Resource: capabilities

The capabilities of an interface

http://{device}/api/mgmt.networking/1.1/interfaces/{name}/capabilities

### JSON

```json
{
    "autoneg": boolean,
    "speed_duplex": [
        string
    ],
    <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>capabilities</td>
<td>&lt;object&gt;</td>
<td>The capabilities of an interface</td>
<td></td>
</tr>
<tr>
<td>capabilities.autoneg</td>
<td>&lt;boolean&gt;</td>
<td>Whether Auto-negotiation is available</td>
<td>Optional;</td>
</tr>
<tr>
<td>capabilities.speed_duplex</td>
<td>&lt;array of &lt;string&gt;&gt;</td>
<td>The speed and duplex combinations available on this interface</td>
<td>Optional;</td>
</tr>
<tr>
<td>capabilities.speed_duplex[items]</td>
<td>&lt;string&gt;</td>
<td>A speed and duplex combination</td>
<td></td>
</tr>
<tr>
<td>capabilities &lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

### Links

capabilities: get

GET http://{device}/api/mgmt.networking/1.1/interfaces/{name}/capabilities

Response Body

Returns a capabilities data object.

Resource: host_settings

General networking setting for the host machine, including the hostname, name servers, and domain names.

http://{device}/api/mgmt.networking/1.1/settings/host

### JSON

```json
{
    "dns_domains": [ string ],
    "dns_servers": [ string ],
    "hostname": string,
    "hosts": [
        { "hostname": string, "ip_address": string, <prop>: any }
    ],
    <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_settings</td>
<td>&lt;object&gt;</td>
<td>General networking setting for the host machine, including the hostname, name servers, and domain names.</td>
<td></td>
</tr>
<tr>
<td>host_settings.dns_domains</td>
<td>&lt;array of &lt;string&gt;&gt;</td>
<td>Ordered list of domain names to append when attempting hostname resolution. If DHCP is enabled, these domains will be prepended to any returned by the DHCP server.</td>
<td>Optional;</td>
</tr>
<tr>
<td>host_settings.dns_domains[items]</td>
<td>&lt;string&gt;</td>
<td>A single domain name</td>
<td></td>
</tr>
</tbody>
</table>
**host_settings: get**

GET http://{device}/api/mgmt.networking/1.1/settings/host

Response Body

Returns a host_settings data object.

**host_settings: set**

PUT http://{device}/api/mgmt.networking/1.1/settings/host

Request Body

Provide a host_settings data object.

Response Body

Returns a host_settings data object.

**Resource: host_state**

The host settings applied to the system, both from values configured in host_settings as well as provided by DHCP, if enabled

http://{device}/api/mgmt.networking/1.1/host_state

**JSON**

```
{
    "dns_domains": [string],
    "dns_servers": [string],
    "ipv4_default_gateway": string,
    "ipv6_default_gateway": string
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_state</td>
<td>&lt;object&gt;</td>
<td>The host settings applied to the system, both from values configured in host_settings as well as provided by DHCP, if enabled</td>
<td>Required properties: [ipv4 default_gateway, ipv6_default_gateway, dns_servers, dns_domains];</td>
</tr>
<tr>
<td>host_state.dns_domains</td>
<td>&lt;array of &lt;string&gt;&gt;</td>
<td>Ordered list of domain names to append when attempting hostname resolution.</td>
<td></td>
</tr>
<tr>
<td>host_state.dns_domains[items]</td>
<td>&lt;string&gt;</td>
<td>A single domain name</td>
<td></td>
</tr>
<tr>
<td>host_state.dns_servers</td>
<td>&lt;array of &lt;string&gt;&gt;</td>
<td>Ordered list of DNS server IP addresses</td>
<td></td>
</tr>
<tr>
<td>host_state.dns_servers[items]</td>
<td>&lt;string&gt;</td>
<td>A single DNS server IP address</td>
<td></td>
</tr>
<tr>
<td>host_state.ipv4_default_gateway</td>
<td>&lt;string&gt;</td>
<td>The IPv4 default gateway</td>
<td></td>
</tr>
<tr>
<td>host_state.ipv6_default_gateway</td>
<td>&lt;string&gt;</td>
<td>The IPv6 default gateway</td>
<td></td>
</tr>
</tbody>
</table>
**host_state: get**

GET http://{device}/api/mgmt.networking/1.1/host_state

Response Body
Returns a host_state data object.

**Resource: interface**

Configuration and current status of a management interface

http://{device}/api/mgmt.networking/1.1/interfaces/{name}

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>&lt;object&gt;</td>
<td>Configuration and current status of a management interface</td>
<td></td>
</tr>
<tr>
<td>interface.configuration</td>
<td>&lt;configuration&gt;</td>
<td>Interface configuration</td>
<td></td>
</tr>
<tr>
<td>interface.name</td>
<td>&lt;string&gt;</td>
<td>Interface name</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>interface.state</td>
<td>&lt;state&gt;</td>
<td>Current interface state</td>
<td>Read-only;</td>
</tr>
<tr>
<td>interface.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**interface: get**

GET http://{device}/api/mgmt.networking/1.1/interfaces/{name}

Response Body
Returns an interface data object.

**interface: renew**

Renew all DHCP leases for this interface. Will return an error if DHCP is not enabled.

POST http://{device}/api/mgmt.networking/1.1/interfaces/{name}/renew_dhcp

Request Body
Provide a request body with the following structure:

```json

```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface.links.renew.request</td>
<td>&lt;number&gt;</td>
<td>DHCP version to renew</td>
<td>Values: 4, 6;</td>
</tr>
</tbody>
</table>

Response Body
On success, the server does not provide any body in the responses.
**interface: set**

```
PUT http://{device}/api/mgmt.networking/1.1/interfaces/{name}
```

**Request Body**

Provide an `interface` data object.

**Response Body**

Returns an `interface` data object.

---

**Relations**

**interface: capabilities**

The capabilities of this interface

**Related resource**

`capabilities`

**Variables**

<table>
<thead>
<tr>
<th>Related var</th>
<th>Data value for replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>0/name</td>
</tr>
</tbody>
</table>

**interface: instances**

All available management interfaces

**Related resource**

`interface`

---

**Resource: interfaces**

Management interfaces available on this device

```
http://{device}/api/mgmt.networking/1.1/interfaces
```

**JSON**

```
[
  {
    string
  }
]
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>interfaces</td>
<td><code>&lt;array of string&gt;</code></td>
<td>Management interfaces available on this device</td>
<td></td>
</tr>
<tr>
<td>interfaces[items]</td>
<td><code>&lt;string&gt;</code></td>
<td>The name of an available management interface</td>
<td></td>
</tr>
</tbody>
</table>

---

**Links**

**interfaces: get**

```
GET http://{device}/api/mgmt.networking/1.1/interfaces
```

**Response Body**

Returns an `interfaces` data object.

---

**Resource: ipv4_route**

A single IPv4 route

```
http://{device}/api/mgmt.networking/1.1/routes/ipv4/{id}
```
**Property Name** | **Type** | **Description** | **Notes**
---|---|---|---
ipv4_route | <object> | A single IPv4 route | Required properties: [network_prefix, gateway_address, interface];
ipv4_route.gateway_address | <string> | Gateway IP address | 
ipv4_route.id | <number> | Route ID | Read-only; Optional;
ipv4_route.interface | <string> | Network interface. At least one of “gateway_address” or “interface needs” to be specified. | 
ipv4_route.network_prefix | <string> | Destination network prefix. A string consisting of an IPv4 address and a mask, separated by a slash (/). For example, "192.168.1.0/24" or "192.168.1.0/255.255.255.0" | 
ipv4_route.<prop> | <any> | Optional; | 

**Links**

**ipv4_route: delete**

DELETE http://{device}/api/mgmt.networking/1.1/routes/ipv4/{id}

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

**ipv4_route: get**

GET http://{device}/api/mgmt.networking/1.1/routes/ipv4/{id}

Response Body
Returns an ipv4_route data object.

**ipv4_route: set**

PUT http://{device}/api/mgmt.networking/1.1/routes/ipv4/{id}

Request Body
Provide an ipv4_route data object.

Response Body
Returns an ipv4_route data object.

**Relations**

**ipv4_route: instances**
All IPv4 routes

Related resource
ipv4_routes

**Resource: ipv4_routes**
Configured IPv4 routes
http://{device}/api/mgmt.networking/1.1/routes/ipv4

JSON

```json
{
  "all": [ route_type ],
  "static": [
    {
      "gateway_address": string,
      "id": number,
      "interface": string,
      "network_prefix": string,
      <prop>: any
    }
  ],
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4_routes</td>
<td>&lt;object&gt;</td>
<td>Configured IPv4 routes</td>
<td></td>
</tr>
<tr>
<td>ipv4_routes.all</td>
<td>&lt;array of &lt;route_type&gt;&gt;</td>
<td>All configured IPv4 routes, including default gateway.</td>
<td>Optional;</td>
</tr>
<tr>
<td>ipv4_routes.all.items</td>
<td>&lt;route_type&gt;</td>
<td>A route</td>
<td>Optional;</td>
</tr>
<tr>
<td>items</td>
<td>&lt;object&gt;</td>
<td>A single IPv4 route</td>
<td>Required properties: [network_prefix, gateway_address, interface];</td>
</tr>
<tr>
<td>Items.gateway_address</td>
<td>&lt;string&gt;</td>
<td>Gateway IP address</td>
<td></td>
</tr>
<tr>
<td>items.id</td>
<td>&lt;number&gt;</td>
<td>Route ID</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>items.interface</td>
<td>&lt;string&gt;</td>
<td>Network interface. At least one of &quot;gateway_address&quot; or &quot;interface needs&quot; to be specified.</td>
<td></td>
</tr>
<tr>
<td>items.network_prefix</td>
<td>&lt;string&gt;</td>
<td>Destination network prefix. A string consisting of an IPv4 address and a mask, separated by a slash (/). For example, &quot;192.168.1.0/24&quot; or &quot;192.168.1.0/255.255.255.0&quot;</td>
<td></td>
</tr>
<tr>
<td>ipv4_routes.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Links**

**ipv4_routes: create**

POST http://{device}/api/mgmt.networking/1.1/routes/ipv4

Request Body

Provide an `ipv4_route` data object.

Response Body

Returns an `ipv4_route` data object.

**ipv4_routes: get**

GET http://{device}/api/mgmt.networking/1.1/routes/ipv4

Response Body

Returns an `ipv4_routes` data object.

**Resource: ipv6_route**

A single IPv6 route

http://{device}/api/mgmt.networking/1.1/routes/ipv6/{id}

JSON
**ipv6_route: delete**

DELETE http://{device}/api/mgmt.networking/1.1/routes/ipv6/{id}

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

**ipv6_route: get**

GET http://{device}/api/mgmt.networking/1.1/routes/ipv6/{id}

**Response Body**
Returns an `ipv6_route` data object.

**ipv6_route: set**

PUT http://{device}/api/mgmt.networking/1.1/routes/ipv6/{id}

**Request Body**
Provide an `ipv6_route` data object.

**Response Body**
Returns an `ipv6_route` data object.

**Relations**

**ipv6_route: instances**
All IPv6 routes

**Related resource**
ipv6_routes

**Resource: ipv6_routes**
Configured IPv6 routes

http://{device}/api/mgmt.networking/1.1/routes/ipv6

JSON
**Property Name** | **Type** | **Description** | **Notes**
---|---|---|---
ipv6_routes | <object> | Configured IPv6 routes | 
ipv6_routes.all | <array of <route_type>> | All configured IPv6 routes | Optional; 
ipv6_routes.all[items] | <route_type> | A route | 
ipv6_routes.static | <array of <items>> | Static, manually configured IPv6 routes. | Optional; 
items | <object> | A single IPv6 route | Required properties: [network_prefix, gateway_address, interface]; 
items.gateway_address | <string> | Gateway IP address | 
items.id | <number> | Route ID | Read-only; Optional; 
items.interface | <string> | Network interface. At least one of "gateway_address" or "interface needs" to be specified. | 
items.network_prefix | <string> | Destination network prefix. A string consisting of an IP address and prefix, separated by a slash (/). | 
ipv6_routes.<prop> | <any> | | Optional; 
ipv6_routes.<prop> | <any> | | Optional;

**Links**

**ipv6_routes: create**

**POST** http://{device}/api/mgmt.networking/1.1/routes/ipv6

Request Body

Provide an ipv6_route data object.

Response Body

Returns an ipv6_route data object.

**ipv6_routes: get**

**GET** http://{device}/api/mgmt.networking/1.1/routes/ipv6

Response Body

Returns an ipv6_routes data object.

**Resource: route_settings**

Global route settings

http://{device}/api/mgmt.networking/1.1/settings/route

JSON
```
{
  "default_gateway": {
    "ipv4": string,
    "ipv6": string,
    <prop>: any
  },
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_settings</td>
<td>&lt;object&gt;</td>
<td>Global route settings</td>
<td></td>
</tr>
<tr>
<td>route_settings.default_gateway</td>
<td>&lt;object&gt;</td>
<td>The default gateway configuration for the host. Ignored if DHCP is enabled. The state can be verified by quering the &quot;ipv4_routes&quot; or &quot;ipv6_routes&quot; resources.</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_settings.default_gateway.ipv4</td>
<td>&lt;string&gt;</td>
<td>IPv4 address for default gateway. Blank string unsets it.</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_settings.default_gateway.ipv6</td>
<td>&lt;string&gt;</td>
<td>IPv6 address for default gateway. Blank string unsets it.</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_settings.default_gateway.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>route_settings.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Links**

**route_settings: get**

```
GET http://{device}/api/mgmt.networking/1.1/settings/route
```

**Response Body**

Returns a `route_settings` data object.

**route_settings: set**

```
PUT http://{device}/api/mgmt.networking/1.1/settings/route
```

**Request Body**

Provide a `route_settings` data object.

**Response Body**

Returns a `route_settings` data object.

**Type: configuration**

Interface configuration

```json```
```json
{
  "autoneg": multiple,
  "description": string,
  "duplex": multiple,
  "enable": boolean,
  "ipv4": {
    "address": string,
    "dhcp": boolean,
    "dynamic_dns": boolean,
    "subnet_mask": string,
    <prop>: any
  },
  "ipv6": {
    "address": string,
    "dhcp": boolean,
    "dynamic_dns": boolean,
    "prefix": string,
    <prop>: any
  },
  "mtu": number,
  "speed": multiple,
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration</td>
<td>&lt;object&gt;</td>
<td>Interface configuration</td>
<td></td>
</tr>
<tr>
<td>configuration.autoneg</td>
<td>&lt;multiple&gt;</td>
<td>Auto-negotiation enabled</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.autoneg.anyOf[0]</td>
<td>&lt;boolean&gt;</td>
<td>Auto-negotiation enabled</td>
<td></td>
</tr>
<tr>
<td>configuration.autoneg.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Not configured, or does not apply to this interface</td>
<td></td>
</tr>
<tr>
<td>configuration.description</td>
<td>&lt;string&gt;</td>
<td>Interface description</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.duplex</td>
<td>&lt;multiple&gt;</td>
<td>Interface duplex setting. Ignored if “autoneg” is True.</td>
<td>Values: full, half;</td>
</tr>
<tr>
<td>configuration.duplex.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Interface duplex setting. Ignored if “autoneg” is True.</td>
<td></td>
</tr>
<tr>
<td>configuration.duplex.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Not configured, or does not apply to this interface</td>
<td></td>
</tr>
<tr>
<td>configuration.enable</td>
<td>&lt;boolean&gt;</td>
<td>Interface enabled</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4</td>
<td>&lt;object&gt;</td>
<td>Set the IPv4 address of the interface. This field is validated but not applied if DHCP is enabled. To disable an interface IP, set address to 0.0.0.0.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4.address</td>
<td>&lt;string&gt;</td>
<td>Set the IPv4 address of the interface. This field is validated but not applied if DHCP is enabled. To disable an interface IP, set address to 0.0.0.0.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4.dhcp</td>
<td>&lt;boolean&gt;</td>
<td>Obtain IP address via DHCP</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4.dynamic_dns</td>
<td>&lt;boolean&gt;</td>
<td>Send hostname with the DHCP request</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4_subnet_mask</td>
<td>&lt;string&gt;</td>
<td>Set the subnet mask of the interface. This field is validated but not applied if DHCP is enabled. Cleared if address is set to 0.0.0.0.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv4.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6</td>
<td>&lt;object&gt;</td>
<td>Set the IPv6 address of the interface. This field is validated but not applied if DHCP is enabled. Remove the IPv6 address by setting this field to an empty string.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6.address</td>
<td>&lt;string&gt;</td>
<td>Set the IPv6 address of the interface. This field is validated but not applied if DHCP is enabled. Remove the IPv6 address by setting this field to an empty string.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6.dhcp</td>
<td>&lt;boolean&gt;</td>
<td>Obtain IP address via DHCP</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6.dynamic_dns</td>
<td>&lt;boolean&gt;</td>
<td>Send hostname with the DHCP request</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6.prefix</td>
<td>&lt;string&gt;</td>
<td>The IPv6 address prefix. This field is validated but not applied if DHCP is enabled, or the address is not set.</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.ipv6.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.mtu</td>
<td>&lt;number&gt;</td>
<td>Interface MTU</td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.speed</td>
<td>&lt;multiple&gt;</td>
<td>Speed to force this interface to run at. Ignored if “autoneg” is True.</td>
<td>Values: 10, 100, 1000, 2500, 10000;</td>
</tr>
<tr>
<td>configuration.speed.anyOf[0]</td>
<td>&lt;number&gt;</td>
<td>Speed to force this interface to run at. Ignored if “autoneg” is True.</td>
<td>Values: 10, 100, 1000, 2500, 10000;</td>
</tr>
<tr>
<td>configuration.speed.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Not configured, or does not apply to this interface</td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Type: route_type**

A route

**JSON**
{  "gateway_address": string,  "interface": string,  "network_prefix": string,  "<prop>": any  }

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_type</td>
<td>&lt;object&gt;</td>
<td>A route</td>
<td></td>
</tr>
<tr>
<td>route_type.gateway_address</td>
<td>&lt;string&gt;</td>
<td>Gateway IP address</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_type.interface</td>
<td>&lt;string&gt;</td>
<td>Network interface. At least one of &quot;gateway_address&quot; or &quot;interface&quot; needs to be specified.</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_type.network_prefix</td>
<td>&lt;string&gt;</td>
<td>Destination network prefix. A string consisting of an IP address and a mask, separated by a slash (/). For example, &quot;192.168.1.0/24&quot; or &quot;192.168.1.0/255.255.255.0&quot;. For ipv6 specify prefix after slash (/).</td>
<td>Optional;</td>
</tr>
<tr>
<td>route_type.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

### Type: state

Current interface state

```json
{  "<autoneg>": multiple,  "<duplex>": multiple,  "<enabled>": boolean,  "<hw_address>": string,  "<if_index>": number,  "<interface_type>": string,  "<ipv4>": {    "<address>": string,    "<subnet_mask>": string,    "<prop>": any  },  "<ipv6>": {    "<address>": string,    "<link_local>": string,    "<prop>": any  },  "<link>": boolean,  "<mtu>": number,  "<speed>": multiple,  "<prop>": any  }
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>&lt;object&gt;</td>
<td>Current interface state</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.autoneg</td>
<td>&lt;multiple&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.autoneg.anyOf[0]</td>
<td>&lt;boolean&gt;</td>
<td>Auto-negotiation status</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.autoneg.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Auto-negotiation is not relevant for this interface</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.duplex</td>
<td>&lt;multiple&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.duplex.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Interface duplex</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.duplex.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Duplex is not relevant for this interface</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.enabled</td>
<td>&lt;boolean&gt;</td>
<td>Interface enabled/up</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.hw_address</td>
<td>&lt;string&gt;</td>
<td>Interface hardware address</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.if_index</td>
<td>&lt;number&gt;</td>
<td>Interface index</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.interface_type</td>
<td>&lt;string&gt;</td>
<td>Interface type</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.ipv4</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.ipv4.address</td>
<td>&lt;string&gt;</td>
<td>Interface IPv4 address</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.ipv4.subnet_mask</td>
<td>&lt;string&gt;</td>
<td>Subnet mask</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.ipv4.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.ipv6</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.ipv6.address</td>
<td>&lt;string&gt;</td>
<td>Interface IPv6 address</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
<td>Readability</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>state.ipv6.link_local</td>
<td>&lt;string&gt;</td>
<td>Link local IPv6 address</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.ipv6.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>state.link</td>
<td>&lt;boolean&gt;</td>
<td>Interface link state</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.mtu</td>
<td>&lt;number&gt;</td>
<td>Interface MTU</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.speed</td>
<td>&lt;multiple&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.speed.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Interface speed</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.speed.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Speed is not relevant for this interface</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
</tr>
</tbody>
</table>