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

Authentication

All HTTP calls (with exceptions noted below) must contain an 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. More details to come.

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

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**
test-request	Command	POST	Request a new meter test
batch-request	Command	POST	Request a list of new meter tests in a batch
test-cancel	Command	DELETE	Cancel a previously requested test
batch-cancel	Command	DELETE	Cancel a previously requested batch of tests
test-status	Query	GET	Read the current status of a previously requested test
batch-status	Query	GET	Read the current status of a previously requested batch of tests
test-search	Query	GET	Search for tests based on criteria such as date, meter type etc

Method	Type	HTTP Verb	Purpose
service-status	Query	GET	Check the current status and version of the MTS server. Note 1.

Notes

1. This method does not require an access token.

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
Authorization: Bearer API-ACCESS-TOKEN
{
  "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
Authorization: Bearer API-ACCESS-TOKEN
```

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.

Response	Meaning	Use
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.
404	Not found	This is only returned if the URL is incorrect and not to convey any application specific information. If e.g. a TestId does not correspond to an existing test 400 will be used instead.
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	Array of strings	Reason or reasons the request failed as a human readable list of strings	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.

test-request command

The test-request command is used to request a new meter test. A unique Test Id will be returned which can be used query the test 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

Name	Type	Value	Mandatory
immediate	Bool	An boolean value. true => test will be run immediately, false => test will be run overnight. If omitted a value of false 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
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
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "requestReference": "ABC",
```

```
  "meterType": "ELSTERA1700",
  "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
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "requestReference": "ABC",
  "meterType": "ELSTERA1700",
  "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"
}
```

batch-request command

The batch-request command is used to request a set of new meter tests in one go. The request will return a BatchId that can be used to check the status of the whole set.

JSON Command Parameters

The batch-request command must contain an array of objects as defined in the test-request command.

JSON Response parameters

Name	Type	Value	Mandatory
batchId	Integer	The MTS batch ID.	YES

Sample - successful case

```
POST https://www.coherent-research.co.uk/MTS/batch-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",
     "surveyDays": 10,
     "surveyDate": "2019-12-01"
  },
  {
     "requestReference": "ABC",
     "meterType": "ELSTERA1700",
     "remoteAddress": "07777000001",
     "outstationAddress": "1",
     "serialNumber": "23456789",
     "password": "AAAA0001",
     "surveyDays": 3,
     "surveyDate": "2019-12-02"
  },
]

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

{
  "batchId": 1234
}
```

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.

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
Accept: application/json
Content-type: application/json
Authorization: Bearer API-ACCESS-TOKEN

{
  "testId": 1234
}

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

{
  "testId": 1234
}
```

batch-cancel command

The client can request the cancellation of any previously requested batch of tests. Cancelling a batch of tests means that all pending tests will be deleted from the system. Optionally, all completed tests may also be deleted from the system.

JSON Request Parameters

Name	Type	Value	Mandatory
batchId	Integer	The batch ID returned by the batch-request command	YES
deleteCompleted	Bool	Delete completed tests. If set to *true* all completed tests in the batch will be deleted. If set to *false* competed tests will be kept by MTS but pending tests will be cancelled and deleted. Default value is *false*.	NO

JSON Response parameters

Name	Type	Value	Mandatory

Name	Type	Value	Mandatory
batchId	String	The batch ID returned by the batch-request command	YES
cancelledCount	Integer	The number of tests cancelled. If *deleteCompleted* is set to *false* completed tests will not be included in this number.	YES

Sample - successful case

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

{
  "batchId": 1234
  "deleteCompleted": true
}

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

{
  "testId": 1234,
  "cancelledCount": 100
}
```

test-status query

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

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

Name	Type	Value	Mandatory
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
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

Name	Type	Value	Mandatory
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	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
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

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
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

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

{
  "testId": 1234,
  "requestReference": "0001",
  "meterType": "ELSTERA1700",
  "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"
        },
      ]
    }
  ]
```

batch-status query

The client can request the status of any previously requested batch of tests using the Batch ID. This query will return a summary of all tests in the batch along with the individual Test IDs. To fetch the actual status (and the completed results) for a given test the test-status query must be used with the Test ID.

URL Request Parameters

Name	Type	Value	Mandatory
batchId	Integer	Batch ID returned by the batch-request command.	YES

JSON Response parameters

Name	Type	Value	Mandatory
batchId	Integer	The MTS Batch ID.	YES
totalCount	Integer	The total number of tests in the batch	YES
completedCount	Integer	The number of completed tests in the batch	YES
status	Array	An array of Status Summary objects	YES

Status Summary Object

A Status Summary Object contains the summary of the status of an individual test.

Name	Type	Value	Mandatory
testId	Integer	The unique Test ID for the test	YES
requestReference	String		NO
meterType	String		YES
remoteAddress	String		YES
resultSummary	String	This parameter will have the same format as in the test-status query.	YES

Sample

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

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

{
  "batchId": 1234,
  "totalCount": 3,
  "completedCount": 1,
  "status": [
    {
      "testId": 1000,
      "requestReference": "0001",
      "meterType": "ELSTERA1700",
      "remoteAddress": "07777000000",
      "resultSummary": "SUCCESS",
    },
    {
      "testId": 1001,
      "requestReference": "0002",
      "meterType": "ELSTERA1700",
      "remoteAddress": "07777000001",
      "resultSummary": "PENDING",
```

```
    },
    {
      "testId": 1002,
      "requestReference": "0003",
      "meterType": "ELSTERA1700",
      "remoteAddress": "07777000002",
      "resultSummary": "PENDING",
    }
  ]
}
```

test-search query

This query allows the client to search the system for tests that match the specified search criteria. The query will return a list of matching tests. Since there may be a large number of results the client may read the results in pages by using the **limit** and **offset** parameters.

> Note that matching tests will be returned irrespective of whether they were initiated via a test-request command or a batch-request command

Name	Type	Value	Mandatory
requestReference	String		NO
fromTime	String	A UTC time in the format YYYY-MM-DDTHH:mm:ssZ	YES
toTime	String	A UTC time in the format YYYY-MM-DDTHH:mm:ssZ	NO
meterType	String	Filter results by meter type.	NO
remoteAddress	String	Filter results by remote address. Note that partial matches are included.	NO
status	String	Filter results by status. Values are "ALL", "COMPLETED", "PENDING". Default = ALL	NO
limit	Integer	The maximum number of results the server will return in one request. Note 1.	NO
offset	Integer	The number of results to skip. If omitted a value of 0 is assumed.	NO
reverseOrder	Bool	Results are normally returned in received order. If this parameter is set to true the results will be returned in the reverse order	NO

Notes

1. If the client does not include the **limit** parameter the server may still limit the number of results returned if it is above a fixed size. See chapter 'Input/Output Limits'. **limit** may be set to 0 which will result in the server just replying with the number of results. This could be used, e.g., to check the total number of pending tests.

JSON Response parameters

Name	Type	Value	Mandatory
totalResultCount	Integer	The total number of matching results.	YES
resultCount	Integer	The number results contained in **results**. See Note 1.	YES
offset	Integer	The current offset	YES
results	Array	Array of Status Summary Objects.	NO

Notes

1. If no data matches the criteria **resultCount** will be 0 and **results** array will be empty.

Status Summary Object

A Status Summary Object contains the summary of the status of an individual test.

Name	Type	Value	Mandatory
testId	Integer	The unique Test ID for the test	YES
receivedTimestamp	String	UTC time in the format YYYY-MM-DDTHH:mm:ssZ	YES
completedTimestamp	String	UTC time in the format YYYY-MM-DDTHH:mm:ssZ	NO
batchId	Integer	The unique Batch ID for the test if applicable	NO
requestReference	String		NO
meterType	String		YES
remoteAddress	String		YES
resultSummary	String	This parameter will have the same format as in the test-status query.	YES

Sample

```
GET https://www.coherent-research.co.uk/MTS/test-search?fromTime=2022-01-
01T00:00:00Z&meterType=ELSTERA1700
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

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

{
  "totalResultCount": 3,
  "resultCount": 3,
  "offset": 0,
  "results":
    [
```

```
      {
        "testId": 1000,
        "requestReference": "0001",
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:02:00Z",
        "remoteAddress": "07777000000",
        "resultSummary": "SUCCESS",
      },
      {
        "testId": 1001,
        "batchId": 123,
        "requestReference": "0002",
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:03:00Z",
        "remoteAddress": "07777000001",
        "resultSummary": "PENDING",
      },
      {
        "testId": 1002,
        "batchId": 123,
        "requestReference": "0003",
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:04:00Z",
        "remoteAddress": "07777000002",
        "resultSummary": "PENDING",
      }
    ]
}
```

Paging Sample

```
GET https://www.coherent-research.co.uk/MTS/test-search?fromTime=2022-01-
01T00:00:00Z&meterType=ELSTERA1700&limit=2&offset=0
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

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

{
  "totalResultCount": 3,
  "resultCount": 2,
  "offset": 0,
  "results":
    [
      {
        "testId": 1000,
        "requestReference": "0001",
```

```
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:02:00Z",
        "remoteAddress": "07777000000",
        "resultSummary": "SUCCESS",
      },
      {
        "testId": 1001,
        "batchId": 123,
        "requestReference": "0002",
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:03:00Z",
        "remoteAddress": "07777000001",
        "resultSummary": "PENDING",
      }
    ]
}

GET https://www.coherent-research.co.uk/MTS/test-search?fromTime=2022-01-
01T00:00:00Z&meterType=ELSTERA1700&limit=2&offset=1
Accept: application/json
Authorization: Bearer API-ACCESS-TOKEN

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

{
  "totalResultCount": 3,
  "resultCount": 1,
  "offset": 1,
  "results":
    [
      {
        "testId": 1002,
        "batchId": 123,
        "requestReference": "0003",
        "meterType": "ELSTERA1700",
        "receivedTimestamp": "2022-01-01T00:01:00Z",
        "completedTimestamp": "2022-01-01T00:04:00Z",
        "remoteAddress": "07777000002",
        "resultSummary": "PENDING",
      }
    ]
}
```

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
Accept: 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. The rate limiting uses a simple fixed window algorithm limiting clients to N calls every X seconds (see Note 1).

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

JSON Response parameters

Name	Type	Value	Mandatory
retryAfter	Integer	A suggested period (in seconds) that the client should wait before retrying	YES

Notes

1. Actual values to be decided.

Input/Output Limits

The following limits are applied by the server:

Limit	Value
The number of tests in a single batch.	Note 1
The maximum number of survey days requested per test.	
The maximum time range in a search query.	
The maximum limit size in a search query.	
The maximum number of results returned by a search query if no limit is specified.	

Notes

1. Actual values to be decided.

CORS

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