Resource: interfaces

http://{device}/api/mgmt.til.networking/1.0/interfaces

JSON

```
{
  "items": [
    string
  ],
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>interfaces</td>
<td>&lt;object&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>interfaces.items</td>
<td>&lt;array of &lt;string&gt;&gt;</td>
<td>Optional;</td>
<td></td>
</tr>
<tr>
<td>interfaces.items.items</td>
<td>&lt;string&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>interfaces.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td>Optional;</td>
<td></td>
</tr>
</tbody>
</table>

Links

**interfaces: get**

GET http://{device}/api/mgmt.til.networking/1.0/interfaces

Response Body

Returns an interfaces data object.

Resource: interface

http://{device}/api/mgmt.til.networking/1.0/interfaces/items/{name}

JSON

```
{
  "name": string,
  "configuration": configuration,
  "state": state,
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>&lt;object&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface.name</td>
<td>&lt;string&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
<td></td>
</tr>
<tr>
<td>interface.configuration</td>
<td>&lt;configuration&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface.state</td>
<td>&lt;state&gt;</td>
<td></td>
<td>Read-only;</td>
<td></td>
</tr>
<tr>
<td>interface.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
<td></td>
</tr>
</tbody>
</table>

Links

**interface: get**

GET http://{device}/api/mgmt.til.networking/1.0/interfaces/items/{name}

Response Body

Returns an interface data object.

**interface: set**
PUT http://{device}/api/mgmt.til.networking/1.0/interfaces/items/{name}

Request Body
Provide an interface data object.

Response Body
Returns an interface data object.

Relations

interface: instances

Related resource
interfaces

Resource: ipv4_routes

http://{device}/api/mgmt.til.networking/1.0/routes/ipv4

Property Name     Type                Description                                                                 Notes
ipv4_routes        <object>            
ipv4_routes.all    <array of          
                    <route_type>>       Optional;
ipv4_routes.all[items] <route_type>   
ipv4_routes.static <array of          
                    <items>>         Required properties: [network_prefix, gateway_address, interface]
items               <object>            
items.id            <string>            Server generated route identifier       Read-only; Optional;
items.network_prefix <string>          Destination Network Prefix. A string consisting of an IP address and a mask, separated by a slash (/). For example 192.168.1.0/24 For IPv6 specify prefix after slash (/).
items.gateway_address <string>          Gateway IP address
items.interface     <string>            Network Interface. At least one of gateway_addresses or interfaces needs to be specified.
items.<prop>        <any>              
ipv4_routes.<prop>  <any>              

Links

ipv4_routes: get

GET http://{device}/api/mgmt.til.networking/1.0/routes/ipv4

Response Body
Returns an ipv4_routes data object.
ipv4_routes: create

POST http://{device}/api/mgmt.til.networking/1.0/routes/ipv4

Request Body
Provide an ipv4_route data object.

Response Body
Returns an ipv4_route data object.

Resource: ipv4_route

http://{device}/api/mgmt.til.networking/1.0/routes/ipv4/{id}

JSON

```
{
  "id": string,
  "network_prefix": string,
  "gateway_address": string,
  "interface": string,
  "<prop>": any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4_route</td>
<td>&lt;object&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipv4_route.id</td>
<td>&lt;string&gt;</td>
<td>Server generated route identifier</td>
<td>Required properties: [network_prefix, gateway_address, interface];</td>
</tr>
<tr>
<td>ipv4_route.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 192.168.1.0/24 For IPv6 specify prefix after slash (/).</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>ipv4_route.gateway_address</td>
<td>&lt;string&gt;</td>
<td>Gateway IP address</td>
<td></td>
</tr>
<tr>
<td>ipv4_route.interface</td>
<td>&lt;string&gt;</td>
<td>Network Interface. At least one of gateway_addresses or interfaces needs to be specified.</td>
<td></td>
</tr>
<tr>
<td>ipv4_route.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

Links

ipv4_route: get

GET http://{device}/api/mgmt.til.networking/1.0/routes/ipv4/{id}

Response Body
Returns an ipv4_route data object.

ipv4_route: delete

Delete a static route

DELETE http://{device}/api/mgmt.til.networking/1.0/routes/ipv4/{id}

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

Relations

ipv4_route: instances

Related resource
ipv4_routes

Resource: ipv6_routes
Get

GET http://{device}/api/mgmt.til.networking/1.0/routes/ipv6

Response Body
Returns an ipv6_routes data object.

Create

POST http://{device}/api/mgmt.til.networking/1.0/routes/ipv6

Request Body
Provide an ipv6_route data object.

Response Body
Returns an ipv6_route data object.

Resource: ipv6_route

http://{device}/api/mgmt.til.networking/1.0/routes/ipv6/{id}


```json
{
  "id": string,
  "network_prefix": string,
  "gateway_address": string,
  "interface": string,
  <prop>: any
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv6_route</td>
<td>&lt;object&gt;</td>
<td>Server generated route identifier</td>
<td>Required properties: [network_prefix, gateway_address, interface];</td>
</tr>
<tr>
<td>ipv6_route.id</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 192.168.1.0/24 For IPv6 specify prefix after slash (/).</td>
<td></td>
</tr>
<tr>
<td>ipv6_route.network_prefix</td>
<td>&lt;string&gt;</td>
<td>Gateway IP address</td>
<td></td>
</tr>
<tr>
<td>ipv6_route.gateway_address</td>
<td>&lt;string&gt;</td>
<td>Network Interface. At least one of gateway_addresses or interfaces needs to be specified.</td>
<td></td>
</tr>
<tr>
<td>ipv6_route.interface</td>
<td>&lt;string&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipv6_route.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>

**Links**

**ipv6_route: get**

GET http://{device}/api/mgmt.til.networking/1.0/routes/ipv6/{id}

Response Body

Returns an `ipv6_route` data object.

**ipv6_route: delete**

Delete a static route

DELETE http://{device}/api/mgmt.til.networking/1.0/routes/ipv6/{id}

Response Body

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

**Relations**

**ipv6_route: instances**

Related resource

`ipv6_routes`

**Resource: route_settings**

This resource configure general route settings.

http://{device}/api/mgmt.til.networking/1.0/settings/route

**JSON**

```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_type</td>
<td>&lt;object&gt;</td>
<td></td>
<td></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 192.168.1.0/24 For IPv6 specify prefix after slash (/).</td>
<td>Optional;</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 gateway_address or interface must be specified.</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: static_route_type**

```json
{
   "id": string,
   "network_prefix": string,
   "gateway_address": string,
   "interface": string,
   <prop>: any
}
```
### Type: state

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

### Type: configuration

<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>Interface enabled/up state</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.enabled</td>
<td>&lt;boolean&gt;</td>
<td>Interface enabled/up state</td>
<td>Read-only; Optional;</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.interface_type</td>
<td>&lt;string&gt;</td>
<td>Type of interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.hw_address</td>
<td>&lt;string&gt;</td>
<td>Hardware address of the interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.if_index</td>
<td>&lt;number&gt;</td>
<td>Interface index of the interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.mtu</td>
<td>&lt;number&gt;</td>
<td>MTU of the interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.speed</td>
<td>&lt;multiple&gt;</td>
<td>Speed of interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.speed.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Speed of interface</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.speed.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Speed is not relevant for interface.</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.duplex</td>
<td>&lt;multiple&gt;</td>
<td>Duplex of interface</td>
<td>Read-only; Optional;</td>
</tr>
<tr>
<td>state.duplex.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Duplex of interface</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.duplex.anyOf[1]</td>
<td>&lt;null&gt;</td>
<td>Duplex is not relevant for interface.</td>
<td>Read-only;</td>
</tr>
<tr>
<td>state.ipv4</td>
<td>&lt;object&gt;</td>
<td>Interface IPv4 address</td>
<td>Read-only; Optional;</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.prefix</td>
<td>&lt;integer&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; Optional;</td>
</tr>
<tr>
<td>state.ipv6</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Read-only; Optional;</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.address</td>
<td>&lt;string&gt;</td>
<td>Interface 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>Read-only; Optional;</td>
</tr>
<tr>
<td>state.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Read-only; Optional;</td>
</tr>
</tbody>
</table>
```json
{
    "enable": boolean,
    "description": string,
    "speed": multiple,
    "duplex": multiple,
    "mtu": number,
    "ipv4": {
        "dhcp": boolean,
        "dynamic_dns": boolean,
        "address": string,
        "prefix": integer,
        <prop>: any
    },
    "ipv6": {
        "dhcp": boolean,
        "dynamic_dns": boolean,
        "address": string,
        "prefix": integer,
        <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>configuration</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Required properties: [enable, description, speed, duplex, mtu, ipv4, ipv6];</td>
</tr>
<tr>
<td>configuration.enable</td>
<td>&lt;boolean&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.description</td>
<td>&lt;string&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.speed</td>
<td>&lt;multiple&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.speed.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Set speed to the interface</td>
<td>Values: auto, 10, 100, 1000, 2500, 10000;</td>
</tr>
<tr>
<td>configuration.duplex</td>
<td>&lt;multiple&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.duplex.anyOf[0]</td>
<td>&lt;string&gt;</td>
<td>Set duplex to the interface</td>
<td>Values: full, half, auto;</td>
</tr>
<tr>
<td>configuration.mtu</td>
<td>&lt;number&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration.ipv4</td>
<td>&lt;object&gt;</td>
<td></td>
<td>Required properties: [dhcp, dynamic_dns, address, prefix];</td>
</tr>
<tr>
<td>configuration.ipv4.dhcp</td>
<td>&lt;boolean&gt;</td>
<td>Obtain IP address from DHCP server.</td>
<td></td>
</tr>
<tr>
<td>configuration.ipv4.dynamic_dns</td>
<td>&lt;boolean&gt;</td>
<td>Send hostname with the DHCP request.</td>
<td></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 set. To disable interface IP, set address to 0.0.0.0</td>
<td></td>
</tr>
<tr>
<td>configuration.ipv4.prefix</td>
<td>&lt;integer&gt;</td>
<td>Set the prefix of the interface.</td>
<td></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></td>
<td>Required properties: [dhcp, dynamic_dns, address, prefix];</td>
</tr>
<tr>
<td>configuration.ipv6.dhcp</td>
<td>&lt;boolean&gt;</td>
<td>Obtain IP address from DHCP server.</td>
<td></td>
</tr>
<tr>
<td>configuration.ipv6.dynamic_dns</td>
<td>&lt;boolean&gt;</td>
<td>Send hostname with the DHCP request.</td>
<td></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 set. To remove IPv6 address set to blank.</td>
<td></td>
</tr>
<tr>
<td>configuration.ipv6.prefix</td>
<td>&lt;integer&gt;</td>
<td>IPv6 address prefix. Ignored if address is set to blank.</td>
<td></td>
</tr>
<tr>
<td>configuration.&lt;prop&gt;</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
<tr>
<td>configuration.</td>
<td>&lt;any&gt;</td>
<td></td>
<td>Optional;</td>
</tr>
</tbody>
</table>