



Restful API Guide

P10GSFPBPFE

10/2025

Release Version: 6.1.22

Copyright © 2025 Garland Technology, LLC. All rights reserved.

No part of this document may be reproduced in any form or by any means without prior written permission of Garland Technology, LLC.

The Garland Technology trademarks, service marks ("Marks") and other Garland Technology trademarks are the property of Garland Technology, LLC. PacketMAX Series products of marks are trademarks or registered trademarks of Garland Technology, LLC. You are not permitted to use these Marks without the prior written consent of Garland Technology.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Garland Technology and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.

Table of Contents

Restful API Basics.....	8
Display Public Information	8
Display Key Press Timeout	8
Reset Key Press Timeout (60-3600 seconds)	8
Logout	8
System	9
Display System Information	9
Display General System Setting.....	9
Edit Chassis Name / Key Press Timeout	9
Display Authentication.....	9
Enable Primary TACACS.....	9
Disable Primary TACACS.....	9
Enable Secondary TACACS.....	10
Disable Secondary TACACS.....	10
Display Privilege	10
Display Group	10
Create Group.....	10
Modify Group	11
Delete Group.....	11
Display User.....	11
Create User	11
Change User Password.....	11
Delete User.....	12
Display Network Setting.....	12
Display Date & Time	12
Display Time Zone	12
Enable NTP No Authentication.....	12
Enable NTP Authentication	12
Disable NTP	13
Set Date/Time Manually	13
Set Network Setting	13

Display Syslog	13
Enable Syslog.....	13
Syslog Test	13
Disable Syslog.....	13
Display SNMP	14
Enable SNMPv2rw.....	14
Enable SNMPv3 MD5/DES	14
Enable SNMPv3 SHA/AES.....	14
SNMP Test	14
Disable SNMP	14
Reboot.....	14
Set Mode	15
Bypass	15
Span.....	15
Span (Packet Inject).....	15
Breakout.....	15
Filter	15
Aggregate	15
Filter Tap.....	16
Bypass Filter	16
Bypass Mode	17
Display Bypass Taps Options (Heartbeat Setting)	17
Modify Bypass Taps Options (Heartbeat Setting).....	17
Display Bypass Tap Configuration.....	17
Modify Tap Configuration.....	17
Display Port Configuration	17
Modify Port Configuration	18
Display Port Availability.....	18
Display Port Statistics	18
Clear Port Statistics	18
Span Mode	19
Display Port Configuration	19

Modify Port Configuration	19
Display Port Availability.....	19
Display Port Statistics	19
Clear Port Statistics	19
Span Packet Inject Mode.....	20
Display Port Configuration	20
Modify Port Configuration	20
Display Port Availability.....	20
Display Port Statistics	20
Clear Port Statistics	20
Breakout Mode	21
Display Port Configuration	21
Modify Port Configuration	21
Display Port Availability.....	21
Display Port Statistics	21
Clear Port Statistics	21
Filter Mode.....	22
Display Packet Broker Configuration	22
Display Filter Template.....	22
Create Filter Template.....	22
Modify Filter Template.....	22
Delete Filter Template	23
Display Config Map Counts	23
Create Config Map / Ingress Filter	23
Modify Config Map Priority.....	23
Enable/Disable Config Map.....	24
Delete Config Map	24
Clear Config Map Counters	24
Refresh Config Map Counters	24
Display Port Configuration	24
Modify Port Configuration	24
Display Port Availability.....	25

Display Port Statistics	25
Clear Port Statistics	25
Aggregate Mode.....	26
Display Port Configuration	26
Modify Port Configuration	26
Display Port Availability.....	26
Display Port Statistics	26
Clear Port Statistics	26
Filter Tap Mode	27
Display Packet Broker Configuration	27
Display Filter Template.....	27
Create Filter Template.....	27
Modify Filter Template.....	27
Delete Filter Template.....	28
Display Config Map Counts	28
Create Config Map / Ingress Filter	28
Modify Config Map Priority.....	28
Enable/Disable Config Map.....	29
Delete Config Map	29
Clear Config Map Counters	29
Refresh Config Map Counters	29
Display Port Configuration	29
Modify Port Configuration	29
Display Port Availability.....	30
Display Port Statistics	30
Clear Port Statistics	30
Bypass Filter Mode.....	31
Display Bypass Taps Options (Heartbeat Setting)	31
Modify Bypass Taps Options (Heartbeat Setting).....	31
Display Bypass Tap Configuration.....	31
Modify Tap Configuration.....	31
Display Packet Broker Configuration	31

Display Filter Template	31
Create Filter Template	32
Modify Filter Template	32
Delete Filter Template	32
Display Config Map Counts	32
Create Config Map / Ingress Filter	33
Modify Config Map Priority	33
Enable/Disable Config Map	33
Delete Config Map	34
Clear Config Map Counters	34
Refresh Config Map Counters	34
Display Port Configuration	34
Modify Port Configuration	34
Display Port Availability	34
Display Port Statistics	34
Clear Port Statistics	34

Restful API Basics

The restful API supports JSON over HTTPS. The restful API data types supported; GET, POST, and PUT. This document contains restful API syntax examples that may be used as a reference. The specific restful API syntax may be captured using the GUI under the Developer Tools / Network / Headers and Preview.

Login

This example uses the default username and password, admin/gtadmin1. Upon successfully logging in the unit will return a Cookie / Session ID. The Session ID must be used as part of all additional commands sent to the unit.

POST <https://xxx.xxx.xxx.xxx/login>

Body Syntax Example

```
{"username": "admin", "password": "gtadmin1", "rememberme": true}
```

Example Session ID:

Set-Cookie: session=id=9iNS4gEJc0sQ9QcgVPd;path=/

Note: Once a session has been established it is controlled by the Key Press Timeout value, 60 to 3600 seconds. The Key Press Timeout value can be modified using the “Edit Chassis Name / Key Press Timeout” option in Section 2. System. The Key Press Timeout may be displayed and reset using the following options.

Display Public Information

GET <https://xxx.xxx.xxx.xxx/sysInfoPublic>

Display Key Press Timeout

GET <https://xxx.xxx.xxx.xxx/active>

Reset Key Press Timeout (60-3600 seconds)

PUT <https://xxx.xxx.xxx.xxx/miscCfg>

Body Syntax Example

```
{"chassisName": "System Test", "keyPressTimeout": "3000"}
```

Logout

POST <https://xxx.xxx.xxx.xxx/logout>

System

Display System Information

GET <https://xxx.xxx.xxx.xxx/sysInfo>

Display General System Setting

GET <https://xxx.xxx.xxx.xxx/miscCfg>

Edit Chassis Name / Key Press Timeout

PUT <https://xxx.xxx.xxx.xxx/miscCfg>

Body Syntax Example

```
{"chassisName":"NewChassisName","keyPressTimeout":"60"}
```

Note: Key press timeout range 60-3600 seconds.

Display Authentication

GET <https://xxx.xxx.xxx.xxx/authentication>

Enable Primary TACACS

PUT <https://xxx.xxx.xxx.xxx/authentication>

Body Syntax Example

```
{"local":true,"tacacs":{"on":true,"server":"192.168.1.166","secret":"xyz","timeout":"5"},  
"tacacs2":{"on":false,"server":"10.10.10.200","secret":"xyz","timeout":"5"}}
```

Disable Primary TACACS

PUT <https://xxx.xxx.xxx.xxx/authentication>

Body Syntax Example

```
{"local":true,"tacacs":{"on":false,"server":"192.168.1.166","secret":"xyz","timeout":"5"},  
"tacacs2":{"on":false,"server":"10.10.10.200","secret":"xyz","timeout":"5"}}
```

Enable Secondary TACACS

PUT <https://xxx.xxx.xxx.xxx/authentication>

Body Syntax Example

```
{
  "local": true,
  "tacacs": {
    "on": false,
    "server": "10.10.10.200",
    "secret": "xyz",
    "timeout": "5"
  },
  "tacacs2": {
    "on": true,
    "server": "192.168.1.173",
    "secret": "richardson1",
    "timeout": "5"
  }
}
```

Disable Secondary TACACS

PUT <https://xxx.xxx.xxx.xxx/authentication>

Body Syntax Example

```
{
  "local": true,
  "tacacs": {
    "on": false,
    "server": "10.10.10.200",
    "secret": "xyz",
    "timeout": "5"
  },
  "tacacs2": {
    "on": false,
    "server": "192.168.1.173",
    "secret": "richardson1",
    "timeout": "5"
  }
}
```

Display Privilege

GET <https://xxx.xxx.xxx.xxx/availablePrivileges>

Display Group

GET <https://xxx.xxx.xxx.xxx/groups>

Create Group

POST <https://xxx.xxx.xxx.xxx/groupAdd>

Body Syntax Example

```
{
  "name": "NewGroup",
  "privileges": [
    "AAA", "USR", "DTC", "DTV", "EXC", "IPC", "IPV", "LGC",
    "LGV", "MIS", "PBC", "PBV", "PTC", "PTV", "RBT", "TPC", "TPV", "UPG", "ADM"
  ]
}
```

Note: Privileges may be added or removed as desired.

AAA	authentication, authorization, account
ADM	user administrator
DTC	date, time, ntp configuration
DTV	date, time, ntp view
EXC	export/import
IPC	maintenance network ip configuration
IPV	maintenance network ip view
LGC	syslog, snmp configuration
LGV	syslog, snmp view
MIS	miscellaneous
PBC	packet broker configuration

PBV	packet broker view
PTC	port configuration
PTV	port view
RBT	chassis reboot
TPC	tap config
TPV	tap view
UPG	software field upgrade
USR	account configuration

Modify Group

POST <https://xxx.xxx.xxx.xxx/groupChange>

Body Syntax Example

```
{"name":"NewGroup","privileges":["MIS","PBC","PBV","PTC"],"oldName":"NewGroup"}
```

Delete Group

POST <https://xxx.xxx.xxx.xxx/groupDelete>

Body Syntax Example

```
{"name":"NewGroup"}
```

Display User

GET <https://xxx.xxx.xxx.xxx/userAll>

Create User

POST <https://xxx.xxx.xxx.xxx/userAdd>

Body Syntax Example

```
{"username":"NewUser","password":"NewUserPW","group":"NewGroup"}
```

Change User Password

POST <https://xxx.xxx.xxx.xxx/userChange>

Body Syntax Example

```
{"username":"User123","password":"NewPW","group":"Group","oldUsername":"User123"}
```

Delete User

POST <https://xxx.xxx.xxx.xxx/userDelete>

Body Syntax Example

```
{"username": "NewUser"}
```

Display Network Setting

GET <https://xxx.xxx.xxx.xxx/maintNetwork>

Display Date & Time

GET <https://xxx.xxx.xxx.xxx/dateTime>

Display Time Zone

GET <https://xxx.xxx.xxx.xxx/timezones>

Enable NTP No Authentication

PUT <https://xxx.xxx.xxx.xxx/dateTime>

Body Syntax Example

```
{"timeZone": "America/Chicago", "ntp": {"on": true, "ipAddress": "192.168.1.132", "authentication": "none", "authSym": {"keynum": "1", "keytype": "MD5", "key": "key"}}, "date": "5/30/2024", "ntpStatus": "off", "time": "10:51"}
```

Enable NTP Authentication

PUT <https://xxx.xxx.xxx.xxx/dateTime>

Body Syntax Example

```
{"timeZone": "America/Chicago", "ntp": {"on": true, "ipAddress": "192.168.1.132", "authentication": "symmetric", "authSym": {"keynum": "1", "keytype": "md5", "key": "%M5zLv16t~/CT\D<JI3`"}}, "date": "5/30/2024", "ntpStatus": "off", "time": "10:52"}
```

Disable NTP

PUT <https://xxx.xxx.xxx.xxx/dateTime>

Body Syntax Example

```
{"timeZone": "America/Chicago", "ntp": {"on": false, "usePool": false, "ipAddress": "xxx.xxx.xx.xxx"}, "date": "2/10/2021", "time": "15:33"}
```

Set Date/Time Manually

PUT <https://xxx.xxx.xxx.xxx/dateTime>

Body Syntax Example

```
{"timeZone": "America/Chicago", "ntp": {"on": false, "ipAddress": "127.0.0.1", "authentication": "none", "authSym": {"keynum": "1", "keytype": "MD5", "key": "key"}}, "date": "5/30/2024", "ntpStatus": "off", "time": "11:26"}
```

Set Network Setting

PUT <https://xxx.xxx.xxx.xxx/maintNetwork>

Body Syntax Example

```
{"ipv4Enable": true, "loadedSslCertUsed": false, "sslCertLoaded": false, "address": "192.168.1.28/24", "mask": "255.255.255.0", "gateway": "192.168.1.1", "ipv6Enable": false, "ipv6Addresses": "", "ipv6Gateway": "", "currentIpv4Address": "192.168.1.28", "currentIpv6Address": ""}
```

Display Syslog

GET <https://xxx.xxx.xxx.xxx/syslogCfg>

Enable Syslog

PUT <https://xxx.xxx.xxx.xxx/syslogCfg>

Body Syntax Example

```
{"on": true, "useUnitId": false, "unitId": "0", "ipAddress": "xxx.xxx.xxx.xxx", "protocol": "UDP", "port": "514"}
```

Syslog Test

POST <https://xxx.xxx.xxx.xxx/syslogTest>

Disable Syslog

PUT <https://xxx.xxx.xxx.xxx/syslogCfg>

Body Syntax Example

```
{"on":false,"useUnitId":false,"unitId":"0","ipAddress":"xxx.xxx.xxx.xxx","protocol":"UDP","port":"514"}
```

Display SNMP

GET <https://xxx.xxx.xxx.xxx/snmpCfg>

Enable SNMPv2rw

PUT <https://xxx.xxx.xxx.xxx/snmpCfg>

Body Syntax Example

```
{"on":true,"accessPort":"161","trapPort":"162","trapIpAddress":"xxx.xxx.xxx.xxx","v3user":"user1234","v3authType":"MD5","v3authPass":"auth1234","v3privPass":"priv1234","v3privProt":"DES","communityPassword":"public","mode":"v2rw"}
```

Enable SNMPv3 MD5/DES

PUT <https://xxx.xxx.xxx.xxx/snmpCfg>

Body Syntax Example

```
{"on":true,"accessPort":"161","trapPort":"162","trapIpAddress":"xxx.xxx.xxx.xxx","v3user":"user1234","v3authType":"MD5","v3authPass":"auth1234","v3privPass":"priv1234","v3privProt":"DES","communityPassword":"public","mode":"v3"}
```

Enable SNMPv3 SHA/AES

PUT <https://xxx.xxx.xxx.xxx/snmpCfg>

Body Syntax Example

```
{"on":true,"accessPort":"161","trapPort":"162","trapIpAddress":"xxx.xxx.xxx.xxx","v3user":"user1234","v3authType":"SHA","v3authPass":"auth1234","v3privPass":"priv1234","v3privProt":"AES","communityPassword":"public","mode":"v3"}
```

SNMP Test

POST <https://xxx.xxx.xxx.xxx/snmpTest>

Disable SNMP

PUT <https://xxx.xxx.xxx.xxx/snmpCfg>

Body Syntax Example

```
{"on":false,"accessPort":"161","trapPort":"162","trapIpAddress":"xxx.xxx.xxx.xxx","v3user":"user","v3authType":"MD5","v3authPass":"auth1234","v3privPass":"prov1234","v3privProt":"DES","communityPassword":"gtpublic","mode":"v2rw"}
```

Reboot

POST <https://xxx.xxx.xxx.xxx/reboot>

Set Mode

Bypass

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "bypass"}
```

Span

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "span1"}
```

Span (Packet Inject)

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "span2"}
```

Breakout

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "breakout"}
```

Filter

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "filter"}
```

Aggregate

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "aggregate"}
```

Filter Tap

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "filtertap"}
```

Bypass Filter

PUT `https://xxx.xxx.xxx.xxx/fourPortMode`

Body Syntax Example

```
{"mode": "bypassfilter"}
```

Bypass Mode

Display Bypass Taps Options (Heartbeat Setting)

GET <https://xxx.xxx.xxx.xxx/gettapoptions>

Modify Bypass Taps Options (Heartbeat Setting)

PUT <https://xxx.xxx.xxx.xxx/settapoptions>

Body Syntax Example

```
{"hbPacketwindow":10,"hbPerSecond":10}
```

Note: No. of Lost Heartbeat Packets, 10-100 / Heartbeats per Second, 10-100.

Display Bypass Tap Configuration

GET <https://xxx.xxx.xxx.xxx/tapGet>

Modify Tap Configuration

PUT <https://xxx.xxx.xxx.xxx/tapPut>

Body Syntax Example

```
{"gt-taps":[{"forceBypass":"false","forceInline":"false","fosFailMode":"closed","lfpEnable":"false","reverseBypass":"false","tapDescription":"","tapElements":[{"description":"element","isGroup":"false","monitorPortsE":[],"monitorPortsI":[],"portA":"1","portB":"1"}, {"description":"element","isGroup":"false","monitorPortsE":[],"monitorPortsI":[],"portA":"3","portB":"4"}, {"description":"element","isGroup":"false","monitorPortsE":[],"monitorPortsI":[],"portA":"2","portB":"2"}]}
```

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpVendorPn": "SFP+SR",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpVendorPn": "SFP+SR",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpVendorPn": "SFP+SR",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpVendorPn": "SFP+SR",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Span Mode

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{ "ports": [ { "portNumber": "1", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "2", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "3", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "4", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" } ] }
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Span Packet Inject Mode

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Breakout Mode

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Filter Mode

Display Packet Broker Configuration

GET <https://xxx.xxx.xxx.xxx/pbConfig>

Display Filter Template

GET <https://xxx.xxx.xxx.xxx/filtTemplates>

Create Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateAdd>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "srcIp": "10.10.10.10", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "innerVlanId": "", "outerVlanId": "", "dscp": "", "ipProtocol": "", "l4SrcStartPort": "", "l4SrcEndPort": "", "l4DstStartPort": "", "l4DstEndPort": "", "egressFilterPort": "" }
```

Note: The filter type may be PASSALL, PASSBY or DENYBY.

Modify Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateChange>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "ipProtocol": "", "srcIp": "10.10.10.20", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "l4SrcStartPort": "", "l4DstStartPort": "", "l4SrcEndPort": "", "l4DstEndPort": "", "outerVlanId": "", "innerVlanId": "", "dscp": "", "egressFilterPort": "", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "oldName": "Example" }
```

Delete Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateRemove>

Body Syntax Example

```
{"name": "NewFilterTemp"}
```

Display Config Map Counts

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Create Config Map / Ingress Filter

POST <https://xxx.xxx.xxx.xxx/configMapAdd>

Body Syntax Example

```
{ "loadBalanceGroup": {}, "configMap": { "ingressPorts": [ "01" ], "egress": { "ports": [ "02" ], "trunkGroup": "" }, "name": "", "description": "", "enabled": true, "ingressFilters": [ { "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "srcIp": "10.10.10.10", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "innerVlanId": "", "outerVlanId": "", "dscp": "", "ipProtocol": "", "l4SrcStartPort": "", "l4SrcEndPort": "", "l4DstStartPort": "", "l4DstEndPort": "", "egressFilterPort": "" } ] }, "egressFilters": { "02": [ ] } }
```

Note: Multiple ingress ports, egress ports and ingress filters may be added.

Modify Config Map Priority

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[ { "name": "CM1", "enabled": true }, { "name": "CM2", "enabled": true }, { "name": "CM3", "enabled": true } ]
```

Note: String the config maps as shown in the above example. The priority is established highest to lowest based on the order listed.

Enable/Disable Config Map

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[{"name": "CM1", "enabled": true}, {"name": "CM2", "enabled": false}, {"name": "CM3", "enabled": true}]
```

Note: String the config maps as shown in the above example. Config maps may be enabled or disabled by modifying the “enabled” option true/false, true = enabled, false = disabled.

Delete Config Map

POST <https://xxx.xxx.xxx.xxx/configMapRemove>

Body Syntax Example

```
["CM1", "CM2", "CM3"]
```

Note: Identify the config map(s) using the “name”.

Clear Config Map Counters

POST <https://xxx.xxx.xxx.xxx/configMapCountsClear>

Refresh Config Map Counters

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Aggregate Mode

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Filter Tap Mode

Display Packet Broker Configuration

GET <https://xxx.xxx.xxx.xxx/pbConfig>

Display Filter Template

GET <https://xxx.xxx.xxx.xxx/filtTemplates>

Create Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateAdd>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "srcIp": "10.10.10.10", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "innerVlanId": "", "outerVlanId": "", "dscp": "", "ipProtocol": "", "l4SrcStartPort": "", "l4SrcEndPort": "", "l4DstStartPort": "", "l4DstEndPort": "", "egressFilterPort": "" }
```

Note: The filter type may be PASSALL, PASSBY or DENYBY.

Modify Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateChange>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "ipProtocol": "", "srcIp": "10.10.10.25", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "l4SrcStartPort": "", "l4DstStartPort": "", "l4SrcEndPort": "", "l4DstEndPort": "", "outerVlanId": "", "innerVlanId": "", "dscp": "", "egressFilterPort": "", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "oldName": "Example" }
```

Delete Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateRemove>

Body Syntax Example

```
{"name": "NewFilterTemp"}
```

Display Config Map Counts

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Create Config Map / Ingress Filter

POST <https://xxx.xxx.xxx.xxx/configMapAdd>

Body Syntax Example

```
{"loadBalanceGroup": {}, "configMap": {"ingressPorts": ["01"], "egress": {"ports": ["03"], "trunkGroup": ""}, "name": "", "description": "", "enabled": true, "ingressFilters": [{"name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "srcIp": "10.10.10.10", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "innerVlanId": "", "outerVlanId": "", "dscp": "", "ipProtocol": "", "l4SrcStartPort": "", "l4SrcEndPort": "", "l4DstStartPort": "", "l4DstEndPort": "", "egressFilterPort": ""}], "egressFilters": {"03": []}}}
```

Note: In the filter tap mode config maps may be made:

Port 1 to Port 3 and/or Port 4

Port 2 to Port 3 and/or Port 4

Multiple ingress filters may be added

Modify Config Map Priority

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[{"name": "CM1", "enabled": true}, {"name": "CM2", "enabled": true}, {"name": "CM3", "enabled": true}]
```

Note: String the config maps as shown in the above example. Priority is established from highest to lowest based on the order listed.

Enable/Disable Config Map

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[{"name": "CM1", "enabled": true}, {"name": "CM2", "enabled": false}, {"name": "CM3", "enabled": true}]
```

Note: String the config maps as shown in the above example. Config maps may be enabled or disabled by modifying the “enabled” option true/false, true = enabled, false = disabled.

Delete Config Map

POST <https://xxx.xxx.xxx.xxx/configMapRemove>

Body Syntax Example

```
["CM1", "CM2", "CM3"]
```

Note: Identify the config map(s) using the “name”.

Clear Config Map Counters

POST <https://xxx.xxx.xxx.xxx/configMapCountsClear>

Refresh Config Map Counters

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{
  "ports": [
    {
      "portNumber": "1",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "2",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "3",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    },
    {
      "portNumber": "4",
      "speedSet": "10G",
      "mode": "normal",
      "description": "port description",
      "sfpvvendorPn": "",
      "vlanIdTagAdd": "0",
      "vlanIdTagStrip": "0",
      "split": "NO"
    }
  ]
}
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>

Bypass Filter Mode

Display Bypass Taps Options (Heartbeat Setting)

GET <https://xxx.xxx.xxx.xxx/gettapoptions>

Modify Bypass Taps Options (Heartbeat Setting)

PUT <https://xxx.xxx.xxx.xxx/settapoptions>

Body Syntax Example

```
{"hbPacketwindow":10,"hbPerSecond":10}
```

Note: No. of Lost Heartbeat Packets, 10-100 / Heartbeats per Second, 10-100.

Display Bypass Tap Configuration

GET <https://xxx.xxx.xxx.xxx/tapGet>

Modify Tap Configuration

PUT <https://xxx.xxx.xxx.xxx/tapPut>

Body Syntax Example

```
{"gt-taps":[{"forceBypass":"true","forceInline":"false","fosFailMode":"closed","lfpEnable":"false","reverseBypass":"false","tapDescription":"","tapElements":[{"description":"element","isGroup":"false","monitorPortsE":[],"monitorPortsI":[],"portA":"1","portB":"1"}, {"description":"element","gElements":[{"portA":"3","portB":"4"}],"groupType":"bypassFilter","isGroup":"true"}, {"description":"element","isGroup":"false","monitorPortsE":[],"monitorPortsI":[],"portA":"2","portB":"2"}]}]}
```

Display Packet Broker Configuration

GET <https://xxx.xxx.xxx.xxx/pbConfig>

Display Filter Template

GET <https://xxx.xxx.xxx.xxx/filtTemplates>

Create Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateAdd>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "srcIp": "10.10.10.10", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "innerVlanId": "", "outerVlanId": "", "dscp": "", "ipProtocol": "", "l4SrcStartPort": "", "l4SrcEndPort": "", "l4DstStartPort": "", "l4DstEndPort": "", "egressFilterPort": "" }
```

Note: The filter type may be PASSALL, PASSBY or DENYBY.

Modify Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateChange>

Body Syntax Example

```
{ "name": "Example", "description": "", "enabled": true, "filterType": "PASSBY", "dstMac": "", "dstMacMask": "ff:ff:ff:ff:ff:ff", "srcMac": "", "srcMacMask": "ff:ff:ff:ff:ff:ff", "etherType": "", "ipProtocol": "", "srcIp": "10.10.10.20", "srcIpMask": "255.255.255.255", "dstIp": "", "dstIpMask": "255.255.255.255", "l4SrcStartPort": "", "l4DstStartPort": "", "l4SrcEndPort": "", "l4DstEndPort": "", "outerVlanId": "", "innerVlanId": "", "dscp": "", "egressFilterPort": "", "srcIp6": "", "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "dstIp6": "", "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF", "oldName": "Example" }
```

Delete Filter Template

POST <https://xxx.xxx.xxx.xxx/filterTemplateRemove>

Body Syntax Example

```
{ "name": "NewFilterTemp" }
```

Display Config Map Counts

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Create Config Map / Ingress Filter

POST <https://xxx.xxx.xxx.xxx/configMapAdd>

Body Syntax Example

```
{
  "loadBalanceGroup": {},
  "configMap": {
    "ingressPorts": ["01"],
    "egress": {
      "ports": ["03"],
      "trunkGroup": "",
      "name": "",
      "description": "",
      "enabled": true,
      "ingressFilters": [
        {
          "name": "Example",
          "description": "",
          "enabled": true,
          "filterType": "PASSBY",
          "srcMac": "",
          "srcMacMask": "ff:ff:ff:ff:ff:ff",
          "dstMac": "",
          "dstMacMask": "ff:ff:ff:ff:ff:ff",
          "etherType": "",
          "srcIp": "10.10.10.10",
          "srcIpMask": "255.255.255.255",
          "dstIp": "",
          "dstIpMask": "255.255.255.255",
          "srcIp6": "",
          "srcIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF",
          "dstIp6": "",
          "dstIp6Mask": "FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF.FFFF",
          "innerVlanId": "",
          "outerVlanId": "",
          "dscp": "",
          "ipProtocol": "",
          "l4SrcStartPort": "",
          "l4SrcEndPort": "",
          "l4DstStartPort": "",
          "l4DstEndPort": "",
          "egressFilterPort": ""
        }
      ],
      "egressFilters": {
        "03": []
      }
    }
  }
}
```

Note: In the bypass filter mode config maps may be made:

Port 1 to Port 3

Port 2 to Port 4

Multiple ingress filters may be added

Modify Config Map Priority

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[
  {
    "name": "CM1",
    "enabled": true
  },
  {
    "name": "CM2",
    "enabled": true
  },
  {
    "name": "CM3",
    "enabled": true
  }
]
```

Note: String the config maps as shown in the above example. Priority is established from highest to lowest based on the order listed.

Enable/Disable Config Map

POST <https://xxx.xxx.xxx.xxx/configMapPriorityEnableChange>

Body Syntax Example

```
[
  {
    "name": "CM1",
    "enabled": true
  },
  {
    "name": "CM2",
    "enabled": false
  },
  {
    "name": "CM3",
    "enabled": true
  }
]
```

Note: String the config maps as shown in the above example. Config maps may be enabled or disabled by modifying the “enabled” option true/false, true = enabled, false = disabled.

Delete Config Map

POST <https://xxx.xxx.xxx.xxx/configMapRemove>

Body Syntax Example

```
["CM1", "CM2", "CM3"]
```

Note: Identify the config map(s) using the "name".

Clear Config Map Counters

POST <https://xxx.xxx.xxx.xxx/configMapCountsClear>

Refresh Config Map Counters

GET <https://xxx.xxx.xxx.xxx/configMapCounts>

Display Port Configuration

GET <https://xxx.xxx.xxx.xxx/portConfigGet>

Modify Port Configuration

PUT <https://xxx.xxx.xxx.xxx/portConfigPut>

Body Syntax Example

```
{ "ports": [ { "portNumber": "1", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "2", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "3", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" }, {  
portNumber": "4", "speedSet": "10G", "mode": "normal", "description": "port  
description", "sfpvendorPn": "", "vlanIdTagAdd": "0", "vlanIdTagStrip": "0", "split": "NO" } ] }
```

Note: Whenever modifying a port(s) all ports must be included in the body syntax.

Display Port Availability

GET <https://xxx.xxx.xxx.xxx/portAvailability>

Display Port Statistics

GET <https://xxx.xxx.xxx.xxx/portCounterGet>

Clear Port Statistics

PUT <https://xxx.xxx.xxx.xxx/portCounterClear>