PacketController API Specification

Version: 7.0.6
Update: Dec. 2020

PacketController Networks

Disclaimer

Table of Contents

Introduction

This document describes the API Interface to the PacketController Product. It describes in detail how to configure the various features of the PacketController product using the API.

Related Firmware Version

Published with PacketController version 7.0.1. This document has not required changes since 7.0.1. However, the content is in sync with the latest PacketController firmware.

Overview

The API works by allowing a user or application to pass HTTPS requests to the PacketController product. The PacketController answers the request with an XML or Json formatted response.

Support HTTP Method

The APIs can be accessed by HTTPs request (either GET or POST method). The API client can be anything which could perform HTTP request, for instance:
Browser
wget
curl

Support Payload Format

Supported payload formats includes:

- XML
- JSON

The default payload format is XML, but it can be changed in runtime by appending JsonFormat=1 to the HTTP request, for instance:

https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getsp&**JsonFormat=1**

URL Format

URL Format is as below:
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getsp&{params}={params}

Property	**Description**
IP	The ip address of PacketController management port.
user	The API user
token	The token for this API user
mod	The module identifier, it includes: qos subscriber report
cmd	The API method
params	The properties of this API method, the format is as below: &param1=value1&param2=value2

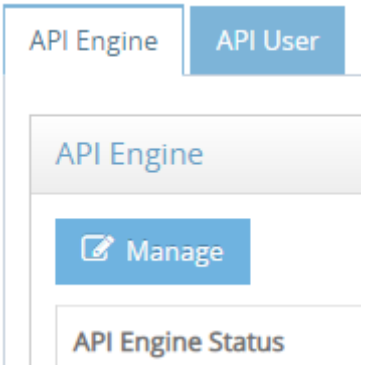
Response Code

For successful API call, the response state is 200 and code is ok.
For failed API call, the response state is 500 and code is error.

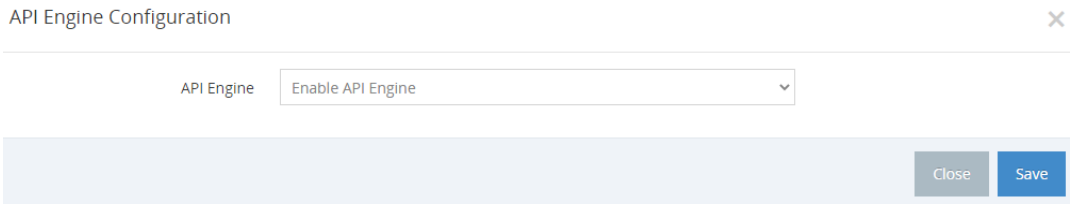
Enable API Engine

The API engine should be enabled before accessing API.

- Navigate to **System->API** and click API Engine tab



- Click **Manage** button

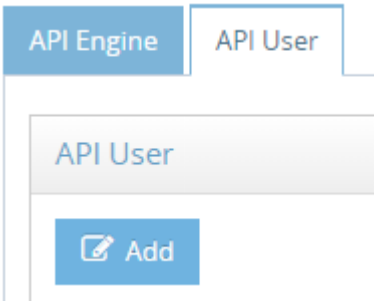


- Select Enable API Engine and click **Save** button

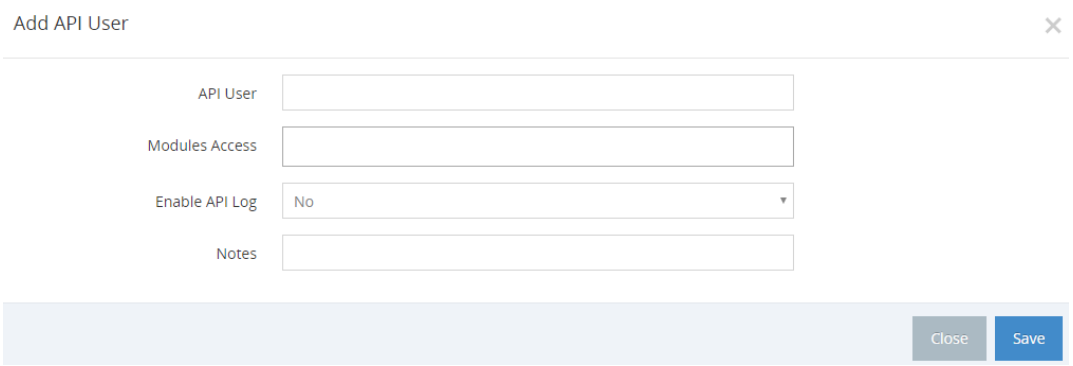
API User Setup

To perform API call, one api user is required to be configured.

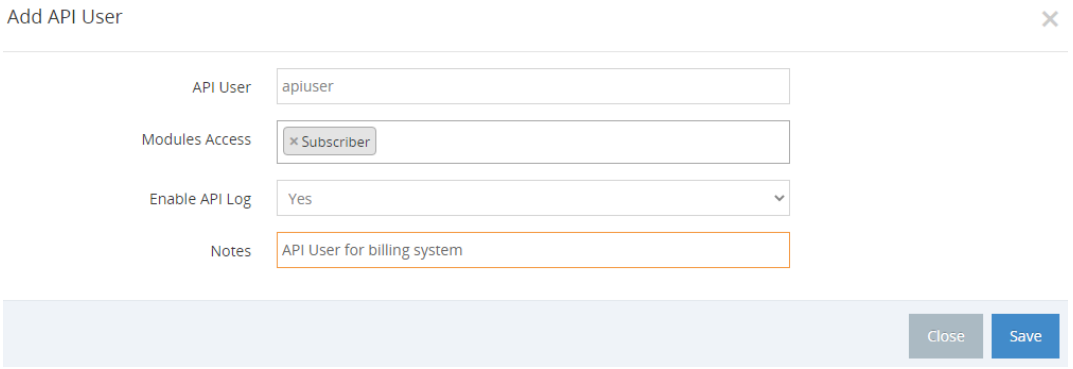
- navigate to **System -> API**, click API User Tab and click Add button



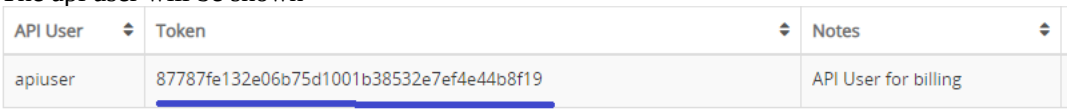
- Add API User page will pop up



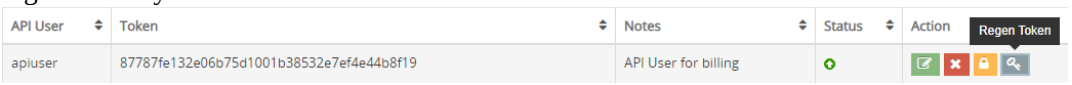
- Input the settings and click Save



- The api user will be shown



Notes: The token is generated automatically after the api user added. And this token can be regenerated by click  icon in Action column.

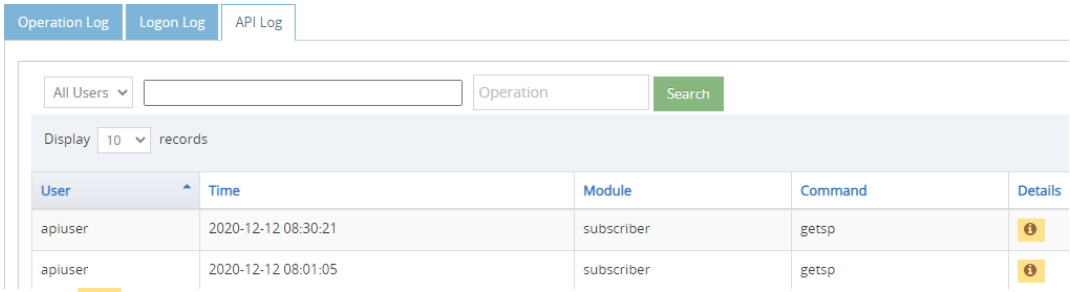


Settings	Description
API User	The name for this api user
Module Access	The modules access for this api user, it includes: QoS Subscriber Logs & Reports
Enable API Log	Whether to log the operation of this api user
Notes	The notes for this api user

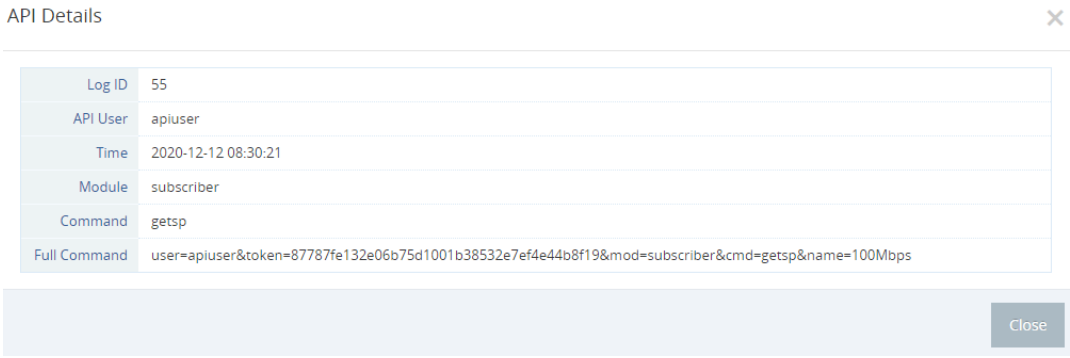
API Logs

The API logs is stored if enabled for API user.

- navigate to **Los & Reports -> Audit Log**, click API Log Tab



- Click icon in Details column



Commands for Subscriber Module

These are commands of subscriber module. The module identifier is "subscriber" The commands currently supported is.

Command	Description
listsp	The command to return all the service plans
getsp	The command to return one specific service plan
addsp	The command to add one service plan
listtp	The command to return all the time plans
gettp	The command to return one specific time plan
listusergrp	The command to return all the user groups
getusergrp	The command to return one specific user group
listcontgrp	The command to return all the contention ratio groups
getcontgrp	The command to return one specific contention ratio group
listsubscriber	The command to return all the subscribers
getsubscriber	The command to return one specific subscriber
addsubscriber	The command to add one subscriber
delsubscriber	The command to delete one subscriber
disablesubscriber	The command to disable one subscriber
enablesubscriber	The command to enable one subscriber
regensubscriber	The command to re-generate the rules of one subscriber

listsp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=listsp

XML Output

```
<response state="200" code="ok">
<data>
<serviceplan>100Mbps</serviceplan>
<serviceplan>10Mbps</serviceplan>
<serviceplan>1Mbps</serviceplan>
<serviceplan>5Mbps</serviceplan>
</data>
</response>
```

Json Output

```
{
    "serviceplan": [
        "100Mbps",
        "10Mbps",
        "1Mbps",
        "5Mbps"
    ],
    "state": 200,
    "code": "ok"
}
```

getsp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getsp&name={service plan name}

Properties	Description	Type	Values
name	The name of service plan	string	String length is 5-40

addsp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=addsp&name={service plan name}&bwin={The download speed in bps}&bwout={The upload speed in bps}

Properties	Description	Type	Values
name	The name of service plan	string	String length is 5-40
bwin	The download speed and the parameter are bps	Integer	0 - 40000000000
bwout	The upload speed and the parameter are bps	integer	0 - 40000000000

listtp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=listtp

XML Output

```
<response state="200" code="ok">
<data>
<timeplan>newtp1</timeplan>
</data>
</response>
```

Json Output

```
{
    "timeplan": [
        "newtp1"
    ],
    "state": 200,
    "code": "ok"
}
```

gettp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=gettp&name={time plan name}

XML Output

```
<response state="200" code="ok">
<data>
<timeplan>
<id>8</id>
<profile>newtp1</profile>
<times>6:00am,6:00pm</times>
<actions>1Mbps,5Mbps</actions>
<days>Sun,Mon,Tue,Wed,Thu,Fri,Sat</days>
</timeplan>
</data>
</response>
```

Json Output

```
{
    "timeplan": {
        "id": "8",
        "profile": "newtp1",
        "times": "6:00am,6:00pm",
        "actions": "1Mbps,5Mbps",
        "days": "Sun,Mon,Tue,Wed,Thu,Fri,Sat"
    },
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of time plan	string	String length is 5-40

listusergrp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=listusergrp

XML Output
```
<response state="200" code="ok">
<data>
<usergrp>statgroup</usergrp>
</data>
</response>
```

Json Output
```
{
    "usergrp": [
        "statgroup"
    ],
    "state": 200,
    "code": "ok"
}
```

getusergrp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getusergrp&name={user group name}

XML Output
```
<response state="200" code="ok">
<data>
<usergrp>
<id>45</id>
<name>statgroup</name>
<balanced_group>0</balanced_group>
<ifname>em0</ifname>
<ptype>0</ptype>
<bwprofile>100Mbps</bwprofile>
<time_profiles/>
<notes>notes</notes>
<premium>0</premium>
</usergrp>
</data>
</response>
```

Json Output
```
{
    "usergrp": {
        "id": "45",
        "name": "statgroup",
        "balanced_group": "0",
        "ifname": "em0",
        "ptype": "0",
```

```
        "bwprofile": "100Mbps",
        "time_profiles": "",
        "notes": "notes",
        "premium": "0"
    },
    "state": 200,
    "code": "ok"
}
```


Properties	Description	Type	Values
name	The name of user group	string	String length is 5-40

listcontgrp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=listcontgrp

XML Output


```
<response state="200" code="ok">
<data>
<contgrp>content101</contgrp>
<contgrp>content51</contgrp>
</data>
</response>
```


Json Output


```
{
    "contgrp": [
        "content101",
        "content51"
    ],
    "state": 200,
    "code": "ok"
}
```


getcontgrp

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getcontgrp&name={contention ratio group name}

XML Output


```
<response state="200" code="ok">
<data>
<contgrp>
<id>7</id>
<name>content101</name>
<ratio>10</ratio>
<balanced_group>0</balanced_group>
<ifname>em0</ifname>
<notes>contention1</notes>
<premium>0</premium>
```

```
</contgrp>
</data>
</response>
```

Json Output

```
{
    "contgrp": {
        "id": "7",
        "name": "content101",
        "ratio": "10",
        "balanced_group": "0",
        "ifname": "em0",
        "notes": "contention1",
        "premium": "0"
    },
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of contention ratio group	string	String length is 5-40

listsubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=listsubscriber

XML Output

```
<response state="200" code="ok">
<data>
<subscriber>demouser</subscriber>
<subscriber>ivanuser</subscriber>
<subscriber>user1</subscriber>
</data>
</response>
```

Json Output

```
{
    "subscriber": [
        "demouser",
        "ivanuser",
        "user1"
    ],
    "state": 200,
    "code": "ok"
}
```

getsubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=getsubscriber&name={subscriber name}

XML Output

```
<response state="200" code="ok">
<data>
<subscriber>
<id>5</id>
<name>demouser</name>
<email>support@packetcontroller.com</email>
<ptype>0</ptype>
<plans>100Mbps</plans>
<gtype>2</gtype>
<ugroup>content101</ugroup>
<addr>192.168.0.225</addr>
<maddr/>
<vlan/>
<password>password</password>
<enabled>1</enabled>
<premium>0</premium>
<port>em0</port>
<tcpopt>0</tcpopt>
</subscriber>
</data>
</response>
```

Json Output

```
{
    "subscriber": {
        "id": "5",
        "name": "demouser",
        "email": "support@packetcontroller.com",
        "ptype": "0",
        "plans": "100Mbps",
        "gtype": "2",
        "ugroup": "content101",
        "addr": "192.168.0.225",
        "maddr": "",
        "vlan": "",
        "password": "password",
        "enabled": "1",
        "premium": "0",
        "port": "em0",
        "tcpopt": "0"
    },
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20

addsubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=addsubscriber&name={subscriber name}&port={port name}&premium={premium flag}&tcpopt={tcp optimization flag}&email={email}&password={password}&ptype={plan type}&plans={plan}&gtype={group type}&ugroup={user group}&maddr={mac address}&vlan={vlan id}&addr={ip address}&notes={notes}

XML Output

```
<response state="200" code="ok">
<data>
<subscriber>
<id>5</id>
<name>demouser</name>
<email>support@packetcontroller.com</email>
<ptype>0</ptype>
<plans>100Mbps</plans>
<gtype>2</gtype>
<ugroup>content101</ugroup>
<addr>192.168.0.225</addr>
<maddr/>
<vlan/>
<password>password</password>
<enabled>1</enabled>
<premium>0</premium>
<port>em0</port>
<tcpopt>0</tcpopt>
</subscriber>
</data>
</response>
```

Json Output

```
{
    "subscriber": {
        "id": "5",
        "name": "demouser",
        "email": "support@packetcontroller.com",
        "ptype": "0",
        "plans": "100Mbps",
        "gtype": "2",
        "ugroup": "content101",
        "addr": "192.168.0.225",
        "maddr": "",
        "vlan": "",
        "password": "password",
        "enabled": "1",
        "premium": "0",
        "port": "em0",
```

```
        "tcpopt": "0"
    },
    "state": 200,
    "code": "ok"
}
```


Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20, it is mandatory The name can only contain alphanumeric characters,underscore and hyphen.
port	The external port of bridge	String	it is mandatory, the external port of the bridge can be seen in Network->Port of WUI. Please note if there are multiple bridges, there will be multiple external ports.
premium	Whether it is premium	String	It is optional Set it to on if it is premium. The format is premium=on
tcpopt	Whether to enable TCP optimization	String	It is optional Set it to on if tcp optimization required. The format is tcpopt=on
email	The primary email account of subscriber	Email	It is optional
password	The password to access subscriber portal	String	It is mandatory. The length is 5-20
ptype	The plan type	Integer	It is mandatory 0:service plan 1:time plan 2:quota plan 3:app plan
plans	The plan	String	It is mandatory Depending on ptype, the actual plan for the subscriber. For instance, if ptype is set to 0, then set one service plan to subscriber. The service plan name list can be retrieved via api getsp

gtype	The group type	Integer	It is mandatory 0:None 1:User group 2:Contention Ratio
ugroup	The group	String	Depending on gtype, the actual group for the subscriber. If gtype is 0, then this field is not required. If gtype is 1, this field is mandatory, the user group name list can be retrieved via api listusergrp. If gtype is 2, this field is mandatory, the contention ratio group name list can be retrieved via api listcontgrp.
maddr	The mac address	Mac address	It is optional The mac address format is in 00:11:22:33:44:55:66
vlan	The vlan ID	Integer	It is optional The vlan id, 2 – 4093 Please note that VLAN must be in 802.1Q format
addr	The ip address	String	It is mandatory Comma-separated ip or subnet list, the following are valid 10.10.10.2,10.10.10.3 Or 10.10.10.0/24,10.10.1.0/24 Or 10.10.10.1
Notes	The comments for this subscriber	String	It is optional. The valid character length 5-250

delsubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=delsubscriber&name={subscriber name}

XML Output

```
<response state="200" code="ok">
<data>
<message>The subscriber user1 deleted</message>
</data>
</response>
```

Json Output

```
{
    "message": "The subscriber user1 deleted",
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20

disablesubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=disablesubscriber&name={subscriber name}

XML Output

```
<response state="200" code="ok">
<data>
<message>The subscriber user1 disabled</message>
</data>
</response>
```

Json Output

```
{
    "message": "The subscriber user1 disabled",
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20

enablesubscriber

API Call
https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=enablesubscriber&name={subscriber name}

XML Output

```
<response state="200" code="ok">
<data>
<message>The subscriber user1 enabled</message>
</data>
</response>
```

Json Output

```
{
    "message": "The subscriber user1 enabled",
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20

regensubscriber

API Call

https://{IP}/api?user={user}&token={token}&mod=subscriber&cmd=regensubscriber&name={subscriber name}

XML Output

```
<response state="200" code="ok">
<data>
<message>The subscriber user1 rule re-generated</message>
</data>
</response>
```

Json Output

```
{
    "message": "The subscriber user1 rule re-generated",
    "state": 200,
    "code": "ok"
}
```

Properties	Description	Type	Values
name	The name of subscriber	string	String length is 5-20