

Содержание

5 Description of the JSON-RPC software interface	3
Description	3
Common Functions	5
Authentication and authorization functions	6
User profile management	9
User management	12
Roles management	15
Device control	18
Notifications management	46
Errors sending	50
Subscriber management	52
Services management	60
Tariff management	60
Protocol prioritization management (DSCP)	60
ASN prioritization management	63

5 Description of the JSON-RPC software interface

Description

All requests to the server are sent by the POST method. The response headers contain Cookie. They have to be transmitted in all the subsequent requests.

An array of [] objects of following type is transferred within the request body

The example of request:

```
{
  "jsonrpc": "2.0",
  "method": "Api_GetAppVersion", -- the method name
  "params": {}, -- Object with parameters
  "id": 1515659482430 - request identifier (it can be any random number)
}
```

If the request contains an array of several objects, then corresponding response will also contain an array. If the request contains an array of just one object, then corresponding response will return one object instead, not an array.

Note: the letter case and the type of parameters matters.

The response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515659482430, -- identifier specified in request
  "result": {
    "success": true, -- result (Boolean value)
    "data": "2.0.0" -- data (string, array, object)
  }
}
```

The response may be returned with an error. Errors are divided into two categories.

- Global - are processed at the json-rpc module level. For example incorrect parameters were passed. The error is displayed in the error field.

```
{
  "jsonrpc": "2.0",
  "id": 1515660273515,
  "error": {
    "code": 6000,
    "message": "Invalid parameters",
    "data": {
      "errors": [
```

```

        {
            "code": "required_field",
            "message": "Mandatory parameter is not passed or it
is empty",
            "object_name": "Test"
        }
    ]
}
}
}
}

```

- Local - are processed at the API level inside the called function. For example, error occurred while writing to the database. The error is displayed in the result.error field. In this case, result.success = false.

```

{
  "jsonrpc": "2.0",
  "id": 1515660902722,
  "result": {
    "success": false,
    "error": {
      "code": -203,
      "msg": "Duplicate entry"
    },
    "data": null
  }
}
}

```

Some example requests with CURL

Get the application version:

```

curl -b cookie.txt -c cookie.txt -X POST -k -i
'https://192.168.1.123/api/jsonrpc' --data
' [{"jsonrpc": "2.0", "method": "Api_GetAppVersion", "params": {}, "id": 15634364952
88}] '

```

Make authorization:

```

curl -b cookie.txt -c cookie.txt -X POST -k -i
'https://192.168.1.123/api/jsonrpc' --data
' [{"jsonrpc": "2.0", "method": "Api_Login", "params": {"username": "admin", "passwo
rd": "vasexperts", "remember": t
rue}, "id": 1563438645838}] '

```

Get DPI configuration file:

```
curl -b cookie.txt -c cookie.txt -X POST -k -i
'https://192.168.1.123/api/jsonrpc' --data
' [{"jsonrpc": "2.0", "method": "Api_GetDpiConfig", "params": {"Id": 1}, "id": 1563448913831}]'
```

Common Functions

Api_GetAppVersion

Displays the application version.

There are no parameters.

The request example:

```
[{
  "jsonrpc": "2.0",
  "method": "Api_GetAppVersion",
  "params": {},
  "id": 1515659482430
}]
```

The response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515659482430,
  "result": {
    "success": true,
    "data": "2.0.0"
  }
}
```

Api_GetDics

Outputs dictionaries.

Parameters:

- DicsKeys - Array: list of dictionary identifiers. Displays all of them by default.

The request example:

```
[
  {
    "jsonrpc": "2.0",
```

```
    "method": "Api_GetDics",
    "params": {},
    "id": 1515921551526
  }
]
```

The response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515921551526,
  "result": {
    "success": true,
    "data": {
      "RolesDic": {
        "1": {
          "dic_id": "1",
          "value": "Administrator"
        }
      },
      "HardwaresDic": {
        "1": {
          "dic_id": "1",
          "value": "Test bed. 34",
          "ip": "x.x.x.x",
          "port": "22",
          "login": "dписu"
        },
        "2": {
          "dic_id": "2",
          "value": "Operational test bed .83 !",
          "ip": "y.y.y.y",
          "port": "22",
          "login": "dписu"
        }
      }
    }
  }
}
```

Authentication and authorization functions

- [Api_Auth](#)
- [Api_Login](#)
- [Api_Logout](#)

Api_Auth

Checks and displays the result: whether the user is authorized or not.

There are no parameters.

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_Auth",
    "params": {
    },
    "id": 1515661809137
  }
]
```

The response example in case the user is not authorized:

```
{
  "jsonrpc": "2.0",
  "id": 1515661809137,
  "error": {
    "code": 7000,
    "message": "Unauthorized"
  }
}
```

The response example in case the user is authorized:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "username": "admin"
    }
  }
}
```

Api_Login

Authorizes the user.

Parameters:

- username - string:the user name
- password - string:password

- remember - bool:password

The request example

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_Login",
    "params": {
      "username": "admin",
      "password": "vasexperts",
      "remember": true
    },
    "id": 1515662219320
  }
]
```

The example of a successful response:

```
{
  "jsonrpc": "2.0",
  "id": 1515662219320,
  "result": {
    "success": true,
    "data": {
      "username": "admin"
    }
  }
}
```

The example of a response if something's gone wrong:

```
{
  "jsonrpc": "2.0",
  "id": 1515662233320,
  "result": {
    "success": false,
    "error": {
      "code": 404,
      "msg": "Invalid credentials"
    },
    "data": null
  }
}
```

If the authorization is successfully completed, cookies will be transmitted in the response headers. They must be transmitted in all subsequent requests.

Api_Logout

Deauthorizes the current user.

There are no parameters.

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_Logout",
    "params": {
    },
    "id": 1515662234646
  }
]
```

User profile management

- [Api_GetMyProfile](#)
- [Api_SaveMyProfile](#)
- [Api_ChangeMyPassword](#)

Api_GetMyProfile

Displays the user profile.

There are no parameters.

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetMyProfile",
    "params": {
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
```

```
"jsonrpc": "2.0",
"id": 1515662165576,
"result": {
  "success": true,
  "data": {
    "user_id": "1",
    "username": "admin",
    "name": "Strogiy S.S.",
    "email": "email@email.ru",
    "phone": "2-2-2",
    "company": "VasExpert",
    "position": "Administrator"
  }
}
}
```

Api_SaveMyProfile

The user profile editing.

Parameters:

- MyProfile - object: the user model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveMyProfile",
    "params": {
      "MyProfile": {
        "user_id": "1",
        "username": "admin",
        "name": "Strogiy S.S.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Administrator"
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
```

```
"jsonrpc": "2.0",
"id": 1515662165576,
"result": {
  "success": true,
  "data": {
    "user_id": "1",
    "username": "admin",
    "name": "Strogiy S.S.",
    "email": "email@email.ru",
    "phone": "2-2-2",
    "company": "VasExpert",
    "position": "Administrator"
  }
}
}
```

Api_ChangeMyPassword

The user password changing.

Parameters:

- OldPassword - object:old password
- NewPassword - string:new password

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveMyProfile",
    "params": {
      "OldPassword": "vasexperts",
      "NewPassword": "vasexperts1"
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
```

```

        "user_id": "1",
        "username": "admin",
        "name": "Strogiy S.S.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Administrator",
        "password":
"$2y$10$t/PdsQVXr927wyunbdET1.EHtuxTBg4iKmtHlJ4jfmU0XR4qPUANu"
    }
}
}

```

User management

- [Api_GetUsers](#)
- [Api_SaveUser](#)
- [Api_DeleteUser](#)

Api_GetUsers

Allows to get list of users.

Parameters:

- Id - int:user identifier(optional parameter)

The request example using the user identifier:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetUsers",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]

```

The example of successful response to request containing the user identifier:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {

```

```

    "success":true,
    "data": {
      "user_id": "1",
      "username": "admin",
      "name": "Strogiy S.S.",
      "email": "email@email.ru",
      "phone": "2-2-2",
      "company": "VasExpert",
      "position": "Administrator",
      "role_sections": "hardware.read,hardware.write,..."
    }
  }
}

```

The example of successful response to request without parameters:

```

{
  "jsonrpc":"2.0",
  "id":1515662165576,
  "result":{
    "success":true,
    "data": [
      {
        "user_id": "1",
        "username": "admin",
        "name": "Strogiy S.S.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Administrator",
        "role_sections": "hardware.read,hardware.write,..."
      },
      ...
    ]
  }
}

```

Api_SaveUser

Create/modify the user data.

Parameters:

- user - object:the user model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveUser",
    "params": {
      "user": {
        "user_id": "1",
        "username": "admin",
        "name": "Strogiy S.S.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Administrator",
        "password": "vasexperts",
        "role": "1",
        "role_sections": "hardware.read,hardware.write,..."
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "user_id": "1",
      "username": "admin",
      "name": "Strogiy S.S.",
      "email": "email@email.ru",
      "phone": "2-2-2",
      "company": "VasExpert",
      "position": "Administrator",
      "password":
"$2y$10$rxlWVJdRybSf9N6nAQE9j.i2LrSTbpGzoiDwsVVsAP90Q5vDY0uhu"
    }
  }
}
```

Api_DeleteUser

Deletes the user.

Parameters:

- user - object:the user model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteUser",
    "params": {
      "user": {
        "user_id": "2",
        "username": "admin",
        "name": "Strogiy S.S.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Administrator",
        "password": "vasexperts",
        "role": "1",
        "role_sections": "hardware.read,hardware.write,..."
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Roles management

- [Api_GetRoles](#)
- [Api_SaveRole](#)
- [Api_DeleteRole](#)

Api_GetRoles

Allows to get a list of users.

Parameters:

- Id - int:role identifier(optional parameter)

The example of request with the role ID:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetRoles",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The example of successful response to request containing the role ID:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "role_id": "1",
        "role": "Администратор",
        "sections": "hardware.read,hardware.write,..."
      }
    ]
  }
}
```

The example of a successful response to a request without parameters:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "role_id": "1",
        "role": "Administrator",

```



```

        "sections": "hardware.read,hardware.write,..."
    },
    ...
]
}
}

```

Api_SaveRole

Creating/modifying the role data.

Parameters:

- role - object:the role model

The request example:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveRole",
    "params": {
      "role": {
        "role_id": "1",
        "role": "Administrator",
        "sections": "hardware.read,hardware.write,..."
      }
    },
    "id": 1515661809137
  }
]

```

The successful response example:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "role_id": "1",
      "role": "Administrator",
      "sections": "hardware.read,hardware.write,..."
    }
  }
}

```

```
}
```

Api_DeleteRole

Deleting a role.

Parameters:

- role - object:the role model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteRole",
    "params": {
      "role": {
        "role_id": "1",
        "role": "Administrator",
        "sections": "hardware.read,hardware.write,..."
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Device control

- [Api_GetHardwares](#)
- [Api_SaveHardware](#)
- [Api_DeleteHardware](#)
- [Api_RestartHardware](#)
- [Api_ReloadHardware](#)
- [Api_TestConnectToHardware](#)
- [Api_GetDpiConfig](#)

- [Api_ValidateDpiConfig](#)
- [Api_SetDpiConfig](#)
- [Api_GetDpiConfigHistory](#)
- [Api_GetDpiConfigFile](#)
- [Api_GetDpiInfo](#)
- [Api_GetDpiResourcesUsageTick](#)
- [Api_GetDpiResourcesStatsUsageForPeriod](#)
- [Api_ProcessDpiStatLog](#)
- [Api_GetStatLogTail](#)
- [Api_DownloadStatLog](#)
- [Api_GetAlertLogTail](#)
- [Api_DownloadAlertLog](#)

Api_GetHardwares

Get the list of equipments.

Parameters:

- Id - int:hardware identifier(optional identifier)

The request using the hardware identifier example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetHardwares",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

Example of a successful response to a request with a hardware identifier:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "hardware_id": "1",
        "name": "Test bed. 34",
        "ip": "188.227.73.34",
        "port": "22",
        "login": "admin",

```

```

        "password": "vasexperts",
        "ssl_key": "",
        "sudocheck": "1",
        "load_cs": 0,
        "port_cs": "1500",
        "protocol_cs": "udp",
        "ip_cs": "217.71.228.148",
        "status_cs": 0
    }
]
}
}

```

The example of successful response to the request without parameters:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "hardware_id": "1",
        "name": "Test bed. 34",
        "ip": "188.227.73.34",
        "port": "22",
        "login": "admin",
        "password": "vasexperts",
        "ssl_key": "",
        "sudocheck": "1",
        "load_cs": 0,
        "port_cs": "1500",
        "protocol_cs": "udp",
        "ip_cs": "217.71.228.148",
        "status_cs": 0
      },
      ...
    ]
  }
}

```

Api_SaveHardware

Creation/modification of the equipment data.

Parameters:

- hardware - object:the equipment model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveHardware",
    "params": {
      "hardware": {
        "hardware_id": "1",
        "name": "Test bed. 34",
        "ip": "188.227.73.34",
        "port": "22",
        "login": "admin",
        "password": "vasexperts",
        "ssl_key": "",
        "sudocheck": "1",
        "load_cs": 0,
        "port_cs": "1500",
        "protocol_cs": "udp",
        "ip_cs": "217.71.228.148",
        "status_cs": 0
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "hardware_id": "1",
      "name": "Test bed. 34",
      "ip": "188.227.73.34",
      "port": "22",
      "login": "admin",
      "password": "vasexperts",
      "ssl_key": "",
      "sudocheck": "1",
      "load_cs": 0,
      "port_cs": "1500",
      "protocol_cs": "udp",
      "ip_cs": "217.71.228.148",

```

```
        "status_cs": 0
    }
}
}
```

Api_DeleteHardware

Equipment removal.

Параметры:

- hardware - object:the equipment model

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteRole",
    "params": {
      "hardware": {
        "hardware_id": "1",
        "name": "Test bed. 34",
        "ip": "188.227.73.34",
        "port": "22",
        "login": "admin",
        "password": "vasexperts",
        "ssl_key": "",
        "sudocheck": "1",
        "load_cs": 0,
        "port_cs": "1500",
        "protocol_cs": "udp",
        "ip_cs": "217.71.228.148",
        "status_cs": 0
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

```
}  
  
}
```

Api_RestartHardware

Equipment restart.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[  
  {  
    "jsonrpc": "2.0",  
    "method": "Api_RestartHardware",  
    "params": {  
      "Id": 1  
    },  
    "id": 1515661809137  
  }  
]
```

The successful response example:

```
{  
  
  "jsonrpc": "2.0",  
  "id": 1515662165576,  
  "result": {  
    "success": true  
  }  
}
```

Api_ReloadHardware

On-the-fly parameters updating.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[
```

```
{
  "jsonrpc": "2.0",
  "method": "Api_ReloadHardware",
  "params": {
    "Id": 1
  },
  "id": 1515661809137
}
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Api_TestConnectToHardware

Equipment availability test.

Parameters:

- Id - int: the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_TestConnectToHardware",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
```



```
"id":1515662165576,
"result":{
  "success":true
}
}
```

Api_GetDpiConfig

Get the DPI equipment configuration.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[
  {
    "jsonrpc":"2.0",
    "method":"Api_GetDpiConfig",
    "params":{
      "Id":1
    },
    "id":1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc":"2.0",
  "id":1515662165576,
  "result":{
    "success":true,
    "data":"string:content"
  }
}
```

Api_ValidateDpiConfig

Validation of the DPI equipment configuration.

Parameters:

- Id - int:the equipment identifier

- Config - string: configuration

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_ValidateDpiConfig",
    "params": {
      "Id": 1,
      "Config": "string:content"
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": "FastDPI 7.4 Minsk (Dec 12 2017) : Check configuration  
'/tmp/dpi/fastdpi.conf' : \n\nResult check  
configuration : SUCCESS\n"
  }
}
```

Api_SetDpiConfig

Setting DPI configuration to the equipment.

Parameters:

- Id - int: the equipment identifier
- Config - string: configuration

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SetDpiConfig",
    "params": {
      "Id": 1,
      "Config": "string:content"
    }
  }
]
```

```
    },
    "id":1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc":"2.0",
  "id":1515662165576,
  "result":{
    "success":true
  }
}
```

Api_GetDpiConfigHistory

Getting the history of DPI configurations on the hardware.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[
  {
    "jsonrpc":"2.0",
    "method":"Api_GetDpiConfigHistory",
    "params":{
      "Id":1
    },
    "id":1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc":"2.0",
  "id":1515662165576,
  "result":{
    "success":true,
    "data": [
      "2015.03.22.03.26.12.000000.fastdpi.conf",

```

```

        "2015.03.22.03.26.22.000000.fastdpi.conf",
        "2015.03.22.19.03.53.000000.fastdpi.conf",
        "2015.03.22.19.44.35.000000.fastdpi.conf",
        "2015.03.23.01.30.11.000000.fastdpi.conf",
        "2015.03.23.01.31.26.000000.fastdpi.conf",
        ...
    ]
}
}

```

Api_GetDpiConfigFile

Get the content of DPI configuration file.

Parameters:

- Id - int:the equipment identifier
- File - string:the file name

The request example:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDpiConfigFile",
    "params": {
      "Id": 1,
      "File": "2015.03.22.03.26.12.000000.fastdpi.conf"
    },
    "id": 1515661809137
  }
]

```

The successful response example:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": "string:content"
  }
}

```

Api_GetDpiInfo

To get the DPI equipment information.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDpiInfo",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "FastDPI": "FastDPI 7.4 Minsk (Dec 12 2017)",
      "Architecture": "x86_64",
      "CPU op-mode(s)": "32-bit, 64-bit",
      "Byte Order": "Little Endian",
      "CPU(s)": "8",
      "On-line CPU(s) list": "0-7",
      "Thread(s) per core": "2",
      "Core(s) per socket": "4",
      "Socket(s)": "1",
      "NUMA node(s)": "1",
      "Vendor ID": "GenuineIntel",
      "CPU family": "6",
      "Model": "58",
      "Stepping": "9",
      "CPU MHz": "3392.160",
      "BogoMIPS": "6784.32",
      "Virtualization": "VT-x",
      "L1d cache": "32K",
      "L1i cache": "32K",
      "L2 cache": "256K",
    }
  }
}
```

```
        "L3 cache": "8192K",
        "NUMA node0 CPU(s)": "0-7"
    }
}
}
```

Api_GetDpiResourcesUsageTick

To get information about the current state of equipment performance.

Parameters:

- Id - int:the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDpiResourcesUsageTick",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "cpus_info": null,
      "cpus_usage": {
        "stat_key": 1,
        "us": 0.40000000000000002,
        "sy": 3.5,
        "ni": 0,
        "idle": 95.299999999999997,
        "wa": 0.69999999999999996,
        "hi": 0,
        "si": 0,
        "st": 0,
        "date": "2018.01.13 10:43:03"
      }
    }
  }
}
```

```
},
"mem_usage": {
  "stat_key": 2,
  "total": 8030376000,
  "used": 2301548000,
  "free": 5728828000,
  "buffers": 138556000,
  "date": "2018.01.13 10:43:03"
},
"swap_usage": {
  "stat_key": 2,
  "total": 2097144000,
  "used": 42040000,
  "free": 2055104000,
  "cached": 795280000,
  "date": "2018.01.13 10:43:03"
},
"top_processes": [
  {
    "pid": "23650",
    "virt": "10.0g",
    "res": "997m",
    "shr": "10m",
    "cpu": 110.59999999999999,
    "mem": 12.699999999999999,
    "command": "fastdpi_main",
    "date": null
  },
  {
    "pid": "13347",
    "virt": "98.0m",
    "res": "3960",
    "shr": "3000",
    "cpu": 9.9000000000000004,
    "mem": 0,
    "command": "sshd",
    "date": null
  },
  {
    "pid": "13342",
    "virt": "15036",
    "res": "1380",
    "shr": "996",
    "cpu": 4,
    "mem": 0,
    "command": "top",
    "date": null
  },
  {
    "pid": "1084",
    "virt": "0",
```

```
    "res": "0",
    "shr": "0",
    "cpu": 1,
    "mem": 0,
    "command": "kauditd",
    "date": null
  },
  {
    "pid": "13351",
    "virt": "98.0m",
    "res": "2004",
    "shr": "1024",
    "cpu": 1,
    "mem": 0,
    "command": "sshd",
    "date": null
  },
  {
    "pid": "1",
    "virt": "19232",
    "res": "884",
    "shr": "748",
    "cpu": 0,
    "mem": 0,
    "command": "init",
    "date": null
  },
  {
    "pid": "2",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "kthreadd",
    "date": null
  },
  {
    "pid": "3",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/0",
    "date": null
  },
  {
    "pid": "4",
    "virt": "0",
    "res": "0",
```



```
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "ksoftirqd/0",
    "date": null
  },
  {
    "pid": "5",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/0",
    "date": null
  },
  {
    "pid": "6",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "watchdog/0",
    "date": null
  },
  {
    "pid": "7",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/1",
    "date": null
  },
  {
    "pid": "8",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/1",
    "date": null
  },
  {
    "pid": "9",
    "virt": "0",
    "res": "0",
    "shr": "0",
```

```
    "cpu": 0,  
    "mem": 0,  
    "command": "ksoftirqd/1",  
    "date": null  
  },  
  {  
    "pid": "10",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "watchdog/1",  
    "date": null  
  },  
  {  
    "pid": "11",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "migration/2",  
    "date": null  
  },  
  {  
    "pid": "12",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "migration/2",  
    "date": null  
  },  
  {  
    "pid": "13",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "ksoftirqd/2",  
    "date": null  
  },  
  {  
    "pid": "14",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,
```



```
    "command": "migration/4",
    "date": null
  },
  {
    "pid": "20",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/4",
    "date": null
  },
  {
    "pid": "21",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "ksoftirqd/4",
    "date": null
  },
  {
    "pid": "22",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "watchdog/4",
    "date": null
  },
  {
    "pid": "23",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/5",
    "date": null
  },
  {
    "pid": "24",
    "virt": "0",
    "res": "0",
    "shr": "0",
    "cpu": 0,
    "mem": 0,
    "command": "migration/5",
```

```

        "date": null
    },
    {
        "pid": "25",
        "virt": "0",
        "res": "0",
        "shr": "0",
        "cpu": 0,
        "mem": 0,
        "command": "ksoftirqd/5",
        "date": null
    },
    {
        "pid": "26",
        "virt": "0",
        "res": "0",
        "shr": "0",
        "cpu": 0,
        "mem": 0,
        "command": "watchdog/5",
        "date": null
    }
],
"date": "2018.01.13 10:43:03",
"cpus_count": "8"
}
}
}
}

```

Api_GetDpiResourcesStatsUsageForPeriod

To get information about the DPI equipment performance for the period.

Параметры:

- Id - int:the equipment identifier
- Period - object:... (optional parameter)
- PeriodType - string:... (optional parameter)

The request example:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDpiResourcesStatsUsageForPeriod",
    "params": {
      "Id": 1,
      "Period": {
        "start": "2018-01-13T21:00:00.000Z",

```

```
        "end": "2018-01-13T21:00:00.000Z"
      },
      "PeriodType": "H"
    },
    "id": 1515921656764
  }
}
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515921656764,
  "result": {
    "success": true,
    "data": {
      "2018-01-14 11": {
        "cpus_usage": {
          "stat_key": "1",
          "us": "0.12121212121212127",
          "sy": "0.47272727272727294",
          "ni": "0",
          "idle": "99.14545454545457",
          "wa": "0.2787878787878788",
          "hi": "0",
          "si": "0",
          "st": "0",
          "date": "2018-01-14 11"
        },
        "mem_usage": {
          "stat_key": "2",
          "virt": "10787827712",
          "res": "3747196928",
          "date": "2018-01-14 11"
        },
        "dnas_usage": {
          "l_0_out": {
            "stat_key": "l_0_out",
            "cluster_id": "1",
            "dna_id": "0",
            "act_rcvd_pkts": "165.87878787878788",
            "act_send_pkts": "0",
            "act_esnd_pkts_err": "0",
            "act_drop_pkts": "0",
            "act_eemt_pkts_err": "0",
            "date": "2018-01-14 11"
          },
          "l_1_in": {
            "stat_key": "l_1_in",
            "cluster_id": "1",
            "dna_id": "1",

```

```
        "act_rcvd_pkts": "0",
        "act_send_pkts": "165.87878787878788",
        "act_esnd_pkts_err": "0",
        "act_drop_pkts": "0",
        "act_eemt_pkts_err": "0",
        "date": "2018-01-14 11"
    }
},
"ifs_usage": {
    "1": {
        "cluster_id": "1",
        "abs_rcvd_pkts": "1755914.8181818181",
        "abs_rcvd_bytes": "1572754593.4545455",
        "abs_rcvd_pkts_dropped": "0",
        "act_captured_pkts_sec": "0.03606060606060608",
        "act_processed_pkts_sec": "0.03606060606060608",
        "act_send_pkts_sec": "0.03606060606060608",
        "date": "2018-01-14 11"
    }
},
"http_usage": {
    "stat_key": "5",
    "url": "1517",
    "url_lock": "0",
    "ssl": "2653",
    "ssl_lock": "0",
    "cna": "8",
    "cna_lock": "0",
    "sni": "2645",
    "sni_lock": "0",
    "quic": "392",
    "quic_lock": "0",
    "chnprc": "0",
    "ccheck": "487579",
    "ccheck_ip_check": null,
    "ccheck_lock": null,
    "ftp": 0,
    "ftp_lock": 0,
    "smtp": 0,
    "smtp_lock": 0,
    "pop3": 0,
    "pop3_lock": 0,
    "imap": 0,
    "imap_lock": 0,
    "xmpp": 0,
    "xmpp_lock": 0,
    "date": "2018-01-14 11"
},
"date": "2018-01-14 11"
},
"2018-01-14 12": {
```

```
"cpus_usage":{
  "stat_key":"1",
  "us":"2.4529953917050675",
  "sy":"1.2198156682027643",
  "ni":"0",
  "idle":"95.97419354838706",
  "wa":"0.3612903225806449",
  "hi":"0",
  "si":"0",
  "st":"0",
  "date":"2018-01-14 12"
},
"mem_usage":{
  "stat_key":"2",
  "virt":"10787827712",
  "res":"3747747245.419355",
  "date":"2018-01-14 12"
},
"dnas_usage":{
  "l1_in":{
    "stat_key":"l1_in",
    "cluster_id":"1",
    "dna_id":"1",
    "act_rcvd_pkts":"23789.79262672811",
    "act_send_pkts":"175827.29493087556",
    "act_esnd_pkts_err":"0",
    "act_drop_pkts":"0",
    "act_eemt_pkts_err":"0",
    "date":"2018-01-14 12"
  },
  "l1_0_out":{
    "stat_key":"l1_0_out",
    "cluster_id":"1",
    "dna_id":"0",
    "act_rcvd_pkts":"175827.29493087556",
    "act_send_pkts":"23789.79262672811",
    "act_esnd_pkts_err":"0",
    "act_drop_pkts":"0",
    "act_eemt_pkts_err":"0",
    "date":"2018-01-14 12"
  }
},
"ifs_usage":{
  "1":{
    "cluster_id":"1",
    "abs_rcvd_pkts":"1795976.133640553",
    "abs_rcvd_bytes":"1597939948.7695854",
    "abs_rcvd_pkts_dropped":"0",
    "act_captured_pkts_sec":"22.814608294930885",
    "act_processed_pkts_sec":"22.814608294930885",
    "act_send_pkts_sec":"22.814608294930885",
```



```
    "jsonrpc": "2.0",
    "method": "Api_ProcessDpiStatLog",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Api_GetStatLogTail

Gets the "tail" of the DPI statistics log.

Parameters:

- Id - int: the equipment identifier
- Tail - int: the "tail" length

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetStatLogTail",
    "params": {
      "Id": 1,
      "Tail": 20
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
```



```
"jsonrpc": "2.0",
"id": 1515662165576,
"result": {
  "success": true,
  "data": "fq4KpGuICfAxICtX5lAC"
}
}
```

When the identifier is received, the file can be obtained from the link

```
http://<server ip address>/api/file/fq4KpGuICfAxICtX5lAC
```

Api_GetAlertLogTail

Gets "tail" of the DPI messages log.

Parameters:

- Id - int: the equipment identifier
- Tail - int: the "tail" length

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetAlertLogTail",
    "params": {
      "Id": 1,
      "Tail": 20
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": "\t\tbras_max_session_duration =604800
seconds\n\t\tbras_dhcp_request_delay =300 seconds\n\tARP
proxy:\n\t\tbras_arp_proxy =0x0000 (local:off,
gateway:off)\n\t\tbras_arp_ip
=0.0.0.0 (not specified)\n\t\tbras_arp_mac
```

```

=00:00:00:00:00:00\n\tBRAS
      services:\n\t\ttbras_dhcp_auth_mix      =1
(enabled)\n\t\ttbras_ip_source_guard      =0
      (disabled)\n\t\ttbras_terminate_local    =0
(disabled)\n\t\ttbras_vlan_terminate      =0
      (disabled)\n\t\ttbras_vlan_subst
=\n\t\ttbras_terminate_l2      =0
      (disabled)\n\t\ttbras_term_by_as        =0
(disabled)\n\t\ttbras_gateway_ip
      =0.0.0.0\n\t\ttbras_gateway_mac
=00:00:00:00:00:00\n[COMMON  ][2018/01/13-13:32:48:364905]
      [0x7fac68409820] BRAS PPPoE: disabled\n[INFO
][2018/01/13-13:32:48:365180][0x7fac68409820] DSCP
      settings is loaded.\n[WARNING
][2018/01/13-13:35:58:302925][0x7faa21306700] NFLW : very long
      operation to send data, duration=11000669 msec,
tmout=10000380 msec, cntr=1\n[ERROR  ]
      [2018/01/13-13:37:13:630075][0x7faa138fe700] bpm : Error
reset_bypass_wd_timer, if='dna0',
      errno=95 : Operation not supported\n"
    }
}
}

```

Api_DownloadAlertLog

To get the messages log file identifier to be used for downloading.

Parameters:

- Id - int:the equipment identifier

The request example:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDpiInfo",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]

```

The successful response example:

```
{
```

```
"jsonrpc": "2.0",
"id": 1515662165576,
"result": {
  "success": true,
  "data": "tjFNBnsRkQmXlJbBsCHM"
}
}
```

When the identifier is received the file can be obtained from the link like following

```
http://<ip адрес сервера>/api/file/fq4KpGuICfAxICtX5lAC
```

Notifications management

- [Api_GetLastNotifications](#)
- [Api_GetUnreadedNotificationsCount](#)
- [Api_SetNotificationsReaded](#)
- [Api_DeleteNotifications](#)

Api_GetLastNotifications

To get the list of the latest notifacations.

Parameters:

- Page - int:the page number
- Size - int:the number of notifications in the response

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetLastNotifications",
    "params": {
      "Page": 1,
      "Size": 3
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
```

```

    "id":1515662165576,
    "result":{
      "success":true,
      "data": {
        "page": 1,
        "size": 3,
        "count": 200,
        "rows": [
          {
            "notif_id": "391439",
            "hardware_hardware_id": "2",
            "users_user_id": null,
            "date": "2018-01-12T09:22:32",
            "from": "Result 'check license', err_code=1",
            "text": " Result 'check license', err_code=1 : Success:
loaded",
            "importance": "COMMON",
            "readed": "0",
            "deleted": "0"
          },
          {
            "notif_id": "391438",
            "hardware_hardware_id": "2",
            "users_user_id": null,
            "date": "2018-01-12T09:22:32",
            "from": "Check license",
            "text": " Check license : devices dna0 <--> dna1 ...",
            "importance": "COMMON",
            "readed": "0",
            "deleted": "0"
          },
          {
            "notif_id": "391437",
            "hardware_hardware_id": "2",
            "users_user_id": null,
            "date": "2018-01-12T09:20:32",
            "from": "bl_updater_thread",
            "text": " bl_updater_thread :
Notifications(updatenotifications) list loaded with result, rc=0
: Success: loaded.",
            "importance": "INFO",
            "readed": "0",
            "deleted": "0"
          }
        ]
      }
    }
  }
}

```

Api_GetUnreadNotificationsCount

To get the number of unread CRITICAL notifications.

There are no parameters.

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetUnreadNotificationsCount",
    "params": {
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": 123
  }
}
```

Api_SetNotificationsReaded

Marks notifications as read.

Parameters:

- Notifs - array:the array of notifications

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SetNotificationsReaded",
    "params": {
      "Notifs": [
        {

```



```

        "notif_id": "391439",
        "hardware_id": "2",
        "users_user_id": null,
        "date": "2018-01-12T09:22:32",
        "from": "Result 'check license', err_code=1",
        "text": " Result 'check license', err_code=1 : Success:
loaded",
        "importance": "COMMON",
        "readed": "0",
        "deleted": "0"
    },
    ...
]
},
"id":1515661809137
}
]

```

The successful response example:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}

```

Api_DeleteNotifications

Deletes notifications.

Parameters:

- Notifs - array:the array of notifications

The request example:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteNotifications",
    "params": {
      "Notifs": [
        {
          "notif_id": "391439",
          "hardware_id": "2",

```

```

        "users_user_id": null,
        "date": "2018-01-12T09:22:32",
        "from": "Result 'check license', err_code=1",
        "text": " Result 'check license', err_code=1 : Success:
loaded",
        "importance": "COMMON",
        "readed": "0",
        "deleted": "0"
    },
    ...
]
},
"id":1515661809137
}
]

```

The successful response example:

```

{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}

```

Errors sending

- [Api_SendError](#)
- [Api_LogError](#)

Api_SendError

Sends an error message to the mail specified in the configuration.

Parameters:

- Params - object: the error message model

Sending configurations:

- subject - string: message subject
- body - string: message body
- hardware_id - int: equipment identifier
- alertLog - bool: to include (or exclude) notification logs file in a message
- statLog - bool: to include (or exclude) equipment status logs file in a message

- uiLog - bool: to include (or exclude) dpiui logs file in a message

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SendError",
    "params": {
      "Params": {
        "hardware_id": "1",
        "subject": "test",
        "body": "test",
        "alertLog": true,
        "statLog": true,
        "uiLog": true
      }
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Api_LogError

Writes the error information to the dpiui log file.

Parameters:

- Content - string: error message body

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_LogError",
    "params": {
```

```
        "Content": "error text"
      },
      "id": 1515661809137
    }
  ]
}
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Subscriber management

- [Api_GetSubscribers](#)
- [Api_SaveSubscriber](#)
- [Api_DeleteSubscriber](#)

Api_GetSubscribers

Gets the list of subscribers.

Parameters:

- Params - object: search options (optional parameter)

Possible search parameters Params:

- page - int: page number
- size - int: dimension of array containing the result
- hardware_id - int: equipment identifier
- login - string: user name
- login_equal - string:...
- ip - string: user ip address
- ip_equal - string:...
- bind_type - ...
- services - ...
- order_by - string: field used for ordering the subscribers list
- order_dir - string: ordering the list in ascending order ("asc") / descending ("desc")

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetSubscribers",
    "params": {
      "Params": {
        "hardware_id": "1",
        "page": 0,
        "size": 100,
        "login": "",
        "ip": "",
        "bind_type": "",
        "services": ""
      }
    },
    "id": 1515929803508
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 6,
      "rows": [
        {
          "hardware_id": "1",
          "login": "llllyyy",
          "ip": "5.5.5.7",
          "bind_type": "1",
          "services": "5,"
        },
        {
          "hardware_id": "1",
          "login": "user11",
          "ip": "1.1.1.1",
          "bind_type": "0",
          "services": "4,"
        },
        {
          "hardware_id": "1",
          "login": "user22",
```



```

    "Id":1,
    "Subscriber":{
      "id":1515929803535,
      "hardware_id":0,
      "login":"test",
      "ip":"77.77.77.77",
      "bind_type":"0",
      "services":"4,6,",
      "tariff":"",
      "valid":true,
      "oldSubscriber":{
        "id":1515929803535,
        "hardware_id":0,
        "login":"",
        "ip":"",
        "bind_type":1,
        "services":"",
        "tariff":"",
        "valid":true
      }
    },
    "SubscriberOld":{
      "id":1515929803535,
      "hardware_id":0,
      "login":"",
      "ip":"",
      "bind_type":1,
      "services":"",
      "tariff":"",
      "valid":true
    }
  },
  "id":1515929803535
}
]

```

The successful response example when creating a subscriber:

```

{
  "jsonrpc":"2.0",
  "id":1515929803535,
  "result":{
    "success":true,
    "data":{
      "bind":{
        "success":true,
        "data":{
          "bind_result":{
            "total":"1",
            "success":"1",

```



```

        "bind_type": "0",
        "services": "1,4,6,",
        "tariff": "",
        "valid": true,
        "oldSubscriber": {
            "id": 1515929804490,
            "hardware_id": "1",
            "login": "test",
            "ip": "77.77.77.77",
            "bind_type": "0",
            "services": "4,6,",
            "tariff": "",
            "valid": true
        }
    },
    "SubscriberOld": {
        "id": 1515929804490,
        "hardware_id": "1",
        "login": "test",
        "ip": "77.77.77.77",
        "bind_type": "0",
        "services": "4,6,",
        "tariff": "",
        "valid": true
    }
},
    "id": 1515929804490
}
]

```

The successful response example when editing a subscriber:

```

{
    "jsonrpc": "2.0",
    "id": 1515929804490,
    "result": {
        "success": true,
        "data": {
            "bind": {
                "success": true,
                "data": {
                    "bind_result": {
                        "total": "1",
                        "success": "1",
                        "fail": "0",
                        "were_set_before": "0",
                        "were_not_set_before": "0",
                        "data": {

```



```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteSubscriber",
    "params": {
      "Id": 1,
      "Subscriber": {
        "id": 1515929804638,
        "hardware_id": "1",
        "login": "test123",
        "ip": "77.77.77.77",
        "bind_type": "0",
        "services": "1,4,6,",
        "tariff": "",
        "valid": true
      }
    },
    "id": 1515929804638
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515929804638,
  "result": {
    "success": true,
    "data": {
      "services": {
        "success": true,
        "data": {
          "bind_result": null,
          "unbind_result": {
            "total": "1",
            "success": "1",
            "fail": "0",
            "were_set_before": "0",
            "were_not_set_before": "0",
            "data": {
            }
          }
        }
      }
    },
    "unbind": {
      "success": true,
      "data": {
        "total": "1",

```



```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDscpProtocolsConfig",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": "ftp cs4\nhttps cs1\nBittorrent cs7\nTCP Unknown
cs7\ndefault keep\n"
  }
}
```

Api_SetDscpProtocolsConfig

Sets the DSCP protocol prioritization to be used.

Parameters:

- Id - int: the equipment identifier
- Config - string: configuration content

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SetDscpProtocolsConfig",
    "params": {
      "Id": 1,
      "Config": "ftp cs4\nhttps cs1\nBittorrent cs7\nTCP Unknown
cs7\ndefault keep\n"
    },
    "id": 1515661809137
  }
]
```

```
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true
  }
}
```

Api_GetDscpProtocolsConfigHistory

Gets the history of DSCP prioritization configuration changes.

Parameters:

- Id - int: the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDscpProtocolsConfigHistory",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      "2015.06.03.03.54.25.000000.protocols.dscp",
      "2015.06.03.04.00.28.000000.protocols.dscp",
      "2015.06.04.15.48.34.000000.protocols.dscp",
      ...
    ]
  }
}
```

```
}  
  
}
```

Api_GetDscpProtocolsConfigFile

Получение файла конфигурации из истории по имени файла.

Parameters:

- Id - int: the equipment identifier
- File - string: the file name

The request example:

```
[  
  {  
    "jsonrpc": "2.0",  
    "method": "Api_GetDscpProtocolsConfigFile",  
    "params": {  
      "Id": 1,  
      "File": "2015.06.03.03.54.25.000000.protocols.dscp"  
    },  
    "id": 1515661809137  
  }  
]
```

The successful response example:

```
{  
  
  "jsonrpc": "2.0",  
  "id": 1515662165576,  
  "result": {  
    "success": true,  
    "data": "dns cs0\nhttp cs1\nhttps cs1\nBittorrent cs7\nICMPv6  
cs0\nICMP cs0\ndefault cs2\n"  
  }  
}
```

ASN prioritization management

- [Api_GetAsnDscpConfig](#)
- [Api_SetAsnDscpConfig](#)
- [Api_GetAsnDscpConfigHistory](#)
- [Api_GetAsnDscpConfigFile](#)

Api_GetAsnDscpConfig

Gets the ASN prioritization configuration.

Parameters:

- Id - int: the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetAsnDscpConfig",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "asn": {
        "success": true,
        "data": "192.168.1.16/32 64512\n192.168.1.111/32 64512\n"
      },
      "asn_dscp": {
        "success": true,
        "data": "64512 pass\n64512 cs0\n64513 drop\n64514
pass\n64514 peer\n"
      }
    }
  }
}
```

Api_SetAsnDscpConfig

Sets the ASN prioritization configuration.

Parameters:

- Id - int: the equipment identifier
- AsnConfig - string: ASN configuration content
- AsnDscpConfig - string: ASNDSCP configuration content

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SetDscpProtocolsConfig",
    "params": {
      "Id": 1,
      "AsnConfig": "192.168.1.16/32 64512\n192.168.1.111/32 64512\n",
      "AsnDscpConfig": "64512 pass\n64512 cs0\n64513 drop\n64514 pass\n64514 peer\n"
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "asn": {
        "success": true
      },
      "asn_dscp": {
        "success": true
      }
    }
  }
}
```

Api_GetAsnDscpConfigHistory

Gets the history of ASN prioritization configuration changes.

Parameters:

- Id - int: the equipment identifier

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetAsnDscpConfigHistory",
    "params": {
      "Id": 1
    },
    "id": 1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      "2015.06.07.08.29.29.000000.asnum.dscp",
      "2015.06.07.08.30.12.000000.asnum.dscp",
      "2015.06.07.08.33.40.000000.asnum.dscp",
      ...
    ]
  }
}
```

Api_GetAsnDscpConfigFile

Gets the configuration file by file name from the history.

Parameters:

- Id - int: the equipment identifier
- File - string: the file name

The request example:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetAsnDscpConfigFile",
    "params": {
      "Id": 1,
      "File": "2015.06.07.08.29.29.000000.asnum.dscp"
    }
  }
]
```

```
    },
    "id":1515661809137
  }
]
```

The successful response example:

```
{
  "jsonrpc":"2.0",
  "id":1515662165576,
  "result":{
    "success":true,
    "data": "11111 pass\n11112 pass\n11113 pass\n11114 pass\n11115
pass\n11116 pass\n11117 pass\n11118
           pass\n11119 pass\n11120 pass\n64514 pass\n"
  }
}
```