# Cisco IOS XE REST API Management Reference Guide

**Updated:** February 21, 2018

## Chapter: IP Interface Configuration Requirements

### Chapter Contents



- Resource Summary for IP Interface
- Interface Resources
- Interface State
- Interface Statistics

# Resource Summary for IP Interface

| | | HTTP Method | | | |
|---|---|---|---|---|---|
| **Resource** | **URL (BaseURL)** | **GET** | **POST** | **PUT** | **DELETE** |
| Interface | /api/v1/interfaces | Y | Y | N | N |
| | /api/v1/interfaces/{if-id}[1] | Y | N | Y | Y* |
| Interface Statistics | /api/v1/interfaces/{if-id}/statistics | Y | Y | N | N |
| Interface State | /api/v1/interfaces/{if-id}/state | Y | N | Y | N |

[1] {if-id} = Interface ID returned from the REST API used to create the interface.

# Interface Resources

History

| Release | Modification |
|---|---|
| IOS XE 3.10 | Introduced for the CSR1000V platform |
| IOS XE 3.11 | Added the following properties:<br>• icmp-redirects<br>• icmp-unreachable<br>• proxy-arp<br>• verify-unicast-source<br>• subinterface-vlan (includes sub-properties described below) |
| IOS XE 3.13 | Enhanced interface API for BDI support: Added a new interface type: "bdi" |
| IOS XE 3.14 | Introduced for ASR1001-X and ASR1002-X platforms |

## Properties

| Property | Type | Required for POST and PUT | Description |
|---|---|---|---|
| kind | string | Not applicable | Object type. Has the fixed value "object#interface" |
| type | string | Mandatory | Interface type. Read-only |
| if-name | string | Mandatory | Interface name. Note that the name follows the usual IOS slot/port convention. |
| description | string | Optional | Interface Description |
| ip-address | ip-address | Mandatory | IP address in the format x.x.x.x |
| subnet-mask | ipsubnet | Mandatory | Subnet mask in the format x.x.x.x |
| nat-direction | string | Mandatory | Indicates if the interface is viewed as "inside" or "outside" from NAT point of view. |
| icmp-redirects | boolean | Optional | ICMP Redirects |
| icmp-unreachable | boolean | Optional | ICMP Unreachable |
| proxy-arp | boolean | Optional | Proxy Arp, enabled or disabled |
| verify-unicast-source | boolean | Optional | Unicast Source Address Verification enabled or disabled |
| subinterface-vlan | object | Optional | This property is only used by a sub-interface; a full interface does not have this property.<br>Includes three sub-properties: encap-type, vlan-id, encapsulated-vlan |
| encap-type | string | Optional | (sub-property of subinterface-vlan)<br>Possible values:<br>• DOT1Q |

| | | | |
|---|---|---|---|
| | | | • QINQ |
| vlan-id | number | Mandatory | (sub-property of subinterface-vlan)<br>vlan-id. Possible values: 1 to 4094 |
| encapsulated-vlan | string | Optional | (sub-property of subinterface-vlan)<br>Used in QINQ subinterface configuration to specify the second vlan-id.<br>Possible values: are 1 to 4094 as a numerical string. |

### JSON Representation

```
{
"if-name": "string",
"type": "string",
"ip-address": "string",
"subnet-mask": "string",
"description": "string",
"nat-direction": "string",
"icmp-redirects":"boolean",
"icmp-unreachable": "boolean",
"proxy-arp": "boolean",
"verify-unicast-source": "boolean",
"subinterface-vlan":
{
"encap-type": "string",
"vlan-id": "number",
"encapsulated-vlan": "string",
},
}
```

# Examples Demonstrating Use of Interface ID

## Examples Using Interface ID

### Resource URI

| Verb | URI |
|---|---|
| [GET PUT] | /api/v1/interfaces/{if-id} |

### Example 1: Creating a Loopback Interface

The following example is for a logical Ethernet network interface, and creates a loopback interface.

```
{
"type": "{string}",
"if-name": "{interface-name}",
"description": "loopback ",
"ip-address": "170.15.15.11",
"subnet-mask": "255.255.255.0",
"nat-direction": ""
}
```

**Note** POST /api/v1/ is available only for loopback. Cisco IOS XE 3.10 does not support POST /api/v1/ on a sub-interface.

### Example 2: Retrieving an Interface

*JSON Request*

```
GET /api/v1/gigabitEthernet1
Accept: application/json
```

*JSON Response*

```
200 OK
Content-Type: application/json

{
"kind" : "object#",
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "outside ",
"ip-address" : "172.15.15.15",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside",
"icmp-redirects" : true,
"icmp-unreachable" : true,
"proxy-arp" : true,
"verify-unicast-source": true
}
```

### Example 3: Modifying an Interface

*JSON Request*

```
PUT /api/v1/gigabitEthernet1
Content-Type: application/json

{
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "outside ",
"ip-address" : "172.15.15.16",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside"
"icmp-redirects" : true,
"icmp-unreachable" : true,
"proxy-arp" : true,
"verify-unicast-source" : true
}
```

*JSON Response*

```
204 No Content
```

## Examples Without Interface ID

### Resource URI

| Verb | URI |
|---|---|
| [GET POST] | /api/v1/interfaces |

### Example: GET

*JSON Request*

```
GET /api/v1/interfaces
Accept: application/json
```

*JSON Response*

```
200 OK
Content-Type: application/json

{
"kind" : "collection#interface",
"items": [
{
"kind" : "object#",
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "outside ",
"ip-address" : "172.15.15.15",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside",
"icmp-redirects" : true,
"icmp-unreachable": true,
"proxy-arp" : true,
"verify-unicast-source" : true
}
]
}
```

### Example: POST

*JSON Request*

```
POST /api/v1/
Content-Type: application/json

{
"type" : "loopback",
"if-name" : "loopback1",
"description" : "outside ",
"ip-address" : "172.15.15.16",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside"
"icmp-redirects" : true,
"icmp-unreachable" : true,
"proxy-arp" : true,
"verify-unicast-source" : true
}
```

*JSON Response*

```
201 Created
Location: https://host/api/v1/interfaces/loopback1
```

## Retrieve Interface Details

### Resource URI

| Verb | URI |
|---|---|
| GET | /api/v1/interfaces/{if-id} |

### Example 1: Retrieve Interface Details

*JSON Request*

```
GET /api/v1/interfaces/gigabitEthernet1
Accept: application/json
```

*JSON Response*

```
200 OK

Content-Type: application/json
{
"kind" : "object#interface",
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "outside interface",
"ip-address" : "172.15.15.15",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside"
}
```

### Example 2: Retrieve Sub-interface Details

**Note** *Available in Cisco IOS XE 3.11 and later*

**JSON Request**

```
GET /api/v1/interfaces/GigabitEthernet2.23
Accept: application/json
```

**JSON Response**

```
200 OK
Content-Type: application/json

{
"kind": "object#interface",
"description": "",
"if-name": "GigabitEthernet2.23",
"proxy-arp": true,
"subnet-mask": "255.255.255.0",
"icmp-unreachable": true,
"nat-direction": "",
"icmp-redirects": true,
"ip-address": "22.10.10.23",
"subinterface-vlan": {"vlan-id": 23, "encap-type": "DOT1Q"},
"type": "ethernet",
"verify-unicast-source": false
}
```

## Retrieve All Interfaces and Details

## Resource URI

| Verb | URI |
|---|---|
| GET | /api/v1/interfaces |

## Properties for Retrieve All

| Property | Type | Description |
|---|---|---|
| kind | string | Object type. Has fixed value " collection#interface" |
| items | array | Array of interface objects |

## Example

*JSON Request*

```
GET /api/v1/interfaces
Accept: application/json
```

*JSON Response*

```
200 OK

Content-Type: application/json
{
"kind" : "collection#interface",
"items": [
{
"kind" : "object#interface",
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "management interface",
"ip-address" : "129.10.10.10",
"subnet-mask" : "255.255.254.0"
},

{
"kind" : "object#interface",
"type" : "ethernet",
"if-name" : "gigabitEthernet2",
"description" : "outside interface",
"ip-address" : "172.15.15.15",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside"
},
{
"kind" : "object#interface",
"type" : "ethernet",
"if-name" : "gigabitEthernet3",
"description" : "inside interface",
"ip-address" : "10.10.10.15",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "inside"
}
]
```

```
}
```

# Modify an Interface Configuration

## Resource URI

| Verb | URI |
|---|---|
| PUT | /api/v1/interfaces/{if-id} |

### Example 1: Changing the IP-address from 172.15.15.15 to 172.15.15.16

**JSON Request**

```
PUT /api/v1/interfaces/gigabitEthernet1

Content-Type: application/json
{
"type" : "ethernet",
"if-name" : "gigabitEthernet1",
"description" : "outside interface",
"ip-address" : "172.15.15.16",
"subnet-mask" : "255.255.254.0",
"nat-direction" : "outside"
}
```

*JSON Response*

```
204 No Content
```

### Example 2: Modify VLAN IDs (Example Includes Sub-interface Property)

**JSON Request**

```
PUT /api/v1/interfaces/GigabitEthernet2.23
Content-Type: application/json

{ "subinterface-vlan":{"vlan-id":230},
"if-name":"GigabitEthernet2.23",
"subnet-mask":"255.255.255.0",
"ip-address":"22.10.10.23",
"type":"ethernet"
}
```

*JSON Response*

```
204 No content
```

# Create an Interface

Enables:

- Creating a loopback or sub-interface and IP address
  - The loopback or sub-interface cannot be on the same network as a physical interface.
  - After a loopback interface is configured, a router-id can be generated from it.
- Changing properties of a physical interface

If the if-name in the HTTP POST body has a dash (for example, myintf-0), the API controller code would add another dash to the if-name to make an if-id (for example, myintf--0). The if-name with one dash should be passed to the 1-P API calls.

### Resource URI

| Verb | URI |
|---|---|
| POST | /api/v1/interfaces |

### Example 1: Create a Loopback Interface

*JSON Request*

```
POST /api/v1/interfaces
Accept: application/json

Content-Type: application/json
{
"type" : "loopback",
"if-name" : "loopback11",
"description" : "loopback ",
"ip-address" : "170.15.15.11",
"subnet-mask" : "255.255.255.0",
"nat-direction" : ""
}
```

*JSON Response: Returning the Interface ID*

```
201 Created
Location: http://host/api/v1/interfaces/loopback11_ifid
```

### Example 2: Create a Sub-interface

**Note** *Available in Cisco IOS XE 3.11 and later*

*JSON Request*

```
POST /api/v1/interfaces
Content-Type: application/json

{
"subinterface-vlan": {"vlan-id":23},
"if-name": "GigabitEthernet2.23",
"subnet-mask": "255.255.255.0",
"ip-address": "22.10.10.23",
"type": "ethernet"
}
```

*JSON Response*

```
201 Created
Location: https://host/api/v1/interfaces/GigabitEthernet2.23
```

## Delete an Interface

### Resource URI

| Verb | URI |
|---|---|
| DELETE | /api/v1/interfaces/{if-id} |

### Example 1: Delete an Interface

*JSON Request*

```
DELETE /api/v1/interfaces/11
```

*JSON Response*

```
204 No Content
```

### Example 2: Delete a Sub-interface

**Note** *Available in Cisco IOS XE 3.11 and later*

**JSON Request**

```
DELETE /api/v1/interfaces/GigabitEthernet2.23
```

**JSON Response**

```
204 No Content
```

# Interface State

### History

| Release | Modification |
|---|---|
| IOS XE 3.10 | Introduced for the CSR1000V platform |
| IOS XE 3.14 | Introduced for ASR1001-X and ASR1002-X platforms |

**Properties**

| Property | Type | Required for POST and PUT | Description |
|---|---|---|---|
| kind | string | Not applicable | Object type. Has the fixed value " object#interface-state" |
| if-name | string | Mandatory | Interface Name. Read-only |
| enabled | boolean | Mandatory | Enables (up) or Disables (down) interface |

# Retrieve Interface State

### Resource URI

| Verb | URI |
|---|---|
| GET | /api/v1/interfaces/{if-id}/state |

### Example

*JSON Request*

```
GET /api/v1/interfaces/gigabitEthernet1/state
Accept: application/json
```

*JSON Response*

```
200 OK

Content-Type: application/json
Accept: application/json


{
"kind" : "object#interface-state",
"if-name" : "gigabitEthernet1",
"enabled" : true
}
```

## Bring an Interface Up or Down

### Resource URI

| Verb | URI |
|---|---|
| PUT | /api/v1/interfaces/{if-id}/state |

### Example: " no shut"  GigabitEthernet1

*JSON Request*

```
PUT /api/v1/interfaces/gigabitEthernet1/state

Content-Type: application/json
Accept: application/json

{
"if-name" : "gigabitEthernet1",
"enabled" : true
}
```

*JSON Response*

```
204 No Content
```

# Interface Statistics

### History

| Release | Modification |
|---|---|
| IOS XE 3.10 | Introduced for the CSR1000V platform |
| IOS XE 3.14 | Introduced for ASR1001-X and ASR1002-X platforms |

### Properties

| Property | Type | Description |
|---|---|---|
| kind | string | Object type. Has the fixed value " object#interface-statistics" |
| if-name | string | Interface Name. Read-only |
| in-errors | number | Sum of all input related errors |

| | | |
|---|---|---|
| in-packet-drops | number | Input packet drop count is caused when the input queue is full. |
| in-current-packets | number | Total packets received since the last reset of statistics |
| in-packet-rate-bps | number | Input packet receive rate in bytes per second |
| in-packet-rate-pps | number | Input packet receive rate in packets per second |
| out-errors | number | Sum of all output related errors |
| out-packet-drops | number | Output packet drop count is caused when the output queue is full. |
| out-current-packets | number | Total packets transmitted since the last statistics |
| out-packet-rate-bps | number | Output packet transmit rate in bytes per second |
| out-packet-rate-pps | number | Output packet transmit rate in packets per second |

## Retrieve Interface Statistics

### Resource URI

| Verb | URI |
|---|---|
| GET | /api/v1/interfaces/{if-id}/statistics |

### Example

*JSON Request*

```
GET /api/v1/interfaces/gigabitEthernet1/statistics
Accept: application/json
```

*JSON Response*

```
200 OK

Content-Type: application/json

{
"kind" : "object#interface-statistics",
"if-name" : "gigabitEthernet1",
"in-errors" : 0,
"in-packet-drops" : 0,
"in-current-packets" : 17,
"in-packet-rate-bps" : 0,
"in-packet-rate-pps" : 0,
"out-errors" : 0,
"out-packet-drops" : 0,
"out-current-packets" : 0,
"out-packet-rate-bps" : 0,
"out-packet-rate-pps" : 0
}
```

## Clear Interface Statistics

This resource also supports clearing of interface statistics by doing a POST on the resource with the following request message. See Resource specific operations for more details & examples.

### Example

**JSON Request**

```
POST /api/v1/interfaces/statistics

Content-Type: application/json
Accept: application/json

{
"action" : "clear"
}
```

*JSON Response*

```
204 No Content
```

---