Yamcs Server API

YAMCS-SA-API-001

December, 8th 2015

www.yamcs.org

Yamcs Server API

YAMCS-SA-API-001

This version was published December, 8th 2015
A later version of this document may be available at www.yamcs.org

Table of Contents

Chapter 1. REST API

1.1. General

Yamcs provides a RESTful web service allowing remote tools to interface with a growing set of internals. All operations accept both JSON and Protobuf as a media type. These pages document the use of the common JSON format.

HTTP Verbs

The supported verbs in the REST API are:

`GET`	Retrieve a resource
`POST`	Create a new resource
`PATCH`	Update an existing resource
`DELETE`	Delete a resource

Many HTTP clients (including Java's `HttpsUrlConnection`) are not able to send request bodies over `GET`. Therefore whenever an operation documents a `GET` request body, it will accept both `GET` and `POST` requests.

In addition, since the use of the HTTP `PATCH` method is not as widespread as it ought to be, Yamcs accepts both `PUT` or `POST` to be used instead.

Time

All timestamps are returned as UTC and formatted according to ISO 8601. E.g.

```
2015-08-26T08:08:40.724
2015-08-26
```

Namespaces

Mission Database resources like parameters and containers are available under different namespaces. When an operation documents the use of `/:namespace/:name` in the URI, the namespace segment can be repeated for every subsystem in case of hierarchical XTCE names. For example, these resources evaluate to the same parameter resource:

```
/api/mdb/simulator/parameters/MDB%3AOPS+Name/SIMULATOR_BatteryVoltage2
/api/mdb/simulator/parameters/YSS/SIMULATOR/BatteryVoltage2
```

Notice as well the use of `%3A` and `+` to URL-encode `MDB:OPS Name` to the ASCII character set. The server supports UTF-8 but your client may not.

Error Handling

When an exception is caught while handling a REST request, the server will try to give some feedback to the client by wrapping it in a generic exception message like so:

```
{
  "exception" : {
    "type": "<short>",
    "msg": "<long>"
  }
}
```

Clients of the REST API should check on whether the status code is between 200 and 299, and if not, interpret the response with the above structure.

Curlability

JSON responses are by default configured for an improved curl experience. This means that responses will be pretty-printed, and that they will contain links to other URL resources where it makes sense. When building an actual client you may want to turn this off for a lighter payload.

Query Parameter	Description
`pretty`	Whether to autoformat JSON responses. Default: `yes`
`nolink`	Whether to hide links to related REST resources. Default: `no`

For example, append all URLs like so, makes responses lighter and faster:

```
?nolink&pretty=no
```

Protobuf

While it is in general recommended to use JSON as documented throughout this chapter, all REST operations also support Google Protocol Buffers for a lighter footprint. Yamcs Studio, for example, uses exclusively binary Protobuf messages. To mark your requests as Protobuf, set this HTTP header:

```
Content-Type: application/protobuf
```

If you also want to server to respond with Protobuf messages, add the `Accept` header:

```
Accept: application/protobuf
```

The `.proto` files that define the contract of the REST endpoint are included in the chapter Proto Files. Using the `protoc` compiler, client code can be generated for Java, Python, C++ and more.

The applicable top-level Protobuf messages are documented for every operation separately. If the response status is not between `200` and `299`, deserialize the response as of type `RestExceptionMessage`.

1.2. Alarms

1.2.1. List Alarms

List the history of alarms:

```
GET /api/archive/:instance/alarms
```

List the history of alarms for the given parameter:

```
GET /api/archive/:instance/alarms/:namespace/:name
```

For each alarm you get full information on the value occurrence that initially triggered the alarm, the most severe value since it originally triggered, and the latest value at the time of your request.

Parameters

Name	Type	Description
`start`	`string`	Filter the lower bound of the alarm's trigger time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the alarm's trigger time. Specify a date string in ISO 8601 format
`pos`	`integer`	The zero-based row number at which to start outputting results. Default: `0`
`limit`	`integer`	The maximum number of returned records per page. Choose this value too high and you risk hitting the maximum response size limit enforced by the server. Default: `100`
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. The sorting is always by trigger time (i.e. the generation time of the trigger value). Default: `desc`

Example

```
Status: 200 OK
```

```
{
  "alarm" : [ {
    "seqNum" : 1,
    "triggerValue" : {
      "id" : {
        "name" : "/YSS/SIMULATOR/BatteryVoltage2"
      },
      "rawValue" : {
        "type" : "UINT32",
        "uint32Value" : 49
      },
      "engValue" : {
```

```
        "type" : "UINT32",
        "uint32Value" : 49
      },
      "acquisitionTime" : 1448229350720,
      "generationTime" : 1448229333628,
      "acquisitionStatus" : "ACQUIRED",
      "processingStatus" : true,
      "monitoringResult" : "WATCH_LOW",
      "acquisitionTimeUTC" : "2015-11-22T21:55:14.720",
      "generationTimeUTC" : "2015-11-22T21:54:57.628",
      "expirationTime" : 1448229357720,
      "expirationTimeUTC" : "2015-11-22T21:55:21.720",
      "alarmRange" : [ {
        "level" : "WATCH",
        "minInclusive" : 50.0
      }, {
        "level" : "WARNING",
        "minInclusive" : 40.0
      }, {
        "level" : "DISTRESS",
        "minInclusive" : 30.0
      }, {
        "level" : "CRITICAL",
        "minInclusive" : 20.0
      }, {
        "level" : "SEVERE",
        "minInclusive" : 10.0
      } ]
    },
    "mostSevereValue" : {
      "id" : {
        "name" : "/YSS/SIMULATOR/BatteryVoltage2"
      },
      "rawValue" : {
        "type" : "UINT32",
        "uint32Value" : 39
      },
      "engValue" : {
        "type" : "UINT32",
        "uint32Value" : 39
      },
      "acquisitionTime" : 1448229413038,
      "generationTime" : 1448229395945,
      "acquisitionStatus" : "ACQUIRED",
      "processingStatus" : true,
      "monitoringResult" : "WARNING_LOW",
      "acquisitionTimeUTC" : "2015-11-22T21:56:17.038",
      "generationTimeUTC" : "2015-11-22T21:55:59.945",
      "expirationTime" : 1448229420038,
      "expirationTimeUTC" : "2015-11-22T21:56:24.038",
```

```
      "alarmRange" : [ {
        "level" : "WATCH",
        "minInclusive" : 50.0
      }, {
        "level" : "WARNING",
        "minInclusive" : 40.0
      }, {
        "level" : "DISTRESS",
        "minInclusive" : 30.0
      }, {
        "level" : "CRITICAL",
        "minInclusive" : 20.0
      }, {
        "level" : "SEVERE",
        "minInclusive" : 10.0
      } ]
    },
    "currentValue" : {
      "id" : {
        "name" : "/YSS/SIMULATOR/BatteryVoltage2"
      },
      "rawValue" : {
        "type" : "UINT32",
        "uint32Value" : 48
      },
      "engValue" : {
        "type" : "UINT32",
        "uint32Value" : 48
      },
      "acquisitionTime" : 1448229356954,
      "generationTime" : 1448229339867,
      "acquisitionStatus" : "ACQUIRED",
      "processingStatus" : true,
      "monitoringResult" : "WATCH_LOW",
      "acquisitionTimeUTC" : "2015-11-22T21:55:20.954",
      "generationTimeUTC" : "2015-11-22T21:55:03.867",
      "expirationTime" : 1448229363954,
      "expirationTimeUTC" : "2015-11-22T21:55:27.954",
      "alarmRange" : [ {
        "level" : "WATCH",
        "minInclusive" : 50.0
      }, {
        "level" : "WARNING",
        "minInclusive" : 40.0
      }, {
        "level" : "DISTRESS",
        "minInclusive" : 30.0
      }, {
        "level" : "CRITICAL",
        "minInclusive" : 20.0
```

```
      }, {
        "level" : "SEVERE",
        "minInclusive" : 10.0
      } ]
    }
  } ]
}
```

Protobuf

Response

rest.proto

```
message ListAlarmsResponse {
  repeated alarms.AlarmData alarm = 1;
}
```

1.2.2. Edit Alarm

```
PATCH
/api/processors/:instance/:processor/parameters/:namespace/:name/alarms/:s
```

Parameters

Name	Type	Description
`state`	`string`	**Required.** The state of the alarm. Either `acknowledged` or `unacknowledged`
`comment`	`string`	Message documenting the alarm change.

Example

```
{
  "state": "acknowledged",
  "comment": "bla bla"
}
```

Protobuf

Request

rest.proto

```
message EditAlarmRequest {
  optional string state = 1;
  optional string comment = 2;
}
```

1.3. Clients

1.3.1. List Clients

List all clients:

```
GET /api/clients
```

List all clients for the given Yamcs instance:

```
GET /api/instances/:instance/clients
```

List all clients for the given processor:

```
GET /api/processors/:instance/:processor/clients
```

Response

```
Status: 200 OK
```

```
todo
```

Protobuf

Response

```
                                                                    rest.proto
```

```
message ListClientsResponse {
  repeated yamcsManagement.ClientInfo client = 1;
}
```

1.3.2. Edit Client

Edit a client:

```
PATCH /api/clients/:id
```

Parameters

Name	Type	Description
`processor`	`string`	The processor. This processor must be part of the instance that the client is connected to.

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Update the client's processor to `replay123`:

```
{
  "processor" : "replay123"
}
```

Protobuf

Request

rest.proto

```
message EditClientRequest {
  optional string processor = 1;
}
```

1.4. Commanding

1.4.1. Issue Command

Issue a new command of the given type:

```
POST /api/processors/:instance/:processor/commands/:namespace/:name
```

After validating the input parameters, the command will be added to the appropriate command queue for further dispatch.

Parameters

Name	Type	Description
`origin`	`string`	The origin of the command. Typically in "user@host" format.
`sequenceNumber`	`integer`	The sequence number as specified by the origin. This gets communicated back in command history and command queue entries, thereby allowing clients to map local with remote command identities.
`dryRun`	`bool`	Whether a response will be returned without actually issuing the command. This is useful when debugging commands. Default `no`
`assignment`	`array of string pairs`	The name/value assignments for this command. **Note:** We enjoy simple curl-able APIs, so have made it possible to provide assignments in the query string in the format `name=value`. However, when building your own client, make sure to use the request body to prevent any potential naming collisions with other URI parameters.

Example

```
{
  "sequenceNumber" : 1,
  "origin" : "user@my-machine",
  "assignment" : [ {
    "voltage_num": 3
  } ],
  "dryRun" : true
}
```

Response

```
Status: 200 OK
```

```
{
  "queue" : "default",
  "source" : "SWITCH_VOLTAGE_ON(voltage_num: 3)",
  "hex" : "1864C00000000000000006A0000000103",
  "binary" : "GGTAAAAAAAAAABqAAAAAQM="
}
```

The binary is encoded in Base64 format.

Protobuf

Request

rest.proto

```
message IssueCommandRequest {
  message Assignment {
    optional string name = 1;
    optional string value = 2;
  }
  repeated Assignment assignment = 1;
  optional string origin = 2;
  optional int32 sequenceNumber = 3;
  optional bool dryRun = 4;
}
```

Response

rest.proto

```
message IssueCommandResponse {
  optional string queue = 1;
  optional string source = 2;
  optional string hex = 3;
  optional bytes binary = 4;
}
```

1.4.2. List Command Queues

List all command queues for the given processor:

```
GET /api/processors/:instance/:processor/cqueues
```

Response

```
Status: 200 OK
```

```
{
  "queue" : [ {
    "instance" : "simulator",
    "processorName" : "realtime",
    "name" : "default",
    "state" : "ENABLED",
    "nbSentCommands" : 0,
    "nbRejectedCommands" : 0,
    "url" :
"http://localhost:8090/api/processors/simulator/realtime/queues/default"

  } ]
}
```

Protobuf

Response

```
                                                                rest.proto
```

```
message ListCommandQueuesResponse {
  repeated commanding.CommandQueueInfo queue = 1;
}
```

1.4.3. Get Command Queue

Get data on a command queue:

```
GET /api/processors/:instance/:processor/cqueues/:name
```

Response

```
{
  "instance" : "simulator",
  "processorName" : "realtime",
  "name" : "default",
  "state" : "BLOCKED",
  "nbSentCommands" : 0,
  "nbRejectedCommands" : 0,
  "entry" : [ {
    "instance" : "simulator",
    "processorName" : "realtime",
    "queueName" : "default",
    "cmdId" : {
```

```
      "generationTime" : 1448782973440,
      "origin" : "000349-WS.local",
      "sequenceNumber" : 5,
      "commandName" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
    },
    "source" : "SWITCH_VOLTAGE_OFF(voltage_num: 2)",
    "binary" : "GGTAAAAAAAAAABqAAAAAgI=",
    "username" : "anonymous",
    "generationTime" : 1448782973440,
    "uuid" : "3e867111-048a-4343-b195-47ba07d07093"
  } ],
  "url" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues/default"

}
```

Protobuf

Response

commanding.proto

```
message CommandQueueInfo {
  required string instance = 1;
  required string processorName = 2;
  required string name = 3;
  optional QueueState state = 4;
  required int32 nbSentCommands = 5;
  required int32 nbRejectedCommands = 6;
  optional int32 stateExpirationTimeS = 7;
  optional string url = 8;
}
```

1.4.4. Edit Command Queue

Edit a command queue:

```
PATCH /api/processors/:instance/:processor/cqueues/:name
```

Parameters

Name	Type	Description
`state`	`string`	The state of the queue. Either `enabled`, `disabled` or `blocked`.

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Block a queue:

```
{
  "state" : "blocked"
}
```

The response contains the updated queue information:

```
Status: 200 OK
```

```
{
  "instance" : "simulator",
  "processorName" : "realtime",
  "name" : "default",
  "state" : "BLOCKED",
  "nbSentCommands" : 0,
  "nbRejectedCommands" : 0,
  "url" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues/default"

}
```

Protobuf

Request

```
                                                                rest.proto
```

```
message EditCommandQueueRequest {
  optional string state = 1;
}
```

Response

```
                                                            commanding.proto
```

```
message CommandQueueInfo {
  required string instance = 1;
  required string processorName = 2;
  required string name = 3;
  optional QueueState state = 4;
  required int32 nbSentCommands = 5;
  required int32 nbRejectedCommands = 6;
  optional int32 stateExpirationTimeS = 7;
```

```
  repeated CommandQueueEntry entry = 8;
  optional string url = 9;
}
```

1.4.5. List Queued Commands

List all queued command entries for the given command queue:

```
GET /api/processors/:instance/:processor/cqueues/:name/entries
```

Response

```
Status: 200 OK
```

```
{
  "entry" : [ {
    "instance" : "simulator",
    "processorName" : "realtime",
    "queueName" : "default",
    "cmdId" : {
      "generationTime" : 1448782973440,
      "origin" : "000349-WS.local",
      "sequenceNumber" : 5,
      "commandName" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
    },
    "source" : "SWITCH_VOLTAGE_OFF(voltage_num: 2)",
    "binary" : "GGTAAAAAAAAAABqAAAAAgI=",
    "username" : "anonymous",
    "generationTime" : 1448782973440,
    "uuid" : "3e867111-048a-4343-b195-47ba07d07093"
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                                rest.proto
```

```
message ListCommandQueueEntries {
  repeated commanding.CommandQueueEntry entry = 1;
}
```

1.4.6. Edit Queued Command

Edit a command queue entry:

```
PATCH
/api/processors/:instance/:processor/cqueues/:cqueue/entries/:uuid
```

Parameters

Name	Type	Description
`state`	`string`	The state of the entry. Either `released` or `rejected`.

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Release an entry:

```
{
  "state" : "released"
}
```

Protobuf

Request

rest.proto

```
message EditCommandQueueEntryRequest {
  optional string state = 1;
}
```

1.5. Events

1.5.1. List Events

List the history of events:

```
GET /api/archive/:instance/events/
```

Parameters

Name	Type	Description
`source`	`array of strings`	The source of the events. Both these notations are accepted: • `?source=DataHandler,CustomAlgorithm` • `?source[]=DataHandler&source[]=CustomAlgorithm` Names must match exactly.
`start`	`string`	Filter the lower bound of the event's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the event's generation time. Specify a date string in ISO 8601 format
`pos`	`integer`	The zero-based row number at which to start outputting results. Default: `0`
`limit`	`integer`	The maximum number of returned records per page. Choose this value too high and you risk hitting the maximum response size limit enforced by the server. Default: `100`
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `desc`

The `pos` and `limit` allow for pagination. Keep in mind that in-between two requests extra data may have been recorded, causing a shift of the results. This stateless operation does not provide a reliable mechanism against that, so address it by overlapping your `pos` parameter with rows of the previous query. In this example we overlap by 4:

```
?pos=0&limit=50&order=desc
?pos=45&limit=50&order=desc
```

An alternative is to download the events instead.

Response

```
Status: 200 OK
```

```
{
  "event" : [ {
    "source" : "AlarmChecker",
    "generationTime" : 1447425863786,
```

```
    "receptionTime" : 1447425863786,
    "seqNumber" : 15,
    "type" : "IN_LIMITS",
    "message" : "Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed
to value 222",
    "severity" : "INFO",
    "generationTimeUTC" : "2015-11-13T14:43:47.786",
    "receptionTimeUTC" : "2015-11-13T14:43:47.786"
  } ]
}
```

Alternative Media Types

CSV

Use HTTP header:

```
Accept: text/csv
```

Or, add this query parameter to the URI: `format=csv`.

Response:

```
Status: 200 OK
Content-Type: text/csv
```

```
Source  Generation Time Reception Time  Event Type      Event Text
AlarmChecker    2015-11-13T14:46:36.029 2015-11-13T14:46:36.029
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 195
AlarmChecker    2015-11-13T14:46:29.784 2015-11-13T14:46:29.784
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 196
AlarmChecker    2015-11-13T14:46:23.571 2015-11-13T14:46:23.571
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 197
```

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Responses are of type:

rest.proto

```
message ListEventsResponse {
  repeated yamcs.Event event = 1;
}
```

1.5.2. Download Events

Download archived events:

```
GET /api/archive/:instance/downloads/events
```

This operation will possibly download a very large file. If you worry about size for your application, check out the support for paged event retrievalsinstead.

Parameters

Name	Type	Description
`source`	`array of strings`	The source of the events. Both these notations are accepted: • `?source=DataHandler,CustomAlgorithm` • `?source[]=DataHandler&source[]=CustomAlgorithm` Names must match exactly.
`start`	`string`	Filter the lower bound of the event's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the event's generation time. Specify a date string in ISO 8601 format
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `asc`

Response

The response will be a stream of self-standing event records.

Alternative Media Types

CSV

Use HTTP header:

```
Accept: text/csv
```

Or, add this query parameter to the URI: `format=csv`.

Response:

```
Status: 200 OK
Content-Type: text/csv
```

```
Source  Generation Time Reception Time  Event Type      Event Text
AlarmChecker    2015-11-13T14:46:36.029 2015-11-13T14:46:36.029
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 195
AlarmChecker    2015-11-13T14:46:29.784 2015-11-13T14:46:29.784
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 196
AlarmChecker    2015-11-13T14:46:23.571 2015-11-13T14:46:23.571
IN_LIMITS       Parameter /YSS/SIMULATOR/BatteryVoltage2 has changed to
value 197
```

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response if a stream of individual Protobuf messages delimited by a `VarInt`. Messages are of type:

yamcs.proto

```
message Event {
  enum EventSeverity {
    INFO = 0;
    WARNING = 1;
    ERROR = 2;
  }
  required string source = 1;
  required int64 generationTime = 2;
  required int64 receptionTime = 3;
  required int32 seqNumber = 4;
  optional string type = 5;
  required string message = 6;
  optional EventSeverity severity = 7[default=INFO];

  optional string generationTimeUTC = 8;
  optional string receptionTimeUTC = 9;

  extensions 100 to 10000;
}
```

1.6. Indexes

1.6.1. Download Packet Index

Download the index of stored packets for the given instance:

```
GET /api/archive/:instance/indexes/packets
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.
`name`	`array of strings`	Exact qualified names of the packets to include in the index. By default all packets will be included. **Partial wildcard matching is not currently supported.**

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
{
  "id" : {
    "name" : "/YSS/SIMULATOR/FlightData"
  },
  "first" : 1448272053375,
  "last" : 1448272089628,
  "num" : 181
}{
  "id" : {
    "name" : "/YSS/SIMULATOR/Power"
  },
  "first" : 1448272059406,
  "last" : 1448272084398,
  "num" : 5
}{
  "id" : {
    "name" : "/YSS/SIMULATOR/DHS"
```

```
  },
  "first" : 1448272059406,
  "last" : 1448272084398,
  "num" : 5
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

yamcs.proto

```
message ArchiveRecord {
  required NamedObjectId id = 1;
  required int64 first = 2;
  required int64 last = 3;
  required int32 num = 4;
  optional string info = 5;
}
```

1.6.2. Download PP Index

Download the index of stored processed parameter groups for the given instance:

```
GET /api/archive/:instance/indexes/pp
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful

for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
{
  "id" : {
    "name" : "simulation"
  },
  "first" : 1448616349067,
  "last" : 1448616467049,
  "num" : 2
}{
  "id" : {
    "name" : "simulation"
  },
  "first" : 1448617368018,
  "last" : 1448617368018,
  "num" : 1
}{
  "id" : {
    "name" : "simulation"
  },
  "first" : 1448617667330,
  "last" : 1448617667330,
  "num" : 1
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

```
                                                                yamcs.proto
```

```
message ArchiveRecord {
  required NamedObjectId id = 1;
  required int64 first = 2;
  required int64 last = 3;
  required int32 num = 4;
  optional string info = 5;
}
```

1.6.3. Download Event Index

Download the index of stored events for the given instance:

```
GET /api/archive/:instance/indexes/events
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
{
  "id" : {
    "name" : "CustomAlgorithm"
  },
  "first" : 1448272052179,
  "last" : 1448272052241,
  "num" : 2
}{
  "id" : {
    "name" : "CustomAlgorithm"
  },
  "first" : 1448272057109,
  "last" : 1448272062209,
  "num" : 3
}{
  "id" : {
    "name" : "CustomAlgorithm"
  },
  "first" : 1448272067107,
  "last" : 1448272072206,
  "num" : 3
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

yamcs.proto

```
message ArchiveRecord {
  required NamedObjectId id = 1;
  required int64 first = 2;
  required int64 last = 3;
  required int32 num = 4;
  optional string info = 5;
}
```

1.6.4. Download Command Index

Download the index of stored commands for the given instance:

```
GET /api/archive/:instance/indexes/commands
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
{
  "id" : {
    "name" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
  },
```

```
  "first" : 1448731973739,
  "last" : 1448731973739,
  "num" : 1
}{
  "id" : {
    "name" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
  },
  "first" : 1448782973440,
  "last" : 1448782973440,
  "num" : 1
}{
  "id" : {
    "name" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
  },
  "first" : 1448783668726,
  "last" : 1448783674298,
  "num" : 3
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

yamcs.proto

```
message ArchiveRecord {
  required NamedObjectId id = 1;
  required int64 first = 2;
  required int64 last = 3;
  required int32 num = 4;
  optional string info = 5;
}
```

1.6.5. Download Completeness Index

Download the completeness index records for the given instance:

```
GET /api/archive/:instance/indexes/completeness
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
todo
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

yamcs.proto

```
message ArchiveRecord {
  required NamedObjectId id = 1;
  required int64 first = 2;
  required int64 last = 3;
  required int32 num = 4;
  optional string info = 5;
}
```

1.6.6. Download Bulk Indexes

Download multiple indexes at the same time for the given instance:

```
GET /api/archive/:instance/indexes/completeness
```

This operation will possibly download a very large file.

Parameters

Name	Type	Description
`start`	`string`	The time at which to start retrieving index records.
`stop`	`string`	The time at which to stop retrieving index records.
`filter`	`array of strings`	The type of indexes to retrieve. Choose out of `tm`, `pp`, `events`, `commands` or `completeness`. By default all indexes will be sent.
`packetname`	`array of strings`	Specify exact names for the TM packets for which you want to retrieve index records. Setting this parameter, automatically implies that `tm` is added to the filter.

Example request URI:

```
/api/archive/simulator/indexes?
filter=commands,events&packetname=/YSS/SIMULATOR/Power
```

Example

You get back a sequence of consecutive self-standing JSON objects. Note that this JSON output can be useful for testing, but you probably want to use the Protobuf media type for decreased network traffic.

```
Status: 200 OK
```

```
{
  "instance" : "simulator",
  "records" : [ {
    "id" : {
      "name" : "/YSS/SIMULATOR/Power"
    },
    "first" : 1448272059406,
    "last" : 1448272084398,
    "num" : 5
  }, {
    "id" : {
      "name" : "/YSS/SIMULATOR/Power"
    },
    "first" : 1448823600003,
    "last" : 1448824423242,
    "num" : 133
  } ],
  "type" : "histogram",
  "tableName" : "tm"
```

```
}{
  "instance" : "simulator",
  "records" : [ {
    "id" : {
      "name" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_OFF"
    },
    "first" : 1448731973739,
    "last" : 1448731973739,
    "num" : 1
  }, {
    "id" : {
      "name" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_ON"
    },
    "first" : 1448791891823,
    "last" : 1448791891823,
    "num" : 1
  } ],
  "type" : "histogram",
  "tableName" : "cmdhist"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of individual Protobuf messages delimited with a `VarInt`. Every message is of type:

yamcs.proto

```
message IndexResult {
  required string instance = 1;
  repeated ArchiveRecord records = 2;
  //type can be histogram or completeness
  optional string type = 3;
  //if type=histogram, the tableName is the table for which the
histogram is sent
  optional string tableName = 4;
}
```

1.7. Instances

1.7.1. List Instances

List all configured Yamcs instances:

```
GET /api/instances
```

Response

```
Status: 200 OK
```


```
{
  "instance" : [ {
    "name" : "simulator",
    "missionDatabase" : {
      "configName" : "landing",
      "name" : "",
      "spaceSystem" : [ {
        "name" : "yamcs",
        "qualifiedName" : "/yamcs"
      }, {
        "name" : "YSS",
        "qualifiedName" : "/YSS",
        "sub" : [ {
          "name" : "SIMULATOR",
          "qualifiedName" : "/YSS/SIMULATOR"
        } ]
      }, {
        "name" : "GS",
        "qualifiedName" : "/GS"
      } ],
      "url" : "http://localhost:8090/api/mdb/simulator",
      "parametersUrl" :
"http://localhost:8090/api/mdb/simulator/parameters{/namespace}
{/name}",
      "containersUrl" :
"http://localhost:8090/api/mdb/simulator/containers{/namespace}
{/name}",
      "commandsUrl" :
"http://localhost:8090/api/mdb/simulator/commands{/namespace}{/name}"
    },
    "processor" : [ {
      "name" : "realtime",
      "url" :
"http://localhost:8090/api/processors/simulator/realtime",
      "clientsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/clients",
```

```
      "parametersUrl" :
"http://localhost:8090/api/processors/simulator/realtime/parameters{/names
{/name}",
      "commandsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/commands{/namespa
{/name}",
      "commandQueuesUrl" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues{/name}"

    } ],
    "url" : "http://localhost:8090/api/instances/simulator",
    "clientsUrl" :
"http://localhost:8090/api/instances/simulator{/processor}/clients",
    "eventsUrl" :
"http://localhost:8090/api/instances/simulator{/processor}/events"
  } ]
}
```

Protobuf

Response

rest.proto

```
message ListInstancesResponse {
  repeated yamcsManagement.YamcsInstance instance = 1;
}
```

1.7.2. Get Instance Detail

Get data on a Yamcs instance:

```
GET /api/instances/:instance
```

Response

```
Status: 200 OK
```

```
{
  "name" : "simulator",
  "missionDatabase" : {
    "configName" : "landing",
    "name" : "",
    "spaceSystem" : [ {
      "name" : "yamcs",
      "qualifiedName" : "/yamcs"
```

```
    }, {
      "name" : "YSS",
      "qualifiedName" : "/YSS",
      "sub" : [ {
        "name" : "SIMULATOR",
        "qualifiedName" : "/YSS/SIMULATOR"
      } ]
    }, {
      "name" : "GS",
      "qualifiedName" : "/GS"
    } ],
    "url" : "http://localhost:8090/api/mdb/simulator",
    "parametersUrl" :
"http://localhost:8090/api/mdb/simulator/parameters{/namespace}
{/name}",
    "containersUrl" :
"http://localhost:8090/api/mdb/simulator/containers{/namespace}
{/name}",
    "commandsUrl" :
"http://localhost:8090/api/mdb/simulator/commands{/namespace}{/name}"
  },
  "processor" : [ {
    "name" : "realtime",
    "url" : "http://localhost:8090/api/processors/simulator/realtime",
    "clientsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/clients",
    "parametersUrl" :
"http://localhost:8090/api/processors/simulator/realtime/parameters{/names
{/name}",
    "commandsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/commands{/namespa
{/name}",
    "commandQueuesUrl" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues{/name}"

  } ],
  "url" : "http://localhost:8090/api/instances/simulator",
  "clientsUrl" :
"http://localhost:8090/api/instances/simulator{/processor}/clients",
  "eventsUrl" :
"http://localhost:8090/api/instances/simulator{/processor}/events"
}
```


If an instance does not have web services enabled, it will be listed among the results, but none of its URLs will be filled in.

Protobuf

Response:

yamcsManagement.proto

```
message YamcsInstance {
  required string name = 1;
  optional MissionDatabase missionDatabase = 3;
  repeated ProcessorInfo processor = 4;
  optional string url = 5;
  optional string clientsUrl = 6;
  optional string commandQueuesUrl = 7;
  optional string eventsUrl = 8;
}
```

1.8. Links

1.8.1. List Links

List all links:

```
GET /api/links
```

List all links for the given Yamcs instance:

```
GET /api/links/:instance
```

Response

```
Status: 200 OK
```

```
{
  "link" : [ {
    "instance" : "simulator",
    "name" : "tm1",
    "type" : "HkDataHandler",
    "spec" : "",
    "stream" : "tm_realtime",
    "disabled" : false,
    "status" : "OK",
    "dataCount" : 34598,
    "detailedStatus" : "reading files from /storage/yamcs-
incoming/simulator/tm"
  } ]
}
```

Protobuf

Response

```
                                                                rest.proto
```

```
message ListLinkInfoResponse {
  repeated yamcsManagement.LinkInfo link = 1;
}
```

1.8.2. Get Link

Get data on a specific link for the given Yamcs instance:

```
GET /api/links/:instance/:name
```

Response

```
Status: 200 OK
```

```
{
  "instance" : "simulator",
  "name" : "tm1",
  "type" : "HkDataHandler",
  "spec" : "",
  "stream" : "tm_realtime",
  "disabled" : false,
  "status" : "OK",
  "dataCount" : 34598,
  "detailedStatus" : "reading files from /storage/yamcs-
incoming/simulator/tm"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                    yamcsManagement.proto
```

```
message LinkInfo {
  required string instance = 1;
  required string name = 2;
  optional string type = 3;
  optional string spec = 4;
  optional string stream = 5;
  optional bool disabled = 6;
  optional string status = 7;
  optional int64 dataCount = 8;
  optional string detailedStatus = 9;
}
```

1.8.3. Edit Link

Edit a link:

```
PATCH /api/links/:instance/:name
```

Parameters

Name	Type	Description
`state`	`string`	The state of the link. Either `enabled` or `disabled`.

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Enable a link:

```
{
  "state" : "enabled"
}
```

Disable a link:

```
{
  "state" : "disabled"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message EditLinkRequest {
  optional string state = 1;
}
```

1.9. Mission Database

1.9.1. Get MDB Detail

Get data on a the Mission Database for the given Yamcs instance:

```
GET /api/mdb/:instance
```

Response

```
Status: 200 OK
```

```
{
  "configName" : "landing",
  "name" : "",
  "spaceSystem" : [ {
    "name" : "YSS",
    "qualifiedName" : "/YSS",
    "version" : "1.2",
    "parameterCount" : 3,
    "containerCount" : 1,
    "commandCount" : 1,
    "sub" : [ {
      "name" : "SIMULATOR",
      "qualifiedName" : "/YSS/SIMULATOR",
      "version" : "1.0",
      "parameterCount" : 59,
      "containerCount" : 9,
      "commandCount" : 8,
      "history" : [ {
        "version" : "1.3",
        "date" : "21-June-2020",
        "message" : "modified this and that"
      } ]
    } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

yamcsManagement.proto

```
message MissionDatabase {
  required string configName = 1;
  required string name = 2;
  optional string version = 3;
  repeated SpaceSystemInfo spaceSystem = 4;
  optional string url = 5;
  optional string parametersUrl = 6;
  optional string containersUrl = 7;
  optional string commandsUrl = 8;
}
```

1.9.2. List Container Info

List all containers defined in the Mission Database for the given Yamcs instance:

```
GET /api/mdb/:instance/containers
```

List all containers defined under the given namespace:

```
GET /api/mdb/:instance/containers/:namespace
```

Parameters

Name	Type	Description
`recurse`	`bool`	If an XTCE `:namespace` is given, specifies whether to list containers of any nested sub systems. Default `no`.
`q`	`string`	The search keywords.

The `q` parameter supports searching on the namespace or name. For example:

```
/api/mdb/simulator/containers?q=yss+dhs&pretty
```

Response

```
Status: 200 OK
```

```
{
  "container" : [ {
    "name": "DHS",
    "qualifiedName" : "/YSS/SIMULATOR/DHS",
```

```
    "alias" : [ {
      "name" : "SIMULATOR_DHS",
      "namespace" : "MDB:OPS Name"
    }, {
      "name" : "DHS",
      "namespace" : "/YSS/SIMULATOR"
    } ],
    "maxInterval" : 1500,
    "baseContainer" : {
      "name": "ccsds-default",
      "qualifiedName" : "/YSS/ccsds-default",
      "url" :
"http://localhost:8090/api/mdb/simulator/containers/YSS/ccsds-default"
    },
    "restrictionCriteria" : [ {
      "parameter" : {
        "name": "ccsds-apid",
        "qualifiedName" : "/YSS/ccsds-apid",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/ccsds-apid"
      },
      "operator" : "EQUAL_TO",
      "value" : "1"
    }, {
      "parameter" : {
        "name": "packet-id",
        "qualifiedName" : "/YSS/packet-id",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/packet-id"
      },
      "operator" : "EQUAL_TO",
      "value" : "2"
    } ],
    "entry" : [ {
      "locationInBits" : 128,
      "referenceLocation" : "CONTAINER_START",
      "parameter" : {
        "name": "PrimBusVoltage1",
        "qualifiedName" : "/YSS/SIMULATOR/PrimBusVoltage1",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/SIMULATOR/PrimBusV

      }
    }, {
      "locationInBits" : 136,
      "referenceLocation" : "CONTAINER_START",
      "parameter" : {
        "name": "PrimBusCurrent1",
        "qualifiedName" : "/YSS/SIMULATOR/PrimBusCurrent1",
        "url" :
```

```
"http://localhost:8090/api/mdb/simulator/parameters/YSS/SIMULATOR/PrimBusC

      }
    } ],
    "url" :
"http://localhost:8090/api/mdb/simulator/containers/YSS/SIMULATOR/DHS"
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message ListContainerInfoResponse {
  repeated mdb.ContainerInfo container = 1;
}
```

1.9.3. Get Container Info

Return the data for the given container:

```
GET /api/mdb/:instance/containers/:namespace/:name
```

Response

```
Status: 200 OK
```

```
{
  "name": "DHS",
  "qualifiedName" : "/YSS/SIMULATOR/DHS",
  "alias" : [ {
    "name" : "SIMULATOR_DHS",
    "namespace" : "MDB:OPS Name"
  }, {
    "name" : "DHS",
    "namespace" : "/YSS/SIMULATOR"
  } ],
```

```
  "maxInterval" : 1500,
  "baseContainer" : {
    "name": "ccsds-default",
    "qualifiedName" : "/YSS/ccsds-default",
    "alias" : [ {
      "name" : "YSS_ccsds-default",
      "namespace" : "MDB:OPS Name"
    }, {
      "name" : "ccsds-default",
      "namespace" : "/YSS"
    } ],
    "entry" : [ {
      "locationInBits" : 5,
      "referenceLocation" : "CONTAINER_START",
      "parameter" : {
        "name": "ccsds-apid",
        "qualifiedName" : "/YSS/ccsds-apid",
        "alias" : [ {
          "name" : "YSS_ccsds-apid",
          "namespace" : "MDB:OPS Name"
        }, {
          "name" : "ccsds-apid",
          "namespace" : "/YSS"
        } ],
        "type" : {
          "engType" : "integer",
          "dataEncoding" : "IntegerDataEncoding(sizeInBits:11,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
        },
        "dataSource" : "TELEMETERED",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/ccsds-apid"
      }
    }, {
      "locationInBits" : 82,
      "referenceLocation" : "CONTAINER_START",
      "parameter" : {
        "name": "packet-type",
        "qualifiedName" : "/YSS/packet-type",
        "alias" : [ {
          "name" : "YSS_packet-type",
          "namespace" : "MDB:OPS Name"
        }, {
          "name" : "packet-type",
          "namespace" : "/YSS"
        } ],
        "type" : {
          "engType" : "integer",
          "dataEncoding" : "IntegerDataEncoding(sizeInBits:4,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
```

```
        },
        "dataSource" : "TELEMETERED",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/packet-type"
      }
    }, {
      "locationInBits" : 96,
      "referenceLocation" : "CONTAINER_START",
      "parameter" : {
        "name": "packet-id",
        "qualifiedName" : "/YSS/packet-id",
        "alias" : [ {
          "name" : "YSS_packet-id",
          "namespace" : "MDB:OPS Name"
        }, {
          "name" : "packet-id",
          "namespace" : "/YSS"
        } ],
        "type" : {
          "engType" : "integer",
          "dataEncoding" : "IntegerDataEncoding(sizeInBits:32,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
        },
        "dataSource" : "TELEMETERED",
        "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/packet-id"
      }
    } ],
    "url" : "http://localhost:8090/api/simulator/containers/YSS/ccsds-
default"
  },
  "restrictionCriteria" : [ {
    "parameter" : {
      "name": "ccsds-apid",
      "qualifiedName" : "/YSS/ccsds-apid",
      "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/ccsds-apid"
    },
    "operator" : "EQUAL_TO",
    "value" : "1"
  }, {
    "parameter" : {
      "name": "packet-id",
      "qualifiedName" : "/YSS/packet-id",
      "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/packet-id"
    },
    "operator" : "EQUAL_TO",
    "value" : "2"
  } ],
```

```
  "entry" : [ {
    "locationInBits" : 128,
    "referenceLocation" : "CONTAINER_START",
    "parameter" : {
      "name": "PrimBusVoltage1",
      "qualifiedName" : "/YSS/SIMULATOR/PrimBusVoltage1",
      "alias" : [ {
        "name" : "SIMULATOR_PrimBusVoltage1",
        "namespace" : "MDB:OPS Name"
      }, {
        "name" : "PrimBusVoltage1",
        "namespace" : "/YSS/SIMULATOR"
      } ],
      "type" : {
        "engType" : "integer",
        "dataEncoding" : "IntegerDataEncoding(sizeInBits:8,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
      },
      "dataSource" : "TELEMETERED",
      "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/SIMULATOR/PrimBusV

    }
  }, {
    "locationInBits" : 136,
    "referenceLocation" : "CONTAINER_START",
    "parameter" : {
      "name": "PrimBusCurrent1",
      "qualifiedName" : "/YSS/SIMULATOR/PrimBusCurrent1",
      "alias" : [ {
        "name" : "SIMULATOR_PrimBusCurrent1",
        "namespace" : "MDB:OPS Name"
      }, {
        "name" : "PrimBusCurrent1",
        "namespace" : "/YSS/SIMULATOR"
      } ],
      "type" : {
        "engType" : "integer",
        "dataEncoding" : "IntegerDataEncoding(sizeInBits:8,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
      },
      "dataSource" : "TELEMETERED",
      "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/SIMULATOR/PrimBusC

    }
  } ],
  "url" :
"http://localhost:8090/api/mdb/simulator/containers/YSS/SIMULATOR/DHS"
}
```

Protobuf

Response:

mdb.proto

```
message ContainerInfo {
  optional string name = 1;
  optional string qualifiedName = 2;
  optional string shortDescription = 3;
  optional string longDescription = 4;
  repeated yamcs.NamedObjectId alias = 5;
  optional int64 maxInterval = 6;
  optional int32 sizeInBits = 7;
  optional ContainerInfo baseContainer = 8;
  repeated ComparisonInfo restrictionCriteria = 9;
  repeated SequenceEntryInfo entry = 10;
  optional string url = 11;
}
```

1.9.4. List Parameter Info

List all parameters defined in the Mission Database for the given Yamcs instance:

```
GET /api/mdb/:instance/parameters
```

List all parameters defined under the given namespace:

```
GET /api/mdb/:instance/parameters/:namespace
```

Parameters

Name	Type	Description
`recurse`	`bool`	If an XTCE `:namespace` is given, specifies whether to list parameters of any nested sub systems. Default `no`.
`type`	`array of strings`	The parameter types to be included in the result. Valid types are `boolean`, `binary`, `enumeration`, `float`, `integer` or `string`. Both these notations are accepted: • `?type=float,integer` • `?type[]=float&type[]=integer` If unspecified, parameters of all types will be included.
`q`	`string`	The search keywords.

The `q` parameter supports searching on namespace or name. For example:

```
/api/mdb/simulator/parameters?q=ccsds+yss&pretty
```

Response

```
Status: 200 OK
```


```
{
  "parameter" : [ {
    "name": "ccsds-apid",
    "qualifiedName" : "/YSS/ccsds-apid",
    "alias" : [ {
      "name" : "YSS_ccsds-apid",
      "namespace" : "MDB:OPS Name"
    }, {
      "name" : "ccsds-apid",
      "namespace" : "/YSS"
    } ],
    "type" : {
      "engType" : "integer",
      "dataEncoding" : "IntegerDataEncoding(sizeInBits:11,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
    },
    "dataSource" : "TELEMETERED",
    "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/ccsds-apid"
  } ]
}
```


Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                                rest.proto
```

```
message ListParameterInfoResponse {
  repeated mdb.ParameterInfo parameter = 1;
}
```

1.9.5. Get Parameter Info

Return the data for the given parameter:

```
GET /api/mdb/:instance/parameters/:namespace/:name
```

Response

```
Status: 200 OK
```


```
{
  "name": "BatteryVoltage2",
  "qualifiedName" : "/YSS/SIMULATOR/BatteryVoltage2",
  "alias" : [ {
    "name" : "SIMULATOR_BatteryVoltage2",
    "namespace" : "MDB:OPS Name"
  }, {
    "name" : "BatteryVoltage2",
    "namespace" : "/YSS/SIMULATOR"
  } ],
  "type" : {
    "engType" : "integer",
    "dataEncoding" : "IntegerDataEncoding(sizeInBits:8,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
  },
  "dataSource" : "TELEMETERED",
  "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/SIMULATOR/BatteryV

}
```


Bulk

Combine multiple parameter queries in one and the same request using this address:

```
GET /api/mdb/:instance/parameters/bulk
```

Specify the parameter IDs in the request body:

```
{
  "id" : [ {
    "name": "YSS_ccsds-apid",
    "namespace": "MDB:OPS Name"
  }, {
    "name": "YSS_packet-type",
```

```
    "namespace": "MDB:OPS Name"
  } ]
}
```

POST requests are also allowed, because some HTTP clients do not support GET with a request body.

In the response the requested parameter ID is returned for every match. Example:

```
{
  "response" : [ {
    "id" : {
      "name" : "YSS_ccsds-apid",
      "namespace" : "MDB:OPS Name"
    },
    "parameter" : {
      "name": "ccsds-apid",
      "qualifiedName" : "/YSS/ccsds-apid",
      "aliases" : [ {
        "name" : "YSS_ccsds-apid",
        "namespace" : "MDB:OPS Name"
      }, {
        "name" : "ccsds-apid",
        "namespace" : "/YSS"
      } ],
      "type" : {
        "engType" : "integer",
        "dataEncoding" : "IntegerDataEncoding(sizeInBits:11,
encoding:unsigned, defaultCalibrator:null byteOrder:BIG_ENDIAN)"
      },
      "dataSource" : "TELEMETERED",
      "url" :
"http://localhost:8090/api/mdb/simulator/parameters/YSS/ccsds-apid"
    }
  } ]
}
```

Alternative Media Types

Protobuf

Use these HTTP headers:

```
Content-Type: application/protobuf
Accept: application/protobuf
```

Response is of type:

mdb.proto

```
message ParameterInfo {
  optional string name = 1;
  optional string qualifiedName = 2;
  optional string shortDescription = 3;
  optional string longDescription = 4;
  repeated yamcs.NamedObjectId alias = 5;
  optional ParameterTypeInfo type = 6;
  optional DataSourceType dataSource = 7;
  optional string url = 8;
}
```

Bulk request is of type:

rest.proto

```
message BulkGetParameterRequest {
  repeated yamcs.NamedObjectId id = 1;
}
```

Bulk response is of type:

rest.proto

```
message BulkGetParameterResponse {
  message GetParameterResponse {
    optional yamcs.NamedObjectId id = 1;
    optional mdb.ParameterInfo parameter = 2;
  }

  repeated GetParameterResponse response = 1;
}
```

1.9.6. List Command Info

List all commands defined in the Mission Database for the given Yamcs instance:

```
GET /api/mdb/:instance/commands
```

List all commands defined under the given namespace:

```
GET /api/mdb/:instance/commands/:namespace
```

Parameters

Name	Type	Description
`recurse`	`bool`	If an XTCE `:namespace` is given, specifies whether to list commands of any nested sub systems. Default `no`.
`q`	`string`	The search keywords.

The `q` parameter supports searching on namespace or name. For example:

```
/api/mdb/simulator/commands?q=volt+on&pretty
```

Response

```
Status: 200 OK
```


```
{
  "command" : [ {
    "name": "SWITCH_VOLTAGE_ON",
    "qualifiedName" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_ON",
    "alias" : [ {
      "name" : "SIMULATOR_SWITCH_VOLTAGE_ON",
      "namespace" : "MDB:OPS Name"
    }, {
      "name" : "SWITCH_VOLTAGE_ON",
      "namespace" : "/YSS/SIMULATOR"
    } ],
    "baseCommand" : {
      "name": "SIM_TC",
      "qualifiedName" : "/YSS/SIMULATOR/SIM_TC",
      "url" :
"http://localhost:8090/api/mdb/simulator/commands/YSS/SIMULATOR/SIM_TC"
    },
    "abstract" : false,
    "argument" : [ {
      "name" : "voltage_num",
      "description" : "voltage number to switch on",
      "type" : "integer",
      "unitSet" : [ {
        "unit" : "V"
      } ]
    } ],
    "argumentAssignment" : [ {
      "name" : "packet-id",
      "value" : "1"
    } ],
    "url" :
"http://localhost:8090/api/mdb/simulator/commands/YSS/SIMULATOR/SWITCH_VOI
```

```
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message ListCommandInfoResponse {
  repeated mdb.CommandInfo command = 1;
}
```

1.9.7. Get Command Info

Return the data for the given command:

```
GET /api/mdb/:instance/commands/:namespace/:name
```

Response

```
Status: 200 OK
```

```
{
  "name": "SWITCH_VOLTAGE_ON",
  "qualifiedName" : "/YSS/SIMULATOR/SWITCH_VOLTAGE_ON",
  "alias" : [ {
    "name" : "SIMULATOR_SWITCH_VOLTAGE_ON",
    "namespace" : "MDB:OPS Name"
  }, {
    "name" : "SWITCH_VOLTAGE_ON",
    "namespace" : "/YSS/SIMULATOR"
  } ],
  "baseCommand" : {
    "name": "SIM_TC",
    "qualifiedName" : "/YSS/SIMULATOR/SIM_TC",
    "alias" : [ {
      "name" : "SIMULATOR_SIM_TC",
      "namespace" : "MDB:OPS Name"
```

```
    }, {
      "name" : "SIM_TC",
      "namespace" : "/YSS/SIMULATOR"
    } ],
    "baseCommand" : {
      "name": "ccsds-tc",
      "qualifiedName" : "/YSS/ccsds-tc",
      "alias" : [ {
        "name" : "YSS_ccsds-tc",
        "namespace" : "MDB:OPS Name"
      }, {
        "name" : "ccsds-tc",
        "namespace" : "/YSS"
      } ],
      "abstract" : true,
      "argument" : [ {
        "name" : "ccsds-apid",
        "type" : "integer"
      }, {
        "name" : "timeId",
        "type" : "integer"
      }, {
        "name" : "checksumIndicator",
        "type" : "integer",
        "initialValue" : "1"
      }, {
        "name" : "packet-type",
        "type" : "integer"
      }, {
        "name" : "packet-id",
        "type" : "integer"
      } ],
      "url" :
"http://localhost:8090/api/mdb/simulator/commands/YSS/ccsds-tc"
    },
    "abstract" : true,
    "argumentAssignment" : [ {
      "name" : "ccsds-apid",
      "value" : "100"
    }, {
      "name" : "timeId",
      "value" : "1"
    }, {
      "name" : "packet-type",
      "value" : "10"
    } ],
    "url" :
"http://localhost:8090/api/mdb/simulator/commands/YSS/SIMULATOR/SIM_TC"
  },
  "abstract" : false,
```

```
  "argument" : [ {
    "name" : "voltage_num",
    "description" : "voltage number to switch on",
    "type" : "integer",
    "unitSet" : [ {
      "unit" : "V"
    } ]
  } ],
  "argumentAssignment" : [ {
    "name" : "packet-id",
    "value" : "1"
  } ],
  "url" :
"http://localhost:8090/api/mdb/simulator/commands/YSS/SIMULATOR/SWITCH_VOI

}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response if of type:

mdb.proto

```
message CommandInfo {
  optional string name = 1;
  optional string qualifiedName = 2;
  optional string shortDescription = 3;
  optional string longDescription = 4;
  repeated yamcs.NamedObjectId alias = 5;
  optional CommandInfo baseCommand = 6;
  optional bool abstract = 7;
  repeated ArgumentInfo argument = 8;
  repeated ArgumentAssignmentInfo argumentAssignment = 9;
  optional SignificanceInfo significance = 10;
  repeated TransmissionConstraintInfo constraint = 11;
  optional string url = 12;
}
```

1.10. Packets

1.10.1. List Packets

List the history of packets:

```
GET /api/archive/:instance/packets
```

List the packets for the specified generation time

```
GET /api/archive/:instance/packets/:gentime
```

The `:gentime` must be in ISO 8601 format. E.g. 2015-10-20T06:47:02.000

Parameters

Name	Type	Description
`name`	`array of strings`	The archived name of the packets. Both these notations are accepted: • `?name=/YSS/SIMULATOR/DHS,/YSS/SIMULATOR/Power` • `?name[]=/YSS/SIMULATOR/DHS&name[]=/YSS/SIMULATOR/Power` Names must match exactly.
`start`	`string`	Filter the lower bound of the packet's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the packet's generation time. Specify a date string in ISO 8601 format
`pos`	`integer`	The zero-based row number at which to start outputting results. Default: `0`
`limit`	`integer`	The maximum number of returned records per page. Choose this value too high and you risk hitting the maximum response size limit enforced by the server. Default: `100`
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `desc`

The `pos` and `limit` allow for pagination. Keep in mind that in-between two requests extra data may have been recorded, causing a shift of the results. This generic stateless operation does not provide a reliable mechanism against that, so address it by overlapping your `pos` parameter with rows of the previous query. In this example we overlap by 4:

```
?pos=0&limit=50&order=desc
?pos=45&limit=50&order=desc
```

An alternative is to download the packets instead.

Response

```
Status: 200 OK
```

```
{
  "packet" : [ {
    "receptionTime" : 1447625895283,
    "packet" :
"CAEAAABFQ3PHAzxFAAAAIUMCgADCcgxKQlS0OUYh0ADDClmaP9AnUj9yMP1DV9wpQ1fcKT85i

    "generationTime" : 1447625878234,
    "sequenceNumber" : 134283264
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                                    rest.proto
```

```
message ListPacketsResponse {
  repeated yamcs.TmPacketData packet = 1;
}
```

1.10.2. Get Packet

Get a single packet:

```
GET /api/archive/:instance/packets/:gentime/:seqnum
```

The gentime must be an exact match of the packet's generation time in ISO 8601 format. Example:

```
/api/archive/simulator/packets/2015-10-20T06:47:02.000/808465467
```

Response

```
Status: 200 OK
```

```
{
  "receptionTime" : 1447880235182,
  "packet" :
"MDA0O1NVOzE7MDsyOzI7MjswOy0yMi43MTMwOysyNC44NTQwOzIwMTUvMTAvMjA7MDY6NDc6M

  "generationTime" : 1445323658000,
  "sequenceNumber" : 808465467
}
```

The binary inside the packet variable is encoded in Base64 format

Protobuf

Response

yamcs.proto

```
message TmPacketData {
  required int64 receptionTime = 1;
  required bytes packet = 2;
  optional int64 generationTime = 3;
  optional int32 sequenceNumber = 4;
  optional NamedObjectId id = 5;
}
```

1.10.3. Download Packets

Download archived packets:

```
GET /api/archive/:instance/downloads/packets
```

This operation will possibly download a very large file. If you worry about size for your application, check out the support for paged packet retrievalsinstead.

Parameters

Name	Type	Description
`name`	`array of strings`	The archived name of the packets. Both these notations are accepted: • `?name=/YSS/SIMULATOR/DHS,/YSS/SIMULATOR/Power` • `?name[]=/YSS/SIMULATOR/DHS&name[]=/YSS/SIMULATOR/Power` Names must match exactly.
`start`	`string`	Filter the lower bound of the packet's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the packet's generation time. Specify a date string in ISO 8601 format
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `asc`

Response

The response will be a stream of self-standing JSON messages.

Alternative Media Types

Raw binary

Use HTTP header:

```
Accept: application/octet-stream
```

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response is a stream of self-standing `VarInt` delimited messages of type:

yamcs.proto

```
message TmPacketData {
  required int64 receptionTime = 1;
  required bytes packet = 2;
  optional int64 generationTime = 3;
  optional int32 sequenceNumber = 4;
  optional NamedObjectId id = 5;
}
```

1.11. Parameter Values

1.11.1. Get Parameter Value

Retrieves the current value of the specified parameter.

```
GET /api/processors/:instance/:processor/parameters/:namespace/:name
```

Parameters

Name	Type	Description
`fromCache`	`bool`	Whether the latest cached value may be returned. Default: `yes`
`timeout`	`number`	Time in milliseconds to wait on a value (only considered if `fromCache=no`). When the timeout is met, the call will return with no or partial data. Default: `10000`

Response

```
Status: 200 OK
```


```
{
  "id" : {
    "name" : "BatteryVoltage2",
    "namespace" : "/YSS/SIMULATOR"
  },
  "rawValue" : {
    "type" : "UINT32",
    "uint32Value" : 184
  },
  "engValue" : {
    "type" : "UINT32",
    "uint32Value" : 184
  },
  "acquisitionTime" : 1445940421269,
  "generationTime" : 1445940404207,
  "acquisitionStatus" : "ACQUIRED",
  "processingStatus" : true,
  "acquisitionTimeUTC" : "2015-10-27T10:06:25.269",
  "generationTimeUTC" : "2015-10-27T10:06:08.207",
  "expirationTime" : 1445940428269,
  "expirationTimeUTC" : "2015-10-27T10:06:32.269"
}
```


Multi-get

Get the values of multiple parameters in one and the same request using this address:

```
GET /api/processors/:instance/:processor/parameters/mget
```

The same options as in the query string can be specified in the request body. Parameter IDs are added to a list:

```
{
  "fromCache" : false,
  "timeout" : 15000,
  "id" : [ {
    "name": "YSS_ccsds-apid",
    "namespace": "MDB:OPS Name"
  }, {
    "name": "/YSS/SIMULATOR/BatteryVoltage2"
  } ]
}
```

POST requests are also allowed, because some HTTP clients do not support GET with a request body.

The response is a list of parameter values:

```
{
  "value" : [ {
    "id" : {
      "name" : "/YSS/SIMULATOR/BatteryVoltage2"
    },
    "rawValue" : {
      "type" : "UINT32",
      "uint32Value" : 206
    },
    "engValue" : {
      "type" : "UINT32",
      "uint32Value" : 206
    },
    "acquisitionTime" : 1445944393466,
    "generationTime" : 1445944376410,
    "acquisitionStatus" : "ACQUIRED",
    "processingStatus" : true,
    "acquisitionTimeUTC" : "2015-10-27T11:12:37.466",
    "generationTimeUTC" : "2015-10-27T11:12:20.410",
    "expirationTime" : 1445944400466,
    "expirationTimeUTC" : "2015-10-27T11:12:44.466"
  } ]
}
```

Protobuf

Response

pvalue.proto

```
message ParameterValue {
  optional yamcs.NamedObjectId id = 1;
  optional yamcs.Value rawValue = 2;
  optional yamcs.Value engValue = 3;
  optional int64 acquisitionTime = 4;
  optional int64 generationTime = 5;
  optional AcquisitionStatus acquisitionStatus = 6;
  optional bool processingStatus = 7;
  optional MonitoringResult monitoringResult = 8;

  optional string acquisitionTimeUTC = 11;
  optional string generationTimeUTC = 12;

  optional int64 expirationTime = 23;
  optional string expirationTimeUTC = 24;

  repeated mdb.AlarmRange alarmRange = 25;
}
```

Bulk Request

rest.proto

```
message BulkGetParameterValueRequest {
  repeated yamcs.NamedObjectId id = 1;
  optional bool fromCache = 2;
  optional uint64 timeout = 3; //if not fromCache, wait this time (in
milliseconds) to receive the parameter
}
```

Bulk Response

rest.proto

```
message BulkGetParameterValueResponse {
  repeated pvalue.ParameterValue value = 1;
}
```

1.11.2. Set Parameter Value

Parameters are usually read-only. But software parameters can be updated:

```
PUT /api/processors/:instance/:processor/parameters/:namespace/:name
```

Example

```
{
  "type" : "UINT32",
  "uint32Value" : 123
}
```

Multi-set

To update multiple parameters at once, send a request to this address:

```
POST /api/processors/:instance/:processor/parameters/mset
```

```
{
  "request" : [ {
    "id": {
      "name": "/YSS/SIMULATOR/AllowCriticalTC1"
    },
    "value": {
      "type" : "BOOLEAN",
      "booleanValue" : true
    }
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Content-Type: application/protobuf
```

Request is of type:

yamcs.proto

```
message Value {
  enum Type {
    FLOAT = 0;
    DOUBLE = 1;
```

```
    UINT32 = 2;
    SINT32 = 3;
    BINARY = 4;
    STRING = 5;
    TIMESTAMP = 6;
    UINT64 = 7;
    SINT64 = 8;
    BOOLEAN = 9;
  }
  required Type type = 1;
  optional float floatValue = 2;
  optional double doubleValue = 3;
  optional sint32 sint32Value = 4;
  optional uint32 uint32Value = 5;
  optional bytes binaryValue = 6;
  optional string stringValue = 7;
  optional int64 timestampValue = 8;
  optional uint64 uint64Value = 9;
  optional sint64 sint64Value = 10;
  optional bool booleanValue = 11;
}
```

Bulk request is of type:

rest.proto

```
message BulkSetParameterValueRequest {
  message SetParameterValueRequest {
    optional yamcs.NamedObjectId id = 1;
    optional yamcs.Value value = 2;
  }
  repeated SetParameterValueRequest request = 1;
}
```

1.11.3. Get Parameter Samples

Sample the history of values for the specified parameter:

```
GET /api/archive/:instance/parameters/:namespace/:name/samples
```

Parameters

Name	Type	Description
`start`	`string`	Filter the lower bound of the parameter's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the parameter's generation time. Specify a date string in ISO 8601 format
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `asc`

Response

```
Status: 200 OK
```

```
{
  "sample" : [ {
    "time" : "2015-11-11T09:11:37.626",
    "avg" : 169.41836734693865,
    "min" : 103.0,
    "max" : 237.0,
    "n" : 98
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                                pvalue.proto
```

```
message TimeSeries {
  message Sample {
    optional string time = 1;
    optional double avg = 2;
    optional double min = 3;
    optional double max = 4;
    optional int32 n = 5;
  }

  repeated Sample sample = 1;
}
```

1.11.4. List Parameter Data

List the history of values for the specified parameter:

```
GET /api/archive/:instance/parameters/:namespace/:name
```

Parameters

Name	Type	Description
`start`	`string`	Filter the lower bound of the parameter's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the parameter's generation time. Specify a date string in ISO 8601 format
`norepeat`	`bool`	Whether to filter out consecutive identical values. Default `no`.
`pos`	`integer`	The zero-based row number at which to start outputting results. Default: `0`
`limit`	`integer`	The maximum number of returned records per page. Choose this value too high and you risk hitting the maximum response size limit enforced by the server. Default: `100`
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `desc`

The `pos` and `limit` allow for pagination. Keep in mind that in-between two requests extra data may have been recorded, causing a shift of the results. This generic stateless operation does not provide a reliable mechanism against that, so address it by overlapping your `pos` parameter with rows of the previous query. In this example we overlap by 4:

```
?pos=0&limit=50&order=desc
?pos=45&limit=50&order=desc
```

Response

```
Status: 200 OK
```

```
{
  "parameter" : [ {
    "id" : {
      "name" : "BatteryVoltage2",
      "namespace" : "/YSS/SIMULATOR"
    },
    "rawValue" : {
      "type" : "UINT32",
      "uint32Value" : 144
    },
    "engValue" : {
      "type" : "UINT32",
      "uint32Value" : 144
```

```
    },
    "acquisitionTime" : 1447417449218,
    "generationTime" : 1447417432121,
    "acquisitionStatus" : "ACQUIRED",
    "processingStatus" : true,
    "monitoringResult" : "IN_LIMITS",
    "acquisitionTimeUTC" : "2015-11-13T12:23:33.218",
    "generationTimeUTC" : "2015-11-13T12:23:16.121",
    "expirationTime" : 1447417456218,
    "expirationTimeUTC" : "2015-11-13T12:23:40.218",
    "alarmRange" : [ {
      "level" : "WATCH",
      "minInclusive" : 50.0
    }, {
      "level" : "WARNING",
      "minInclusive" : 40.0
    }, {
      "level" : "DISTRESS",
      "minInclusive" : 30.0
    }, {
      "level" : "CRITICAL",
      "minInclusive" : 20.0
    }, {
      "level" : "SEVERE",
      "minInclusive" : 10.0
    } ]
  } ]
}
```

Alternative Media Types

CSV

Use HTTP header:

```
Accept: text/csv
```

Or, add this query parameter: `format=csv`.

```
Status: 200 OK
Content-Type: text/csv
```

```
Time    BatteryVoltage2
2015-11-13T12:21:55.199 157
2015-11-13T12:21:48.972 158
2015-11-13T12:21:42.750 159
```

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response if of type:

pvalue.proto

```
message ParameterData {
  repeated ParameterValue parameter = 1;
}
```

1.11.5. Download Parameter Data

Download archived parameters:

```
GET /api/archive/:instance/downloads/parameters/:namespace/:name
```

This operation will possibly download a very large file. If you worry about size for your application, check out the support for paged parameter retrievals instead.

Parameters

Name	Type	Description
`start`	`string`	Filter the lower bound of the parameter's generation time. Specify a date string in ISO 8601 format
`stop`	`string`	Filter the upper bound of the parameter's generation time. Specify a date string in ISO 8601 format
`norepeat`	`bool`	Whether to filter out consecutive identical values. Default `no`.
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `asc`

Response

The response will be a stream of individual parameters.

Alternative Media Types

CSV

Use HTTP header:

```
Accept: application/protobuf
```

Or add the query parameter `format=csv`.

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

The response if a stream of individual Protobuf messages delimited by a `VarInt`.

1.12. Processors

1.12.1. List Processors

List all processors, across all Yamcs instances:

```
GET /api/processors
```

List all processors for the given Yamcs instance:

```
GET /api/processors/:instance
```

Parameters

Name	Type	Description
`type`	`string`	Indicates the type of the processors to return. Can be either `replay`, `realtime` or `all`. Default: `all`

Response

```
Status: 200 OK
```

```
{
  "processor" : [ {
    "instance" : "simulator",
    "name" : "realtime",
    "type" : "realtime",
    "creator" : "system",
    "hasCommanding" : true,
    "state" : "RUNNING",
    "url" : "http://localhost:8090/api/processors/simulator/realtime",
    "clientsUrl":
"http://localhost:8090/api/processors/simulator/realtime/clients",
    "parametersUrl" :
"http://localhost:8090/api/processors/simulator/realtime/parameters{/names
{/name}",
    "commandsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/commands{/namespa
{/name}",
    "commandQueuesUrl" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues{/name}"

  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message ListProcessorsResponse {
  repeated yamcsManagement.ProcessorInfo processor = 1;
}
```

1.12.2. Get Processor Info

Get info on a specific Yamcs processor:

```
GET /api/processors/:instance/:name
```

Response

```
Status: 200 OK
```

```
{
  "instance" : "simulator",
  "name" : "realtime",
  "type" : "realtime",
  "creator" : "system",
  "hasCommanding" : true,
  "state" : "RUNNING",
  "url" : "http://localhost:8090/api/processors/simulator/realtime",
  "clientsUrl":
"http://localhost:8090/api/processors/simulator/realtime/clients",
  "parametersUrl" :
"http://localhost:8090/api/processors/simulator/realtime/parameters{/names
{/name}",
  "commandsUrl" :
"http://localhost:8090/api/processors/simulator/realtime/commands{/namespa
{/name}",
  "commandQueuesUrl" :
"http://localhost:8090/api/processors/simulator/realtime/cqueues{/name}"

}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

yamcsManagement.proto

```
message ProcessorInfo {
  optional string instance = 1;
  optional string name = 2;
  optional string type = 3;
  optional string spec = 4;
  optional string creator = 5;
  optional bool hasCommanding = 6;
  optional ServiceState state = 7;
  optional yamcs.ReplayRequest replayRequest = 8;
  optional yamcs.ReplayStatus.ReplayState replayState = 9;
  optional string url = 10;
  optional string clientsUrl = 11;
  optional string parametersUrl = 12;
  optional string commandsUrl = 13;
  optional string commandQueuesUrl = 14;
}
```

1.12.3. Create Processor

Create a replay processor for the given Yamcs instance:

```
POST /api/processors/:instance
```

Replay processors allow for synchronized playback of different types of archived data.

Parameters

Name	Type	Description
`name`	`string`	**Required.** The name of the processor. Must be unique for the Yamcs instance.
`start`	`string`	**Required.** The time at which the replay should start. Must be a date string in ISO 8601 format.
`stop`	`string`	The time at which the replay should stop. Must be a date string in ISO 8601 format. If unspecified, the replay will keep going as long as there is remaining data.
`loop`	`bool`	Whether the processing should restart at the end of the replay. Default: `no`
`speed`	`string`	The speed of the processor. One of: • `afap` • a speed factor relative to the original speed. Example: `2x` • a fixed delay value in milliseconds. Example: `2000` Default: `1x`
`clientId`	`array of integers`	The client IDs that should be connected to this processor.
`paraname`	`array of strings`	Name patterns of parameters that are included in the replay. Patterns are matched on the qualified names and support wildcard expansion. If the pattern matches the name of a space system, all parameters directly within that system are included.
`ppgroup`	`array of strings`	Exact names of the groups of processed parameters to include in the replay. **Partial wildcard matching is not currently supported.**
`packetname`	`array of strings`	Exact qualified names of the packets to include in the replay. Specify `*` to include all packets. **Partial wildcard matching is not currently supported.**
`cmdhist`	`bool`	Whether or not to replay Command History. Default: `no`

The criteria `paraname`, `ppgroup`, `packetname` and `cmdhist` are additive. At least one of them should be filled in, or there will be nothing to replay.

Example

Replay everything except PPs as of January 1st 2015 at 4.5x the original speed, and add client 12 to the replay:

```
{
  "name" : "A sample processor",
  "start" : "2015-01-01T00:00:00.000",
  "clientId" : [ 12 ],
  "speed": "4.5x",
  "paraname": [ "*" ],
  "packetname": [ "*" ],
```

```
  "cmdhist": true
}
```

Replay parameters directly under `/GC` or anywhere under `/YSS` at a fixed delay of 1 second, and switch client 12 to this replay processor:

```
{
  "name" : "A lighter processor",
  "start" : "2015-01-01T00:00:00.000",
  "clientId" : [ 12 ],
  "speed": "1000",
  "paraname": [ "/YSS/*", "/GC" ]
}
```

Notice how the speed value must always be encoded as a string to keep our parser happy.

Protobuf

Request

rest.proto

```
message CreateProcessorRequest {
  optional string name = 1;
  optional string start = 2;
  optional string stop = 3;
  optional bool loop = 4;
  optional string speed = 5;
  repeated int32 clientId = 6;
  repeated string paraname = 7;
  repeated string ppgroup = 8;
  repeated string packetname = 9;
  optional bool cmdhist = 10;
  optional bool persistent = 11;
}
```

1.12.4. Edit Processor

Edit a processor:

```
PATCH /api/processors/:instance/:name
```

Parameters

Name	Type	Description
`state`	`string`	The state this processor should be updated to. Either `PAUSED` or `RUNNING`.
`seek`	`string`	The time where the processing needs to jump towards. Must be a date string in ISO 8601 format.
`speed`	`string`	The speed of the processor. One of: • `afap` • a speed factor relative to the original speed. Example: `2x` • a fixed delay value in milliseconds. Example: `2000`

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Pause the processor:

```
{
  "state" : "PAUSED"
}
```

Resume the processor, and set speed to 2.5x:

```
{
  "state" : "RUNNING",
  "speed" : "2.5x"
}
```

Make processor move according to original speed:

```
{
  "speed" : "1x"
}
```

Notice that the speed value must be encoded as a string to keep our parser happy.

Protobuf

Request

rest.proto

```
message EditProcessorRequest {
  optional string state = 1;
```

```
  optional string seek = 2;
  optional string speed = 3;
}
```

1.13. Streams

1.13.1. List Streams

List all streams for the given instance:

```
GET /api/archive/:instance/streams
```

This is low-level API for those cases where access to the internal streams of Yamcs is wanted. It is recommended to use other API operations for any of the default built-in streams.

Example

```
Status: 200 OK
```

```
{
  "stream" : [ {
    "name" : "tm_realtime",
    "column" : [ {
      "name" : "gentime",
      "type" : "TIMESTAMP"
    }, {
      "name" : "seqNum",
      "type" : "INT"
    }, {
      "name" : "rectime",
      "type" : "TIMESTAMP"
    }, {
      "name" : "packet",
      "type" : "BINARY"
    } ]
  } ]
}
```

Note that this will only list the fixed columns of the stream. Tuples may always have extra columns.

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message ListStreamsResponse {
  repeated archive.StreamInfo stream = 1;
}
```

1.13.2. Get Stream Info

Get info on a Yamcs stream:

```
GET /api/archive/:instance/streams/:name
```

This is low-level API for those cases where access to an internal stream of Yamcs is wanted. It is recommended to use other API operations for any of the default built-in streams.

Response

```
Status: 200 OK
```

```
{
  "name" : "tm_realtime",
  "column" : [ {
    "name" : "gentime",
    "type" : "TIMESTAMP"
  }, {
    "name" : "seqNum",
    "type" : "INT"
  }, {
    "name" : "rectime",
    "type" : "TIMESTAMP"
  }, {
    "name" : "packet",
    "type" : "BINARY"
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response if of type:

archive.proto

```
message StreamInfo {
  optional string name = 1;
  repeated ColumnInfo column = 2;
}
```

1.14. Tables

1.14.1. List Tables

List all tables for the given instance:

```
GET /api/archive/:instance/tables
```

This is low-level API for those cases where access to the internal key/value tables of Yamcs is wanted. It is recommended to use other API operations for any of the default built-in tables.

Example

```
Status: 200 OK
```

```
{
  "table" : [ {
    "name" : "tm",
    "keyColumn" : [ {
      "name" : "gentime",
      "type" : "TIMESTAMP"
    }, {
      "name" : "seqNum",
      "type" : "INT"
    } ],
    "valueColumn" : [ {
      "name" : "rectime",
      "type" : "TIMESTAMP"
    }, {
      "name" : "packet",
      "type" : "BINARY"
    }, {
      "name" : "pname",
      "type" : "ENUM"
    } ]
  } ]
}
```

Note that this will only list the fixed columns of the table. Tuples may always add extra value columns.

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message ListTablesResponse {
  repeated archive.TableInfo table = 1;
}
```

1.14.2. Get Table Info

Get info on a Yamcs table:

```
GET /api/archive/:instance/tables/:name
```

This is low-level API for those cases where access to an internal key/value table of Yamcs is wanted. It is recommended to use other API operations for any of the default built-in tables.

Response

```
Status: 200 OK
```

```
{
  "name" : "tm",
  "keyColumn" : [ {
    "name" : "gentime",
    "type" : "TIMESTAMP"
  }, {
    "name" : "seqNum",
    "type" : "INT"
  } ],
  "valueColumn" : [ {
    "name" : "rectime",
    "type" : "TIMESTAMP"
  }, {
    "name" : "packet",
    "type" : "BINARY"
  }, {
    "name" : "pname",
    "type" : "ENUM"
  } ]
}
```

Protobuf

Response

archive.proto

```
message TableInfo {
  optional string name = 1;
  repeated ColumnInfo keyColumn = 2;
  repeated ColumnInfo valueColumn = 3;
}
```

1.14.3. List Table Data

List the most recent data of a Yamcs table:

```
GET /api/archive/:instance/tables/:table/data
```

This is low-level API for those cases where access to an internal key/value table of Yamcs is wanted. It is recommended to use other API operations for any of the default built-in tables.

Parameters

Name	Type	Description
`cols`	`array of strings`	The columns to be included in the result. Both these notations are accepted: • `?cols=rectime,gentime,pname` • `?cols[]=rectime&cols[]=gentime&cols[]=pname` If unspecified, all table and/or additional tuple columns will be included.
`start`	`integer`	The zero-based row number at which to start outputting results. Default: `0`
`limit`	`integer`	The maximum number of returned records per page. Choose this value too high and you risk hitting the maximum response size limit enforced by the server. Default: `100`
`order`	`string`	The direction of the sort. Sorting is always done on the key of the table. Can be either `asc` or `desc`. Default: `desc`

The `start` and `limit` allow for pagination. Keep in mind that in-between two requests extra data may have been added to the table, causing a shift of the results. This generic stateless operation does not provide a reliable mechanism against that, so address it by overlapping your `start` parameter with rows of the previous query. In this example we overlap by 4:

```
?start=0&limit=50&order=desc
?start=45&limit=50&order=desc
```

Response

```
Status: 200 OK
```

```
{
  "record" : [ {
    "column" : [ {
      "name" : "gentime",
      "value" : {
        "type" : "TIMESTAMP",
        "timestampValue" : 1446650363464
      }
    }, {
      "name" : "pname",
      "value" : {
        "type" : "STRING",
        "stringValue" : "/YSS/SIMULATOR/FlightData"
      }
    } ]
  }, {
    "column" : [ {
      "name" : "gentime",
      "value" : {
        "type" : "TIMESTAMP",
        "timestampValue" : 1446650363667
      }
    }, {
      "name" : "pname",
      "value" : {
        "type" : "STRING",
        "stringValue" : "/YSS/SIMULATOR/FlightData"
      }
    } ]
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                    archive.proto
```

```
message TableData {
```

```
  message TableRecord {
    repeated ColumnData column = 1;
  }
  repeated TableRecord record = 1;
}
```

1.14.4. Download Table Data

Download archived table data:

```
GET /api/archive/:instance/downloads/tables/:table
```

This operation will possibly download a very large file. If you worry about size for your application, check out the support for paged table data instead.

Parameters

Name	Type	Description
`cols`	`array of strings`	The columns to be included in the result. Both these notations are accepted: • `?cols=rectime,gentime,pname` • `?cols[]=rectime&cols[]=gentime&cols[]=pname` If unspecified, all table and/or additional tuple columns will be included.
`order`	`string`	The order of the returned results. Can be either `asc` or `desc`. Default: `asc`

Response

The response will be a stream of individual table records. When using Protobuf, every table record is delimited by its byte size.

1.15. Tags

1.15.1. List Tags

List all tags for the given instance:

```
GET /api/archive/:instance/tags
```

Example

```
Status: 200 OK
```

```
{
  "tag" : [ {
    "id" : 1,
    "name" : "My annotation",
    "start" : 1449128432000,
    "stop" : 1449174255000,
    "description" : "blabla",
    "color" : "#ffc800"
  } ]
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

```
                                                                rest.proto
```

```
message ListTagsResponse {
  repeated yamcs.ArchiveTag tag = 1;
}
```

1.15.2. Get Tag

Get info on a specific tag for the given archive instance:

```
GET /api/archive/:instance/tags/:start/:id
```

Response

```
Status: 200 OK
```

```
{
  "id" : 1,
  "name" : "My annotation",
  "start" : 1449128432000,
  "stop" : 1449174255000,
  "description" : "blabla",
  "color" : "#ffc800"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

yamcs.proto

```
message ArchiveTag {
  optional int32 id = 1;
  required string name = 2;
  optional int64 start = 3;
  optional int64 stop = 4;
  optional string description = 5;
  optional string color = 6;
}
```

1.15.3. Create Tag

Create a tag for the given archive instance:

```
POST /api/archive/:instance/tags
```

Parameters

Name	Type	Description
`name`	`string`	**Required.** The name of the tag.
`description`	`string`	The description of the tag.
`start`	`string`	The start time of the tag. Default is unbounded.
`stop`	`string`	The stop time of the tag. Default is unbounded.
`color`	`string`	The color of the tag. Must be an RGB hex color, e.g. `#ff0000`

Example

Create a red tag covering January 1st 2015, onwards:

```
{
  "name" : "My archive annotation",
  "start" : "2015-01-01T00:00:00.000",
  "color" : "#ff0000"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Content-Type: application/protobuf
```

Request is of type:

rest.proto

```
message CreateTagRequest {
  optional string name = 1;
  optional string start = 2;
  optional string stop = 3;
  optional string description = 4;
  optional string color = 5;
}
```

1.15.4. Edit Tag

Edit a tag:

```
PATCH /api/archive/:instance/tags/:start/:id
```

Parameters

Name	Type	Description
`name`	`string`	The name of the tag.
`description`	`string`	The description of the tag.
`start`	`string`	The start time of the tag. Must be a date string in ISO 8601 format. Set to empty to indicate unbounded.
`stop`	`string`	The stop time of the tag. Must be a date string in ISO 8601 format. Set to empty to indicate unbounded.
`color`	`string`	The color of the tag. Must be an RGB hex color, e.g. `#ff0000`

The same parameters can also be specified in the request body. In case both query string parameters and body parameters are specified, they are merged with priority being given to query string parameters.

Example

Change the color, and the description:

```
{
  "color" : "#00ff00",
  "description": "a sample description"
}
```

Alternative Media Types

Protobuf

Use HTTP header:

```
Accept: application/protobuf
```

Response is of type:

rest.proto

```
message EditTagRequest {
  optional string name = 1;
  optional string start = 2;
  optional string stop = 3;
  optional string description = 4;
  optional string color = 5;
}
```

1.16. User

Get information on the authenticated user:

```
GET /api/user
```

Response

```
Status: 200 OK
```

```
{
  "login" : "anonymous",
  "roles" : [ "admin" ],
  "tmParaPrivileges" : [ ".*" ],
  "tmParaSetPrivileges" : [ ".*" ],
  "tmPacketPrivileges" : [ ".*" ],
  "tcPrivileges" : [ ".*" ],
  "systemPrivileges" : [ "MayControlYProcessor",
"MayModifyCommandHistory", "MayControlCommandQueue",
"MayCommandPayload", "MayGetMissionDatabase", "MayControlArchiving" ]
}
```

Protobuf

Response

```
                                                          yamcsManagement.proto
```

```
message UserInfo {
  optional string login = 1;
  repeated ClientInfo clientInfo = 2;
  repeated string roles = 3;
  repeated string tmParaPrivileges = 4;
  repeated string tmParaSetPrivileges = 5;
  repeated string tmPacketPrivileges = 6;
  repeated string tcPrivileges = 7;
  repeated string systemPrivileges = 8;
}
```

Chapter 2. WebSocket API

Yamcs provides a WebSocket API for data subscriptions. A typical use case would be a display tool subscribing to parameter updates.

All operations support binary WebSocket frames encoded using Google Protocol Buffers, as well as textual WebSocket frames encoded in JSON. The server distinguishes between Protobuf and JSON based on the type of the first received client frame. If it is a binary websocket frame, all further communication will default to Protobuf. If it is a textual websocket frame, all further communication will default to JSON.

Wrapper

WebSocket calls should be directed to a URL of the form:

```
http://localhost:8090/:instance/_websocket
```

Replace `:instance` with your Yamcs instance name. The frame must contain a text array like so:

```
[x,y,z,{"<request-type>":"<request>"}]
```

Where:

`x`	the version of the protocol (currently fixed at 1)
`y`	the message type. One of: • `1` Request • `2` Reply • `3` Exception • `4` Data
`z`	a sequence counter. Enables clients to couple a response with the original request

The `request-type` and `request` criteria vary for every type of resource, and are each time indicated in further pages.

Java Client

WebSockets have good language support and are supported by all major browsers. If you are developing a Java application integrated with Yamcs, we recommend using the LGPL Java WebSocket client distributed as part of the `yamcs-api` jar.

2.1. Parameter Updates

Subscribe to parameter updates:

```
[ 1, 1, :seq, {
    "parameter": "subscribe",
    "data": {
        "list": [
            { "namespace": ":namespace", "name": ":name" }
        ]
    }
} ]
```

Subscribing is an additive operation, where new parameter IDs are appended to any existing subscription.

Example

Subscribe to BatteryVoltage1 through a qualified name, and BatteryVoltage2 using an OPS name:

```
[ 1, 1, 789, {
    "parameter": "subscribe",
    "data": {
        "list": [
            { "name": "/YSS/SIMULATOR/BatteryVoltage1" },
            { "namespace": "MDB:OPS Name", "name":
"SIMULATOR_BatteryVoltage2" }
        ]
    }
} ]
```

Response

You first get an empty reply message confirming the positive receipt of your request:

```
[ 1, 2, 789 ]
```

Further messages will be marked as type `PARAMETER_DATA`. Directly after you subscribe, you will receive the latest cached values – if applicable.

```
todo
```

Unsubscribe

Unsubscribe from all currently subscribed parameters:

```
[ 1, 1, 790, { "parameter": "unsubscribe" } ]
```

This will be confirmed with an empty reply message:

```
[ 1, 2, 790 ]
```

2.2. Alarm Notices

Subscribe to alarm notices:

```
[ 1, 1, :seq, { "alarms": "subscribe" } ]
```

Response

You first get an empty reply message confirming the positive receipt of your request:

```
[ 1, 2, 789 ]
```

Further messages will be marked as type `ALARM_DATA`. Directly after you subscribe, you will receive get the active set of alarms – if applicable.

```
[1,4,0,{"dt":"ALARM_DATA","data":{"id":0,"type":1,"triggerValue":
{"id":{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":227},"engValue":
{"type":2,"uint32Value":227},"acquisitionTime":1440576556724,"generationT
08-26T08:08:40.724","generationTimeUTC":"2015-08-
26T08:08:23.714","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
08-26T08:08:42.224"},"mostSevereValue":{"id":
{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":227},"engValue":
{"type":2,"uint32Value":227},"acquisitionTime":1440576556724,"generationT
08-26T08:08:40.724","generationTimeUTC":"2015-08-
26T08:08:23.714","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
08-26T08:08:42.224"},"currentValue":{"id":
{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":258},"engValue":
{"type":2,"uint32Value":258},"acquisitionTime":1440576955780,"generationT
08-26T08:15:19.780","generationTimeUTC":"2015-08-
26T08:15:02.777","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
08-26T08:15:21.280"},"violations":65}}]
[1,4,1,{"dt":"ALARM_DATA","data":{"id":0,"type":4,"triggerValue":
{"id":{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":227},"engValue":
{"type":2,"uint32Value":227},"acquisitionTime":1440576556724,"generationT
08-26T08:08:40.724","generationTimeUTC":"2015-08-
26T08:08:23.714","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
08-26T08:08:42.224"},"mostSevereValue":{"id":
{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":227},"engValue":
{"type":2,"uint32Value":227},"acquisitionTime":1440576556724,"generationT
08-26T08:08:40.724","generationTimeUTC":"2015-08-
26T08:08:23.714","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
```

```
08-26T08:08:42.224"},"currentValue":{"id":
{"name":"/YSS/SIMULATOR/O2TankTemp"},"rawValue":
{"type":2,"uint32Value":280},"engValue":
{"type":2,"uint32Value":280},"acquisitionTime":1440576962013,"generationT
08-26T08:15:26.013","generationTimeUTC":"2015-08-
26T08:15:09.011","watchLow":10.0,"watchHigh":12.0,"warningLow":30.0,"warn
08-26T08:15:27.513"},"violations":66}}]
```


Notice how we are first getting an alarm of type `ACTIVE` that triggered somewhere before we connected, and only then a further update on that alarm of type `PVAL_UPDATED`.

Unsubscribe

Unsubscribe from all further alarm updates:

```
[ 1, 1, 790, { "alarms": "unsubscribe" } ]
```

This will be confirmed with an empty reply message:

```
[ 1, 2, 790 ]
```

2.3. Event Updates

The `events` resource type within the WebSocket API allows subscribing to event updates.

Subscribe

Within the WebSocket request envelope use these values:

- request-type `events`
- request `subscribe`

This will make your web socket connection receive updates of the type `ProtoDataType.EVENT`.

Here's example output in JSON (with Protobuf, there's an applicable getter in the `WebSocketSubscriptionData`).

```
[1,2,3]
[1,4,0,{"dt":"EVENT","data":
{"source":"CustomAlgorithm","generationTime":1440823760490,"receptionTime

[1,4,1,{"dt":"EVENT","data":
{"source":"CustomAlgorithm","generationTime":1440823765491,"receptionTime

[1,4,2,{"dt":"EVENT","data":
{"source":"CustomAlgorithm","generationTime":1440823770490,"receptionTime
```

Unsubscribe

Within the WebSocket request envelope use these values:

- request-type `events`
- request `unsubscribe`

This will stop your WebSocket connection from getting further event updates.

2.4. Management Updates

The `management` resource type within the WebSocket API groups general Yamcs info that does not fit under a specific other resource type.

Subscribe to Management Updates

Within the websocket request envelope use these values:

- request-type `management`
- request `subscribe`

This will make your web socket connection receive updates on the following Yamcs data types (aka `ProtoDataType`):

CLIENT_INFO

Updates on a client that is or was connected to Yamcs. Directly after sending a `management/subscribe` request, you will also get an update for all the clients that are connected at that point. As soon as a client is disconnected, you will also get an update on that.

Here's example output in JSON (with Protobuf, there's an applicable getter in the `WebSocketSubscriptionData`), where there are two anonymous clients connected to a non-secured deployment of Yamcs:

```
[1,2,3]
[1,4,1,{"dt":"CLIENT_INFO","data":
{"instance":"simulator","id":1,"username":"unknown","applicationName":"Un

[1,4,2,{"dt":"CLIENT_INFO","data":
{"instance":"simulator","id":3,"username":"unknown","applicationName":"Un
```

After the empty initial ACK, we receive two data messages, one for each connected client at that point. Notice the `currentClient` field which indicates whether this client info concerns your own web socket session. The `state` field indicates either CONNECTED (0) or DISCONNECTED (1). In case the other use would close its connection, we would thus get an update on that too.

PROCESSOR_INFO

Updates on the lifecycle of Yamcs TM/TC processors. Directly after sending a `management/subscribe` request, you will also get an update for all the processors at that point.

Here's an example output in JSON (with Protobuf, there's an applicable getter in the `WebSocketSubscriptionData`), where there is just one processor `realtime`.

```
[1,2,3]
[1,4,0,{"dt":"PROCESSOR_INFO","data":
```

```
{"instance":"simulator","name":"realtime","type":"realtime","creator":"sy
```

PROCESSING_STATISTICS

General statistics on processors. For every high-level packet, shows you the the current processing time, an increasing data count, and many more info. Updates on this data type are a bit noisier than the client_info/processor_info updates, so we might decide to move this out to a different subscription request in the future.

Here's an example output in JSON (with Protobuf, there's an applicable getter in the `WebSocketSubscriptionData`):

```
[1,4,3,{"dt":"PROCESSING_STATISTICS","data":
{"instance":"simulator","yProcessorName":"realtime","tmstats":
[{"packetName":"RCS","receivedPackets":2378,"lastReceived":1438235693322,
{"packetName":"FlightData","receivedPackets":73718,"lastReceived":1438235
{"packetName":"ccsds-
default","receivedPackets":2378,"lastReceived":1438235693322,"lastPacketT
{"packetName":"Power","receivedPackets":2378,"lastReceived":1438235693322
{"packetName":"DHS","receivedPackets":2378,"lastReceived":1438235693322,"

[1,4,4,{"dt":"PROCESSING_STATISTICS","data":
{"instance":"simulator","yProcessorName":"realtime","tmstats":
[{"packetName":"RCS","receivedPackets":2378,"lastReceived":1438235693322,
{"packetName":"FlightData","receivedPackets":73723,"lastReceived":1438235
{"packetName":"ccsds-
default","receivedPackets":2378,"lastReceived":1438235693322,"lastPacketT
{"packetName":"Power","receivedPackets":2378,"lastReceived":1438235693322
{"packetName":"DHS","receivedPackets":2378,"lastReceived":1438235693322,"
```

Get Client-Info

Within the websocket request envelope use these values:

- request-type `management`
- request `getClientInfo`

This will return you a *one-time* data frame containing your own client info. The format is exactly the same as for the above described subscription updates.

2.5. Stream Updates

The `stream` resource type within the WebSocket API groups low-level publish/subscribe operations on Yamcs Streams.

The documented operations work on one of the built-in streams (like `tm`, `tm_realtime`, `tm_dump`, `pp_realtime`, `cmdhist_realtime`, etc). Or, if your Yamcs deployment defines any other streams, they would work as well.

Subscribe to a Stream

Within the websocket request envelope use these values:

- request-type `stream`
- request `subscribe`
- data
  - With Protobuf: an object of type `Yamcs.StreamSubscribeRequest` where at least the `stream` name is filled in.
  - With JSON: an object literal where at least the `stream` key is set.

Here's a full request example in JSON-notation

```
[1,1,3,{"stream": "subscribe", "data": {"stream": "tm_realtime"}}]
```

We are thinking of maybe adding other options, like a way to limit the data by filtering on one of the stream's columns, but this is pending an actual use case.

As a result of the above call you will get updates whenever anybody publishes data to the specified stream. With Protobuf, the data can be fetched with the `getStreamData()`-method on in the `WebSocketSubscriptionData` object. With JSON, you might see something like this sample output:

```
[1,2,3]
[1,4,0,{"dt":"STREAM_DATA","data":
{"stream":"tm_realtime","columnValue":[{"columnName":"gentime","value":
{"type":6,"timestampValue":1438608491320}},
{"columnName":"seqNum","value":{"type":3,"sint32Value":134283264}},
{"columnName":"rectime","value":
{"type":6,"timestampValue":1438608508323}},
{"columnName":"packet","value":
{"type":4,"binaryValue":"CAEAAAAPQuou2FJFAAAABOcAAAAAAA=="}}]}}]
[1,4,1,{"dt":"STREAM_DATA","data":
{"stream":"tm_realtime","columnValue":[{"columnName":"gentime","value":
{"type":6,"timestampValue":1438608491320}},
{"columnName":"seqNum","value":{"type":3,"sint32Value":134283264}},
{"columnName":"rectime","value":
{"type":6,"timestampValue":1438608508323}},
{"columnName":"packet","value":
```

```
{"type":4,"binaryValue":"CAEAAAAPQuou2FJFAAAABOcAAAAAAA=="}}]}}]
```

In the case we were receiving some simulated data from the `tm_realtime` stream, this is a built-in stream with columns `gentime`, `rectime`, `seqNum` and `packet`. This last column is of binary format (it's the raw TM packet), which is why it is Base64-encoded in the JSON output.

Other streams would have different columns.

Publish to a Stream

Within the websocket request envelope use these values:

- request-type `stream`
- request `publish`
- data
  - With Protobuf: an object of type `Yamcs.StreamData` where the `stream` key is set to the targeted stream, and where column-values are added for every column of that stream's definition.
  - With JSON: an object literal where the `stream` key is set to the targeted stream, and where `columnValue` is an array of `columnName`-`value` literals.

  The provided type has to match the type of the actual stream's definition.

Here's a full publish-request to the `tm_realtime`-stream in JSON-notation. Notice the similarities with the return type on the subscribe-operation.

```
[1,1,3,{"stream":"publish","data":
{"stream":"tm_realtime","columnValue":[{"columnName":"gentime","value":
{"type":6,"timestampValue":1438608491320}},
{"columnName":"seqNum","value":{"type":3,"sint32Value":134283264}},
{"columnName":"rectime","value":
{"type":6,"timestampValue":1438608508323}},
{"columnName":"packet","value":
{"type":4,"binaryValue":"CAEAAAAPQuou2FJFAAAABOcAAAAAAA=="}}]}}]
```

As as result of the above call the server will write that data to the requested stream (hence notifying anybody that might be subscribed to that stream), and return with just an empty ACK:

```
[1,2,3]
```

Chapter 3. Proto Files

3.1. alarms.proto

```
package alarms;
option java_package = "org.yamcs.protobuf";

import "pvalue.proto";

message AcknowledgeInfo {
  optional string acknowledgedBy = 1;
  optional string acknowledgeMessage = 2;
  optional int64 acknowledgeTime = 3;
  optional string acknowledgeTimeUTC = 4;
}

message AlarmData {
  enum Type {
    ACTIVE = 1;  // Initial active alarms at the moment of request
    TRIGGERED = 2;  // Whenever a new alarm triggers
    SEVERITY_INCREASED = 3;  // Whenever an alarm jumps severity
    PVAL_UPDATED = 4;  // Whenever a pval is updated (even if that pval
is not a violation by itself)
    ACKNOWLEDGED = 5;  // Whenever somebody acknowledged an alarm (it
could be that it is still OOL)
    CLEARED = 6; // When the alarm was really cleared by the server
(acknowledged && not OOL)
  }
  optional uint32 seqNum = 1;
  optional Type type = 2;
  optional pvalue.ParameterValue triggerValue = 3;
  optional pvalue.ParameterValue mostSevereValue = 4;
  optional pvalue.ParameterValue currentValue = 5;
  optional uint32 violations = 6;
  optional AcknowledgeInfo acknowledgeInfo = 7;
}
```

3.2. archive.proto

```
package archive;
option java_package = "org.yamcs.protobuf";

import "yamcs.proto";
import "commanding.proto";
import "pvalue.proto";

message ColumnData {
  optional string name = 1;
  optional yamcs.Value value = 2;
}

message StreamData {
  optional string stream = 1;
  repeated ColumnData column = 2;
}

message TableData {
  message TableRecord {
    repeated ColumnData column = 1;
  }
  repeated TableRecord record = 1;
}

message ColumnInfo {
  optional string name = 1;
  optional string type = 2;
}

message TableInfo {
  optional string name = 1;
  repeated ColumnInfo keyColumn = 2;
  repeated ColumnInfo valueColumn = 3;
}

message StreamInfo {
  optional string name = 1;
  repeated ColumnInfo column = 2;
}
```

3.3. commanding.proto

```
package commanding;
option java_package = "org.yamcs.protobuf";

import "yamcs.proto";
import "mdb.proto";


message CommandId {
  required int64 generationTime = 1;
  required string origin = 2;
  required int32 sequenceNumber = 3; //this has to be unique in
relation to the generation time and origin
  optional string commandName = 4;
}

/************* Command Queue Control ********************/
enum QueueState {
  BLOCKED = 1;
  DISABLED = 2;
  ENABLED = 3;
}

message CommandQueueInfo {
  required string instance = 1;
  required string processorName = 2;
  required string name = 3;
  optional QueueState state = 4;
  required int32 nbSentCommands = 5;
  required int32 nbRejectedCommands = 6;
  optional int32 stateExpirationTimeS = 7;
  repeated CommandQueueEntry entry = 8;
  optional string url = 9;
}

/*One entry (command) in the command queue*/
message CommandQueueEntry {
  required string instance = 1;
  required string processorName = 2;
  required string queueName = 3;
  required CommandId cmdId = 4;
  optional string source = 5;
  optional bytes binary = 6;
  optional string username = 7;
  optional int64 generationTime = 8;
  optional string uuid = 9;
}
```

```
message CommandQueueEvent {
  enum Type {
    COMMAND_ADDED = 1;
    COMMAND_REJECTED = 2;
    COMMAND_SENT = 3;
  }
  optional Type type = 1;
  optional CommandQueueEntry data = 2;
}


message CommandQueueRequest {
  optional CommandQueueInfo queueInfo = 1; // for SetQueueState
  optional CommandQueueEntry queueEntry = 2; //for SendCommand and
RejectCommand
  optional bool rebuild = 3[default=false]; //if rebuild is true, the
binary packet will be recreated to include new time and sequence count
}

/* this message is sent as response to validate, in case the
significance is defined for a commands*/
message CommandSignificance {
  optional int32 sequenceNumber = 1; //the sequence number of the
command sent
  optional mdb.SignificanceInfo significance = 2;
}


/********************* Command History ***********************/

message CommandHistoryAttribute {
  optional string name = 1;
  optional yamcs.Value value = 2;
  optional int64 time = 3;
}

message CommandHistoryEntry {
  required CommandId commandId = 1;
  //required string cmdName = 2; //removed because it's part of the
CommandId
  repeated CommandHistoryAttribute attr = 3;
}
```

3.4. mdb.proto

{% highlight proto %} {% include api/mdb.proto %} {% endhighlight %}

3.5. pvalue.proto

{% highlight proto %} {% include api/pvalue.proto %} {% endhighlight %}

3.6. rest.proto

{% highlight proto %} {% include api/rest.proto %} {% endhighlight %}

3.7. web.proto

{% highlight proto %} {% include api/web.proto %} {% endhighlight %}

3.8. yamcs.proto

{% highlight proto %} {% include api/yamcs.proto %} {% endhighlight %}

3.9. yamcsManagement.proto

{% highlight proto %} {% include api/yamcsManagement.proto %} {% endhighlight %}