EDS API v1

General Information

/login endpoint creates session and returns session id. Client should provide session token using Authorization header in the form of *Authorization: Bearer <token>*
See errors for error handling description.

/api/v1/authenticate

Checks user name and password. Does not create session.

POST

REQUEST

```
{
    "username": "myname",
    "password": "mypass"
}
```

SCHEMA

```
{
    "type": "object",
    "properties": {
        "username": {"type": "string"},
        "password": {"type": "string"}
    },
    "required": ["username", "password"]
}
```

RESPONSE

```
{
    "authenticated": true
}
```

/api/v1/login

Creates new session.

POST

REQUEST

```
{
    "username": "myname",
    "password": "mypass",
    "type": "myapp"       // client type "portal", "mobile" or custom name
}
```

SCHEMA

```
{
    "type": "object",
    "properties": {
        "username": {"type": "string"},
        "password": {"type": "string"},
        "type": {"type": "string"}
    },
    "required": ["username", "password", "ty
}
```

RESPONSE

```
{
    "sessionId": 00000000-0000-0000-0000-000000000000
}
```

/api/v1/logout

Closes session.

POST

/api/v1/ping

EDS API v1

/api/v1/points/query

Query points matching selected criteria or a pre-defined filter.

GET - query points matching a pre-defined filter.

Parameters:

- "source" - source name
- "order" - semicolon-separated list of point field names e.g. "zd;iess;sid" or "aux;-sid". Adding "-" before a field name reverses the order. Full list of fields that may appear in the order parameter: sid, value, dts, tss, at, atss, quality, iess, desc, rt, zd, idcs, ar, ap, aux, un, dp, artd, ard, tb, bb, hl, ll, dun, ddp, dartd, dard, dtb, dbb, dhl, dll, sd, rd.
- "fields" - (optional) comma-separated list of fields returned in response
- "page" - (optional) page number
- "pagesize" - (optional) points per page (default: 50, max: 1000)

POST - query points matching the filters.

Point value for ANALOG and DOUBLE points can be a string with "Inf", "-Inf", "NaN" value.
The "order" parameter should be a list of point field names e.g. ["zd", "iess", "sid"] or ["aux", "-sid"]. Adding "-" before a field name reverses the order. Full list of fields that may appear in the order parameter: sid, value, dts, tss, at, atss, quality, iess, desc, rt, zd, idcs, ar, ap, aux, un, dp, artd, ard, tb, bb, hl, ll sd, rd. }

GET

REQUEST

```
/api/v1/points/query?source=zd1&filter=myfitler1&order=iess
```

RESPONSE

```
see "points/query" ...
```

POST

REQUEST

```
{
  "filters": [{                          // (optional) filters array joined by OR
    "sid": [1, 2],                       // (optional) sid array
    "iess": ["iess1", "iess2"],          // (optional) iess array
    "iessRe": "^abc",                    // (optional) iess regular expression
    "idcs": ["idcs1", "idcs2"],          // (optional) idcs array
    "idcsRe": "^abc",                    // (optional) idcs regular expression
    "zd": ["zd1", "zd2"],                // (optional) source array
    "zdRe": "^abc",                      // (optional) source regular expression
    "descRe": "^abc",                    // (optional) description regular expression
    "auxRe": "^abc",                     // (optional) aux regular expression
    "rt": ["ANALOG", "DOUBLE"],          // (optional) record type array
    "ts": {                              // (optional) time modification range
      "from": 1603305950,
      "till": 1603305959
    },
    "quality": ["GOOD", "FAIR"],  // (optional) quality array
    "stSet": 0,                          // (optional) status bits set
    "stUnset": 0,                        // (optional) status bits unset
    "dfSet": 0,                          // (optional) definition flags set
    "dfUnset": 0,                        // (optional) definition flags unset
    "xst1Set": 0,                        // (optional) xst1 bits set
    "xst1Unset": 0,                      // (optional) xst1 bits unset
    "xst2Set": 0,                        // (optional) xst2 bits set
    "xst2Unset": 0,                      // (optional) xst2 bits unset
    "xst3Set": 0,                        // (optional) xst3 bits set
    "xst3Unset": 0,                      // (optional) xst3 bits unset
    "ar": ["LONG_TERM"],                 // (optional) archiving type array
    "artd": ["NONE", "PCT_RANGE"] // (optional) archiving deadband type array
    "sg": [0, 1],                        // (optional) security groups array
    "tg": [0, 1],                        // (optional) technological groups array
    "ap": [1, 2]                         // (optional) alarm priority array
  }]
  },
  "order": ["zd", "iess"],        // (optional) order, see description for more details
  "fields": ["rt" ,"iess"], // (optional) list of fields returned in response
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filters": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "sid": {"type": "
          "iess": {"type":
          "iessRe": {"type"
          "idcs": {"type":
          "idcsRe": {"type"
          "zd": {"type": "a
          "zdRe": {"type":
          "descRe": {"type"
          "auxRe": {"type":
          "rt": {"type": "a
          "ts": {
            "type": "object
            "properties": {
            "required": ["f
          },
          "quality": {"type
          "stSet": {"type":
          "stUnset": {"type
          "dfSet": {"type":
          "dfUnset": {"type
          "xst1Set": {"type
          "xst1Unset": {"ty
          "xst2Set": {"type
          "xst2Unset": {"ty
          "xst3Set": {"type
          "xst3Unset": {"ty
          "ar": {"type": "a
          "artd": {
```

EDS API v1

          enum : [NON
                   "GEO
          }
        },
        "sg": {"type": "a
        "tg": {"type": "a
        "ap": {"type": "a
      }
    }
  },
  "order": {"type": "arra
  "page": {"type": "integ
  "pagesize": {"type": "i
  "fields": {
    "type": "array",
    "items": {
      "type": "string",
      "enum": ["sid", "ie
                "desc", "s
                "dp", "ard
    }
  }
 }
}

RESPONSE

```
{
  "points": [{
    "sid": 1,            // (optional) sid
    "iess": "iess1",     // (optional) iess
    "idcs": "idcs1",     // (optional) idcs
    "zd": "zd1",         // (optional) source
    "rt": "ANALOG",      // (optional) record type
    "value": 1.23,       // (optional) value
    "quality": "GOOD",   // (optional) quality
    "ts": 1603305959,    // (optional) last read timestamp
    "lts": 1603305959,   // (optional) long term modification timestamp
    "tss": 0,            // (optional) timestamp shift
    "at": 0,             // (optional) alarm timestamp
    "atss": 0,           // (optional) alarm timestamp shift
    "desc": "desc1",     // (optional) description
    "st": 0,             // (optional) status bits
    "xst1": 0,           // (optional) external status 1 bits
    "xst2": 0,           // (optional) external status 2 bits
    "xst3": 0,           // (optional) external status 3 bits
    "ar": "LONG_TERM",   // (optional) archiving type
    "artd": "PCT_RANGE", // (optional) archiving deadband type
    "ard": 1.0,          // (optional) archiving deadband
    "sg": [0, 1],        // (optional) security groups
    "tg": [],            // (optional) technological groups
    "df": 0,             // (optional) definition flags
    "ap": 0,             // (optional) alarm priority
    "aux": "",           // (optional) aux
    "un": "m",           // (optional) unit
    "dp": 2,             // (optional) display precision
    "tb": 2,             // (optional) top bar
    "bb": 0,             // (optional) bottom bar
    "hl": 2,             // (optional) high limit
    "ll": 0,             // (optional) low limit
    "sd": "NORMAL",      // (optional) set description (BINARY)
    "rd": "ALARM",       // (optional) reset description (BINARY)
    "foreground": 123,   // (optional) foreground color
    "background": 123    // (optional) background color
  }],
  "matchCount": 1,       // matched elements count
  "totalCount": 100000,  // total point count
}
```

/api/v1/points/export

Export points matching selected criteria or a pre-defined filter.

GET - export points matching a pre-defined filter.

Parameters:

- "zd" - zd name
- "iess" - iess simple regular expression
- "idcs" - idcs simple regular expression
- "desc" - desc simple regular expression
- "aux" - aux simple regular expression

EDS API v1

name reverses the order. Full list of fields that may appear in the order parameter: sid, value, dts, tss, at, atss, quality, iess, desc, rt, zd, idcs, ar, ap, aux, un, dp, artd, ard, tb, bb, hl, ll, dun, ddp, dartd, dard, dtb, dbb, dhl, dll, sd, rd.

- "separator" - space is the default separator
- "encoding" - one of: iso-8859-1, iso8859-1, iso-latin-1, latin-1, latin1, iso-8859-2, iso8859-2, iso-latin-2, latin-2, latin2, windows-1250, windows1250, windows-1251, windows1251, windows-1252, windows1252, utf-8, utf8, utf-16be, utf16be, utf-16le, utf16le, unicode
- "flags" - (optional) 0x0002 - export with SIDs
- "page" - (optional) page number
- "pagesize" - (optional) points per page (default: 1000000, max: 1000000)

POST - export points matching the filters.

Point value for ANALOG and DOUBLE points can be a string with "Inf", "-Inf", "NaN" value.
The "order" parameter should be a list of point field names e.g. ["zd", "iess", "sid"] or ["aux", "-sid"]. Adding "-" before a field name reverses the order. Full list of fields that may appear in the order parameter: sid, value, dts, tss, at, atss, quality, iess, desc, rt, zd, idcs, ar, ap, aux, un, dp, artd, ard, tb, bb, hl, ll sd, rd. }

GET

REQUEST

```
/api/v1/points/export?zd=zd1&iess=^A&order=iess
```

RESPONSE

```
see "points/export" ...
```

POST

REQUEST

```
{
  "filters": [{                      // (optional) filters array joined by OR
    "sid": [1, 2],                   // (optional) sid array
    "iess": ["iess1", "iess2"],      // (optional) iess array
    "iessRe": "^abc",                // (optional) iess regular expression
    "idcs": ["idcs1", "idcs2"],      // (optional) idcs array
    "idcsRe": "^abc",                // (optional) idcs regular expression
    "zd": ["zd1", "zd2"],            // (optional) source array
    "zdRe": "^abc",                  // (optional) source regular expression
    "descRe": "^abc",                // (optional) description regular expression
    "auxRe": "^abc",                 // (optional) aux regular expression
    "rt": ["ANALOG", "DOUBLE"],      // (optional) record type array
    "ts": {                          // (optional) time modification range
      "from": 1603305950,
      "till": 1603305959
    },
    "quality": ["GOOD", "FAIR"],     // (optional) quality array
    "stSet": 0,                      // (optional) status bits set
    "stUnset": 0,                    // (optional) status bits unset
    "dfSet": 0,                      // (optional) definition flags set
    "dfUnset": 0,                    // (optional) definition flags unset
    "xst1Set": 0,                    // (optional) xst1 bits set
    "xst1Unset": 0,                  // (optional) xst1 bits unset
    "xst2Set": 0,                    // (optional) xst2 bits set
    "xst2Unset": 0,                  // (optional) xst2 bits unset
    "xst3Set": 0,                    // (optional) xst3 bits set
    "xst3Unset": 0,                  // (optional) xst3 bits unset
    "ac": "DEFAULT",                 // (optional) description regular expression
    "acRe": "^DEF",                  // (optional) description regular expression
    "ar": ["LONG_TERM"],             // (optional) archiving type array
    "artd": ["NONE", "PCT_RANGE"]    // (optional) archiving deadband type array
    "sg": [0, 1],                    // (optional) security groups array
    "tg": [0, 1],                    // (optional) technological groups array
    "ap": [1, 2]                     // (optional) alarm priority array
  }]
  },
  "separator": " ",                  // (optional) separator, see description for more details
  "encoding": "utf-8",               // (optional) encoding, see description for more details
  "order": ["zd", "iess"],           // (optional) order, see description for more details
  "fields": ["rt" ,"iess"], // (optional) list of fields returned in response
  "flags": 0x0002,                   // (optional) flags, default: 0x0002 - export with SIDs
  "page": 2,                         // (optional) page number, default: 1
  "pagesize": 50                     // (optional) results per page, default: 1000000, max: 1000000
}
```

EDS API v1

RESPONSE

```
# ENCODING='utf-8'
# SAVE_TIMESTAMP=1680275316.0 SAVE_TIMESTAMP_LOCAL=2023-03-31_17:08:36
# POINTS_AND_CONFIGS_MD5=1f410743b28acf59bf94a3b6f10a3d06
# POINTS_TIMESTAMP=1677784798.4 POINTS_TIMESTAMP_LOCAL=2023-03-02_20:19:58
# CONFIGS_TIMESTAMP=1647328961.7 CONFIGS_TIMESTAMP_LOCAL=2022-03-15_08:22:42
CONFIG NAME='DEFAULT' AR='F' ARTD='V' ARD=0 ARP=' SELECTIVE_SOURCE_MERGE=\'0\' TREND_FORCEABLE=\'0\' USE_IDCS=\'1\'' SO
POINT RT=ANALOG SID=45631 DF=0x0124 IESS='UHN17CF201_M' ZD='Ovation' IDCS='UHN17CF201_M' DESC='F.spalin w kominie D_MOD
POINT RT=ANALOG SID=45632 DF=0x0124 IESS='UHN18CQ203' ZD='Ovation' IDCS='UHN18CQ203' DESC='NOx (6%) w spalinach - kom.C
POINT RT=ANALOG SID=45633 DF=0x0124 IESS='UHN18CT201' ZD='Ovation' IDCS='UHN18CT201' DESC='TEMPERATURA SPALIN KOMIN C'
POINT RT=ANALOG SID=45634 DF=0x0124 IESS='UHN19CQ203' ZD='Ovation' IDCS='UHN19CQ203' DESC='NOx (6%) w spalinach - kom.D
POINT RT=ANALOG SID=45635 DF=0x0124 IESS='UHN19CT201' ZD='Ovation' IDCS='UHN19CT201' DESC='TEMPERATURA SPALIN KOMIN D'
# END
```

/api/v1/points/sources

Returns a list of all point sources.
Note: point sources and object sources might be completely different, even though in many cases they have similar names.

GET

REQUEST

```
/api/v1/points/sources
```

RESPONSE

```
["zd1", "zd2"]
```

/api/v1/points/operate

Changes current points values.

POST

REQUEST

```
[{
  "sid": 1,              // (optional) sid
  "iess": "iess1",       // (optional) iess
  "idcs": "idcs1",       // (optional) idcs
  "zd": "zd1",           // (optional) source
  "value": 1.1,          // (optional) value
  "quality": "GOOD"      // (optional) quality
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "sid": {"type": "integer", "minimum": 0},
      "iess": {"type": "string"},
      "idcs": {"type": "string"},
      "zd": {"type": "string"},
      "value": {"type": ["number", "string", "boolean"]},
      "quality": {"type": "string", "enum": ["GOOD", "FAIR", "POOR",
    }
  }
}
```

EDS API v1

Automatically refresh point values for 'duration' time after the call

POST

REQUEST

```
[{
  "sid": 1,             // (optional) sid
  "iess": "iess1",      // (optional) iess
  "idcs": "idcs1",      // (optional) idcs
  "zd": "zd1",          // (optional) source
  "value": 1.1,         // (optional) value
  "quality": "GOOD",    // (optional) quality
  "duration": 60,       // (optional) duration
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "sid": {"type": "integer", "minimum": 0},
      "iess": {"type": "string"},
      "idcs": {"type": "string"},
      "zd": {"type": "string"},
      "value": {"type": ["number", "string", "boolean"]},
      "quality": {"type": "string", "enum": ["GOOD", "FAIR", "POOR"
      "duration": {"type": "integer", "minimum": 1}
    }
  }
}
```

/api/v1/points/unpublish

Turns off automatic refresh of points values.

POST

REQUEST

```
[{
  "sid": 1,             // (optional) sid
  "iess": "iess1",      // (optional) iess
  "idcs": "idcs1",      // (optional) idcs
  "zd": "zd1"           // (optional) source
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "sid": {"type": "integer", "minimum": 0},
      "iess": {"type": "string"},
      "idcs": {"type": "string"},
      "zd": {"type": "string"}
    }
  }
}
```

/api/v1/objects/query

Query objects metadata matching selected criteria.
The "order" parameter should be a list of fields names, for example: ["file", "-name"]. Adding "-" before a field name reverses the order.
Full list of fields that may appear in the order parameter: file, name, sourceName, sourceId, modified, sg, tg, md5.

POST

REQUEST

```
{
  "filters": [{              // (optional) filters array joined by OR
    "id": [1, 2],            // (optional) id array
    "fileRe": "^abc",        // (optional) filename regular expression
    "nameRe": "^abc",        // (optional) name regular expression
    "sourceNameRe": "^abc",  // (optional) source name regular expression
    "sourceId": [1],         // (optional) source ids array
    "modified": {            // (optional) time modification range
      "from": 1603305950,
      "till": 1603305959
    },
    "sg": [0, 1],            // (optional) security groups array
    "tg": [0, 1],            // (optional) technological groups array
    "md5": "abcdefg"         // (optional) md5 hash
  }],
  "order": ["file"],         // (optional) order, see description for more details
  "fields": ["file", "name"], // (optional) list of fields returned in response
  "page": 2,                 // (optional) page number, default: 1
  "pagesize": 50             // (optional) results per page, default: 100, max: 1000
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filters": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {"type": "ar
          "fileRe": {"type":
          "nameRe": {"type":
          "sourceNameRe": {"
          "sourceId": {"type
          "modified": {
            "type": "object"
            "properties": {"
          },
          "required": ["fr
        },
        "sg": {"type": "ar
        "tg": {"type": "ar
```

EDS API v1

```
},
"order": {"type": "array
"page": {"type": "intege
"pagesize": {"type": "in
"fields": {
  "type": "array",
  "items": {
    "type": "string",
    "enum": ["id", "file
  }
 }
}
}
```

RESPONSE

```
{
  "objects": [{
    "id": 1,                  // (optional) id
    "file": "file1.edf",      // (optional) file
    "name": "name1",          // (optional) name
    "sourceName": "src1",     // (optional) source name
    "sourceId": 3,            // (optional) source id
    "modified": 1603305959,   // (optional) modification timestamp
    "sg": [0, 1],             // (optional) security groups
    "tg": [],                 // (optional) technological groups
    "md5": "abcd"             // (optional) md5 sum
  }],
  "matchCount": 1,            // matched count
  "totalCount": 100000,       // total count
}
```

/api/v1/objects

POST - create object

PUT - update object

DELETE - delete object

POST

REQUEST

```
[{
  "sourceId": 123,          // source id
  "file": "myfile.edf",     // filename
  "name": "myobject",       // (optional) name
  "sg": [0, 1],             // (optional) security groups array
  "tg": []                  // (optional) technological groups array
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "sourceId": {"type": "integer", "minimum":
      "file": {"type": "string"},
      "name": {"type": "string"},
      "sg": {"type": "array", "items": {"type":
      "tg": {"type": "array", "items": {"type":
    },
    "required": ["sourceId", "file"]
  }
}
```

PUT

REQUEST

```
[{
  "id": 123,                // source id
  "name": "myobject",       // (optional) name
  "sg": [0, 1],             // (optional) security groups array
  "tg": [],                 // (optional) technological groups array
  "force": false            // (optional) force when removing own permission, default: false
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {"type": "inte
      "name": {"type": "st
      "sg": {"type": "arra
      "tg": {"type": "arra
      "force": {"type": "b
    },
    "required": ["id"]
  }
}
```

EDS API v1

REQUEST

```
[{
    "id": 1  // id
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {"type": "integer", "minimum": 0}
    },
    "required": ["id"]
  }
}
```

/api/v1/objects/sources/query

Returns a list of all object sources if filter is not set or all object sources matching filter. Note: point sources and object sources might be completely different, even though in many cases they have similar names.
The order parameter should be a list of field names, in example: ["id"] or ["location", "-name"]. Adding "-" before a field name reverses the order.
Full list of fields that may appear in the order parameter: id, name, desc, sg, tg, location, host, prefix, suffix, options.

POST

REQUEST

```
{
  "filters": [{                        // (optional) filters array joined by OR
    "id": [1, 2],                      // (optional) id array
    "nameRe": "^abc",                  // (optional) name regular expression
    "descRe": "^abc",                  // (optional) description regular expression
    "modified": 1603305950,            // (optional) minimum modification timestamp
    "sg": [0, 1],                      // (optional) security groups array
    "tg": [0, 1],                      // (optional) technological groups array
    "location": "DB",              // (optional) source location
    "hostRe": "^abc",                  // (optional) host regular expression
    "prefixRe": "^abc",                // (optional) prefix regular expression
    "suffixRe": "^abc",                // (optional) suffix regular expression
    "optionsSet": 0,                   // (optional) xst1 bits set
    "optionsUnset": 0                  // (optional) xst1 bits unset
  }],
  "order": ["id", "name"],             // (optional) order, see description for more details
  "fields": ["name" ,"desc"],          // (optional) list of fields returned in response
  "page": 2,                           // (optional) page number, default: 1
  "pagesize": 50                       // (optional) results per page, default: 100, max: 1000
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filters": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {"type":
          "nameRe": {"typ
          "descRe": {"typ
          "modified": {"t
          "sg": {"type":
          "tg": {"type":
          "location": {"t
          "hostRe": {"typ
          "prefixRe": {"t
          "suffixRe": {"t
          "optionsSet": {
          "optionsUnset":
        }
      }
    },
    "order": {"type": "ar
    "page": {"type": "int
    "pagesize": {"type":
    "fields": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": ["id", "n
      }
    }
  }
}
```

EDS API v1

```
{
  "sources": [{
    "id": 1,                  // (optional) id
    "name": "iess1",          // (optional) name
    "desc": "desc1",          // (optional) description
    "sg": [0, 1],             // (optional) security groups
    "tg": [],                 // (optional) technological groups
    "location": "DB",         // (optional) location "DISK", "FTP" or "DB"
    "host": "",               // (optional) host
    "prefix": "",             // (optional) prefix
    "suffix": "",             // (optional) suffix
    "options": 0              // (optional) options bits
  }],
  "matchCount": 1,        // matched count
  "totalCount": 100000,   // total count
}
```

/api/v1/objects/sources

POST - create object sources

PUT - update object sources

DELETE - delete object sources

POST

REQUEST

```
[{
  "name": "mysource",  // name
  "desc": "",            // (optional) description
  "sg": [0, 1],          // (optional) security groups array
  "tg": [],              // (optional) technological groups array
  "location": "DISK",    // (optional) source location "DISK", "FTP" or "DB"
  "host": "",            // (optional) host
  "prefix": "",          // (optional) IESS prefix
  "suffix": ".a@NET1"  // (optional) IESS suffix
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "name": {"type": "string"},
      "desc": {"type": "string"},
      "sg": {"type": "array", "items":
      "tg": {"type": "array", "items":
      "location": {"type": "string", "e
      "host": {"type": "string"},
      "prefix": {"type": "string"},
      "suffix": {"type": "string"}
    },
    "required": ["name"]
  }
}
```

PUT

REQUEST

```
[{
  "id": 123,             // id
  "name": "myobject",  // (optional) name
  "desc": "",            // (optional) description
  "sg": [0, 1],          // (optional) security groups array
  "tg": [],              // (optional) technological groups array
  "location": "DISK",    // (optional) source location "DISK", "FTP" or "DB"
  "host": "",            // (optional) host
  "prefix": "",          // (optional) IESS prefix
  "suffix": ".a@NET1",   // (optional) IESS suffix
  "force": false         // (optional) force when removing own permission, default: false
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {"type": "int
      "name": {"type": "s
      "desc": {"type": "s
      "sg": {"type": "arr
      "tg": {"type": "arr
      "location": {"type"
      "host": {"type": "s
      "prefix": {"type":
      "suffix": {"type":
      "force": {"type": "
    },
    "required": ["id"]
  }
}
```

DELETE

REQUEST

```
[{
  "id": 1  // id
```

SCHEMA

```
{
  "type": "array",
```

EDS API v1

```
            id : { type : "integer", minimum : 0}
        },
        "required": ["id"]
    }
}
```

/api/v1/diagram/open

Starts diagram rendering session on Graphics Server (GfxSrv). Returns root URL in form http://GFX-HOST:GFX-PORT/UUID/. This URL is also an identifier of renderer session (URI) and should be used as a base url for communication with GfxSrv. For more details please refer to GfxSrv API documentation.

POST

REQUEST

```
{
  "source": "src1",                          // diagram source
  "file": "diag1.edf",                        // diagram file
  "redrawInterval": 3,                        // (optional) redraw interval, default: 3
  "httpUrl": "http://mydomain.com:43090/",    // (optional) GfxSrv address
  "previousUrl": "http://mydomain.com:43090/",  // (optional) re-use previous url
  "pointGroup": "pts1"                        // (optional) point group
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "source": {"type": "stri
    "file": {"type": "string
    "redrawInterval": {"type
    "httpUrl": {"type": "str
    "previousUrl": {"type":
    "pointGroup": {"type": "
  },
  "required": ["source", "fi
}
```

RESPONSE

```
{
  "url": "http://mydomain.com:43090//"
}
```

/api/v1/requests

Long running requests status.

GET - get requests status

DELETE - drop requests

GET

REQUEST

```
/api/v1/requests?ids=3214,2415,7896
```

RESPONSE

```
{
  "3214": {
    "status": "EXECUTING",  // status: "QUEUED", "EXECUTING", "SUCCESS" or "FAILURE"
    "progress": 78.12,      // (optional) progress 0 - 100
    "message": ""            // (optional) error message
  },
  "2415": null
}
```

DELETE

REQUEST

```
[{
  "id": 1  // request id
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
```

EDS API v1

```
    required : [ id ]
  }
}
```

/api/v1/trend

GET - Retrieves trend result

POST - Requests trend

Executes graphical trend. "pixelCount" is a trend width resolution. When requesting trend for long time range 1 pixel can correspond to e.g. 1 day, that's why trend request may return up to 5 samples per pixel (value at the start of the range, maximum, minimum, value at the end, hole marker). See also /api/v1/requests.

GET

```
REQUEST

/api/v1/trend?id=1234
```

```
RESPONSE

[{
  "items": [  // list of items results
    [         // list of samples
      [
        1603305950,  // timestamp
        2.1,         // value
        "G",         // quality
        0,           // (optional) timestamp shift
        0,           // (optional) sample origin (default: DataSource):
                     //   0 - DataSource
                     //   1 - Filter
                     //   2 - Function
                     //   3 - Processor
                     //   4 - Test
                     //   5 - Other
        1,           // (optional) sample type (default: Regular):
                     //   0 - Regular
                     //   1 - RegularZeroOrder
                     //   2 - ShadeBegin
                     //   3 - ShadeEnd
      ],
      [1603305951, 2.2, "G"],
      [1603305952, 2.3, "G"]
    ],
    [
      [1603305950, 0.5, "G"]
    ]
  ],
  "averages": [2.15, 0.5], // (optional) averages available with "LAST" chunk
  "status": "LAST"
}]
```

POST

```
REQUEST

[{
  "pointId": {                  // point id
    "sid": 1,                   // (optional) point sid
    "iess": "iess1"             // (optional) point iess
  },
  "period": {                   // time range
    "from": 1603305950,
    "till": 1603305959
  },
  "pixelCount": 1024,           // pixel count
  "shadePriority": "DEFAULT"    // (optional) shade priority
}]
```

```
SCHEMA

{
  "type": "array",
  "items":  {
    "type": "object",
    "properties": {
      "pointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "period": {
        "type": "object",
        "properties": {"from": {"type": "integer"}, "ti
        "required": ["from", "till"]
      },
      "pixelCount": {"type": "integer", "minimum": 1},
      "shadePriority": {"type": "string", "enum": ["DEF.
    },
    "required": ["pointId", "period", "pixelCount"]
```

EDS API v1

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/trend/tabular

GET - Retrieves tabular trend result

POST - Requests tabular trend

See also /api/v1/requests.

GET

REQUEST

```
/api/v1/trend/tabular?id=1234
```

RESPONSE

```
[{
  "items": [  // list of items results
    [          // list of samples
      [
        1603305950,  // timestamp
        2.1,         // value
        "G",         // quality
        0,           // (optional) timestamp shift
      ],
      [1603305951, 2.3, "G"],
      [1603305952, 2.4, "G"]
    ],
    [
      [1603305950, 0.5, "G"]
    ]
  ],
  "status": "LAST"
}]
```

POST

REQUEST

```
{
  "period": {                      // time range
    "from": 1603305950,
    "till": 1603305959
  },
  "step": 60,                      // step [s]
  "items": [{
    "pointId": {                   // point id
      "sid": 1,                    // (optional) point sid
      "iess": "iess1"              // (optional) point iess
    },
    "shadePriority": "DEFAULT",    // (optional) shade priority
    "function": "VALUE",           // (optional) function (default: VALUE)
    "params": [2.1]                // (optional) function parameters
  }]
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "period": {
      "type": "object",
      "properties": {"from": {"type": "in
      "required": ["from", "till"]
    },
    "step": {"type": "integer", "minimum"
    "items": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "pointId": {
            "type": "object",
            "properties": {
              "sid": {"type": "integer",
              "iess": {"type": "string"}
            }
          },
          "shadePriority": {"type": "stri
          "function": {"type": "string",
            "AVG", "AVG_QUAL", "BETWEEN_T
            "F_OVER_TIME", "F_UNDER_TIME"
            "L_MAX_DT", "L_MIN_DT", "L_OV
            "TOGGLE", "TOGGLE_OVER", "TOG
          "params": {"type": "array", "it
        },
        "required": ["pointId"]
      }
    }
  },
```

EDS API v1

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/trend/groups

Retrieves a list of trend groups migrated to the specified configuration version. If configuration version is unspecified, it defaults to the current version.
Parameters:

- "ver" - configuration version

GET

REQUEST

```
/api/v1/trend/groups?ver=9.2.0.26
```

RESPONSE

```
{
  "Unit1": {
    "ovation_trends": {
      "": {
        "test-123456": {
          "": {
            "[@config_block@]": {
              "": {
                "id": "2",
                "version": "9.2.0.36"
              }
            },
            "chart": {
              "": {
                "0": {
                  "S25B-U1-L.PCC@SFWMD": {
                    "bit": "0",
                    "colors": "FF0000 FFFF0000 FFFF0000 FFFF00",
                    "id": "1",
                    "regular_to_shade": "default",
                    "yrang": "auto"
                  }
                },
                "Configurations": {
                  "": {
                    "default": {
                      "default": {
                        "archive": "0",
                        "autoScaleMarginPercentage": "0",
                        "backgroundColor": "0",
                        "description": "",
                        "futureColor": "FFFFFF00",
                        "minimumVerticalGridLines": "4",
                        "numberOfGridLines": "7",
                        "numberOfSubGridLines": "1",
                        "pointXItemId": "-1",
                        "subwindows": {
                          "": {
                            "count": "0",
                            "layout": "quadrant",
                            "titles": {"": {"A": "", "B": "", "C": "", "D": ""}}
                          }
                        },
                        "thickLines": "0",
                        "timeRange": {
                          "": {
                            "endTime": "1637268223",
                            "startTime": "1637181823",
                            "range": "86400"
                          }
                        },
                        "type": "norm",
                        "verticalGridType": "1",
                        "verticalGridUnit": "10",
                        "viewerConfig": "1 0 0 0 1 1 0 0 0 0 0",
                        "viewerEventsConfig": "FF FFFF",
                        "viewerHaxisConfig": {"": {"0": "0 1 0", "1": "", "2": ""}}
                      }
                    }
                  }
```

EDS API v1

```
                          futureColor : FFFFFF00 ,
                        "thickLines": "0",
                        "timeRange": {
                          "": {
                            "endTime": "1637268223",
                            "startTime": "1637181823",
                            "range": "86400"
                          }
                        },
                        "type": "norm"
                      }
                    }
                  }
                }
              }
            }
          }
        }
```

/api/v1/events/read

GET - Retrieves events read results.

Parameters:

- "id" - request id
- "fields" - (optional) comma-separated list of fields returned in response. Full list of fields: "category", "type", "priority", "message", "sid", "iess", "floatValue", "intValue", "st", "ts", "tss", "aux", "foreground", "background"

POST - Requests event read.

All returned Events will be ordered in reverse chronological order. See also /api/v1/requests.

GET

REQUEST

```
/api/v1/events/read?id=1234
```

RESPONSE

```
{
  "events": [{
    "category": "ALARM",     // (optional) category: "SYSTEM", "ALARM", "EXTERNAL", "CUSTOM", "SET_POINT"
    "type": "ALARM_ANALOG", // (optional) type:
      // "SYSTEM_STARTUP", "SYSTEM_SHUTDOWN","GRANT_LOGIN", "DENY_LOGIN","LOGOUT", "TIMEOUT", "DIAGRAM_CHANGE",
      // "SCANNER", "ZD_SHUTDOWN", "REPORT_EXEC_SUCC", "REPORT_EXEC_RETR", "REPORT_EXEC_FAIL", "FORCE_FILLIN",
      // "ALARM_ANALOG", "ALARM_BINARY", "ALARM_PACKED", "ALARM_DOUBLE", "ALARM_INT64",
      // "SET_ANALOG", "SET_BINARY", "SET_PACKED", "SET_DOUBLE", "SET_INT64",
      // "EXTERNAL_ALARM_ANALOG", "EXTERNAL_ALARM_BINARY", "EXTERNAL_ALARM_PACKED", "EXTERNAL_ALARM_DROP",
      // "EXTERNAL_ALARM_HARDWARE", "EXTERNAL_ALARM_MODULE", "EXTERNAL_ALARM_NODE", "EXTERNAL_ALARM_DEVICE",
      // "EXTERNAL_ALARM_COS", "EXTERNAL_ALARM_SOE", "EXTERNAL_ALARM_SYSTEM", "EXTERNAL_ALARM_OEL",
      // "EXTERNAL_ALARM_TEXT", "CUSTOM_MESSAGE",
    "priority": 5,          // (optional) priority
    "message": "abcd",      // (optional) message
    "sid": 1234,            // (optional) point sid
    "iess": "iess1",        // (optional) point iess
    "floatValue": 5.23,     // (optional) float value
    "intValue": 32,         // (optional) int value
    "st": 1,                // (optional) status bits
    "ts": 1603305950,       // (optional) timestamp
    "tss": 0,               // (optional) timestamp shift
    "aux": 0,               // (optional) aux
    "foreground": 123,      // (optional) foreground color ARGB bytes
    "background": 123       // (optional) background color ARGB bytes
  }]
}
```

POST

REQUEST

```
{
  "filter": {                    // (optional) filter
    "period": {                  // (optional) time range
      "from": 1603305950,
      "till": 1603305959
    },
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filter": {
      "type": "object",
      "properties": {
```

EDS API v1

```
    sg : [3, 4],            // (optional) technological groups array
    "tg": [5, 6],           // (optional) technological groups array
    "zd": ["source1"],      // (optional) source array
    "iessRe": "^abc",       // (optional) iess regular expression
    "messageRe": "^abc",    // (optional) message regular expression
    "pointId": [{           // (optional) point id array
      "sid": 1,             // (optional) point sid
      "iess": "iess1"       // (optional) point iess
    }]
  },
  "maxCount": 50            // (optional) maximum number of results, default: 100
}
```

```
    required : [ from
  },
  "category": {"type":
  "rt": {"type": "arra
  "priority": {"type":
  "sg": {"type": "arra
  "tg": {"type": "arra
  "zd": {"type": "arra
  "iessRe": {"type": "
  "messageRe": {"type"
  "pointId": {
    "type": "array",
    "items": {
      "type": "object"
      "properties": {
        "sid": {"type"
        "iess": {"type
      }
    }
  }
  },
  "maxCount": {"type": "in
  },
  "required": ["filter"]
}
```

RESPONSE

```
{
  "requestId": 1234   // request id
}
```

/api/v1/events

Creates events of type CUSTOM_MESSAGE.

POST

REQUEST

```
[{
  "priority": 1,      // alarm priority
  "message": "msg1",  // (optional) message
  "pointId": {
    "sid": 2,         // (optional) point sid
    "iess": "iess1"   // (optional) point iess
  },
  "floatValue": 2.5,  // (optional) float value
  "intValue": 3,      // (optional) integer value
  "st": 4,            // (optional) status
  "ts": 1603305950,   // timestamp
  "tss": 0,           // (optional) timestamp shift
  "aux": 6            // (optional) aux
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "priority": {"type": "integer", "minimum": 1, "maximum":
      "message": {"type": "string"},
      "pointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "floatValue": {"type": "number"},
      "intValue": {"type": "integer"},
      "st": {"type": "integer", "minimum": 0},
      "ts": {"type": "integer", "minimum": 0},
      "tss": {"type": "integer", "minimum": 0},
      "aux": {"type": "integer", "minimum": 0}
    },
    "required": ["priority", "ts"]
  },
  "maxItems": 128
}
```

/api/v1/report/configs/query

Query report configurations.

POST

REQUEST

```
{
  "filters": [{                    // (optional) filters array joined by OR
```

SCHEMA

```
{
  "type": "object",
```

EDS API v1

```
    nameRe : "abc",              // (optional) name regular expression
    "sourceNameRe": "^abc",      // (optional) source name regular expression
    "sourceId": [1],             // (optional) source ids array
    "modified": {                // (optional) time modification range
      "from": 1603305950,
      "till": 1603305959
    },
    "sg": [0, 1],                // (optional) security groups array
    "tg": [0, 1],                // (optional) technological groups array
    "md5": "abcdefg"             // (optional) md5 hash
  },
  "outputType": "FILE_RDF",      // (optional) output type
  "executionCondition": "CYCLIC" // (optional) execution condition
}],
  "page": 2,                     // (optional) page number, default: 1
  "pagesize": 50                 // (optional) results per page, default: 100, max: 1000
}
```

```
    items : {
      "type": "object",
      "properties": {
        "objectFilter": {
          "id": {"type": "
          "fileRe": {"type
          "nameRe": {"type
          "sourceNameRe":
          "sourceId": {"ty
          "modified": {
            "type": "objec
            "properties":
            "required": ["
          },
          "sg": {"type": "
          "tg": {"type": "
          "md5": {"type":
        },
        "outputType": {
          "type": "string"
          "enum": ["FILE_T
                   "DATABA
        },
        "executionConditio
      }
    }
  },
  "page": {"type": "intege
  "pagesize": {"type": "in
  }
}
```

RESPONSE

```
{
  "results": [{
    "id": 1,                           // id
    "reportDefinitionSource": "src1",  // source
    "reportDefinitionFile": "r1.edf",  // file
    "referenceTimeShift": "",          // reference time shift
    "runDelay": 60,                    // run delay [s]
    "timeMaskExpression": "0 * * * *", // cron mask
    "eventsExpression": "",            // events expression
    "inputValues": "",                 // input values
    "outputMaskFileTxt": "",           // (optional) output mast file txt
    "outputMaskFileRdf": "",           // (optional) output mast file rdf
    "outputMaskFileEdf": "",           // (optional) output mast file edf
    "outputMaskFileHtml": "",          // (optional) output mast file html
    "outputMaskFilePdf": "",           // (optional) output mast file pdf
    "outputMaskFileCsv": "",           // (optional) output mast file csv
    "outputMaskDatabaseRdf": "",       // (optional) output mast database rdf
    "outputMaskDatabaseEdf": "",       // (optional) output mast database edf
    "outputMaskDatabaseHtml": ""       // (optional) output mast database html
  }],
  "matchCount": 1,         // matched count
  "totalCount": 100000,    // total count
}
```

/api/v1/report/configs

POST - create report configs

PUT - update report configs

DELETE - delete report configs

POST

REQUEST

```
[{
  "sourceId": 123,                 // source id
  "rdfFileName": "myfile.rdf",  // filename
  "config": {
    "referenceTimeShift": "",
    "runDelay": 60,
    "timeMaskExpression": "0 * * * *",
    "eventsExpression": "",
    "inputValues": "",
    "outputMaskFileTxt": "",
    "outputMaskFileRdf": "",
    "outputMaskFileEdf": "ab",
    "outputMaskFileHtml": "",
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "sourceId": {"type": "integer", "minimum": 0},
      "rdfFileName": {"type": "string"},
      "config": {
        "type": "object",
        "properties": {
          "referenceTimeShift": {"type": "string"},
          "runDelay": {"type": "integer", "minimum": 0},
          "timeMaskExpression": {"type": "string"},
```

EDS API v1

```
          outputMaskDatabaseEdf":  ,
        "outputMaskDatabaseHtml": ""
    }
}]
```

```
          outputMaskFileRdf": {"type": "string"},
        "outputMaskFileEdf": {"type": "string"},
        "outputMaskFileHtml": {"type": "string"},
        "outputMaskFilePdf": {"type": "string"},
        "outputMaskFileCsv": {"type": "string"},
        "outputMaskDatabaseRdf": {"type": "string"},
        "outputMaskDatabaseEdf": {"type": "string"},
        "outputMaskDatabaseHtml": {"type": "string"}
      }
    }
  },
  "required": ["sourceId", "rdfFileName", "config"]
}
```

PUT

REQUEST

```
[{
  "id": 123,   // config id
  "config": {
    "referenceTimeShift": "",
    "runDelay": 60,
    "timeMaskExpression": "0 * * * *",
    "eventsExpression": "",
    "inputValues": "",
    "outputMaskFileTxt": "",
    "outputMaskFileRdf": "",
    "outputMaskFileEdf": "",
    "outputMaskFileHtml": "",
    "outputMaskFilePdf": "",
    "outputMaskFileCsv": "",
    "outputMaskDatabaseRdf": "",
    "outputMaskDatabaseEdf": "",
    "outputMaskDatabaseHtml": ""
  }
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {"type": "integer", "minimum": 0},
      "config": {
        "type": "object",
        "properties": {
          "referenceTimeShift": {"type": "string"},
          "runDelay": {"type": "integer", "minimum": 0},
          "timeMaskExpression": {"type": "string"},
          "eventsExpression": {"type": "string"},
          "inputValues": {"type": "string"},
          "outputMaskFileTxt": {"type": "string"},
          "outputMaskFileRdf": {"type": "string"},
          "outputMaskFileEdf": {"type": "string"},
          "outputMaskFileHtml": {"type": "string"},
          "outputMaskFilePdf": {"type": "string"},
          "outputMaskFileCsv": {"type": "string"},
          "outputMaskDatabaseRdf": {"type": "string"},
          "outputMaskDatabaseEdf": {"type": "string"},
          "outputMaskDatabaseHtml": {"type": "string"}
        }
      }
    }
  },
  "required": ["id", "config"]
}
```

DELETE

REQUEST

```
[{
  "id": 1  // config id
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {"type": "integer", "minimum": 0}
    },
    "required": ["id"]
  }
}
```

/api/v1/report/custom

GET - Retrieves custom report result

POST - Requests custom report

See also /api/v1/requests.

GET

REQUEST

EDS API v1

```
[
  ["abc", "aaa"],
  ["1.123", "2.3"]
]
```

POST

REQUEST

```
{
  "rdf": {                                // report definition
    "localTime": true,                    // (optional) local or UTC time, default: false
    "showDstTransition": true,            // (optional) show DST transition, default: false
    "showQuality": true,                  // (optional) show value qualities, default: true
    "precision": 2,                       // (optional) default precision, default: 6
    "timeMode": "RELATIVE",               // (optional) time mode, default: "RELATIVE"
    "addressingType": "A1",               // (optional) addressing type, default: "A1"
    "shadePriority": "DEFAULT",           // (optional) shade priority, default: "DEFAULT"
    "rows": [
      [
        "abc"                             // cells as text
      ],
      [{
        "content": "aaa",                 // content
        "showQuality": true,             // (optional) show quality
        "precision": 3                    // (optional) precision
      }]
    ]
  },
  "dtRef": 1603305950,                     // DT_REF reference time
  "args": {                                // additional arguments
    "myarg": {                             // name
      "boolean": true,                     // (optional) boolean value
      "number": 5,                         // (optional) number value
      "packed": 1,                         // (optional) packed value
      "string": "x",                       // (optional) boolean value
      "timestamp": 1603305950,             // (optional) timestamp value
      "point": "pt1",                      // (optional) point name value
      "quality": "GOOD"                    // (optional) value quality, default: GOOD
    }
  }
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "rdf": {
      "type": "object",
      "properties": {
        "localTime": {"type": "
        "showDstTransition": {"
        "showQuality": {"type":
        "precision": {"type": "
        "timeMode": {"type": "s
        "addressingType": {"typ
        "shadePriority": {"type
        "rows": {
          "type": "array",
          "items": {
            "type": "array",
            "items": {
              "anyOf": [
                {
                  "type": "stri
                },
                {
                  "type": "obje
                  "properties":
                    "content":
                    "showQualit
                    "precision"
                  },
                  "required": [
                ]
              }
            }
          }
        }
      },
      "required": ["rows"]
    },
    "dtRef": {"type": "integer"
    "args": {
      "type": "object",
      "additionalProperties": {
        "type": "object",
        "properties": {
          "boolean": {"type": "
          "number": {"type": "n
          "packed": {"type": "i
          "string": {"type": "s
          "timestamp": {"type":
          "point": {"type": "st
          "quality": {"type": "
        }
      }
    }
  },
  "required": ["rdf", "dtRef"]
}
```

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/report/global

Creates global report.
See also /api/v1/requests.

POST

EDS API v1

```
{
  "sourceId": 123,               // source id
  "file": "myfile.edf",          // filename
  "name": "myobject",            // (optional) name
  "rdf": {                       // report definition
    "localTime": true,           // (optional) local or UTC time, default: false
    "showDstTransition": true,   // (optional) show DST transition, default: false
    "showQuality": true,         // (optional) show value qualities, default: true
    "precision": 2,              // (optional) default precision, default: 6
    "timeMode": "RELATIVE",      // (optional) time mode, default: "RELATIVE"
    "addressingType": "A1",      // (optional) addressing type, default: "A1"
    "shadePriority": "DEFAULT",  // (optional) shade priority, default: "DEFAULT"
    "rows": [
      [
        "abc"                    // cells as text
      ],
      [{                         // cells as objects
        "content": "aaa"         // content
        "showQuality": true,     // (optional) show quality
        "precision": 3           // (optional) precision
      }]
    ]
  },
  "sg": [0, 1],                  // (optional) security groups array
  "tg": []                       // (optional) technological groups array
}
```

```
{
  "type": "object",
  "properties": {
    "sourceId": {"type": "integ
    "file": {"type": "string"},
    "name": {"type": "string"},
    "rdf": {
      "type": "object",
      "properties": {
        "localTime": {"type": "
        "showDstTransition": {"
        "showQuality": {"type":
        "precision": {"type": "
        "timeMode": {"type": "s
        "addressingType": {"typ
        "shadePriority": {"type
        "rows": {
          "type": "array",
          "items": {
            "type": "array",
            "items": {
              "anyOf": [
                {
                  "type": "stri
                },
                {
                  "type": "obje
                  "properties":
                    "content":
                    "showQualit
                    "precision"
                  },
                  "required": [
                }
              ]
            }
          }
        }
      },
      "required": ["rows"]
    },
    "sg": {"type": "array", "it
    "tg": {"type": "array", "it
  },
  "required": ["sourceId", "fil
}
```

RESPONSE

```
{
  "id": 1234   // request id
}
```

/api/v1/report/global/run

Executes global report.
See also /api/v1/requests.

POST

REQUEST

```
{
  "configId": 123,               // report config id
  "dtRef": 1603305950,           // DT_REF reference time
  "args": {                      // additional arguments
    "myarg": {                   // name
      "boolean": true,           // (optional) boolean value
      "number": 5,               // (optional) number value
      "packed": 1,               // (optional) packed value
      "string": "x",             // (optional) boolean value
      "timestamp": 1603305950,   // (optional) timestamp value
      "point": "pt1",            // (optional) point name value
      "quality": "GOOD"          // (optional) value quality, default: GOOD
    }
  }
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "configId": {"type": "integer", "mi
    "dtRef": {"type": "integer"},
    "args": {
      "type": "object",
      "additionalProperties": {
        "type": "object",
        "properties": {
          "boolean": {"type": "boolean"
          "number": {"type": "number"},
          "packed": {"type": "integer"}
          "string": {"type": "string"},
          "timestamp": {"type": "intege
          "point": {"type": "string"},
          "quality": {"type": "string",
        }
      }
    }
  },
}
```

EDS API v1

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/shades/points

Query points with existing shade values.
The "order" parameter should be a list of fields names, for example: ["sid", "-iess"]. Adding "-" before a field name reverses the order.

POST

REQUEST

```
{
  "filters": [{            // (optional) filters array joined by OR
    "sid": [1, 2],          // (optional) point sid array
    "pointType": "ANALOG",  // (optional) point type
    "iessRe": "^abc"        // (optional) point name regular expression
  }],
  "order": ["iess"],        // (optional) order, see description for more details
  "page": 2,                // (optional) page number, default: 1
  "pagesize": 50            // (optional) results per page, default: 100, max: 1000
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filters": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "sid": {"type": "arr
          "pointType": {"type"
          "iessRe": {"type": "
        }
      }
    },
    "order": {"type": "array",
    "page": {"type": "integer"
    "pagesize": {"type": "inte
  }
}
```

RESPONSE

```
{
  "results": [{
    "sid": 1,         // (optional) id
    "iess": "iess1",  // (optional) name
    "type": "ANALOG"  // (optional) description
  }],
  "matchCount": 1      // match count
}
```

/api/v1/shades/read

Read points shades for a specific time periods. See also /api/v1/requests.

GET

REQUEST

```
/api/v1/shades/read?id=1234
```

RESPONSE

```
[{
  "items": [  // list of items results
    [        // list of samples
      [
        1603305950,  // start timestamp
        1603305960,  // end timestamp
        2.1,          // value
        "G",          // quality
      ],
      [1603305971, 1603305972, 2.2, "G"],
      [1603305981, 1603305982, 2.3, "G"]
    ],
    [
```

EDS API v1

```
      status : LAST
}]
```

POST

REQUEST

```
[{
  "pointId": {                    // point id
    "sid": 1,                     // (optional) point sid
    "iess": "iess1"               // (optional) point iess
  },
  "period": {                     // time range
    "from": 1603305950,
    "till": 1603305959
  }
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "pointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "period": {
        "type": "object",
        "properties": {"from": {"type": "integer"}, "till":
        "required": ["from", "till"]
      }
    },
    "required": ["pointId", "period"]
  }
}
```

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/shades/write

Overwrite points shades for a specific time periods. See also /api/v1/requests.

POST

REQUEST

```
[{
  "pointId": {              // point id
    "sid": 1,               // (optional) point sid
    "iess": "iess1"         // (optional) point iess
  },
  "period": {               // time range
    "from": 1603305950,
    "till": 1603305959
  },
  "value": 1.1              // (optional) value
  "quality": "GOOD",        // (optional) quality
}]
```

SCHEMA

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "pointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "period": {
        "type": "object",
        "properties": {"from": {"type": "integer"}, "till": {"ty
        "required": ["from", "till"]
      },
      "value": {"type": ["number", "string", "boolean"]},
      "quality": {"type": "string", "enum": ["GOOD", "FAIR", "PO
    },
    "required": ["pointId", "period", "value", "quality"]
  }
}
```

RESPONSE

```
{
  "id": 1234  // request id
}
```

EDS API v1

Clear points shades for a specific time periods. See also /api/v1/requests.

POST

REQUEST

```
[{
  "pointId": {              // point id
    "sid": 1,               // (optional) point sid
    "iess": "iess1"         // (optional) point iess
  },
  "period": {               // time range
    "from": 1603305950,
    "till": 1603305959
  }
}]
```

SCHEMA

```
{
  "type": "array",
  "items":  {
    "type": "object",
    "properties": {
      "pointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "period": {
        "type": "object",
        "properties": {"from": {"type": "integer"}, "till": {"ty
        "required": ["from", "till"]
      }
    },
    "required": ["pointId", "period"]
  }
}
```

RESPONSE

```
{
  "id": 1234  // request id
}
```

/api/v1/shades/copy

Copy shades between points for a specific time periods. The points must be of the same type. See also /api/v1/requests.

POST

REQUEST

```
[{
  "srcPointId": {           // source point id
    "sid": 1,               // (optional) point sid
    "iess": "iess1"         // (optional) point iess
  },
  "dstPointId": {           // destination point id
    "sid": 2,               // (optional) point sid
    "iess": "iess2"         // (optional) point iess
  },
  "period": {               // time range
    "from": 1603305950,
    "till": 1603305959
  }
}]
```

SCHEMA

```
{
  "type": "array",
  "items":  {
    "type": "object",
    "properties": {
      "srcPointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "dstPointId": {
        "type": "object",
        "properties": {
          "sid": {"type": "integer", "minimum": 0},
          "iess": {"type": "string"}
        }
      },
      "period": {
        "type": "object",
        "properties": {"from": {"type": "integer"}, "till": {"ty
        "required": ["from", "till"]
      }
    },
    "required": ["srcPointId", "dstPointId", "period"]
  }
}
```

RESPONSE

EDS API v1

/api/v1/sg

Returns a list of all technological groups (tg).

GET

REQUEST

```
/api/v1/sg
```

RESPONSE

```
[{
    "id": 0,
    "name": "admin",
    "desc": ""
}]
```

/api/v1/tg

Returns a list of all technological groups (tg).

GET

REQUEST

```
/api/v1/tg
```

RESPONSE

```
[{
    "id": 0,
    "name": "admin",
    "desc": ""
}]
```

/api/v1/user/sg

Returns a list of all technological groups (tg).

GET

REQUEST

```
/api/v1/user/sg
```

RESPONSE

```
[{
    "id": 0,
    "name": "admin",
    "desc": ""
}]
```

/api/v1/user/profile

Returns user effective profile. Effective profile is composition of all profiles with user security groups access.

GET

EDS API v1

```
/api/v1/user/profile
```

RESPONSE

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive" version="19">
<Root class_id="0" tracking_level="1" version="0" object_id="_0">
        <mValue></mValue>
        <mEditable>1</mEditable>
        <mCanHaveExtraSubNodes>1</mCanHaveExtraSubNodes>
        <mSubNodes class_id="1" tracking_level="0" version="0">
                <count>0</count>
                <item_version>0</item_version>
        </mSubNodes>
</Root>
</boost_serialization>
```

/api/v1/users/query

Query users matching selected criteria. Requires admin SG.
The "order" parameter should be a list of fields names, for example: ["created", "-id"]. Adding "-" before a field name reverses the order.

POST

REQUEST

```
{
  "filters": [{              // (optional) filters array joined by OR
    "id": [1, 2],            // (optional) id array
    "nameRe": "^abc"         // (optional) name regular expression
  }],
  "order": ["name"],         // (optional) order, see description for more details
  "fields": ["id", "name"],  // (optional) list of fields returned in response
  "page": 2,                 // (optional) page number, default: 1
  "pagesize": 50             // (optional) results per page, default: 100, max: 1000
}
```

SCHEMA

```
{
  "type": "object",
  "properties": {
    "filters": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {"type": "ar
          "nameRe": {"type":
        }
      }
    },
    "order": {"type": "array
    "page": {"type": "intege
    "pagesize": {"type": "in
    "fields": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": ["id", "name
      }
    }
  }
}
```

RESPONSE

```
{
  "results": [{
    "id": 1,                 // (optional) id
    "name": "user1",         // (optional) name
    "description": "abc",    // (optional) description
    "locked ": false,        // (optional) is locked
    "created": 1603305950,   // (optional) creation timestamp
    "modified": 1603305959,  // (optional) modification timestamp
    "sg": [0, 1]             // (optional) security groups
  }],
  "matchCount": 1,           // match count
  "totalCount": 100,         // total count
}
```

/api/v1/status

Returns information about current server status

EDS API v1

REQUEST

```
/api/v1/status
```

RESPONSE

```
{
  "time": 1641550268,                                 // server time
  "timezone_offset": 3600,                            // server timezone offset [s]
  "srv_connection": "LOGGED_IN | SYNCHRONIZED
| STATIC_CHANGED | DYNAMIC_CHANGED | UPDATE_CYCLE",   // online server connection
  "object_srv_connection": "CONNECTED",               // object server connection
  "archive_srv_connection": "CONNECTED",              // archive server connection
  "report_srv_connection": "CONNECTED",               // report server connection
  "objects_count": 4,                                 // objects count
  "objects_pending_count": 0,                         // pending objects count
  "http_connections": 1,                              // API HTTP connection count
  "https_connections": 0,                             // API HTTPS connection count
  "soap_http_connections": 0,                         // SOAP HTTP connection count
  "soap_https_connections": 0,                        // SOAP HTTPS connection count
  "live_data_connections": 0,                         // live data connection count
  "session_count": 1,                                 // session count
  "request_count": 0,                                 // request count
  "request_running_count": 0                          // running request count
}
```

/api/v1/license

Returns EDS server license information

GET

REQUEST

```
/api/v1/license
```

RESPONSE

```
{
  "AllowedNetworkAddresses": "0.0.0.0/0",
  "ClientType:DBA": "0.0.0.0/0",
  "ClientType:TERM": "0.0.0.0/0",
  "Clustering": "Yes",
  "DatabaseName": "eds",
  "ExpiryDate": "unlimited",
  "ImportAlarms": "Yes",
  "LicenseType": "Commercial",
  "LocalAlarms": "Yes",
  "MSExcelPluginEnabled": "Yes",
  "MaxArchivedPoints": "200000",
  "MaxClients": "128",
  "MaxControlDiagrams": "50000",
  "MaxDiagramSources": "10",
  "MaxPoints": "200000",
  "MaxProcessDiagrams": "50000",
  "MaxReports": "5000",
  "MaxWEBApiSessions": "unlimited",
  "MaxZDs": "20",
  "MaxZIPs": "50000",
  "MobileLicenseExpiryDate": "19/07/2022",
  "MobileLicenseMultiServer": "No",
  "MultiServerConnectivity": "Yes",
  "NotificationServiceEnabled": "No",
  "OneWayScanners": "Yes",
  "OnlineCalculationsEnabled": "Yes",
  "Product": "EnterpriseServer",
  "ReportsEnabled": "Yes",
  "TabularTrendsEnabled": "Yes",
  "Vendor": "Transition Technologies S.A.",
  "WEBApiSupportsGraphics": "Yes",
  "WinNTOwner": "xxx",
  "WinNTSerialNbr": "xxxxx-xxxxx-xxxxx-xxxxx"
}
```

Errors

EDS API v1

Response body will contain JSON with following structure:

```
Error body

{
  "error": 1,                    // error code
  "message": "License invalid"  // error message
}
```

- 0 -
- 1 - License invalid
- 2 - License expired
- 3 - Mobile license invalid
- 4 - Mobile license expired
- 5 - License max session reached
- 6 - Object server not connected
- 7 - Authentication error
- 8 - Authentication timout
- 9 - User not in 'webapi' nor 'admin' group
- 10 - Cannot remove own permission
- 11 - Invalid request ID
- 30 - Points' data not synchronized yet.
- 31 - Point not found
- 32 - Invalid point type
- 33 - Invalid point value
- 34 - No point alter permission
- 35 - Cannot operate point
- 36 - No operate permission
- 37 - Operate point error
- 400 - Bad request
- 401 - Unauthorized
- 403 - Access denied
- 404 - Not found
- 500 - Internal server error