# System Time/NTP/PTP configuration module v1.0

Created Jan 16, 2024 at 02:01 PM

# Resource: current_time

Current system time

http://{device}/api/mgmt.time/1.0/now

**JSON**

```
{
  "is_dst": boolean,
  "timestamp": integer,
  "tz_name": string,
  "utc_offset": integer
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *current_time* | *<object>* | Current system time | |
| *current_time*.is_dst | *<boolean>* | Daylight saving status | Read-only; Optional; |
| *current_time*.timestamp | *<integer>* | Current system time, in Unix epoch seconds | Optional; |
| *current_time*.tz_name | *<string>* | Time zone abbreviation, e.g., PST | Read-only; Optional; |
| *current_time*.utc_offset | *<integer>* | The UTC offset in seconds | Read-only; Optional; |

## Links

### current_time: get

GET http://{device}/api/mgmt.time/1.0/now

Response Body
  Returns a current_time data object.

### current_time: set

PUT http://{device}/api/mgmt.time/1.0/now

Request Body
  Provide a current_time data object.

Response Body
  Returns a current_time data object.

# Resource: ntp_server

An NTP server

http://{device}/api/mgmt.time/1.0/ntp/servers/items/{server_id}

**JSON**

```
{
  "address": string,
  "encryption": string,
  "key_id": integer,
  "prefer": boolean,
  "secret": string,
  "server_id": integer,
  "version": integer
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *ntp_server* | *<object>* | An NTP server | |
| *ntp_server*.address | *<string>* | NTP server address | Optional; |
| *ntp_server*.encryption | *<string>* | Encryption method to use | Optional; Values: none, sha1, md5; |

| | | | |
|---|---|---|---|
| *ntp_server*.key_id | *<integer>* | Encryption key ID | Optional; Range: 0 to 65534; |
| *ntp_server*.prefer | *<boolean>* | Prefer this server | Optional; |
| *ntp_server*.secret | *<string>* | Encryption secret, if encryption is enabled | Optional; |
| *ntp_server*.server_id | *<integer>* | Unique NTP server ID | Read-only; Optional; |
| *ntp_server*.version | *<integer>* | NTP version | Optional; Range: 1 to 4; |
| *ntp_server*.oneOf[0] | *<object>* | | Required properties: [address, encryption, prefer, version]; |
| *ntp_server*.oneOf[0].encryption | *<string>* | | Values: none; |
| *ntp_server*.oneOf[0].<prop> | *<any>* | | Optional; |
| *ntp_server*.oneOf[1] | *<object>* | | Required properties: [address, encryption, key_id, secret, prefer, version]; |
| *ntp_server*.oneOf[1].encryption | *<string>* | | Values: sha1, md5; |
| *ntp_server*.oneOf[1].<prop> | *<any>* | | Optional; |

## Links

### ntp_server: delete

DELETE http://{device}/api/mgmt.time/1.0/ntp/servers/items/{server_id}

Response Body
On success, the server does not provide any body in the responses.

### ntp_server: get

GET http://{device}/api/mgmt.time/1.0/ntp/servers/items/{server_id}

Response Body
Returns a [ntp_server](#) data object.

### ntp_server: set

PUT http://{device}/api/mgmt.time/1.0/ntp/servers/items/{server_id}

Request Body
Provide a [ntp_server](#) data object.

Response Body
Returns a [ntp_server](#) data object.

## Relations

### ntp_server: instances

List of all configured NTP servers

Related resource
[ntp_servers](#)

## Resource: ntp_servers

Configured NTP servers

http://{device}/api/mgmt.time/1.0/ntp/servers

**JSON**

```
{
  "items": [
    {
      "address": string,
      "encryption": string,
      "key_id": integer,
      "prefer": boolean,
      "secret": string,
      "server_id": integer,
      "version": integer
    }
  ],
  <prop>: any
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| ntp_servers | <object> | Configured NTP servers | |
| ntp_servers.items | <array of <items>> | List of configured NTP servers | Optional; |
| items | <object> | An NTP server | |
| items.address | <string> | NTP server address | Optional; |
| items.encryption | <string> | Encryption method to use | Optional; Values: none, sha1, md5; |
| items.key_id | <integer> | Encryption key ID | Optional; Range: 0 to 65534; |
| items.prefer | <boolean> | Prefer this server | Optional; |
| items.secret | <string> | Encryption secret, if encryption is enabled | Optional; |
| items.server_id | <integer> | Unique NTP server ID | Read-only; Optional; |
| items.version | <integer> | NTP version | Optional; Range: 1 to 4; |
| items.oneOf[0] | <object> | | Required properties: [address, encryption, prefer, version]; |
| items.oneOf[0].encryption | <string> | | Values: none; |
| items.oneOf[0].<prop> | <any> | | Optional; |
| items.oneOf[1] | <object> | | Required properties: [address, encryption, key_id, secret, prefer, version]; |
| items.oneOf[1].encryption | <string> | | Values: sha1, md5; |
| items.oneOf[1].<prop> | <any> | | Optional; |
| ntp_servers.<prop> | <any> | | Optional; |

# Links

## ntp_servers: create

POST http://{device}/api/mgmt.time/1.0/ntp/servers

Request Body
Provide a ntp_server data object.

Response Body
Returns a ntp_server data object.

## ntp_servers: get

GET http://{device}/api/mgmt.time/1.0/ntp/servers

Response Body
Returns a ntp_servers data object.

# Resource: ntp_status

Status of configured NTP servers

http://{device}/api/mgmt.time/1.0/ntp/status

JSON

```
[
  {
    "active_sync": boolean,
    "auth": boolean,
    "conf": boolean,
    "delay": number,
    "jitter": number,
    "key_id": integer,
    "offset": number,
    "refid": string,
    "remote_peer": string
  }
]
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *ntp_status* | *<array of <object>>* | Status of configured NTP servers | |
| *ntp_status*[items] | *<object>* | Status of a single NTP server | Read-only; |
| *ntp_status*[items].active_sync | *<boolean>* | This is the active NTP server. Only one may be active at a time. | Read-only; Optional; |
| *ntp_status*[items].auth | *<boolean>* | This peer has been authenticated | Read-only; Optional; |
| *ntp_status*[items].conf | *<boolean>* | True if this peer was configured. False indicates it was learned. | Read-only; Optional; |
| *ntp_status*[items].delay | *<number>* | Roundtrip time to source, in milliseconds | Read-only; Optional; |
| *ntp_status*[items].jitter | *<number>* | The difference between two samples, in milliseconds | Read-only; Optional; |
| *ntp_status*[items].key_id | *<integer>* | Encryption key id | Read-only; Optional; Range: 0 to 65534; |
| *ntp_status*[items].offset | *<number>* | Time difference between client and source, in milliseconds | Read-only; Optional; |
| *ntp_status*[items].refid | *<string>* | Remote peer's reference ID | Read-only; Optional; |
| *ntp_status*[items].remote_peer | *<string>* | Hostname/IP address of the remote NTP peer | Read-only; Optional; |

## Links

## ntp_status: get

```
GET http://{device}/api/mgmt.time/1.0/ntp/status
```

Response Body
   Returns a [ntp_status](#) data object.

# Resource: ptp_config

PTP configuration

```
http://{device}/api/mgmt.time/1.0/ptp/config
```

**JSON**

```
{
  "interface": string,
  "transport": string
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *ptp_config* | *<object>* | PTP configuration | Required properties: [interface, transport]; |
| *ptp_config*.interface | *<string>* | Name of the interface to use for PTP synchronization | |
| *ptp_config*.transport | *<string>* | PTP transport type | Values: ipv4, ipv6, ethernet; |

## Links

## ptp_config: get

```
GET http://{device}/api/mgmt.time/1.0/ptp/config
```

Response Body
Returns a [ptp_config](#) data object.

## ptp_config: set

```
PUT http://{device}/api/mgmt.time/1.0/ptp/config
```

Request Body
Provide a [ptp_config](#) data object.

Response Body
Returns a [ptp_config](#) data object.

# Resource: ptp_status

PTP status

```
http://{device}/api/mgmt.time/1.0/ptp/status
```

**JSON**

```json
{
  "active_sync": boolean,
  "offset": number,
  "server": string
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *ptp_status* | *<object>* | PTP status | Read-only; |
| *ptp_status*.active_sync | *<boolean>* | PTP is synchronized with the master | Read-only; Optional; |
| *ptp_status*.offset | *<number>* | PTP offset, in milliseconds, from the master | Read-only; Optional; |
| *ptp_status*.server | *<string>* | Server address (IPv4/IPv6/MAC address) | Read-only; Optional; |

## Links

## ptp_status: get

```
GET http://{device}/api/mgmt.time/1.0/ptp/status
```

Response Body
Returns a [ptp_status](#) data object.

# Resource: time_configuration

Time source configuration

```
http://{device}/api/mgmt.time/1.0/configuration
```

**JSON**

```json
{
  "mode": string,
  <prop>: any
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *time_configuration* | *<object>* | Time source configuration | |
| *time_configuration*.mode | *<string>* | Mode for automatic clock synchronization | Optional; Values: none, ntp, ptp; |
| *time_configuration*.<prop> | *<any>* | | Optional; |

## Links

### time_configuration: get

```
GET http://{device}/api/mgmt.time/1.0/configuration
```

Response Body
Returns a [time_configuration](#) data object.

### time_configuration: set

```
PUT http://{device}/api/mgmt.time/1.0/configuration
```

Request Body
Provide a [time_configuration](#) data object.

Response Body
Returns a [time_configuration](#) data object.

# Resource: time_set_event

The time has been changed. This event will only be sent when the system is on local clock.

```
http://{device}/api/mgmt.time/1.0/event_queues/time_set_events/{id}
```

**JSON**

```json
{
  "hdr": event_header,
  "new_timestamp": integer,
  "old_timestamp": integer
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *time_set_event* | *<object>* | The time has been changed. This event will only be sent when the system is on local clock. | Required properties: [hdr, old_timestamp, new_timestamp]; |
| *time_set_event*.hdr | *<event_header>* | Common properties for all events | |
| *time_set_event*.new_timestamp | *<integer>* | New system time, in Unix epoch time | |
| *time_set_event*.old_timestamp | *<integer>* | Previous system time, in Unix epoch time | |

## Links

### time_set_event: get

```
GET http://{device}/api/mgmt.time/1.0/event_queues/time_set_events/{id}
```

Response Body
Returns a [time_set_event](#) data object.

## Relations

### time_set_event: instances

Related resource
[time_set_events](#)

# Resource: time_set_events

All time change events that have occurred

```
http://{device}/api/mgmt.time/1.0/event_queues/time_set_events
```

**JSON**

```
{
  "queue": [ time_set_event ],
  <prop>: any
}
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *time_set_events* | *<object>* | All time change events that have occurred | |
| *time_set_events*.queue | *<array of <time_set_event>>* | The time change events | Optional; |
| *time_set_events*.queue[items] | *<time_set_event>* | The time has been changed. This event will only be sent when the system is on local clock. | |
| *time_set_events*.<prop> | *<any>* | | Optional; |

## Links

## time_set_events: get

```
GET http://{device}/api/mgmt.time/1.0/event_queues/time_set_events
```

Response Body
  Returns a time_set_events data object.

# Resource: timezone

The configured timezone

```
http://{device}/api/mgmt.time/1.0/timezone
```

**JSON**

```
string
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *timezone* | *<string>* | The configured timezone | |

## Links

## timezone: get

```
GET http://{device}/api/mgmt.time/1.0/timezone
```

Response Body
  Returns a timezone data object.

## timezone: set

```
PUT http://{device}/api/mgmt.time/1.0/timezone
```

Request Body
  Provide a timezone data object.

Response Body
  Returns a timezone data object.

# Resource: timezones

The list of known timezones

http://{device}/api/mgmt.time/1.0/timezones

**JSON**

```
[
  string
]
```

| Property Name | Type | Description | Notes |
|---|---|---|---|
| *timezones* | *<array of <string>>* | The list of known timezones | |
| *timezones*[items] | *<string>* | A timezone | |

## Links

## timezones: get

GET http://{device}/api/mgmt.time/1.0/timezones

Response Body
  Returns a <u>timezones</u> data object.