

Management networking configuration v1.1

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

Resource: capabilities

The capabilities of an interface

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

JSON

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

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

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

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

Property Name	Type	Description	Notes
<i>host_settings</i>	<object>	General networking setting for the host machine, including the hostname, name servers, and domain names.	
<i>host_settings.dns_domains</i>	<array of <string>>	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.	Optional;
<i>host_settings.dns_domains[items]</i>	<string>	A single domain name	

<code>host_settings.dns_servers</code>	<code><array of <string>></code>	Ordered list of DNS server IP addresses. A maximum of three servers may be configured. If DHCP is enabled, these servers will be prepended to any returned by the DHCP server.	Optional;
<code>host_settings.dns_servers[items]</code>	<code><string></code>	A single DNS server IP address	
<code>host_settings.hostname</code>	<code><string></code>	Hostname of the device	Optional;
<code>host_settings.hosts</code>	<code><array of <object>></code>	Additional host entries	Optional;
<code>host_settings.hosts[items]</code>	<code><object></code>	A single host to IP mapping	
<code>host_settings.hosts[items].hostname</code>	<code><string></code>	The hostname to associate with this IP address	Optional;
<code>host_settings.hosts[items].ip_address</code>	<code><string></code>	The IP address of the host	Optional;
<code>host_settings.hosts[items].<prop></code>	<code><any></code>		Optional;
<code>host_settings.<prop></code>	<code><any></code>		Optional;

Links

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
}
```

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

<code>host_state.ipv6_default_gateway</code>	<code><string></code>	The IPv6 default gateway	
--	-----------------------------	--------------------------	--

Links

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}`

JSON

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

Property Name	Type	Description	Notes
<code>interface</code>	<code><object></code>	Configuration and current status of a management interface	
<code>interface.configuration</code>	<code><configuration></code>	Interface configuration	
<code>interface.name</code>	<code><string></code>	Interface name	Read-only; Optional;
<code>interface.state</code>	<code><state></code>	Current interface state	Read-only;
<code>interface.<prop></code>	<code><any></code>		Optional;

Links

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

```
number
```

Property Name	Type	Description	Notes
<code>interface.links.renew.request</code>	<code><number></code>	DHCP version to renew	Values: 4, 6;

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

Related var	Data value for replacement
name	0/name

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  
]
```

Property Name	Type	Description	Notes
<i>interfaces</i>	<i><array of <string>></i>	Management interfaces available on this device	
<i>interfaces[items]</i>	<i><string></i>	The name of an available management interface	

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}
```

JSON

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

Property Name	Type	Description	Notes
<code>ipv4_route</code>	<code><object></code>	A single IPv4 route	Required properties: [network_prefix, gateway_address, interface];
<code>ipv4_route.gateway_address</code>	<code><string></code>	Gateway IP address	
<code>ipv4_route.id</code>	<code><number></code>	Route ID	Read-only; Optional;
<code>ipv4_route.interface</code>	<code><string></code>	Network interface. At least one of "gateway_address" or "interface needs" to be specified.	
<code>ipv4_route.network_prefix</code>	<code><string></code>	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"	
<code>ipv4_route.<prop></code>	<code><any></code>		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

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

Property Name	Type	Description	Notes
<i>ipv4_routes</i>	<object>	Configured IPv4 routes	
<i>ipv4_routes.all</i>	<array of <route_type>>	All configured IPv4 routes, including default gateway.	Optional;
<i>ipv4_routes.all[items]</i>	<route_type>	A route	
<i>ipv4_routes.static</i>	<array of <items>>	Static, manually configured IPv4 routes.	Optional;
<i>items</i>	<object>	A single IPv4 route	Required properties: [network_prefix, gateway_address, interface];
<i>items.gateway_address</i>	<string>	Gateway IP address	
<i>items.id</i>	<number>	Route ID	Read-only; Optional;
<i>items.interface</i>	<string>	Network interface. At least one of "gateway_address" or "interface needs" to be specified.	
<i>items.network_prefix</i>	<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"	
<i>items.<prop></i>	<any>		Optional;
<i>ipv4_routes.<prop></i>	<any>		Optional;

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

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

Property Name	Type	Description	Notes
<i>ipv6_route</i>	<object>	A single Ipv6 route	Required properties: [network_prefix, gateway_address, interface];
<i>ipv6_route.gateway_address</i>	<string>	Gateway IP address	
<i>ipv6_route.id</i>	<number>	Route ID	Read-only; Optional;
<i>ipv6_route.interface</i>	<string>	Network interface. At least one of "gateway_address" or "interface needs" to be specified.	
<i>ipv6_route.network_prefix</i>	<string>	Destination network prefix. A string consisting of an IP address and prefix, separated by a slash (/).	
<i>ipv6_route.<prop></i>	<any>		Optional;

Links

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


```

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

```

Property Name	Type	Description	Notes
<i>ipv6_routes</i>	<object>	Configured IPv6 routes	
<i>ipv6_routes.all</i>	<array of <route_type>>	All configured IPv6 routes	Optional;
<i>ipv6_routes.all[items]</i>	<route_type>	A route	
<i>ipv6_routes.static</i>	<array of <items>>	Static, manually configured IPv6 routes.	Optional;
<i>items</i>	<object>	A single Ipv6 route	Required properties: [network_prefix, gateway_address, interface];
<i>items.gateway_address</i>	<string>	Gateway IP address	
<i>items.id</i>	<number>	Route ID	Read-only; Optional;
<i>items.interface</i>	<string>	Network interface. At least one of "gateway_address" or "interface needs" to be specified.	
<i>items.network_prefix</i>	<string>	Destination network prefix. A string consisting of an IP address and prefix, separated by a slash (/).	
<i>items.<prop></i>	<any>		Optional;
<i>ipv6_routes.<prop></i>	<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
}
```

Property Name	Type	Description	Notes
<i>route_settings</i>	<i><object></i>	Global route settings	
<i>route_settings.default_gateway</i>	<i><object></i>	The default gateway configuration for the host. Ignored if DHCP is enabled. The state can be verified by querying the "ipv4_routes" or "ipv6_routes" resources.	Optional;
<i>route_settings.default_gateway.ipv4</i>	<i><string></i>	IPv4 address for default gateway. Blank string unsets it.	Optional;
<i>route_settings.default_gateway.ipv6</i>	<i><string></i>	IPv6 address for default gateway. Blank string unsets it.	Optional;
<i>route_settings.default_gateway.<prop></i>	<i><any></i>		Optional;
<i>route_settings.<prop></i>	<i><any></i>		Optional;

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

```

{
  "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
}

```

Property Name	Type	Description	Notes
<i>configuration</i>	<object>	Interface configuration	
<i>configuration.autoneg</i>	<multiple>		Optional;
<i>configuration.autoneg.anyOf[0]</i>	<boolean>	Auto-negotiation enabled	
<i>configuration.autoneg.anyOf[1]</i>	<null>	Not configured, or does not apply to this interface	
<i>configuration.description</i>	<string>	Interface description	Optional;
<i>configuration.duplex</i>	<multiple>		Optional;
<i>configuration.duplex.anyOf[0]</i>	<string>	Interface duplex setting. Ignored if "autoneg" is True.	Values: full, half;
<i>configuration.duplex.anyOf[1]</i>	<null>	Not configured, or does not apply to this interface	
<i>configuration.enable</i>	<boolean>	Interface enabled	Optional;
<i>configuration.ipv4</i>	<object>		Optional;
<i>configuration.ipv4.address</i>	<string>	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.	Optional;
<i>configuration.ipv4.dhcp</i>	<boolean>	Obtain IP address via DHCP	Optional;
<i>configuration.ipv4.dynamic_dns</i>	<boolean>	Send hostname with the DHCP request	Optional;
<i>configuration.ipv4.subnet_mask</i>	<string>	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.	Optional;
<i>configuration.ipv4.<prop></i>	<any>		Optional;
<i>configuration.ipv6</i>	<object>		Optional;
<i>configuration.ipv6.address</i>	<string>	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.	Optional;
<i>configuration.ipv6.dhcp</i>	<boolean>	Obtain IP address via DHCP	Optional;
<i>configuration.ipv6.dynamic_dns</i>	<boolean>	Send hostname with the DHCP request	Optional;
<i>configuration.ipv6.prefix</i>	<string>	The IPv6 address prefix. This field is validated but not applied if DHCP is enabled, or the address is not set.	Optional;
<i>configuration.ipv6.<prop></i>	<any>		Optional;
<i>configuration.mtu</i>	<number>	Interface MTU	Optional;
<i>configuration.speed</i>	<multiple>		Optional;
<i>configuration.speed.anyOf[0]</i>	<number>	Speed to force this interface to run at. Ignored if "autoneg" is True.	Values: 10, 100, 1000, 2500, 10000;
<i>configuration.speed.anyOf[1]</i>	<null>	Not configured, or does not apply to this interface	
<i>configuration.<prop></i>	<any>		Optional;

Type: route_type

A route

JSON

```

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

```

Property Name	Type	Description	Notes
<i>route_type</i>	<object>	A route	
<i>route_type.gateway_address</i>	<string>	Gateway IP address	Optional;
<i>route_type.interface</i>	<string>	Network interface. At least one of "gateway_address" or "interface" needs to be specified.	Optional;
<i>route_type.network_prefix</i>	<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" or "192.168.1.0/255.255.255.0". For ipv6 specify prefix after slash (/).	Optional;
<i>route_type.<prop></i>	<any>		Optional;

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
}

```

Property Name	Type	Description	Notes
<i>state</i>	<object>	Current interface state	Read-only;
<i>state.autoneg</i>	<multiple>		Read-only; Optional;
<i>state.autoneg.anyOf[0]</i>	<boolean>	Auto-negotiation status	Read-only;
<i>state.autoneg.anyOf[1]</i>	<null>	Auto-negotiation is not relevant for this interface	Read-only;
<i>state.duplex</i>	<multiple>		Read-only; Optional;
<i>state.duplex.anyOf[0]</i>	<string>	Interface duplex	Read-only;
<i>state.duplex.anyOf[1]</i>	<null>	Duplex is not relevant for this interface	Read-only;
<i>state.enabled</i>	<boolean>	Interface enabled/up	Read-only; Optional;
<i>state.hw_address</i>	<string>	Interface hardware address	Read-only; Optional;
<i>state.if_index</i>	<number>	Interface index	Read-only; Optional;
<i>state.interface_type</i>	<string>	Interface type	Read-only; Optional;
<i>state.ipv4</i>	<object>		Read-only; Optional;
<i>state.ipv4.address</i>	<string>	Interface IPv4 address	Read-only; Optional;
<i>state.ipv4.subnet_mask</i>	<string>	Subnet mask	Read-only; Optional;
<i>state.ipv4.<prop></i>	<any>		Read-only; Optional;
<i>state.ipv6</i>	<object>		Read-only; Optional;
<i>state.ipv6.address</i>	<string>	Interface IPv6 address	Read-only; Optional;

<i>state.ipv6.link_local</i>	<string>	Link local IPv6 address	Read-only; Optional;
<i>state.ipv6.<prop></i>	<any>		Read-only; Optional;
<i>state.link</i>	<boolean>	Interface link state	Read-only; Optional;
<i>state.mtu</i>	<number>	Interface MTU	Read-only; Optional;
<i>state.speed</i>	<multiple>		Read-only; Optional;
<i>state.speed.anyOf[0]</i>	<string>	Interface speed	Read-only;
<i>state.speed.anyOf[1]</i>	<null>	Speed is not relevant for this interface	Read-only;
<i>state.<prop></i>	<any>		Read-only; Optional;