
Accepts: application/json
Content-type: application/json
Cookie: COHERENT-MTS-API...

{
  "requestReference": "ABC",
  "meterType": "ELSTER_A1700",
  "remoteAddress": "07777000000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "surveyDays": 10,
  "surveyDate": "2019-12-01"
}

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

{
  "testId": 1234
}
```

Sample - error case

```
POST https://www.coherent-research.co.uk/MTS/test-request
Accepts: application/json
Content-type: application/json
Cookie: COHERENT-MTS-API...

{
  "requestReference": "ABC",
  "meterType": "ELSTER_A1700",
  "remoteAddress": "abc",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "surveyDays": 10,
  "surveyDate": "2019-12-01"
}

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

{
  "details": "Remote address is not in a recognised format"
}
```

test-cancel command

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

The test-cancel command will use an HTTP POST message and the parameters will be sent in the body of the message as a JSON object.

JSON Request Parameters

Name	Type	Value	Mandatory
testId	Integer	The test ID returned by the test-request command	YES

JSON Response parameters

Name	Type	Value	Mandatory
testId	String	The test ID returned by the test-request command	YES

Sample - successful case

```
DELETE https://www.coherent-research.co.uk/MTS/test-cancel
Accepts: application/json
Content-type: application/json
```

```
Cookie: COHERENT-MTS-API...

{
  "testId": 1234
}

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

{
  "testId": 1234
}
```

test-status query

The client can request the status of any previously requested test.

The test-status query will use an HTTP GET message and the parameters will be sent as part of the URL and the response will contain the information in JSON format in the response body.

URL Request Parameters

Name	Type	Value	Mandatory
testId	Integer	Test ID returned by the test-request command.	YES

JSON Response parameters

Name	Type	Value	Mandatory
testId	Integer	The MTS Test 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
surveyDays	Number	Note 1	YES
surveyDate	String	Note 1	NO
result	String	Indicates the overall result of the collection request. Values are "PENDING" (i.e. the collection has not been performed yet), "SUCCESS", "PARTIAL SUCCESS" (note 3) or "ERROR: details" where details describe the problem that caused the test to fail.	YES

Name	Type	Value	Mandatory
testRequestTime	String	The time the test-request command was received	NO
testStartTime	String	The time the test started. In the case of multiple retries this will be the start of the first attempt. Note 1	NO
testEndTime	String	The time the test 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
meterTime	String	The meter date/time received from the meter with an offset in seconds from the time the test took place in the format YYYY-MM-DDTHH:mm:ss +/- Ns, e.g. 2014-10-31T23:33:32 -203s (indicates that the time was 203 seconds behind the real time when it was read). If no time was read this will contain "ERROR: details" where details describe the problem.	NO
statusEvents	Array of strings	An array of strings representing status events indicated by the meter. These will vary by meter type	NO
registerValues	Array	An array of Register Value objects. See below	NO
surveyData	Array	An array of Register Survey Data objects. If no survey data was requested this property will be omitted. See below	NO

Notes:

1. The parameters from the original test-request command are repeated here.
2. All times are in UTC and in the format YYYY-MM-DDTHH:mm:ssZ
3. A test will be considered partially successful if some, but not all, of the data was collected.

Register Value Object

A Register Value Object contains an instantaneous value read from the meter.

Name	Type	Value	Mandatory

Name	Type	Value	Mandatory
name	String	The register name (depending on the meter type)	YES
value	Float	The value of the register	
units	String	The units of the register	

Register Survey Data Object

A Register Survey Value Object contains survey data for an individual register/channel from the meter:

Name	Type	Value	Mandatory
name	String	The register name (depending on the meter type)	YES
units	String	The units of the register	YES
readings	JSON object	An array of readings	YES

Survey Reading Object

Name	Type	Value	Mandatory
timestamp	String	The time of the reading (i.e. start of the half hour period).	YES
value	Number	The value of the register	YES

Sample - pending test

```
GET https://www.coherent-research.co.uk/MTS/test-status?testid=1234
Accepts: application/json
Cookie: COHERENT-MTS-API...

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

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

Sample - successfully completed test with survey data collection

```
GET https://www.coherent-research.co.uk/MTS/test-status?testid=1234
Accepts: application/json
Cookie: COHERENT-MTS-API...

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

```
{
  "testId": 1234,
  "requestReference": "0001",
  "meterType": "ELSTER_A1700",
  "remoteAddress": "07777000000",
  "outstationAddress": "1",
  "serialNumber": "12345678",
  "password": "AAAA0000",
  "surveyDays": 1,
  "surveyDate": "2019-01-01",
  "resultSummary": "SUCCESS",
  "testStartTime": "2019-01-02T04:00:00",
  "testEndTime": "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",
  "statusEvents" : [
    "Lid/terminal cover tamper"
  ],
  "registerValues": [
    {
      "name": "kWh Import",
      "timestamp": "2019-01-02T04:02:10Z",
      "value": "758",
      "units": "kWh"
    },
    {
      "name": "kvarh Q1",
      "timestamp": "2019-01-02T04:02:11Z",
      "value": "1190",
      "units": "kvarh"
    }
  ],
  "surveyData": [
    {
      "name": "kWh Import",
      "units": "kWh",
      [
        {
          "timestamp": "2019-01-01T00:00:00Z",
          "value": "640"
        },
        {
          "timestamp": "2019-01-01T00:30:00Z",
          "value": "645"
        },
        ...
        {
          "timestamp": "2019-01-01T23:30:00Z",
          "value": "700"
        }
      ]
   },
   {
     "name": "kvarh Q1",
     "units": "kvarh",
     [
       {
         "timestamp": "2019-01-01T00:00:00Z",
         "value": "900"
       },
      {
        "timestamp": "2019-01-01T00:30:00Z",
        "value": "910"
      },
      ...
      {
        "timestamp": "2019-01-01T23:30:00Z,
        "value": "920"
      },
    ]
  }
]
```

service-status query

MTS provides a method to check the status of the service itself. If the service is available the server will respond with code 200.

URL Request parameters

This method takes no parameters.

JSON Response parameters

Name	Type	Value	Mandatory
serviceVersion	String	The version of the service in the format X.Y.Z	YES
status	String	A string indicating the status of the service if it is running. Normally this will be OK	YES

Sample

```
GET https://www.coherent-research.co.uk/MTS/service-status
GET https://www.coherent-research.co.uk/MTS/test-status?testid=1234
Accepts: application/json

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

{
"serviceVersion": "3.0.0",
  "status": "OK"
}
```

> This method can be called without an access token.

Rate limiting

Rate limiting will be applied to the API.

When the rate is exceeded the request will be rejected with the HTTP status code 429.

JSON Response parameters

Name	Type	Value	Mandatory
backOffSuggestion	Integer	A suggested period that the caller should wait before retrying in seconds	YES

CORS

The MTS API is designed to be called by a non-browser client and as such CORS will not be enabled.