

# Table of Contents

<b><i>JSON-RPC API Описание программного интерфейса</i></b> .....	3
Описание .....	3
Общие функции .....	5
Функции аутентификации и авторизации .....	6
Управление профилем пользователя .....	9
Управление пользователями .....	12
Управление ролями .....	15
Управление оборудованием .....	18
Управление нотификациями .....	46
Отправка ошибок .....	50
Управление абонентами .....	52
Управление услугами .....	61
Управление тарифами .....	93
Управление приоретизацией по протоколам (DSCP) .....	99
Управление приоретизацией по ASN .....	102
Управление QoE .....	106
Управление рекламными кампаниями .....	121
Управление абонентами рекламных кампаний .....	131



# JSON-RPC API Описание программного интерфейса

## Описание

Все запросы на сервер отправляются методом POST по адресу [https://your\\_host/api/jsonrpc](https://your_host/api/jsonrpc). В заголовках ответа передаются "печеньки". Их необходимо передавать во всех последующих запросах.

В теле запроса передается массив [] объектов типа

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "Api_GetAppVersion", -- Имя метода
  "params": {}, -- Объект с параметрами
  "id": 1515659482430 - идентификатор запроса (может быть любое случайное число)
}
```

Если в запросе передан массив из нескольких объектов, в ответе будет массив. Если в запросе передан массив из одного объекта, в ответе вернется один объект, не массив.

Примечание: регистр и тип параметров имеет значение.

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515659482430, -- идентификатор заданный при запросе
  "result": {
    "success": true, -- результат (булево значение)
    "data": "2.0.0" -- данные (строка, массив, объект)
  }
}
```

Ответ может вернуться с ошибкой. Ошибки подразделяются на две категории:

- Глобальные - обрабатываются на уровне модуля json-rpc. Например, переданы неверные параметры. Ошибка выводится в поле error.

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

```
{
  "code": "required_field",
  "message": "Не передан либо пустой обязательный параметр",
  "object_name": "Test"
}
]
```

- Локальные - обрабатываются на уровне Api внутри вызываемой функции. Например, ошибки записи в БД. Ошибка выводится в поле result.error. При этом result.success = false.

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

## Несколько примеров запросов с использованием CURL

Получить версию приложения:

```
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": 1563436495288}] '
```

Выполнить авторизацию:

```
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", "password": "vasexperts", "remember": true}, "id": 1563438645838}] '
```

Получить файл конфигурации DPI:

```
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}] '
```

## Общие функции

### Api\_GetAppVersion

Выводит версию приложения.

Параметры отсутствуют.

Пример запроса:

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

Пример ответа:

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

### Api\_GetDics

Выводит словари.

Параметры:

- DicsKeys - Array: список идентификаторов словарей. По умолчанию выводит все.

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetDics",
    "params": {},
  }
]
```

```
    "id":1515921551526
  }
]
```

Пример ответа:

```
{
  "jsonrpc":"2.0",
  "id":1515921551526,
  "result":{
    "success":true,
    "data":{
      "RolesDic":{
        "1":{
          "dic_id":"1",
          "value":"Администратор"
        }
      },
      "HardwaresDic":{
        "1":{
          "dic_id":"1",
          "value":"Тестовый стенд. 34",
          "ip":"x.x.x.x",
          "port":"22",
          "login":"dpisu"
        },
        "2":{
          "dic_id":"2",
          "value":"Рабочий стенд .83 !",
          "ip":"y.y.y.y",
          "port":"22",
          "login":"dpisu"
        }
      }
    }
  }
}
```

## Функции аутентификации и авторизации

- [Api\\_Auth](#)
- [Api\\_Login](#)
- [Api\\_Logout](#)

### Api\_Auth

Проверяет и выводит результат: авторизован пользователь или нет.

Параметры отсутствуют.

Пример запроса:

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

Пример ответа, если пользователь не авторизован:

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

Пример ответа, если пользователь авторизован:

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

## Api\_Login

Авторизует пользователя

Параметры:

- username - string:имя пользователя
- password - string:пароль
- remember - bool:пароль

Пример запроса:

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

Пример успешного ответа:

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

Пример неуспешного ответа:

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

Если авторизация прошла успешно, в заголовках ответа передаются "печеньки". Их необходимо передавать во всех последующих запросах.

## Api\_Logout

Деавторизацует текущего пользователя.

Параметры отсутствуют.

Пример запроса:

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

## Управление профилем пользователя

- [Api\\_GetMyProfile](#)
- [Api\\_SaveMyProfile](#)
- [Api\\_ChangeMyPassword](#)

### Api\_GetMyProfile

Выводит профиль пользователя.

Параметры отсутствуют.

Пример запроса:

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

Пример успешного ответа:

```
{
```

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

## Api\_SaveMyProfile

Изменение профиля пользователя.

Параметры:

- MyProfile - object: модель пользователя

Пример запроса:

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

Пример успешного ответа:

```
{
```

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

## Api\_ChangeMyPassword

Изменение пароля пользователя.

Параметры:

- OldPassword - object:старый пароль
- NewPassword - string:новый пароль

Пример запроса:

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

Пример успешного ответа:

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

```
        "user_id": "1",
        "username": "admin",
        "name": "Строгий С.С.",
        "email": "email@email.ru",
        "phone": "2-2-2",
        "company": "VasExpert",
        "position": "Админ",
        "password":
"$2y$10$t/PdsQVXr927wyunbdET1.EHtuxTBg4iKmtHlJ4jfmU0XR4qPUANu"
    }
}
}
```

## Управление пользователями

- [Api\\_GetUsers](#)
- [Api\\_SaveUser](#)
- [Api\\_DeleteUser](#)

### Api\_GetUsers

Получение списка пользователей.

Параметры:

- Id - int:идентификатор пользователя(необязательный параметр)

Пример запроса с идентификатором пользователя:

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

Пример успешного ответа на запрос с идентификатором пользователя:

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

```
    "success":true,
    "data": {
      "user_id": "1",
      "username": "admin",
      "name": "Строгий С.С.",
      "email": "email@email.ru",
      "phone": "2-2-2",
      "company": "VasExpert",
      "position": "Админ",
      "role_sections": "hardware.read,hardware.write,..."
    }
  }
}
```

Пример успешного ответа на запрос без параметров:

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

## Api\_SaveUser

Создание/изменение данных пользователя.

Параметры:

- user - object:модель пользователя

Пример запроса:

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

Пример успешного ответа:

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

### **Api\_DeleteUser**

Удаляет пользователя.

Параметры:

- user - object: модель пользователя

Пример запроса:

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

Пример успешного ответа:

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

## Управление ролями

- [Api\\_GetRoles](#)
- [Api\\_SaveRole](#)
- [Api\\_DeleteRole](#)

### Api\_GetRoles

Получение списка пользователей.

Параметры:

- Id - int:идентификатор роли(необязательный параметр)

Пример запроса с идентификатором роли:

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

Пример успешного ответа на запрос с идентификатором роли:

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

Пример успешного ответа на запрос без параметров:

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

```

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

## Api\_SaveRole

Создание/изменение данных роли.

Параметры:

- role - object: модель роли

Пример запроса:

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

Пример успешного ответа:

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

```
}
```

## Api\_DeleteRole

Удаление роли.

Параметры:

- role - объект: модель роли

Пример запроса:

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

Пример успешного ответа:

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

## Управление оборудованием

- [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

Получить список оборудований.

Параметры:

- Id - int:идентификатор оборудования(необязательный параметр)

Пример запроса с идентификатором оборудования:

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

Пример успешного ответа на запрос с идентификатором оборудования:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "hardware_id": "1",
        "name": "Тестовый стенд. 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
    }
]
}
}
```

Пример успешного ответа на запрос без параметров:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": [
      {
        "hardware_id": "1",
        "name": "Тестовый стенд. 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

Создание/изменение данных оборудования.

Параметры:

- hardware - object: модель оборудования

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveHardware",
    "params": {
      "hardware": {
        "hardware_id": "1",
        "name": "Тестовый стенд. 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
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": {
      "hardware_id": "1",
      "name": "Тестовый стенд. 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

Удаление оборудования.

Параметры:

- hardware - объект: модель оборудования

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteRole",
    "params": {
      "hardware": {
        "hardware_id": "1",
        "name": "Тестовый стенд. 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
  }
]
```

Пример успешного ответа:

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

```
}  
  
}
```

## Api\_RestartHardware

Перезапуск оборудования.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

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

## Api\_ReloadHardware

Обновление горячих параметров.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

```
[
```

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

Пример успешного ответа:

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

### **Api\_TestConnectToHardware**

Проверка доступности оборудования.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

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

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

### **Api\_GetDpiConfig**

Получение DPI-конфигурации оборудования.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

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

### **Api\_ValidateDpiConfig**

Валидация DPI-конфигурации оборудования.

Параметры:

- Id - int:идентификатор оборудования

- Config - string: конфигурация

Пример запроса:

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

Пример успешного ответа:

```
{
  "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

Установка DPI-конфигурации на оборудование.

Параметры:

- Id - int: идентификатор оборудования
- Config - string: конфигурация

Пример запроса:

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

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

Пример успешного ответа:

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

### Api\_GetDpiConfigHistory

Получение истории установки DPI-конфигураций на оборудовании.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

```
{
  "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

Получение содержимого файла DPI-конфигурации.

Параметры:

- Id - int:идентификатор оборудования
- File - string:имя файла

Пример запроса:

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

Пример успешного ответа:

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

## Api\_GetDpiInfo

Получение информации об оборудовании.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

```
{
  "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

Получение информации о текущем состоянии производительности оборудования.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

```
{
  "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,
```

```
    "mem": 0,  
    "command": "watchdog/2",  
    "date": null  
  },  
  {  
    "pid": "15",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "migration/3",  
    "date": null  
  },  
  {  
    "pid": "16",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "migration/3",  
    "date": null  
  },  
  {  
    "pid": "17",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "ksoftirqd/3",  
    "date": null  
  },  
  {  
    "pid": "18",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 0,  
    "command": "watchdog/3",  
    "date": null  
  },  
  {  
    "pid": "19",  
    "virt": "0",  
    "res": "0",  
    "shr": "0",  
    "cpu": 0,  
    "mem": 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

Получение информации о состоянии производительности оборудования за период.

Параметры:

- Id - int:идентификатор оборудования
- Period - object:...(необязательный параметр)
- PeriodType - string:...(необязательный параметр)

Пример запроса:

```

[
  {
    "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
  }
}
```

Пример успешного ответа:

```
{
  "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
  }
]
```

Пример успешного ответа:

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

## Api\_GetStatLogTail

Получение "хвоста" лога статистики DPI.

Параметры:

- Id - int: идентификатор оборудования
- Tail - int: длина "хвоста"

Пример запроса:

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

Пример успешного ответа:

```
{
```



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

Получив идентификатор, файл можно получить по ссылке виде

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

## Api\_GetAlertLogTail

Получение "хвоста" лога сообщений DPI.

Параметры:

- Id - int: идентификатор оборудования
- Tail - int: длина "хвоста"

Пример запроса:

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

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515662165576,
  "result": {
    "success": true,
    "data": "\t\ttbras_max_session_duration =604800
seconds\n\t\ttbras_dhcp_request_delay =300 seconds\n\t\tARP
proxy:\n\t\ttbras_arp_proxy =0x0000 (local:off,
gateway:off)\n\t\ttbras_arp_ip
=0.0.0.0 (not specified)\n\t\ttbras_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

Получение идентификатора лог-файла сообщений для скачивания.

Параметры:

- Id - int:идентификатор оборудования

Пример запроса:

```

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

```

Пример успешного ответа:

```
{
```

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

Получив идентификатор, файл можно получить по ссылке виде

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

## Управление нотификациями

- [Api\\_GetLastNotifications](#)
- [Api\\_GetUnreadedNotificationsCount](#)
- [Api\\_SetNotificationsReaded](#)
- [Api\\_DeleteNotifications](#)

### Api\_GetLastNotifications

Получение списка последних нотификаций.

Параметры:

- Page - int: номер страницы
- Size - int: количество нотификаций в ответе

Пример запроса:

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

Пример успешного ответа:

```
{
  "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\_GetUnreadedNotificationsCount

Получение количества непрочитанных CRITICAL нотификаций.

Параметры отсутствуют.

Пример запроса:

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

Пример успешного ответа:

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

## Api\_SetNotificationsReaded

Отметить нотификации как прочитанные.

Параметры:

- Notifs - array: массив нотификаций

Пример запроса:

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

```

```

        "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"
    },
    ...
]
},
"id":1515661809137
}
]

```

Пример успешного ответа:

```

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

```

## Api\_DeleteNotifications

Удалить нотификации.

Параметры:

- Notifs - array:массив нотификаций

Пример запроса:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteNotifications",
    "params": {
      "Notifs": [
        {
          "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"
    },
    ...
]
},
"id":1515661809137
}
]

```

Пример успешного ответа:

```

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

```

## Отправка ошибок

- [Api\\_SendError](#)
- [Api\\_LogError](#)

### Api\_SendError

Отправка сообщения об ошибке на почту указанную в конфигурации.

Параметры:

- Params - объект: модель сообщения об ошибке

Параметры конфигурации отправки:

- subject - string: тема сообщения
- body - string: тело сообщения
- hardware\_id - int: идентификатор оборудования
- alertLog - bool: включение файла с логами уведомлений в сообщение
- statLog - bool: включение файла с логами состояния оборудования в сообщение

- uiLog - bool: включение файла с логами dpiui в сообщение

Пример запроса:

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

Пример успешного ответа:

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

## Api\_LogError

Запись информации об ошибке в dpiui-лог.

Параметры:

- Content - string: тело сообщения об ошибке

Пример запроса:

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

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

Пример успешного ответа:

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

## Управление абонентами

- [Api\\_GetSubscribers](#)
- [Api\\_SaveSubscriber](#)
- [Api\\_DeleteSubscriber](#)

### Api\_GetSubscribers

Получение списка абонентов. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- hardware\_id - int: идентификатор оборудования
- login - string: имя пользователя
- login\_equal - string:...
- ip - string: ip-адрес пользователя
- ip\_equal - string:...
- bind\_type - ...
- services - ...
- policings - ...
- order\_by - string: поле, по которому будет упорядочен список абонентов
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
```

```

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

```

Пример успешного ответа:

```

{
  "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,",
          "policing": "278"
        },
        {
          "hardware_id": "1",
          "login": "user11",
          "ip": "1.1.1.1",
          "bind_type": "0",
          "services": "4,",
          "policing": "279"
        },
        {
          "hardware_id": "1",

```



```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveSubscriber",
    "params": {
      "Id": 1,
      "Subscriber": {
        "id": 1515929803535,
        "hardware_id": 0,
        "login": "test",
        "ip": "77.77.77.77",
        "bind_type": "0",
        "services": "4,6,",
        "policing": "278",
        "valid": true,
        "oldSubscriber": {
          "id": 1515929803535,
          "hardware_id": 0,
          "login": "",
          "ip": "",
          "bind_type": 1,
          "services": "",
          "policing": "0",
          "valid": true
        }
      },
      "SubscriberOld": {
        "id": 1515929803535,
        "hardware_id": 0,
        "login": "",
        "ip": "",
        "bind_type": 1,
        "services": "",
        "policing": "",
        "valid": true
      }
    },
    "id": 1515929803535
  }
]
```

Пример успешного ответа при создании абонента:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803535,
  "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":{
          }
        },
      "unbind_result":null
    }
  },
  "services":{
    "success":true,
    "data":{
      "bind_result":{
        "total":"1",
        "success":"1",
        "fail":"0",
        "were_set_before":"0",
        "were_not_set_before":"0",
        "data":{
          }
        },
      "unbind_result":null
    }
  },
  "sync_bind":{
    "success":true,
    "data":0.0988309383392334
  },
  "sync_services":{
    "success":true,
    "data":0.08402705192565918
  },
  "policing":{
    "success": true,
    "data":{
      "bind":{
        "success":true
      },
      "unbind":{
        "success":true
      }
    }
  },
  "sync_policing":{
    "success": true
  }
}
```

```
}
}
}
```

Пример запроса при редактировании абонента:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveSubscriber",
    "params": {
      "Id": 1,
      "Subscriber": {
        "id": 1515929804490,
        "hardware_id": "1",
        "login": "test123",
        "ip": "77.77.77.77",
        "bind_type": "0",
        "services": "1,4,6,",
        "policing": "277",
        "valid": true,
        "oldSubscriber": {
          "id": 1515929804490,
          "hardware_id": "1",
          "login": "test",
          "ip": "77.77.77.77",
          "bind_type": "0",
          "services": "4,6,",
          "policing": "",
          "valid": true
        }
      },
      "SubscriberOld": {
        "id": 1515929804490,
        "hardware_id": "1",
        "login": "test",
        "ip": "77.77.77.77",
        "bind_type": "0",
        "services": "4,6,",
        "policing": "278",
        "valid": true
      }
    },
    "id": 1515929804490
  }
]
```

Пример успешного ответа при редактировании абонента:

```
{
  "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": {
            }
          },
          "unbind_result": null
        }
      },
      "services": {
        "success": true,
        "data": {
          "bind_result": {
            "total": "1",
            "success": "1",
            "fail": "0",
            "were_set_before": "0",
            "were_not_set_before": "0",
            "data": {
            }
          },
          "unbind_result": {
            "total": "1",
            "success": "1",
            "fail": "0",
            "were_set_before": "0",
            "were_not_set_before": "0",
            "data": {
            }
          }
        }
      },
      "sync_bind": {
        "success": true,
        "data": 0.1651449203491211
      },
      "sync_services": {
        "success": true,

```

```
        "data":0.1676170825958252
    },
    "policing":{
        "success": true,
        "data":{
            "bind":{
                "success":true
            },
            "unbind":{
                "success":true
            }
        }
    },
    "sync_policing":{
        "success": true
    }
}
}
```

## Api\_DeleteSubscriber

Удаление абонента.

Параметры:

- Id - int: идентификатор оборудования
- Subscriber - object: модель абонента

Пример запроса:

```
[
  {
    "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,",
        "policing": "278",
        "valid": true
      }
    }
  },
]
```

```
    "id":1515929804638
  }
]
```

Пример успешного ответа:

```
{
  "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",
          "success":"1",
          "fail":"0",
          "were_set_before":"0",
          "were_not_set_before":"0",
          "data":{
          }
        }
      },
      "sync_bind":{
        "success":true,
        "data":0.07761001586914062
      },
      "sync_services":{
        "success":true,
        "data":0.0842139720916748
      },
      "policing":{
        "success": true,

```

```
    "data":{
        "unbind":{
            "success":true
        }
    },
    "sync_policing":{
        "success": true
    }
}
}
```

## Управление услугами

- [Управление черными/белыми списками](#)
- [Управление записью PCAP](#)
- [Управление мини Firewall-ом](#)
- [Управление L2 BRAS](#)
- [Управление L3 BRAS](#)
- [Управление CG NAT](#)
- [Управление защитой от DOS и DDOS атак](#)
- [Управление кешированием](#)
- [Управление уведомлениями абонентов](#)
- [Управление блокировкой и заменой рекламы](#)
- [Управление бонусной программой](#)

## Управление черными/белыми списками

- [Api\\_GetBITasks](#)
- [Api\\_SaveBITask](#)
- [Api\\_DeleteBITask](#)
- [Api\\_RunBITask](#)
- [Api\\_StopBITask](#)
- [Api\\_GetBIListItems](#)
- [Api\\_SaveBIListItem](#)
- [Api\\_DeleteBIListItem](#)
- [Api\\_ImportBIListItems](#)
- [Api\\_ImportBIListItemsFromQoE](#)
- [Api\\_GetBITargets](#)
- [Api\\_SaveBITargets](#)
- [Api\\_SaveBITarget](#)
- [Api\\_DeleteBITarget](#)

### Api\_GetBITasks

Получение списка профилей Черных/Белых списков. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- hardware\_id - int: идентификатор оборудования
- title - string: имя профиля
- white - int:0|1, если 0 - то профиль услуги Черного списка
- task\_id - int: идентификатор профиля
- tasks\_ids - string: идентификаторы профилей через запятую
- not\_in\_tasks\_ids - string: идентификаторы профилей через запятую
- parent\_ids\_is\_null - int: 0|1
- status - int: 0|1 ⇒ Выкл|Вкл
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetBlTasks",
    "params": {
      "Params": {
        "hardware_id": 1,
        "parent_id_is_null": 1,
        "order_by": "task_id",
        "order_dir": "desc",
        "page": 0,
        "tasks_ids": null,
        "size": 100,
        "title": ""
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "page": 0,

```

```

        "size":100,
        "count":6,
        "rows":[
            {
                "date": "2019-03-23 06:28:25",
                "form_data": "{\"title\":\"All
range\",\"white\":0,\"redirect_enabled\":\"\",\"redirect\":\"spectredpi.com\",\"target\":
\"\"}\",
                "hardware_id": "1",
                "lists_updated": "0",
                "parent_id": null,
                "status": "1",
                "targets_updated": "0",
                "task_id": "234",
                "title": "All range",
                "user_id": "1",
                "white": "0",
            },
            ...
        ]
    }
}
}
}

```

### Api\_SaveBlTask

Сохранение профиля Черных/Белых списков. Параметры:

- Form - object: модель данных из формы
- Task - object: модель профиля

Пример запроса:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveBlTask",
    "params": {
      "Form": {
        "title": "All range",
        "redirect": "specredpi.com",
        "white": 0,
        "target": "",
        "redirect_enabled": ""
      },
      "Task": {
        "data": [],
        "date": "2019-03-22 06:28:25",

```

```

        "form_data":{"title":"All
range1","white":1,"redirect_enabled":"","redirect":"spectredpi.com
","target":""},
        "hardware_id": 1,
        "parent_id":null,
        "status":1,
        "task_id":234,
        "title":"All range1",
        "user_id":1,
        "white":1
    }
},
    "id":1515929803508
}
]

```

Пример успешного ответа:

```

{
    "jsonrpc":"2.0",
    "id":1515929803508,
    "result":{
        "success":true,
        "data":{
            "task":{
                "data":[],
                "date":"2019-03-22 06:28:25",
                "form_data":{"title":"All
range","white":0,"redirect_enabled":"","redirect":"spectredpi.com\
","target":""},
                "hardware_id": 1,
                "parent_id":null,
                "status":1,
                "task_id":234,
                "title":"All range",
                "user_id":1,
                "white":0,
                "lists_updated":1,
                "targets_updated":1
            }
        }
    }
}

```

**Api\_DeleteBITask**

Удаление профиля Черных/Белых списков. Параметры:

- Task - object: модель профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteBlTask",
    "params": {
      "Task": {
        "data": [],
        "date": "2019-03-22 06:28:25",
        "form_data": "{\"title\": \"All range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com\", \"target\": \"\"}",
        "hardware_id": 1,
        "parent_id": null,
        "status": 1,
        "task_id": 234,
        "title": "All range1",
        "user_id": 1,
        "white": 1
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true
  }
}
```

### **Api\_RunBlTask**

Включение профиля Черных/Белых списков. Параметры:

- Task - object: модель профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_RunBlTask",
    "params": {
      "Task": {
        "data": [],
        "date": "2019-03-22 06:28:25",
        "form_data": "{\"title\": \"All
range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com
\", \"target\": \"\"}",
        "hardware_id": 1,
        "parent_id": null,
        "status": 0,
        "task_id": 234,
        "title": "All range1",
        "user_id": 1,
        "white": 1
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "data": [],
      "date": "2019-03-22 06:28:25",
      "form_data": "{\"title\": \"All
range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com
\", \"target\": \"\"}",
      "hardware_id": 1,
      "parent_id": null,
      "status": 1,
      "task_id": 234,
      "title": "All range1",
      "user_id": 1,
      "white": 1
    }
  }
}
```

```
}
```

## Api\_StopBlTask

Выключение профиля Черных/Белых списков. Параметры:

- Task - object: модель профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_StopBlTask",
    "params": {
      "Task": {
        "data": [],
        "date": "2019-03-22 06:28:25",
        "form_data": "{\"title\": \"All
range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com
\", \"target\": \"\"}",
        "hardware_id": 1,
        "parent_id": null,
        "status": 1,
        "task_id": 234,
        "title": "All range1",
        "user_id": 1,
        "white": 1
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "data": [],
      "date": "2019-03-22 06:28:25",
      "form_data": "{\"title\": \"All
range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com
\", \"target\": \"\"}",
    }
  }
}
```

```
        "hardware_id": 1,
        "parent_id": null,
        "status": 0,
        "task_id": 234,
        "title": "All range1",
        "user_id": 1,
        "white": 1
    }
}
```

### Api\_GetBLListItems

Получение списка ресурсов профилей Черных/Белых списков. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- uri - string: ресурс
- type - string: sni|ip|url|cn
- task\_id - int: идентификатор профиля
- enabled - int: 0|1 ⇒ Выкл|Вкл
- order\_by - string: поле, по которому будет упорядочен список ресурсов
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetBLListItems",
    "params": {
      "Params": {
        "order_by": "uri",
        "order_dir": "asc",
        "page": 0,
        "tasks_id": 234,
        "size": 100
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 6,
      "rows": [
        {
          "enabled": 1,
          "item_id": 2098644,
          "task_id": "234",
          "type": "sni",
          "uri": "*.detik.com"
        },
        ...
      ]
    }
  }
}
```

### Api\_SaveBlListItem

Сохранение ресурса профиля Черных/Белых списков. Параметры:

- ListItem - object: модель ресурса

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveBlListItem",
    "params": {
      "ListItem": {
        "enabled": 1,
        "item_id": 2098644,
        "task_id": "234",
        "type": "sni",
        "uri": "*.detik.com"
      }
    },
    "id": 1515929803508
  }
]
```

```
}  
]
```

Пример успешного ответа:

```
{  
  "jsonrpc": "2.0",  
  "id": 1515929803508,  
  "result": {  
    "success": true,  
    "data": {  
      "list_item": {  
        "enabled": 1,  
        "item_id": 2098644,  
        "task_id": "234",  
        "type": "sni",  
        "uri": "*.detik.com"  
      }  
    }  
  }  
}
```

#### Api\_DeleteBlListItem

Удаление ресурса профиля Черных/Белых списков.

Параметры:

- ListItem - object: модель ресурса

Пример запроса:

```
[  
  {  
    "jsonrpc": "2.0",  
    "method": "Api_DeleteBlListItem",  
    "params": {  
      "ListItem": {  
        "enabled": 1,  
        "item_id": 2098644,  
        "task_id": "234",  
        "type": "sni",  
        "uri": "*.detik.com"  
      }  
    },  
    "id": 1515929803508  
  }  
]
```

```
}  
]
```

Пример успешного ответа:

```
{  
  "jsonrpc": "2.0",  
  "id": 1515929803508,  
  "result": {  
    "success": true  
  }  
}
```

### **Api\_ImportBLListItems**

Импорт массива моделей ресурсов профиля Черных/Белых списков.

Параметры:

- Items - array: массив моделей ресурсов

Пример запроса:

```
[  
  {  
    "jsonrpc": "2.0",  
    "method": "Api_ImportBLListItems",  
    "params": {  
      "Items": [  
        {  
          "enabled": 1,  
          "item_id": 0,  
          "task_id": "234",  
          "type": "sni",  
          "uri": "*.detik.com"  
        },  
        ...  
      ]  
    },  
    "id": 1515929803508  
  }  
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true
  }
}
```

### Api\_ImportBLListItemsFromQoE

Импорт ресурсов по категориям из QoE.

Параметры:

- Task - object: модель профиля
- Categories - string: идентификаторы категорий ресурсов из QoE через запятую

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_ImportBLListItemsFromQoE",
    "params": {
      "Categories": "7,8,9,10",
      "Task": {
        "data": [],
        "date": "2019-03-22 06:28:25",
        "form_data": "{\"title\": \"All range1\", \"white\": 1, \"redirect_enabled\": \"\", \"redirect\": \"spectredpi.com\", \"target\": \"\"}",
        "hardware_id": 1,
        "parent_id": null,
        "status": 1,
        "task_id": 234,
        "title": "All range1",
        "user_id": 1,
        "white": 1
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true
  }
}
```

### Api\_GetBITargets

Получение списка абонентов профилей Черных/Белых списков.

Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- task\_id - int: идентификатор профиля
- target - string: логин абонента
- status - int: статус абонента
- order\_by - string: поле, по которому будет упорядочен список абонентов
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetBITargets",
    "params": {
      "Params": {
        "order_by": "target",
        "order_dir": "desc",
        "page": 0,
        "tasks_id": 234,
        "size": 100
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 6,
      "rows": [
        {
          "status": 1,
          "task_id": "234",
          "target": "4.4.4.4"
        },
        ...
      ]
    }
  }
}

```

### Api\_SaveBlTargets

Сохранение нескольких абонентов профиля Черных/Белых списков. Параметры:

- Targets - array: массив моделей абонентов профиля

Пример запроса:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveBlTargets",
    "params": {
      "Targets": [
        {
          "status": 1,
          "task_id": "234",
          "target": "4.4.4.4"
        },
        {
          "status": 1,
          "task_id": "234",
          "target": "10.64.66.101"
        }
      ]
    }
  },

```

```
    "id":1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc":"2.0",
  "id":1515929803508,
  "result":{
    "success":true
  }
}
```

### **Api\_SaveBlTarget**

Сохранение абонента профиля Черных/Белых списков. Параметры:

- NewTarget - object: новая модель абонента профиля
- OldTarget - object: старая модель абонента профиля

Пример запроса:

```
[
  {
    "jsonrpc":"2.0",
    "method":"Api_SaveBlTarget",
    "params":{
      "NewTarget":{
        "status": 1,
        "task_id": "234",
        "target": "4.4.4.4"
      },
      "OldTarget":null
    },
    "id":1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc":"2.0",
  "id":1515929803508,
```

```
"result":{
  "success":true
}
}
```

### **Api\_DeleteBlTarget**

Удаление абонента профиля Черных/Белых списков. Параметры:

- Target - object: модель абонента профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteBlTarget",
    "params": {
      "Target": {
        "status": 1,
        "task_id": "234",
        "target": "4.4.4.4"
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true
  }
}
```

### **Управление записью PCAP**

В разработке.

## Управление мини Firewall-ом

- [Api\\_GetMiniFirewallProfiles](#)
- [Api\\_SaveMiniFirewallProfile](#)
- [Api\\_DeleteMiniFirewallProfile](#)
- [Api\\_RunMiniFirewallProfile](#)
- [Api\\_StopMiniFirewallProfile](#)

### Api\_GetMiniFirewallProfiles

Получение списка профилей Мини Firewall. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- hardware\_id - int: идентификатор оборудования
- title - string: имя профиля
- title\_equal - string: точное имя профиля
- profile\_id - int: Идентификатор профиля
- profile\_id\_not\_eqaul - int: Идентификатор профиля
- profile\_ids - string: Строка с перечислением идентификаторов профилей через запятую
- status - int: 0|1 ⇒ Выкл|Вкл
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[{
  "jsonrpc": "2.0",
  "method": "Api_GetMiniFirewallProfiles",
  "params":
  {
    "Params":
    {
      "page": 0,
      "size": 100,
      "order_by": "profile_id",
      "order_dir": "desc",
      "profiles_ids": null,
      "hardware_id": 10
    }
  },
  "id": 1570094137904
}]
```

Пример успешного ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1570094137904,
  "result":
  {
    "success": true,
    "data":
    {
      "page": 0,
      "size": 100,
      "count": 7, "rows":
      [
        {
          "profile_id": "1202",
          "hardware_id": "10",
          "title": "qq",
          "date": "2019-10-02 16:22:42",
          "form_data": "{\"title\": \"qq\", \"max_port\": 1000, \"port_holes\": [], \"out_port\": [555]}", "status": "0", "sync_marker": "0"
        },
        {...}
      ]
    }
  }
}

```

### Api\_SaveMiniFirewallProfile

Сохранение профиля Мини Firewall. Параметры:

- Form - object: модель данных из формы
- Task - object: модель профиля

Пример запроса:

```

[ {
  "jsonrpc": "2.0",
  "method": "Api_SaveMiniFirewallProfile",
  "params":
  {
    "Profile":
    {
      "id": 1570094137951,
      "profile_id": 0,
      "hardware_id": "10",
      "title": "",
      "date": 0,
      "status": 0,
      "form_data": "{\"title\": \"\", \"max_port\": 0, \"port_holes\": [], \"out_port\": [], \"valid\": true}"
    }
  }
} ]

```

```

        "sync_marker":0,
        "valid":true
    },
    "Form":
    {
        "title":"test",
        "max_port":10240,
        "port_holes":[80,8080],
        "out_port":[25,465],
        "valid":true
    }
},
"id":1570094137951
}]

```

Пример успешного ответа:

```

{
  "jsonrpc":"2.0",
  "id":1570094137951,
  "result":
  {
    "success":true,
    "data":
    {
      "profile":
      {
        "profile_id":"1509",
        "hardware_id":"10",
        "title":"test",
        "date":"2019-10-03 09:44:22",
        "form_data":"{\\"title\\":\\"test\\",\\"max_port\\":10240,\\"port_holes\\": [80,8080]
,\\"out_port\\": [25,465]}",
        "status":"0",
        "sync_marker":"0"
      }
    }
  }
}

```

### Api\_DeleteMiniFirewallProfile

Удаление профиля Мини Firewall. Параметры:

- Profile - object: модель профиля

Пример запроса:

```

[ {
  "jsonrpc":"2.0",

```

```
"method": "Api_DeleteMiniFirewallProfile",
"params":
{
  "Profile":
  {
    "id": 1570094137926,
    "profile_id": "1202",
    "hardware_id": "10",
    "title": "qq",
    "date": "2019-10-02 16:22:42",
    "status": 0,
"form_data": "{\"title\": \"qq\", \"max_port\": 1000, \"port_holes\": [], \"out_port\": [555]}",
    "sync_marker": "0",
    "valid": true
  }
},
  "id": 1570094137926
}]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1570094137926,
  "result": {
    "success": true
  }
}
```

### **Api\_RunMiniFirewallProfile**

Запуск профиля Мини Firewall. Параметры:

- Profile - object: модель профиля

Пример запроса:

```
[{
  "jsonrpc": "2.0",
  "method": "Api_RunMiniFirewallProfile",
  "params":
  {
    "Profile":
    {
      "id": 1570094137951,
      "profile_id": "1509",
      "hardware_id": "10",
```

```

        "title":"test",
        "date":"2019-10-03 09:44:22",
        "status":0,
"form_data":{"title":"test","max_port":10240,"port_holes":[80,8080]
,\out_port":[25,465]}",
        "sync_marker":"0",
        "valid":true
    }
},
    "id":1570094137951
}]

```

Пример успешного ответа:

```

{
  "jsonrpc":"2.0",
  "id":1570094137951,
  "result":
  {
    "success":true,
    "data":
    {
      "profile_id":"1509",
      "hardware_id":"10",
      "title":"test",
      "date":"2019-10-03 09:44:22",
"form_data":{"title":"test","max_port":10240,"port_holes":[80,8080]
,\out_port":[25,465]}",
      "status":1,
      "sync_marker":"0"
    }
  }
}

```

### Api\_StopMiniFirewallProfile

Остановка профиля Мини Firewall. Параметры:

- Profile - object: модель профиля

Пример запроса:

```

[ {
  "jsonrpc":"2.0",
  "method":"Api_StopMiniFirewallProfile",
  "params":
  {
    "Profile":
    {
      "id":1570094137951,

```

```
        "profile_id": "1509",
        "hardware_id": "10",
        "title": "test",
        "date": "2019-10-03 09:44:22",
        "status": 1,
"form_data": "{\"title\": \"test\", \"max_port\": 10240, \"port_holes\": [80, 8080]
, \"out_port\": [25, 465]}",
        "sync_marker": "0",
        "valid": true
    }
},
    "id": 1570094137951
}]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1570094137951,
  "result": {
    "success": true,
    "data": {
      "profile_id": "1509",
      "hardware_id": "10",
      "title": "test",
      "date": "2019-10-03 09:44:22",
"form_data": "{\"title\": \"test\", \"max_port\": 10240, \"port_holes\": [80, 8080]
, \"out_port\": [25, 465]}",
      "status": 0,
      "sync_marker": "0"
    }
  }
}
```

## Управление L2 BRAS

В разработке.

## Управление L3 BRAS

В разработке.

## Управление CG NAT

- [Api\\_GetCgNatProfiles](#)

- [Api\\_SaveCgNatProfile](#)
- [Api\\_DeleteCgNatProfile](#)
- [Api\\_RunCgNatProfile](#)
- [Api\\_StopCgNatProfile](#)
- [Api\\_GetCgNatDetailStatus](#)
- [Api\\_GetCgNatFullStatus](#)
- [Api\\_GetCgNatSubscribersStatus](#)

### Api\_GetCgNatProfiles

Получение списка CG NAT профилей. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- hardware\_id - int: идентификатор оборудования
- title - string: имя профиля
- title\_equal - string: точное имя профиля
- profile\_id - int: Идентификатор профиля
- profile\_id\_not\_eqaul - int: Идентификатор профиля
- profile\_ids - string: Строка с перечислением идентификаторов профилей через запятую
- status - int: 0|1 ⇒ Выкл|Вкл
- nat\_type - int: Тип профиля (0 ⇒ CG NAT, 1 ⇒ 1:1)
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetCgNatProfiles",
    "params": {
      "Params": {
        "page": 0,
        "size": 100,
        "order_by": "profile_id",
        "order_dir": "desc",
        "profiles_ids": null,
        "hardware_id": 1
      }
    },
    "id": 1564488173195
  }
]
```

Пример успешного ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1564488173195,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 4,
      "rows": [
        {
          "profile_id": "755",
          "hardware_id": "1",
          "title": "stress",
          "date": "2019-07-22 20:24:25",
          "form_data": "{\"title\": \"stress\", \"nat_ip_pool\": \"5.199.42.0/28\", \"nat_type\": 1, \"nat_tcp_max_sessions\": 0, \"nat_udp_max_sessions\": 0}\",
          "status": "0",
          "nat_type": "1",
          "subscribers_updated": "1",
          "sync_marker": "2039079626"
        },
        {...},
        ...
      ]
    }
  }
}

```

### Api\_SaveCgNatProfile

Сохранение CG NAT профиля. Параметры:

- Form - object: модель данных из формы
- Task - object: модель профиля

Пример запроса:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_SaveCgNatProfile",
    "params": {
      "Profile": {
        "id": 1564488173202,
        "profile_id": "752",
        "hardware_id": "1",
        "title": "stress",
        "date": "2019-07-22 20:24:25",
        "status": 0,

```

```

        "nat_type":1,
"form_data":{"title\":"stress\","nat_ip_pool\":"5.200.44.0/32\","nat_t
ype\":1,"nat_tcp_max_sessions\":0,"nat_udp_max_sessions\":0},
        "subscribers_updated":1,
        "sync_marker":"2039079626",
        "valid":true
    },
    "Form":{
        "title":"stress",
        "nat_ip_pool":"5.200.44.0/32",
        "nat_type":1,
        "nat_tcp_max_sessions":0,
        "nat_udp_max_sessions":0,
        "valid":true
    }
},
    "id":1564488173202
}
]

```

Пример успешного ответа:

```

{
    "jsonrpc":"2.0",
    "id":1564488173202,
    "result":{
        "success":true,
        "data":{
            "profile":{
                "profile_id":"752",
                "hardware_id":"1",
                "title":"stress",
                "date":"2019-07-30 13:07:56",
"form_data":{"title\":"stress\","nat_ip_pool\":"5.200.44.0\\\/32\","na
t_type\":1,"nat_tcp_max_sessions\":0,"nat_udp_max_sessions\":0},
                "status":"0",
                "nat_type":"1",
                "subscribers_updated":"1",
                "sync_marker":"2039079626"
            }
        }
    }
}

```

#### Api\_DeleteCgNatProfile

Удаление CG NAT профиля. Параметры:

- Profile - object: модель профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeleteCgNatProfile",
    "params": {
      "Profile": {
        "id": 1564488173351,
        "profile_id": "773",
        "hardware_id": "1",
        "title": "test_delete",
        "date": "2019-07-30 13:16:12",
        "status": 0,
        "nat_type": 0,
        "form_data": "{\"title\": \"test_delete\", \"nat_ip_pool\": \"192.168.1.90\\/24\", \"nat_type\": 0, \"nat_tcp_max_sessions\": 1500, \"nat_udp_max_sessions\": 1500 }",
        "subscribers_updated": 0,
        "sync_marker": "0",
        "valid": true
      }
    },
    "id": 1564488173351
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1564488173351,
  "result": {
    "success": true
  }
}
```

### Api\_RunCgNatProfile

Запуск CG NAT профиля. Параметры:

- Profile - object: модель профиля

Пример запроса:

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

```

    "method": "Api_RunCgNatProfile",
    "params": {
      "Profile": {
        "id": 1564494014854,
        "profile_id": "752",
        "hardware_id": "1",
        "title": "stress",
        "date": "2019-07-30 13:21:06",
        "status": 0,
        "nat_type": 1,
        "form_data": "{\\"title\\":\\"stress\\",\\"nat_ip_pool\\":\\"5.200.44.0\\\/32\\",\\"nat_type\\":1,\\"nat_tcp_max_sessions\\":0,\\"nat_udp_max_sessions\\":0}",
        "subscribers_updated": 1,
        "sync_marker": "2039079626",
        "valid": true
      }
    },
    "id": 1564494014854
  }
]

```

Пример успешного ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1564494014854,
  "result": {
    "success": true,
    "data": {
      "profile_id": "752",
      "hardware_id": "1",
      "title": "stress",
      "date": "2019-07-30 13:21:06",
      "form_data": "{\\"title\\":\\"stress\\",\\"nat_ip_pool\\":\\"5.200.44.0\\\/32\\",\\"nat_type\\":1,\\"nat_tcp_max_sessions\\":0,\\"nat_udp_max_sessions\\":0}",
      "status": 1,
      "nat_type": "1",
      "subscribers_updated": "1",
      "sync_marker": "2039079626"
    }
  }
}

```

### Api\_StopCgNatProfile

Остановка CG NAT профиля. Параметры:

- Profile - object: модель профиля

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_StopCgNatProfile",
    "params": {
      "Profile": {
        "id": 1564494014854,
        "profile_id": "752",
        "hardware_id": "1",
        "title": "stress",
        "date": "2019-07-30 13:21:06",
        "status": 1,
        "nat_type": 1,
        "form_data": "{\\"title\\":\\"stress\\",\\"nat_ip_pool\\":\\"5.200.44.0\\\\"/32\\",\\"nat_type\\":1,\\"nat_tcp_max_sessions\\":0,\\"nat_udp_max_sessions\\":0}",
        "subscribers_updated": 1,
        "sync_marker": "2039079626",
        "valid": true
      }
    },
    "id": 1564494014854
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1564494014854,
  "result": {
    "success": true,
    "data": {
      "profile_id": "752",
      "hardware_id": "1",
      "title": "stress",
      "date": "2019-07-30 13:21:06",
      "form_data": "{\\"title\\":\\"stress\\",\\"nat_ip_pool\\":\\"5.200.44.0\\\\"/32\\",\\"nat_type\\":1,\\"nat_tcp_max_sessions\\":0,\\"nat_udp_max_sessions\\":0}",
      "status": 0,
      "nat_type": "1",
      "subscribers_updated": "1",
      "sync_marker": "2039079626"
    }
  }
}
```

### Api\_GetCgNatDetailStatus

Получение подробного статуса CG NAT профиля. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")
- hardware\_id - int: идентификатор оборудования
- profile\_id - int: Идентификатор профиля
- profile\_ids - string: Строка с перечислением идентификаторов профилей через запятую
- proto - string: Используемый протокол (TCP / UDP)
- ip\_mask - string: IP маска( 0x0 / 0x1)
- nthr - int: Номер потока (0 / 1)
- whip - string: Белый IP адрес
- num\_port - int: Общее количество портов ( >= )
- use\_port - int: Используемое количество портов ( >= )

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetCgNatDetailStatus",
    "params": {
      "Params": {
        "page": 0,
        "size": 100,
        "order_by": "profile_id",
        "order_dir": "desc",
        "profile_ids": "753",
        "hardware_id": "1"
      }
    },
    "id": 1564494015645
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1564494015645,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 16,
      "rows": [
        {
```

```

        "profile_id": "753",
        "proto": "TCP",
        "nthr": "0",
        "ip_mask": "0x0",
        "whip": "8.8.8.10",
        "num_port": "64512",
        "use_port": "0",
        "sync_marker": "2039079626",
        "title": "1st_profile1"
    },
    {...},
    ...
]
}
}
}

```

### Api\_GetCgNatFullStatus

Получение общего статуса CG NAT профиля. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")
- hardware\_id - int: идентификатор оборудования
- profile\_id - int: Идентификатор профиля
- profile\_ids - string: Строка с перечислением идентификаторов профилей через запятую
- proto - string: Используемый протокол (TCP / UDP)
- ip\_mask - string: IP маска( 0x0 / 0x1)
- nthr - int: Номер потока (0 / 1)
- whip - string: Белый IP адрес
- num\_port - int: Общее количество портов ( >= )
- use\_port - int: Используемое количество портов ( >= )
- num\_addr - int: Общее количество адресов ( >= )
- use\_addr - int: Используемое количество адресов ( >= )

Пример запроса:

```

[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetCgNatFullStatus",
    "params": {
      "Params": {
        "page": 0,

```

```
        "size":100,  
        "order_by":"profile_id",  
        "order_dir":"desc",  
        "profile_ids":"753",  
        "hardware_id":"1"  
    }  
},  
"id":1564494015647  
}  
]
```

Пример успешного ответа:

```
{  
  "jsonrpc":"2.0",  
  "id":1564494015647,  
  "result":{  
    "success":true,  
    "data":{  
      "page":0,"size":100,  
      "count":2,  
      "rows":[  
        {  
          "profile_id":"753",  
          "proto":"TCP",  
          "nthr":"0",  
          "ip_mask":"0x0",  
          "num_addr":"8",  
          "use_addr":"0",  
          "num_port":"516096",  
          "use_port":"0",  
          "sync_marker":"2039079626",  
          "title":"1st_profile1"  
        },  
        {  
          "profile_id":"753",  
          "proto":"UDP",  
          "nthr":"0",  
          "ip_mask":"0x0",  
          "num_addr":"8",  
          "use_addr":"0",  
          "num_port":"516096",  
          "use_port":"0",  
          "sync_marker":"2039079626",  
          "title":"1st_profile1"  
        }  
      ]  
    }  
  }  
}
```

## Api\_GetCgNatSubscribersStatus

Получение статуса абонентов CG NAT профиля. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- order\_by - string: поле, по которому будет упорядочен список профилей
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")
- hardware\_id - int: идентификатор оборудования
- profile\_id - int: Идентификатор профиля
- profile\_ids - string: Строка с перечислением идентификаторов профилей через запятую
- ipv4 - string: Серый IP адрес
- whiteip - string: Белый IP адрес
- sess\_tcp - int: Количество TCP сессий ( >= )
- sess\_udp - int: Количество UDP сессий ( >= )

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetCgNatSubscribersStatus",
    "params": {
      "Params": {
        "page": 0,
        "size": 100,
        "order_by": "profile_id",
        "order_dir": "desc",
        "profile_ids": "753",
        "hardware_id": "1"
      }
    },
    "id": 1564494015649
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1564494015649,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 15,

```

```
    "rows": [
      {
        "profile_id": "753",
        "ipv4": "192.168.15.25",
        "whiteip": "5.200.44.1",
        "sess_tcp": "134",
        "sess_udp": "62",
        "sync_marker": "2039079626"
      },
      {...},
      ...
    ]
  }
}
```

### **Управление защитой от DOS и DDOS атак**

В разработке.

### **Управление кешированием**

В разработке.

### **Управление уведомлениями абонентов**

В разработке.

### **Управление блокировкой и заменой рекламы**

В разработке.

### **Управление бонусной программой**

В разработке.

### **Управление тарифами**

- [Api\\_GetPolicingProfiles](#)
- [Api\\_SavePolicingProfile](#)
- [Api\\_SavePolicingProfileConfig](#)
- [Api\\_DeletePolicingProfile](#)

## Api\_GetPolicingProfiles

Получение списка профилей тарифных планов. Параметры:

- Params - object: параметры поиска(необязательный параметр)

Возможные параметры поиска Params:

- page - int: номер страницы
- size - int: размерность массива с результатом
- hardware\_id - int: идентификатор оборудования
- title - string: имя профиля
- individual - int:0|1
- profile\_id - int: идентификатор профиля тарифного плана
- order\_by - string: поле, по которому будет упорядочен список абонентов
- order\_dir - string: упорядочивание списка по возрастанию("asc")/убыванию("desc")

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetPolicingProfiles",
    "params": {
      "Params": {
        "hardware_id": 1,
        "individual": "0",
        "order_by": "profile_id",
        "order_dir": "desc",
        "page": 0,
        "profile_id": null,
        "size": 100,
        "title": ""
      }
    },
    "id": 1515929803508
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929803508,
  "result": {
    "success": true,
    "data": {
      "page": 0,

```

```

        "size":100,
        "count":6,
        "rows":[
            {
                "config": "{\"type\":\"HTB\",...}"
                "config_updated": "0"
                "dpi_name": "at9Eneeg29"
                "enabled": "0"
                "hardware_id": "1"
                "individual": "0"
                "profile_id": "276"
                "status": "0"
                "subs_updated": "0"
                "title": "10mbit"
                "user_id": "1"
            },
            ...
        ]
    }
}

```

## Api\_SavePolicingProfile

Сохранение модели абонента.

Параметры:

- Profile - object: модель профиля тарифного плана с новыми данными
- Config - string: конфигурация тарифного плана. Если данный параметр указан, производится обновление конфигурация тарифного плана.

Пример запроса при сохранении профиля:

```

[
  {
    "jsonrpc":"2.0",
    "method":"Api_SavePolicingProfile",
    "params":{
      "Config": "htb_inbound_root=rate
10mbit<htb_inbound_class0=rate 8bit ceil 10mbit<htb_inbound_class1=rate 8bit
ceil...",
      "Profile":{
        "config": "",
        "config_updated": 0,
        "dpi_name": "",
        "enabled": 0,
        "hardware_id": "1",
        "individual": 0,

```

```

        "profile_id": 0,
        "status": 0,
        "subs_updated": 0,
        "title": "test_save"
        "user_id": 0,
        "valid": true
    }
},
    "id":1515929803535
}
]

```

Пример успешного ответа при создании профиля:

```

{
  "jsonrpc":"2.0",
  "id":1553521000824,
  "result":{
    "success":true,
    "data":{
      "profile":{
        "profile_id":"280",
        "title":"test_save",
        "hardware_id":"1",
        "user_id":"1",
        "enabled":"0",
"config":{"type":"HTB",inbound":{"root_rate":"10mbit",root_ceil":"10mbit",root_burst":"5mbit",root_cburst":"5mbit",classes":["...]},outbound":{"root_rate":"10mbit",root_ceil":"10mbit",root_burst":"5mbit",root_cburst":"5mbit",classes":["...]}}",
        "subs_updated":"0",
        "config_updated":"0",
        "dpi_name":"4t0KYYU83Z",
        "status":"0",
        "individual":"0"
      }
    }
  }
}

```

## Api\_SavePolicingProfileConfig

Сохранение конфигурации профиля тарифного плана.

Параметры:

- Profile - object: модель профиля тарифного плана.
- Config - string: конфигурация тарифного плана.

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_SavePolicingProfile",
    "params": {
      "Config": "htb_inbound_root=rate
20mbit<htb_inbound_class0=rate 8bit ceil 20mbit<htb_inbound_class1=rate 8bit
ceil...",
      "Profile": {
        "config":
"{\"type\": \"HTB\", \"inbound\": {\"root_rate\": \"10mbit\", \"root_ceil\": \"10m
bit\", \"root_burst\": \"5mbit\", \"root_cburst\": \"5mbit\", \"classes\": [...]},
\"outbound\": {\"root_rate\": \"10mbit\", \"root_ceil\": \"10mbit\", \"root_burst
\": \"5mbit\", \"root_cburst\": \"5mbit\", \"classes\": [...]}}",
        "config_updated": 0,
        "dpi_name": "ztovKBaiZs",
        "enabled": 0,
        "hardware_id": "1",
        "individual": 0,
        "profile_id": 278,
        "status": 0,
        "subs_updated": 0,
        "title": "test_save"
        "user_id": 1,
        "valid": true
      }
    },
    "id": 1515929804638
  }
]
```

Пример успешного ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1515929804638,
  "result": {
    "success": true,
    "data": {
      "profile": {
        config:
{"type": "HTB", "inbound": {"root_rate": "20mbit", "root_ceil": "20mbit", "root_bu
rst": "10mbit", "root_cburst": "10mbit", "classes": [...]}, "outbound": {"root_rate
": "20mbit", "root_ceil": "20mbit", "root_burst": "10mbit", "root_cburst": "10mbit"
, "classes": [...]}}
        config_updated: "0"
        dpi_name: "ztovKBaiZs"
        enabled: "0"
        hardware_id: "1"
        individual: "0"
      }
    }
  }
}
```

```
        profile_id: "278"
        status: "0"
        subs_updated: "0"
        title: "test_save"
        user_id: "1"
    }
}
```

## Api\_DeletePolicingProfile

Удаление профиля тарифного плана.

Параметры:

- Profile - объект: модель профиля тарифного плана.

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_DeletePolicingProfile",
    "params": {
      "Profile": {
        config:
        '{"type": "HTB", "inbound": {"root_rate": "20mbit", "root_ceil": "20mbit", "root_bu
rst": "10mbit", "root_cburst": "10mbit", "classes": [...]}, "outbound": {"root_rate
": "20mbit", "root_ceil": "20mbit", "root_burst": "10mbit", "root_cburst": "10mbit"
, "classes": [...]}'
        config_updated: "0"
        dpi_name: "ztovKBaiZs"
        enabled: "0"
        hardware_id: "1"
        individual: "0"
        profile_id: "278"
        status: "0"
        subs_updated: "0"
        title: "test_save"
        user_id: "1"
      }
    },
    "id": 1515929804638
  }
]
```

Пример успешного ответа:

```
{
```

```

"jsonrpc": "2.0",
"id": 1515929804638,
"result": {
  "success": true,
  "data": {
    "profile": {
      config:
{"type": "HTB", "inbound": {"root_rate": "20mbit", "root_ceil": "20mbit", "root_bu
rst": "10mbit", "root_cburst": "10mbit", "classes": [...]}, "outbound": {"root_rate
": "20mbit", "root_ceil": "20mbit", "root_burst": "10mbit", "root_cburst": "10mbit"
, "classes": [...]}"}
      config_updated: "0"
      dpi_name: "ztovKBaiZs"
      enabled: "0"
      hardware_id: "1"
      individual: "0"
      profile_id: "278"
      status: "0"
      subs_updated: "0"
      title: "test_save"
      user_id: "1"
    }
  }
}

```

## Управление приоритизацией по протоколам (DSCP)

- [Api\\_GetDscpProtocolsConfig](#)
- [Api\\_SetDscpProtocolsConfig](#)
- [Api\\_GetDscpProtocolsConfigHistory](#)
- [Api\\_GetDscpProtocolsConfigFile](#)

### Api\_GetDscpProtocolsConfig

Получение конфигурации приоритизации по протоколу DSCP.

Параметры:

- Id - int: идентификатор оборудования

Пример запроса:

```

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

```

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

Пример успешного ответа:

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

### Api\_SetDscpProtocolsConfig

Установка конфигурации приоретизации по протоколу DSCP.

Параметры:

- Id - int: идентификатор оборудования
- Config - string: тело конфигурации

Пример запроса:

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

Пример успешного ответа:

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

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

## Api\_GetDscpProtocolsConfigHistory

Получение истории изменения конфигураций приоретизации по протоколу DSCP.

Параметры:

- Id - int: идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

```
{
  "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",
      ...
    ]
  }
}
```



Параметры:

- Id - int: идентификатор оборудования

Пример запроса:

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

Пример успешного ответа:

```
{
  "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

Установка конфигурации приоретизации по ASN.

Параметры:

- Id - int: идентификатор оборудования
- AsnConfig - string: тело конфигурации ASN
- AsnDscpConfig - string: тело конфигурации ASNDSCP

Пример запроса:

```
[
  {
    "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
  }
]
```

Пример успешного ответа:

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

### Api\_GetAsnDscpConfigHistory

Получение истории изменения конфигураций приоретизации по ASN.

Параметры:

- Id - int: идентификатор оборудования

Пример запроса:

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

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

Пример успешного ответа:

```
{
  "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

Получение файла конфигурации из истории по имени файла.

Параметры:

- Id - int: идентификатор оборудования
- File - string: имя файла

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetAsnDscpConfigFile",
    "params": {
      "Id": 1,
      "File": "2015.06.07.08.29.29.000000.asnum.dscp"
    },
    "id": 1515661809137
  }
]
```

```
]
```

Пример успешного ответа:

```
{
  "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"
  }
}
```

## Управление QoE

- [Api\\_GetQoEClientPredefinedReport](#)
- [Api\\_GetQoEFilters](#)
- [Api\\_SaveQoEFilter](#)
- [Api\\_DeleteQoEFilter](#)
- [Api\\_StartExpQoESubscribersToAdvcJob](#)

### Api\_GetQoEClientPredefinedReport

Получение QoE отчета.

Параметры:

- Config - Object: конфигурация отчета
- Params - Object: параметры отчета (фильтры)

**Пример: Топ хостов (Кликстрим)**

Пример запроса:

```
[
  {
    "jsonrpc": "2.0",
    "method": "Api_GetQoEClientPredefinedReport",
    "params": {
      "Config": {
        "type": "ClientPredefined",
        "table": "clickstream_report",
        "values": [
          {
```

```

        "col": "",
        "alias": "total",
        "func": "count"
    },
    {
        "col": "session_id",
        "alias": "sessions_uniq",
        "func": "uniq"
    },
    {
        "col": "subscriber",
        "alias": "subscribers_uniq",
        "func": "uniq"
    },
    {
        "col": "user_agent",
        "alias": "user_agent_uniq",
        "func": "uniq"
    },
    {
        "col": "device_other",
        "alias": "device_other_uniq",
        "func": "uniq"
    },
    {
        "col": "host_ip",
        "alias": "host_ip_uniq",
        "func": "uniq"
    }
],
"rows": [
    "host"
],
"sort": {
    "id": "total",
    "dir": "desc"
},
"with_totals": true
},
"Params": {
    "table": "clickstream_report",
    "page": 0,
    "size": 0,
    "period": null,
    "filter": {
        "period": {
            "id": "time",
            "operator": "between",
            "value": [
                "2018-08-20 14:19:43",
                "2018-09-20 14:19:43"
            ]
        }
    }
}

```

```

        ]
      },
      "host":""
    },
    "sort":{
      "id":"total",
      "dir":"desc"
    }
  }
},
"id":1537438732453
}
]

```

Пример ответа:

```

{
  "jsonrpc":"2.0",
  "id":1537438732453,
  "result":{
    "success":true,
    "data":{
      "page":0,
      "size":100,
      "count":2317,
      "series":[

    ],
    "rows":[
      {
        "host":"www.google.com",
        "total":"3287",
        "sessions_uniq":"1817",
        "subscribers_uniq":"3",
        "user_agent_uniq":"1",
        "device_other_uniq":"1",
        "host_ip_uniq":"58"
      },
      {
        "host":"*.google.com.ru",
        "total":"2838",
        "sessions_uniq":"2838",
        "subscribers_uniq":"3",
        "user_agent_uniq":"1",
        "device_other_uniq":"1",
        "host_ip_uniq":"39"
      },
      {
        "host":"*.google.com",
        "total":"2328",
        "sessions_uniq":"2328",

```

```

        "subscribers_uniq":"3",
        "user_agent_uniq":"1",
        "device_other_uniq":"1",
        "host_ip_uniq":"158"
    },
    ...
    {
        "host":"edge.skype.com",
        "total":"130",
        "sessions_uniq":"109",
        "subscribers_uniq":"3",
        "user_agent_uniq":"1",
        "device_other_uniq":"1",
        "host_ip_uniq":"1"
    }
],
"totals":{
    "host":"",
    "total":"74141",
    "sessions_uniq":"45680",
    "subscribers_uniq":"3",
    "user_agent_uniq":"91",
    "device_other_uniq":"42",
    "host_ip_uniq":"3053"
}
}
}
}
}

```

#### Пример: RTT распределение (Нетфлю)

Пример запроса:

```

[
  {
    "jsonrpc":"2.0",
    "method":"Api_GetQoEClientPredefinedReport",
    "params":{
      "Config":{
        "type":"ClientPredefined",
        "table":"fullflow_report",
        "values":[
          {
            "col":"",
            "alias":"RTT распределение",
            "func":"countIf(rtt <> 0)"
          },
          {
            "col":"",

```

```

        "alias": "RTT распределение от абонента",
        "func": "countIf(rtt_from_subscriber <> 0)"
    },
    {
        "col": "",
        "alias": "RTT распределение к абоненту",
        "func": "countIf(rtt_to_subscriber <> 0)"
    }
],
"rows": [
    {
        "col": "rtt",
        "alias": "rtt",
        "func": ""
    }
],
"sort": {
    "id": "rtt",
    "dir": "asc"
},
"size": 1000
},
"Params": {
    "table": "fullflow_report",
    "page": 0,
    "size": 1000,
    "period": null,
    "filter": {
        "period": {
            "id": "time",
            "operator": "between",
            "value": [
                "2018-08-20 15:47:13",
                "2018-09-20 15:47:13"
            ]
        }
    },
    "sort": {
        "id": "rtt",
        "dir": "asc"
    }
}
},
"id": 1537447636531
}
]

```

Пример ответа:

```

{
    "jsonrpc": "2.0",

```

```
"id":1537447636531,
"result":{
  "success":true,
  "data":{
    "page":0,
    "size":1000,
    "count":1262,
    "series":{

  },
  "rows":[
    {
      "rtt":0,
      "RTT распределение":"0",
      "RTT распределение от абонента":"0",
      "RTT распределение к абоненту":"0"
    },
    {
      "rtt":1,
      "RTT распределение":"7641",
      "RTT распределение от абонента":"6160",
      "RTT распределение к абоненту":"1481"
    },
    {
      "rtt":2,
      "RTT распределение":"20061",
      "RTT распределение от абонента":"15859",
      "RTT распределение к абоненту":"4202"
    },
    {
      "rtt":3,
      "RTT распределение":"12266",
      "RTT распределение от абонента":"10672",
      "RTT распределение к абоненту":"1594"
    },
    {
      "rtt":4,
      "RTT распределение":"3922",
      "RTT распределение от абонента":"3287",
      "RTT распределение к абоненту":"635"
    },
    {
      "rtt":5,
      "RTT распределение":"2347",
      "RTT распределение от абонента":"1501",
      "RTT распределение к абоненту":"846"
    },
    .....
    {
      "rtt":3856,
      "RTT распределение":"2",
```



```
"rows":[
  {
    "filter_id":"52",
    "parent_id":null,
    "user_id":"1",
    "title":"\u0420\u0435\u0437\u0443\u043b\u044c\u0442\u0430\u0442\u044b
speedtest",
    "date":"2018-09-13 11:27:36",
    "form_data":{"\filter_state\":[{\id\:1536837377540,\f_enabled\":0,\f_id
\":"host\","f_operator\":"like\","f_value\":"\","f_value_type\":"strin
g\","f_value_editor\":"text\"},{\id\:1536837377541,\f_enabled\":0,\f_
id\":"subscriber\","f_operator\":"like\","f_value\":"\","f_value_type\
":"string\","f_value_editor\":"text\"},{\id\:1536837377542,\f_enabled\
":0,\f_id\":"login\","f_operator\":"like\","f_value\":"\","f_value_ty
pe\":"string\","f_value_editor\":"text\"},{\id\:1536837377543,\f_enabl
ed\":1,\f_id\":"url\","f_operator\":"like\","f_value\":"speedtest.net\
\/result\","f_value_type\":"string\","f_value_editor\":"text\"},{\id\
":1536837377544,\f_enabled\":0,\f_id\":"device\","f_operator\":"like\","
f_value\":"\","f_value_type\":"string\","f_value_editor\":"text\"},{\
id\:1536837377545,\f_enabled\":0,\f_id\":"user_agent\","f_operator\":"
like\","f_value\":"\","f_value_type\":"string\","f_value_editor\":"tex
t\"},{\id\:1536837377546,\f_enabled\":0,\f_id\":"host_ip\","f_operato
r\":"like\","f_value\":"\","f_value_type\":"string\","f_value_editor\
":"text\"},{\id\:1536837377547,\f_enabled\":0,\f_id\":"device_other_uniq
\","f_operator\":">=\","f_value\":"\","f_value_type\":"string\","f_va
lue_editor\":"text\"},{\id\:1536837377548,\f_enabled\":0,\f_id\":"user
_agent_uniq\","f_operator\":">=\","f_value\":"\","f_value_type\":"stri
ng\","f_value_editor\":"text\"},{\id\:1536837377549,\f_enabled\":0,\f_
id\":"hosts_uniq\","f_operator\":">=\","f_value\":"\","f_value_type\
":"string\","f_value_editor\":"text\"},{\id\:1536837377550,\f_enabled\
":0,\f_id\":"host_ip_uniq\","f_operator\":">=\","f_value\":"\","f_value
_type\":"string\","f_value_editor\":"text\"}]}",
    "type":"subs_clickstream"
  },
  .....
  {
    "filter_id":"18",
    "parent_id":null,
    "user_id":"1",
    "title":"test",
    "date":"2018-08-20 17:24:37",
    "form_data":{"\filter_state\":[{\id\:1534776214196,\f_enabled\":1,\f_id
\":"hosts_uniq\","f_operator\":">=\","f_value\":"100\","f_value_type\
":"string\","f_value_editor\":"text\"},{\id\:1534776214197,\f_enabled\
":0,\f_id\":"device_other_uniq\","f_operator\":">=\","f_value\":"10\","
f_value_type\":"string\","f_value_editor\":"text\"},{\id\:153477621419
8,\f_enabled\":0,\f_id\":"host\","f_operator\":"like\","f_value\":"te
st\","f_value_type\":"string\","f_value_editor\":"text\"},{\id\:153477
6214199,\f_enabled\":0,\f_id\":"subscriber\","f_operator\":"like\","f_
value\":"qq\","f_value_type\":"string\","f_value_editor\":"text\"},{\i
d\:1534776214200,\f_enabled\":0,\f_id\":"device_other_uniq\","f_operato
```

```

r\":">=",\f_value\":"10",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1534776214201,\f_enabled\:0,\f_id\":"device",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1534776214202,\f_enabled\:0,\f_id\":"host_ip",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"}]}",
      "type":"subs_clickstream"
    }
  ]
}
}
}
}

```

## Api\_SaveQoEFilter

Сохранить QoE фильтр.

Пример запроса:

```

[
  {
    "jsonrpc":"2.0",
    "method":"Api_SaveQoEFilter",
    "params":{
      "Filter":{
        "id":1537447973674,
        "filter_id":"52",
        "parent_id":null,
        "title":"Результаты speedtest",
        "date":"2018-09-13 11:27:36",
        "form_data":"{\f_filter_state\":[{\id\:1536837377540,\f_enabled\:0,\f_id\":"host",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377541,\f_enabled\:0,\f_id\":"subscriber",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377542,\f_enabled\:0,\f_id\":"login",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377543,\f_enabled\:1,\f_id\":"url",\f_operator\":"like",\f_value\":"speedtest.net\/result",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377544,\f_enabled\:0,\f_id\":"device",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377545,\f_enabled\:0,\f_id\":"user_agent",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377546,\f_enabled\:0,\f_id\":"host_ip",\f_operator\":"like",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377547,\f_enabled\:0,\f_id\":"device_other_uniq",\f_operator\":">=",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377548,\f_enabled\:0,\f_id\":"user_agent_uniq",\f_operator\":">=",\f_value\":"",\f_value_type\":"string",\f_value_editor\":"text"},{\id\:1536837377549,\f_enabled\:0,\f_i

```

```
d\":"hosts_uniq\","\f_operator\":">=\","\f_value\":"\","\f_value_type\":"\
"string\","\f_value_editor\":"text\"},{\id\":1536837377550,\f_enabled\":0
,\f_id\":"host_ip_uniq\","\f_operator\":">=\","\f_value\":"\","\f_value_
type\":"string\","\f_value_editor\":"text\"]]}",
  "type":"subs_clickstream",
  "data":[

  ],
  "badge":1,
  "open":true,
  "valid":true
},
"Form":{
  "filter_state":[
    {
      "id":1536837377540,
      "f_enabled":0,
      "f_id":"host",
      "f_operator":"like",
      "f_value":"",
      "f_value_type":"string",
      "f_value_editor":"text"
    },
    {
      "id":1536837377541,
      "f_enabled":0,
      "f_id":"subscriber",
      "f_operator":"like",
      "f_value":"",
      "f_value_type":"string",
      "f_value_editor":"text"
    },
    {
      "id":1536837377542,
      "f_enabled":0,
      "f_id":"login",
      "f_operator":"like",
      "f_value":"",
      "f_value_type":"string",
      "f_value_editor":"text"
    },
    {
      "id":1536837377543,
      "f_enabled":1,
      "f_id":"url",
      "f_operator":"like",
      "f_value":"speedtest.net/result",
      "f_value_type":"string",
      "f_value_editor":"text"
    }
  ]
}
```

```
"id":1536837377544,
"f_enabled":0,
"f_id":"device",
"f_operator":"like",
"f_value":"",
"f_value_type":"string",
"f_value_editor":"text"
},
{
  "id":1536837377545,
  "f_enabled":0,
  "f_id":"user_agent",
  "f_operator":"like",
  "f_value":"",
  "f_value_type":"string",
  "f_value_editor":"text"
},
{
  "id":1536837377546,
  "f_enabled":0,
  "f_id":"host_ip",
  "f_operator":"like",
  "f_value":"",
  "f_value_type":"string",
  "f_value_editor":"text"
},
{
  "id":1536837377547,
  "f_enabled":0,
  "f_id":"device_other_uniq",
  "f_operator":>=",
  "f_value":"",
  "f_value_type":"string",
  "f_value_editor":"text"
},
{
  "id":1536837377548,
  "f_enabled":0,
  "f_id":"user_agent_uniq",
  "f_operator":>=",
  "f_value":"",
  "f_value_type":"string",
  "f_value_editor":"text"
},
{
  "id":1536837377549,
  "f_enabled":0,
  "f_id":"hosts_uniq",
  "f_operator":>=",
  "f_value":"",
  "f_value_type":"string",
```



```

_agent_uniq\", \"f_operator\": \">=\", \"f_value\": \"\", \"f_value_type\": \"string\", \"f_value_editor\": \"text\"}, {\"id\": 1536837377549, \"f_enabled\": 0, \"f_id\": \"hosts_uniq\", \"f_operator\": \">=\", \"f_value\": \"\", \"f_value_type\": \"string\", \"f_value_editor\": \"text\"}, {\"id\": 1536837377550, \"f_enabled\": 0, \"f_id\": \"host_ip_uniq\", \"f_operator\": \">=\", \"f_value\": \"\", \"f_value_type\": \"string\", \"f_value_editor\": \"text\"}}\",
    \"type\": \"subs_clickstream\"
  }
}
}
}

```

## Api\_DeleteQoEFilter

Удалить сохраненный QoE фильтр.

Пример запроса:

```

[
  {
    \"jsonrpc\": \"2.0\",
    \"method\": \"Api_DeleteQoEFilter\",
    \"params\": {
      \"Filter\": {
        \"id\": 1537447973810,
        \"filter_id\": \"53\",
        \"parent_id\": null,
        \"title\": \"test\",
        \"date\": \"2018-09-20 13:11:29\",
        \"form_data\": \"{\\\"filter_state\\\": [{\\\"id\\\": 1537447973833, \\\"f_enabled\\\": 1, \\\"f_id\\\": \\\"host\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973834, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"subscriber\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973835, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"login\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973836, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"url\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973837, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"device\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973838, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"user_agent\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973839, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"host_ip\\\", \\\"f_operator\\\": \\\"like\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973840, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"device_other_uniq\\\", \\\"f_operator\\\": \\\">=\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973841, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"user_agent_uniq\\\", \\\"f_operator\\\": \\\">=\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_editor\\\": \\\"text\\\"}, {\\\"id\\\": 1537447973842, \\\"f_enabled\\\": 0, \\\"f_id\\\": \\\"hosts_uniq\\\", \\\"f_operator\\\": \\\">=\\\", \\\"f_value\\\": \\\"\\\", \\\"f_value_type\\\": \\\"string\\\", \\\"f_value_ed

```

```
itor\":"text\"},{\"id\":1537447973843,\"f_enabled\":0,\"f_id\":\"host_ip_uniq\", \"f_operator\": \">=\", \"f_value\": \"\", \"f_value_type\": \"string\", \"f_value_editor\": \"text\"}}\",
    \"type\": \"subs_clickstream\",
    \"data\": [
        ],
    \"badge\": 1,
    \"open\": true,
    \"valid\": true
    }
},
\"id\": 1537447973810
}
]
```

Пример ответа:

```
{\"jsonrpc\": \"2.0\", \"id\": 1537447973810, \"result\": {\"success\": true}}
```

### Api\_StartExpQoESubscribersToAdvJob

Запустить фоновую задачу импорта абонентов в рекламную кампанию на основе сохраненного QoE фильтра.

Пример запроса:

```
[
  {
    \"jsonrpc\": \"2.0\",
    \"method\": \"Api_StartExpQoESubscribersToAdvJob\",
    \"params\": {
      \"Params\": {
        \"Config\": {
          \"type\": \"ClientPredefined\",
          \"table\": \"clickstream_report\",
          \"values\": [
            {
              \"col\": \"login\",
              \"alias\": \"login_top\",
              \"func\": \"topK(1)(login)\"
            },
            {
              \"col\": \"\",
              \"alias\": \"total\",
              \"func\": \"count\"
            },
            {
              \"col\": \"session_id\",
              \"alias\": \"sessons_uniq\",
```

```
        "func": "uniq"
    },
    {
        "col": "host",
        "alias": "hosts_uniq",
        "func": "uniq"
    },
    {
        "col": "user_agent",
        "alias": "user_agent_uniq",
        "func": "uniq"
    },
    {
        "col": "device_other",
        "alias": "device_other_uniq",
        "func": "uniq"
    },
    {
        "col": "host_ip",
        "alias": "host_ip_uniq",
        "func": "uniq"
    }
],
"rows": [
    "subscriber"
],
"sort": {
    "id": "total",
    "dir": "desc"
},
"with_totals": true
},
"Params": {
    "filter": {
        "period": {
            "id": "time",
            "operator": "between",
            "value": [
                "2018-08-20 15:54:10",
                "2018-09-20 15:54:10"
            ]
        },
        "1534931185441": {
            "id": "hosts_uniq",
            "operator": ">=",
            "value": "100",
            "sql_type": "having"
        }
    },
    "table": "clickstream_report"
},
}
```

```
        "Type": "by_ip",
        "Max": "0",
        "HwCampaignData": {
            "adc_id": "7",
            "hardware_id": "6"
        }
    },
    "id": 1537447974127
}
]
```

Пример ответа:

```
{"jsonrpc": "2.0", "id": 1537447974127, "result": {"success": true}}
```

## Управление рекламными кампаниями

- [Api\\_GetAdvProfiles](#)
- [Api\\_SaveAdvProfile](#)
- [Api\\_DeleteAdvProfile](#)
- [Api\\_GetAdvCompanies](#)
- [Api\\_SaveAdvCompany](#)
- [Api\\_DeleteAdvCompany](#)
- [Api\\_EnableAdvCompany](#)
- [Api\\_DisableAdvCompany](#)

### Api\_GetAdvProfiles

Получение профилей кампаний.

Параметры:

- Params - Object: параметры поиска (фильтры)

### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_GetAdvProfiles",
  "params": {
    "Params": {
      "hardware_id": 1,
      "size": 100,
      "page": 0,
      "title": ""
    }
  }
},
```

```
"id": 1081
}
```

#### Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 1,
      "rows": [
        {
          "profile_id": "6",
          "parent_id": null,
          "hardware_id": "1",
          "title": "test_profile_4",
          "user_id": "1",
          "deleted": "0",
          "data": []
        }
      ]
    }
  }
}
```

#### Api\_SaveAdvProfile

Создание/изменение данных профиля кампаний.

Параметры:

- Profile - Object: модель профиля кампании.

#### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_SaveAdvProfile",
  "params": {
    "Profile": {
      "profile_id": 0,
      "parent_id": null,
      "hardware_id": "1",

```

```
        "title": "save_profile"
    }
},
"id": 1081
}
```

#### Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "profile": {
        "profile_id": "20",
        "parent_id": null,
        "hardware_id": "1",
        "title": "save_profile",
        "user_id": "1",
        "deleted": "0",
        "data": []
      }
    }
  }
}
```

#### Api\_DeleteAdvProfile

Удаление профиля компании.

Параметры:

- Profile - Object: модель профиля компании.

#### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_DeleteAdvProfile",
  "params": {
    "Profile": {
      "profile_id": "20",
      "parent_id": null,
      "hardware_id": "1",
      "title": "save_profile",
      "user_id": "1",

```

```
        "deleted": "0",
        "data": []
    }
},
"id": 1081
}
```

#### Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "profile_id": "20",
      "parent_id": null,
      "hardware_id": "1",
      "title": "save_profile",
      "user_id": "1",
      "deleted": "0",
      "data": []
    }
  }
}
```

#### Api\_GetAdvCompanies

Получение рекламных кампаний.

Параметры:

- Params - Object: параметры поиска (фильтры)

#### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_GetAdvCompanies",
  "params": {
    "Params": {
      "profile_id": 6,
      "title": "test",
      "status": 4
    }
  }
},
"id": 1081
```

```
}
```

### Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 1,
      "rows": [
        {
          "adc_id": "3",
          "profile_id": "6",
          "title": "test_profile_4",
          "user_id": "1",
          "from_date": "2018-09-18",
          "to_date": "2018-10-30",
          "from_time": "06:16:00",
          "to_time": "23:55:00",
          "total": "16",
          "released": "16",
          "redirect": "google.com",
          "pack": "0",
          "days_of_week": "1,2,3,4,5,6,7",
          "status": "4",
          "params": "[]",
          "need_to_update": "1",
          "processed": "1,3,8",
          "prepared_url": "google.com?",
          "deleted": "0",
          "enabled": "0"
        }
      ]
    }
  }
}
```

### Api\_SaveAdvCompany

Создание/изменение данных рекламной кампании.

Параметры:

- Company - Object: модель рекламной кампании.

### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_SaveAdvCompany",
  "params": {
    "Company":{
      "adc_id": "0",
      "profile_id": "6",
      "title": "test_save_company",
      "user_id": "1",
      "from_date": "2018-09-18",
      "to_date": "2018-10-30",
      "from_time": "06:16:00",
      "to_time": "23:55:00",
      "redirect": "google.com",
      "days_of_week": "1,2,3,4,5,6,7",
      "params": "[]",
      "enabled": "0"
    }
  },
  "id": 1081
}
```

### Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "adc": {
        "adc_id": "19",
        "profile_id": "6",
        "title": "test_save_company",
        "user_id": "1",
        "from_date": "2018-09-18",
        "to_date": "2018-10-30",
        "from_time": "06:16:00",
        "to_time": "23:55:00",
        "total": null,
        "released": null,
        "redirect": "google.com",
        "pack": "0",
        "days_of_week": "1,2,3,4,5,6,7",
        "status": "1",
        "params": "[]",
        "need_to_update": "1",

```

```
        "processed": null,
        "prepared_url": "google.com?",
        "deleted": "0",
        "enabled": "0"
    }
}
}
```

## Api\_DeleteAdvCompany

Удаление рекламной кампании.

Параметры:

- Company - Object: модель рекламной кампании.

### Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_DeleteAdvCompany",
  "params": {
    "Company": {
      "adc_id": "19",
      "profile_id": "6",
      "title": "test_save_company",
      "user_id": "1",
      "from_date": "2018-09-18",
      "to_date": "2018-10-30",
      "from_time": "06:16:00",
      "to_time": "23:55:00",
      "total": null,
      "released": null,
      "redirect": "google.com",
      "pack": "0",
      "days_of_week": "1,2,3,4,5,6,7",
      "status": "1",
      "params": "[]",
      "need_to_update": "1",
      "processed": null,
      "prepared_url": "google.com?",
      "deleted": "0",
      "enabled": "0"
    }
  },
  "id": 1081
}
```

## Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "adc_id": "19",
      "profile_id": "6",
      "title": "test_save_company",
      "user_id": "1",
      "from_date": "2018-09-18",
      "to_date": "2018-10-30",
      "from_time": "06:16:00",
      "to_time": "23:55:00",
      "total": null,
      "released": null,
      "redirect": "google.com",
      "pack": "0",
      "days_of_week": "1,2,3,4,5,6,7",
      "status": "1",
      "params": "[]",
      "need_to_update": "1",
      "processed": null,
      "prepared_url": "google.com?",
      "deleted": "0",
      "enabled": "0"
    }
  }
}
```

## Api\_EnableAdvCompany

Запуск рекламной кампании.

Параметры:

- Company - Object: модель рекламной кампании.

## Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_EnableAdvCompany",
  "params": {
    "Company": {
      "adc_id": "4",

```

```

        "profile_id": "6",
        "title": "fake_company",
        "user_id": "1",
        "from_date": "2018-08-20",
        "to_date": "2018-09-30",
        "from_time": "00:00:00",
        "to_time": "23:55:00",
        "total": "16",
        "released": "0",
        "redirect": "abracadabra.ru",
        "pack": "0",
        "days_of_week": "1,2,3,4,5,6,7",
        "status": "3",
        "params":
"[\{"data_name\":"routing\","data_type\":"enum\","default_value\":"321\
","\id\":1535918599902}]",
        "need_to_update": "1",
        "processed": "4,16,17",
        "prepared_url": "abracadabra.ru?",
        "deleted": "0",
        "enabled": "0"
    }
},
"id": 1081
}

```

#### Пример ответа

```

{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "adc_id": "4",
      "enabled": 1
    }
  }
}

```

#### Api\_DisableAdvCompany

Остановка рекламной кампании.

Параметры:

- Company - Object: модель рекламной кампании.

## Пример запроса

```
{
  "jsonrpc": "2.0",
  "method": "Api_DisableAdvCompany",
  "params": {
    "Company":{
      "adc_id": "4",
      "profile_id": "6",
      "title": "fake_company",
      "user_id": "1",
      "from_date": "2018-08-20",
      "to_date": "2018-09-30",
      "from_time": "00:00:00",
      "to_time": "23:55:00",
      "total": "16",
      "released": "0",
      "redirect": "abracadabra.ru",
      "pack": "0",
      "days_of_week": "1,2,3,4,5,6,7",
      "status": "3",
      "params":
        "[{\\"data_name\\":\\"routing\\",\\"data_type\\":\\"enum\\",\\"default_value\\":\\"321\\",\\"id\\":1535918599902}]",
      "need_to_update": "1",
      "processed": "4,16,17",
      "prepared_url": "abracadabra.ru?",
      "deleted": "0",
      "enabled": "1"
    }
  },
  "id": 1081
}
```

## Пример ответа

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "adc_id": "4",
      "enabled": 0
    }
  }
}
```

## Управление абонентами рекламных кампаний

- [Api\\_GetAdvProcessedSubscribers](#)
- [Api\\_SaveAdvProcessedSubscriber](#)
- [Api\\_SaveAdvProcessedSubscribers](#)
- [Api\\_DeleteAdvProcessedSubscriber](#)

### Api\_GetAdvProcessedSubscribers

Получения списка абонентов рекламной кампании.

Параметры:

- Params - Object: параметры поиска абонентов (фильтры)

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "Api_GetAdvProcessedSubscribers",
  "params": {
    "Params": {
      "adc_id": 18,
      "hardware_id": 1,
      "status": 1,
      "login": "username1"
    }
  },
  "id": 1081
}
```

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "page": 0,
      "size": 100,
      "count": 8,
      "rows": [
        {
          "adc_id": "18",
          "login": "username1",
          "hardware_id": "1",
          "status": "1",
          "notification_date": null,
          "url": "",

```

```

        "deleted": "0",
        "params": "{}"
    },
    {
        "adc_id": "18",
        "login": "username10",
        "hardware_id": "1",
        "status": "1",
        "notification_date": null,
        "url": "",
        "deleted": "0",
        "params": "{\"routing\":\"10\",\"param2\":\"7\"}"
    },
    ...
]
}
}
}
}

```

## Api\_SaveAdvProcessedSubscriber

Создание/обновления данных абонента рекламной кампании.

Параметры:

- NewSubscriber - Object: модель абонента рекламной кампании с новыми данными.
- OldSubscriber - Object: модель абонента рекламной кампании со старыми данными(может быть null).

Пример запроса:

```

{
  "jsonrpc": "2.0",
  "method": "Api_SaveAdvProcessedSubscriber",
  "params": {
    "NewSubscriber":{
      "adc_id":"18",
      "hardware_id":"1",
      "status":"1",
      "params":"{}",
      "notification_date":null,
      "login":"username1"
    },
    "OldSubscriber":{
      "adc_id":"18",
      "login":"username1",
      "hardware_id":"1",
      "status":"1",
      "params":"{\"routing\":\"1\",\"param2\":\"16\"}",
      "notification_date":null
    }
  }
}

```

```
    }
  },
  "id": 1081
}
```

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true
  }
}
```

## Api\_SaveAdvProcessedSubscribers

Создание/обновления данных абонентов рекламной кампании.

Параметры:

- Subscribers - Array: массив моделей абонентов рекламной кампании.

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "Api_SaveAdvProcessedSubscribers",
  "params": {
    "Subscribers": [
      {
        "adc_id": "18",
        "login": "username1",
        "hardware_id": "1",
        "params": "{\"routing\": \"1\", \"param2\": \"16\"}",
        "status": 1,
        "notification_date": null
      },
      {
        "id": 1538142108942,
        "adc_id": "18",
        "login": "username2",
        "hardware_id": "1",
        "params": "{\"routing\": \"2\", \"param2\": \"15\"}",
        "status": 1,
        "notification_date": null
      },
      ...
    ]
  },
}
```

```
"id": 1081
}
```

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true
  }
}
```

### **Api\_DeleteAdvProcessedSubscriber**

Удаление абонента рекламной кампании.

Параметры:

- Subscriber - Object: модель абонента рекламной кампании.

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "Api_DeleteAdvProcessedSubscriber",
  "params": {
    "Subscriber":{
      "adc_id": "18",
      "login": "username1",
      "hardware_id": "1",
      "status": "1",
      "notification_date": null,
      "url": "",
      "deleted": "0",
      "params": "{}"
    }
  },
  "id": 1081
}
```

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1081,
  "result": {
    "success": true,
    "data": {
      "adc_id": "18",

```

```
    "login": "username1",
    "hardware_id": "1",
    "status": "1",
    "notification_date": null,
    "url": "",
    "deleted": "0",
    "params": "{}"
  }
}
```