

Welcome! This is documentation dedicated to MediaCore v4 API (0.39). Here you can find the descriptions of API functions, their usage and get description of JSON protocol that uses MediaCore API.

Table of content

1. [Mediacore API usage](#)
 - [Introduction](#)
 - [Common rules](#)
2. [Functions description](#)
 - [a number list add](#)
 - [a number list del](#)
 - [a number list del by password](#)
 - [a number list get](#)
 - [a number list get by password](#)
 - [a number list set](#)
 - [a numbers add](#)
 - [a numbers get](#)
 - [accucore config get](#)
 - [active calls get](#)
 - [active calls points get](#)
 - [active sessions get](#)
 - [active smpp sessions get](#)
 - [actual session get](#)
 - [agg data today](#)
 - [allowed ip add](#)
 - [allowed ip check](#)
 - [allowed ip del](#)
 - [allowed ips get](#)
 - [asr acd get](#)
 - [asr acd get ac](#)
 - [audit tables names get](#)
 - [audites get](#)
 - [auto test add](#)
 - [auto test del](#)
 - [auto test email send](#)
 - [auto test get](#)
 - [auto test result get](#)
 - [available routes by providers get](#)
 - [average stat get](#)
 - [balance add](#)
 - [balance get](#)
 - [balance get on date](#)
 - [balance report](#)
 - [balance set](#)
 - [balancer relation set](#)
 - [balancer relations get](#)
 - [balancer rtp relation set](#)
 - [balancer rtp relations get](#)
 - [balancers get](#)
 - [balances get](#)
 - [bind smpp session](#)
 - [call chart diagram points get](#)

- [call chart diagram points get old](#)
- [call trace get](#)
- [cdr classes get](#)
- [cdr daily gen for accucore](#)
- [cdr daily gen get status](#)
- [cdr request add](#)
- [cdr request del](#)
- [cdr request ftp get](#)
- [cdr request ftp set](#)
- [cdr request get](#)
- [change dests min max](#)
- [check is password expired](#)
- [client add](#)
- [client get](#)
- [client note add](#)
- [client notes history get](#)
- [client point get](#)
- [client set](#)
- [client vat history get](#)
- [codecs get](#)
- [config get](#)
- [config set](#)
- [contexts get](#)
- [correct balance by paypal](#)
- [countries get](#)
- [countries regexp get](#)
- [country code add](#)
- [country code bulk add](#)
- [country code bulk delete](#)
- [country codes del](#)
- [country codes get](#)
- [country codes get grouped](#)
- [country codes get grouped equals](#)
- [cpm points get](#)
- [currency add](#)
- [currency course add](#)
- [currency course get](#)
- [currency del](#)
- [currency get](#)
- [currency history get](#)
- [currency requisite by owner get](#)
- [currency requisite by owner set](#)
- [currency set](#)
- [customized report get](#)
- [destinations by tp get](#)
- [dialpeer add](#)
- [dialpeer copy](#)
- [dialpeer groups get](#)
- [dialpeer set](#)
- [dialpeer tp set](#)
- [dialpeer tps op get](#)
- [dialpeers free group get](#)
- [dialpeers from groups del](#)

- [dialpeers_get](#)
- [dialpeers_tp_get](#)
- [dialpeers_tps_get](#)
- [email_list_get](#)
- [email_list_owners_get](#)
- [email_list_set](#)
- [email_owner_set](#)
- [email_send](#)
- [email_settings_get](#)
- [email_settings_set](#)
- [export_price_list](#)
- [export_price_list_jurisdictional](#)
- [export_price_list_sms](#)
- [fin_summary_get](#)
- [firewall_control_add](#)
- [firewall_control_del](#)
- [firewall_control_get](#)
- [get_half_hour_stat_ac](#)
- [group_add](#)
- [group_copy](#)
- [group_del](#)
- [group_dialpeers_get](#)
- [group_dp_tp_dnls_get](#)
- [group_set](#)
- [groups_free_dialpeer_get](#)
- [groups_free_op_get](#)
- [groups_free_tp_get](#)
- [groups_get](#)
- [groups_to_dialpeers_add](#)
- [guardian_blocked_dest_get](#)
- [guardian_blocked_dest_set](#)
- [guardian_conditions_get](#)
- [guardian_event_add](#)
- [guardian_event_copy](#)
- [guardian_event_del](#)
- [guardian_event_get](#)
- [guardian_event_set](#)
- [guardian_event_trace](#)
- [guardian_global_settings_get](#)
- [guardian_global_settings_set](#)
- [guardian_report_send](#)
- [guardian_template_add](#)
- [guardian_template_del](#)
- [guardian_template_preview](#)
- [guardian_template_set](#)
- [guardian_templates_get](#)
- [guardian_tt_add](#)
- [guardian_tt_cancel](#)
- [guardian_tt_get](#)
- [guardian_tt_preview](#)
- [guardian_tt_send](#)
- [guardian_tt_set](#)
- [guardian_warning_add](#)

- [guardian warning del](#)
- [guardian warning set](#)
- [guardian warnings get](#)
- [guardian warnings replace](#)
- [guardian warnings report get](#)
- [guardian wizard countries get](#)
- [guardian wizard events add](#)
- [guardian wizard status get](#)
- [guardian wizard status set](#)
- [gw auth add](#)
- [gw auth del](#)
- [gw auth set](#)
- [gw auths get](#)
- [host add](#)
- [host del](#)
- [host dumper relations get](#)
- [host from balancer del](#)
- [host set](#)
- [host to balancer add](#)
- [host types get](#)
- [hosts get](#)
- [incoming call pathes get](#)
- [incoming calls get](#)
- [incoming calls get old](#)
- [incoming invoice add](#)
- [incoming invoice get](#)
- [incoming invoice set](#)
- [incoming sms get](#)
- [information messages get](#)
- [internal sms disconnect cause get](#)
- [invoice amount claimed get](#)
- [invoice data cancel](#)
- [invoice data get](#)
- [invoice data set](#)
- [invoice default email get](#)
- [invoice del](#)
- [invoice email send](#)
- [invoice periods get](#)
- [invoice report](#)
- [invoice settings get](#)
- [invoice settings set](#)
- [invoice statuses get](#)
- [invoices gen](#)
- [ips get](#)
- [jurisdictional incoming call pathes get](#)
- [jurisdictional outgoing call pathes get](#)
- [jurisdictional price status get](#)
- [jurisdictional price upload](#)
- [jurisdictional prices list get](#)
- [jurisdictional tariff add](#)
- [jurisdictional tariff set](#)
- [jurisdictional tariffs op get](#)
- [jurisdictional tariffs tp get](#)

- [kill smpp session](#)
- [lib disconnect cause get](#)
- [lib local disconnect cause get](#)
- [lib local disconnect class get](#)
- [login](#)
- [logout](#)
- [maintenance command](#)
- [manager get](#)
- [mcc mnc add](#)
- [mcc mnc bulk add](#)
- [mcc mnc del](#)
- [mcc mnc get](#)
- [mcc mnc get grouped](#)
- [monitoring options get](#)
- [month sms count get](#)
- [notification email add](#)
- [numbers by point get](#)
- [op add or update](#)
- [op add simple](#)
- [op call path analysis get](#)
- [op dest capacity add](#)
- [op dest capacity del](#)
- [op dest capacity get](#)
- [op dest capacity set](#)
- [op group set](#)
- [op ip add](#)
- [op ip del](#)
- [op ip set](#)
- [op ips get](#)
- [ops free group get](#)
- [ops free tp get](#)
- [ops from groups del](#)
- [ops get](#)
- [ops groups get](#)
- [ops to groups add](#)
- [ops tp get](#)
- [ops tp simple get](#)
- [outgoing call pathes get](#)
- [outgoing calls get](#)
- [outgoing invoice add](#)
- [outgoing invoice get](#)
- [outgoing invoice set](#)
- [outgoing sms get](#)
- [owner add](#)
- [owner del](#)
- [owner info by invoice get](#)
- [owner info get](#)
- [owner info set](#)
- [owner set](#)
- [owners get](#)
- [payment add](#)
- [payment calendar get](#)
- [payment del](#)

- [payment_get](#)
- [payment history_get](#)
- [payment_set](#)
- [payment type_get](#)
- [payments_get_ac](#)
- [payments total_get_ac](#)
- [pc_notification_send](#)
- [pc_weekly_get](#)
- [pc_weekly_set](#)
- [permission_set](#)
- [permissions_enabled_get](#)
- [permissions_get](#)
- [points_enabled_get](#)
- [price_analyze request_add](#)
- [price_analyze request_del](#)
- [price_analyze requests_get](#)
- [price_analyze top_routes_get](#)
- [price_status_get](#)
- [price_upload](#)
- [prices_creation_get](#)
- [prices_format_get](#)
- [prices_list_get](#)
- [problem_calls_get_out](#)
- [problem_types_get](#)
- [process_new_price](#)
- [radiuses_by_context_get](#)
- [regexp_test](#)
- [regions_by_tp_get](#)
- [rel_op_tp_add](#)
- [rel_op_tp_del](#)
- [role_add](#)
- [role_del](#)
- [role_set](#)
- [roles_get](#)
- [routing_types_get](#)
- [services_get](#)
- [sip_authorization_add](#)
- [sip_authorization_del](#)
- [sip_authorization_get](#)
- [sip_authorization_set](#)
- [sms_activity_points_get](#)
- [sms_change_dests_min_max](#)
- [sms_content_get](#)
- [sms_content_replacement_rule_add](#)
- [sms_content_replacement_rule_del](#)
- [sms_content_replacement_rule_get](#)
- [sms_content_replacement_rules_get_all](#)
- [sms_dialpeer_add](#)
- [sms_dialpeer_groups_get](#)
- [sms_dialpeer_set](#)
- [sms_dialpeer_tp_set](#)
- [sms_dialpeer_tps_op_get](#)
- [sms_dialpeers_free_group_get](#)

- [sms dialpeers from groups del](#)
- [sms dialpeers get](#)
- [sms dialpeers tp get](#)
- [sms dialpeers tps get](#)
- [sms end upload](#)
- [sms fin summary get](#)
- [sms group add](#)
- [sms group copy](#)
- [sms group del](#)
- [sms group dialpeers get](#)
- [sms group dp tp dnis get](#)
- [sms group set](#)
- [sms groups free dialpeer get](#)
- [sms groups free op get](#)
- [sms groups free tp get](#)
- [sms groups get](#)
- [sms groups to dialpeers add](#)
- [sms hlr report get](#)
- [sms hosts get](#)
- [sms numbers by point get](#)
- [sms op add or update](#)
- [sms op authorization add](#)
- [sms op authorization del](#)
- [sms op authorization get](#)
- [sms op authorization set](#)
- [sms op group set](#)
- [sms op settings copy](#)
- [sms ops free group get](#)
- [sms ops free tp get](#)
- [sms ops from groups del](#)
- [sms ops get](#)
- [sms ops groups get](#)
- [sms ops to groups add](#)
- [sms ops tp get](#)
- [sms ops tp simple get](#)
- [sms points enabled get](#)
- [sms price upload](#)
- [sms prices list get](#)
- [sms process new price](#)
- [sms radiuses by context get](#)
- [sms rel op tp add](#)
- [sms rel op tp del](#)
- [sms routing table get](#)
- [sms routing table get old](#)
- [sms start page get clients close to credit limit](#)
- [sms start page get closed companies](#)
- [sms tariff add](#)
- [sms tariff set](#)
- [sms tariffs get](#)
- [sms tariffs op get](#)
- [sms tariffs tp get](#)
- [sms top countries get](#)
- [sms tp add or update](#)

- [sms tp authorization add](#)
- [sms tp authorization del](#)
- [sms tp authorization get](#)
- [sms tp authorization set](#)
- [sms tp free dialpeers get](#)
- [sms tp settings copy](#)
- [sms tps free dialpeer get](#)
- [sms tps free group get](#)
- [sms tps free op get](#)
- [sms tps from dialpeers del](#)
- [sms tps from groups del](#)
- [sms tps get](#)
- [sms tps groups get](#)
- [sms tps op get](#)
- [sms tps to dialpeers add](#)
- [sms tps to groups add](#)
- [sms trace get](#)
- [spam calls get](#)
- [start page call charts get](#)
- [start page credit limit report get](#)
- [start page credit limit report sms get](#)
- [start page get clients close to credit limit](#)
- [start page get closed companies](#)
- [start page sms activity get](#)
- [start page top countries get](#)
- [start page top countries sms get](#)
- [start page top offers get](#)
- [start page top offers sms get](#)
- [start page top targets get](#)
- [start page top targets sms get](#)
- [stat get](#)
- [sum active calls points get](#)
- [sum cpm points get](#)
- [sum sms activity points get](#)
- [swap deal add](#)
- [swap deal del](#)
- [swap deal fin summary get](#)
- [swap deal get](#)
- [swap deal set](#)
- [system info get](#)
- [target by prices get](#)
- [tariff add](#)
- [tariff group add](#)
- [tariff group del](#)
- [tariff group get](#)
- [tariff set](#)
- [tariff to group add](#)
- [tariffs by group del](#)
- [tariffs by group get](#)
- [tariffs by group set](#)
- [tariffs get](#)
- [tariffs op get](#)
- [tariffs tp get](#)

- [tariffs without group get](#)
- [tcp_dump_del](#)
- [tcp_dump_file link get](#)
- [tcp_dump_get](#)
- [tcp_dump_set](#)
- [tcp_dumper files get](#)
- [time zones get](#)
- [top countries get](#)
- [top duration regions get](#)
- [top profit regions get](#)
- [total balance get](#)
- [tp_add_or_update](#)
- [tp_add_simple](#)
- [tp_auth_add](#)
- [tp_auth_del](#)
- [tp_auth_get](#)
- [tp_auth_set](#)
- [tp_dest_capacity_add](#)
- [tp_dest_capacity_del](#)
- [tp_dest_capacity_get](#)
- [tp_dest_capacity_set](#)
- [tp_free_dialpeers get](#)
- [tp_ip_add](#)
- [tp_ip_del](#)
- [tp_ip_set](#)
- [tp_ips_get](#)
- [tps_by_destination get](#)
- [tps_free_dialpeer get](#)
- [tps_free_group get](#)
- [tps_free_op get](#)
- [tps_from_dialpeers del](#)
- [tps_from_groups del](#)
- [tps_get](#)
- [tps_groups get](#)
- [tps_op get](#)
- [tps_to_dialpeers add](#)
- [tps_to_groups add](#)
- [traffic_classes get](#)
- [traffic_share get](#)
- [traffic_stat get](#)
- [turn_of_changes get](#)
- [user_add](#)
- [user_companies get](#)
- [user_set](#)
- [users_get](#)
- [util_should_stop](#)
- [util_stop_job](#)
- [vat_rate_add](#)
- [vat_rates_get](#)
- [xdr_request add](#)

3. [Appendix](#)

- [Regular expressions abbreviations](#)
- [Validator abbreviations](#)

1. Mediacore API usage.

1.1 Introduction

Mediacore API uses JSON protocol for data exchanging with user applications.

JSON (JavaScript Object Notation) - is a simple format of data exchange, convenient for reading and writing as for human as well as for computer. It based on subset of JavaScript programming language, defined in standard ECMA-262 3rd Edition - December 1999. JSON is a text format completely independent from realization, but it uses agreements familiar to C-like languages programmers. These properties make JSON perfect language for data exchange (from Wikipedia).

More about JSON read here <http://json.org/>.

Protocol's main task is provision reliable operation of client applications for managing MediaCore class 4 softswitch. Data exchange occurs through http POST queries which based on JSON protocol.

Request (hereinafter) - request from client's application to the softswitch's API.

Response (hereinafter) - reply of server's API on the client's query.

1.2 Common rules of the MediaCore JSON protocol usage.

1) All queries, besides one which doesn't require authorization (function "login"), must contain "session_id" identifier - "session_id". Session identifier ("session_id") or in other words "session Hash key VALUE" can be obtained at "login" and represents MD5 Hash of the string that consists of current_date user_login and user_password.

Query format with "session_id":

```
Request: {"session_id":"value", "data":  
{"key1":"value1", "key2":"value2", "key3":"value3", ...}}
```

- "session_id" - is actual session identifier. Identifier is actual during 2 hours after "login" or after last query with given "session_id".
- "data" - associative array which contains query parameters

Query format without "session_id":

```
Request: { "key1":"value1", "key2":"value2", "key3":"value3", ....}
```

Let's look on the example of "login" Request:

```
Request: {"login":"login_VALUE", "password":"password_VALUE", "ip":"IP_VALUE",  
"description":"description_VALUE"}
```

- "login" - mandatory option, user login
- "password" - mandatory option, user password

- "ip" - mandatory option, user IP address
- "description" - some description, optional parameter

From example above you can see that request is a simple dictionary in which Request keys separates by comma, key names are written in double quotes, some keys are optional (filters as usually).

2) Query VALUES can be following data types:

- Number - integer or float(integer and fractional parts are separates with point) number. A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.
- String - is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.
- Array - is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma). Array example: [1,2,3]
- Object - is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).
- Value - can be a *string* in double quotes, or a *number*, or *true* or *false* or *null*, or an *object* or an *array*. These structures can be nested.

3) In case of any ERROR server response is following:

Response: **{"status":!=200(is not equal 200), "message": "error message", "details": "error description VALUE"}**

There are 3 types of unsuccessful statuses:

- 3XX - errors raised by API functions
- 4XX - request error, errors caused by incorrect protocol usage
- 5XX - servers error, errors caused by server side faults

Successful server answer has "status" == 200 (successful answer - status is equal 200).

4) In each selection query (functions with "_get" suffixes) is available filter by fields: {"fields":[fields ARRAY]}. In case when this filter is present - server Response contains only fields specified in filter in chosen order.

If in selection query fields filter is absent - API returns maximal possible data set. All possible fields listed in the "default_fields" section of function description. List of requested fields (parameter {"fields":[fields ARRAY]}) is always present in server's Response on selection query.

5) Date in the client's Requests must have following format:

yyyy-MM-dd'T'HH:mm:ss

This format is a simplification of format described by ECMAScript (which based upon a simplification of the ISO 8601 Extended Format).

Where the fields are as follows:

YYYY - is the decimal digits of the year 0000 to 9999 in the Gregorian calendar.
 "-" - (hyphen) appears literally twice in the string.

MM - is the month of the year from 01 (January) to 12 (December).
DD - is the day of the month from 01 to 31.
T - appears literally in the string, to indicate the beginning of the time element .
HH - is the number of complete hours that have passed since midnight as two decimal digits from 00 to 24.
":" - (colon) appears literally twice in the string.
mm - is the number of complete minutes since the start of the hour as two decimal digits from 00 to 59.
ss - is the number of complete seconds since the start of the minute as two decimal digits from 00 to 59.

6) In some selection queries option "sort_by" is available. This option allows to sort output by single field or a few fields (rows with the same first field will be sorted by second field, then by third and so on) listed in the "default_fields". "sort_by" is an array that consists of one-dimensional objects (with single pair "key" & "value"). Sorting may be ascending ("asc") or descending ("desc")

Examples:

[{"disconnect_cause": "desc"}] - will sort by field "disconnect_cause" descending.

[{"priority": "asc"}, {"dp_name": "desc"}] - will sort at first by "priority" ascending and then fields with the same one will sorted by "dp_name" descending.

2. Functions description

2.1 a_number_list_add

Docstring:

Creates new a-number list.

Example:

- Client Request

```
{"session_id": "88fa42f23bea3330780e802a44f53864",  
  "data": {"name": "test_list",  
           "destination": ["1234"],  
           "from_cdr": true}  
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "id": 80,  
  "time_elapsed": 0.881779}
```

Request structure and fields description:

```
{'data': {'destination': ['String'],  
          'from_cdr': 'Bool',  
          'modification_password': 'String',  
          'name': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
from_cdr	False	True	Bool	If True - numbers from CDR will be aggregated X times(the X value is defined in configuration file conf.ini) every day for the previous 24 hours; otherwise - won't.
destination	[]	True	List([String(allow_	Destinations.

			blank: True)], min_length: 0)	
name	N/A	False	String(allow_blank : False, min_length: 1, max_length: 100)	A-number replacement list name.
modification_password	N/A	True	String(allow_blank : False)	password for data modification (if set in API configuration).

2.2 a_number_list_del

Docstring:

Deletes existing a-number list.

Example:

- Client Request

```
{"session_id": "88fa42f23bea3330780e802a44f53864",
  "data": {"id": 69}
}
```

- Server Response

```
{"status": 200,
  "message": "OK",
  "time_elapsed": 1.642563}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'modification_password': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gte: 1)	DB record id.

modification_password	N/A	True	String(allow_blank : False)	password for data modification (if set in API configuration).
-----------------------	-----	------	-----------------------------	---

2.3 a_number_list_del_by_password

Docstring:

Deletes existing a-number list by link from web (MCD-3624).
Function works only if api->anumbers_modification_password
API configuration parameter is set

Example:

- Client Request

```
{
  "data": {"id": 69,
           "modification_password": "*****"}
}
```

- Server Response

```
{"status": 200,
 "message": "OK",
 "time_elapsed": 1.642563}
```

Request structure and fields description:

```
{'id': 'Integer', 'modification_password': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
modification_password	N/A	False	String(allow_blank : False)	password for data modification (if set in API configuration).
id	N/A	False	Integer(gte: 1)	DB record id.

2.4 a_number_list_get

Docstring:

Returns existing a-number list.

Example:

- Client Request

```
{"session_id": "88fa42f23bea3330780e802a44f53864",  
  "data": {"sort_by": [{"name": "asc"}],  
           "fields": ["id", "name", "destination", "from_cdr"]  
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "fields": ["id", "name", "destination", "from_cdr"],  
  "data": [  
    [34, "29", ["012345678998765432101234"], false],  
    [1, "anton_test_1_", [], false],  
    [10, "A-Number TP", ["9198"], true]  
  ],  
  "time_elapsed": 0.057094}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('from_cdr', 'destination', 'changed', 'id', 'name')"],  
          'id': 'Integer',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('from_cdr', 'destination', 'changed', 'id', 'name')": "Enum('asc', 'desc', '')"}]},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	True	Integer(gte: 1)	Filter by id.
name	N/A	True	String(allow_blank: True)	Filter by list name.

offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['from_cdr', 'destination', 'changed', 'id', 'name']	True	List([Enum('from_cdr', 'destination', 'changed', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('from_cdr', 'destination', 'changed', 'id', 'name'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result sorts
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
from_cdr	Bool
destination	List(String)
changed	DateTime
id	Int
name	String

2.5 a_number_list_get_by_password

Docstring:

Returns existing a-number list.

Example:

- Client Request

```
{
  "data": {"sort_by": [{"name": "asc"}],
```

```

        "modification_password": "*****",
        "fields":["id","name","destination","from_cdr"]
    }
}

- Server Response

{"status": 200,
 "message": "OK",
 "fields": ["id", "name", "destination", "from_cdr"],
 "data": [
     [34, "29", ["012345678998765432101234"], false],
     [1, "anton_test_1_", [], false],
     [10, "A-Number TP", ["9198"], true]
 ],
 "time_elapsed": 0.057094}

```

Request structure and fields description:

```

{'fields': ["Enum('from_cdr', 'destination', 'changed', 'id', 'name')"],
 'id': 'Integer',
 'limit': 'Integer',
 'modification_password': 'String',
 'name': 'String',
 'offset': 'Integer',
 'sort_by': [{"Enum('from_cdr', 'destination', 'changed', 'id', 'name')": "Enum('asc',
'desc', '')"}]}

```

data validation description:

Name	Default value	Optional	Type	Description
modification_password	N/A	False	String(allow_blank: False)	password for data modification (if set in API configuration).
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
name	N/A	True	String(allow_blank: True)	Filter by list name.
fields	['from_cdr', 'destination', 'changed', 'id', 'name']	True	List([Enum('from_cdr', 'destination', 'changed', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

limit	0	True	Integer(gte: 0)	says how many rows will be in response
id	N/A	True	Integer(gte: 1)	Filter by id.
sort_by	[]	True	List([{Enum('from_cdr', 'destination', 'changed', 'id', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result sorts

Default fields:

Name	Type & Comments
from_cdr	Bool
destination	List(String)
changed	DateTime
id	Int
name	String

2.6 a_number_list_set

Docstring:

Changes existing a-number list.

Example:

- Client Request

```
{
  "session_id": "88fa42f23bea3330780e802a44f53864",
  "data": {
    "name": "testing",
    "destination": ["99", "99"],
    "from_cdr": false,
    "id": 51
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
}
```

```
"time_elapsed": 0.189702}
```

Request structure and fields description:

```
{'data': {'destination': ['String'],  
          'from_cdr': 'Bool',  
          'id': 'Integer',  
          'name': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
from_cdr	N/A	True	Bool	If True - numbers from CDR will be aggregated X times(the X value is defined in configuration file conf.ini) every day for the previous 24 hours; otherwise - won't.
destination	N/A	True	List([String(allow_blank: False)], min_length: 0)	Destinations.
id	N/A	False	Integer(gte: 1)	DB record id.
name	N/A	True	String(allow_blank: False, min_length: 1, max_length: 100)	A-number replacement list name.

2.7 a_numbers_add

Docstring:

Function inserts a_numbers into given list.

Example:

- Client Request

```
{"session_id": "88fa42f23bea3330780e802a44f53864",  
 "data": {"list_id": 8,  
          "numbers": ["79888345460", "79670799737", "79236524304"]}}
```

```
}  
}
```

- Server Response

```
{"status": 200,  
 "message": "OK",  
 "time_elapsed": 0.057094}
```

Request structure and fields description:

```
{'data': {'append': 'Bool',  
         'list_id': 'Integer',  
         'modification_password': 'String',  
         'numbers': ['String']},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
numbers	N/A	False	List([String(allow_blank: False)], min_length: 0)	List of a_numbers.
list_id	N/A	False	Integer(gte: 1)	List ID where a_numbers will be inserted to.
append	False	True	Bool	Attribute for appending numbers. If True a_numbers will be added into list.
modification_password	N/A	True	String(allow_blank: False)	password for data modification (if set in API configuration).

2.8 a_numbers_get

Docstring:

Function returns a_numbers from list by ID.

Example:

- Client Request

```
{
  "session_id": "88fa42f23bea3330780e802a44f53864",
  "data": {
    "list_id": 8
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": [
    "79888345460", "79670799737", "79236524304", "79081778523",
    "79067708765", "79210327373", "79608638987", "79644264403",
    "79067708765", "79210327373", "79608638987", "79644264403"
  ],
  "time_elapsed": 0.057094
}
```

Request structure and fields description:

```
{
  'data': {
    'list_id': 'Integer',
    'modification_password': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
modification_password	N/A	True	String(allow_blank: False)	password for data modification (if set in API configuration).
list_id	N/A	False	Integer(gte: 1)	List ID where a_numbers will be inserted to.

2.9 accucore_config_get

Docstring:

Function for Accucore use ONLY!

Function returns specified config keys.

- Client Request

```
{
  "session_id": "832a3d94029e3c1f64f5bc569e1b42c4",
  "data": {
    "fields": [
      "ips_edit_flag"
    ]
  }
}
```

```
}
```

- Server Response

```
{  
  "status": 200,  
  "message": "OK",  
  "data": {  
    "ips_edit_flag": true  
  },  
  "time_elapsed": 6.025933  
}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.10 active_calls_get

Docstring:

Returns information about calls which are passing through the given RTP host at the moment and number of connected calls(calls with duration greater than 0:00).

Filtering by client and points ids:

- if company is selected and points aren't selected - shows calls which are passing via all client's points
- if company and some it's point(s) are selected - shows calls which are going through chosen point(s)
- if company is not selected but selected OP and TP - shows calls which are passing between chosen OP and TP
- if company is not selected but selected OP or TP - shows calls which are passing via chosen point

Function filters result by user's permissions (companies locked by user).

Example:

- Client Request:

```
{"session_id": "75effd728d6ac74d4fdd626ed5bd2f6d",
  "data": {"sort_by": {"number": ""},
    "duration_min": "60",
    "fields": ["margin", "client_name", "op_name", "op_ip", "prefix_in",
      "tp_name", "tp_ip", "prefix_out", "ani", "number",
      "duration_short", "op_proto_tp_proto", "op_price",
      "tp_price", "op_codec", "tp_codec", "transcoding", "nas",
      "dialpeer_name"]}
}
```

- Server Response:

```
{"status": 200,
  "fields": ["margin", "client_name", "op_name", "op_ip", "prefix_in",
    "tp_name", "tp_ip", "prefix_out", "ani", "number",
    "duration_short", "op_proto_tp_proto", "op_price",
    "tp_price", "op_codec", "tp_codec", "transcoding", "nas",
    "dialpeer_name"],
  "message": "OK",
  "data": [[0.01, "Test_calls", "OP_test", "194.28.166.1:5060",
    "200", "Termination_Test", "193.104.107.10:5060", "400",
    "+1959", "923005934146", 214, "sip/sip", "0.04000",
    "0.03000", "g729,g723,alaw,mulaw", "g729", false,
    "193.161.85.79", "Test"],
    [0.01, "Test_calls", "OP_test",
    "193.104.107.14:5060", "200", "Termination_Test",
    "193.104.107.10:5060", "400", "966594164589",
    "923139505010", 95, "sip/sip", "0.04000", "0.03000",
    "alaw,mulaw,g729,g723", "g723", false, "193.161.85.79",
    "Test"]
  ],
  "connected_calls": 2,
  "time_elapsed": 0.169904}
```

Request structure and fields description:

```
{'data': {'clients_ids': ['Integer'],
  'codes_exc': ['String'],
  'codes_inc': ['String'],
  'context_id': 'Integer',
  'currency_id': 'Integer',
  'detect_loop_calls': 'Bool',
  'dialpeer_id': 'Integer',
  'duration_max': 'Integer',
  'duration_min': 'Integer',
  'fields': ["Enum('op_ip', 'tp_currency_id', 'op_balance_id', 'tp_client_id',
    'prefix_out', 'ani', 'number', 'duration_short', 'op_id', 'dialpeer_name', 'op_codec',
    'op_tarif_id', 'dialpeer_name_context', 'tp_client_name', 'transcoding', 'destination_n
ame', 'tp_ip', 'op_price', 'op_proto_tp_proto', 'tp_tarif_id', 'prefix_in', 'tp_price',
    'client_name', 'op_client_id', 'op_name', 'op_currency_id', 'tp_name', 'nas', 'tp_codec
', 'tp_id', 'margin', 'tp_balance_id', 'is_loop_call')"],
  'hosts': ['String'],
  'limit': 'Integer',
  'offset': 'Integer',
  'op_ids': ['Integer'],
  'session_id': 'String',
```



```

'sort_by': {"Enum('op_ip', 'tp_currency_id', 'op_balance_id', 'tp_client_id',
'prefix_out', 'ani', 'number', 'duration_short', 'op_id', 'dialpeer_name', 'op_codec',
'op_tarif_id', 'dialpeer_name_context', 'tp_client_name', 'transcoding', 'destination_name',
'tp_ip', 'op_price', 'op_proto_tp_proto', 'tp_tarif_id', 'prefix_in', 'tp_price',
'client_name', 'op_client_id', 'op_name', 'op_currency_id', 'tp_name', 'nas', 'tp_codec',
'tp_id', 'margin', 'tp_balance_id', 'is_loop_call')": "Enum('asc', 'desc', '')"},
'tp_ids': ['Integer']},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['op_ip', 'tp_currency_id', 'op_balance_id', 'tp_client_id', 'prefix_out', 'ani', 'number', 'duration_short', 'op_id', 'dialpeer_name', 'op_codec', 'op_tarif_id', 'dialpeer_name_context', 'tp_client_name', 'transcoding', 'destination_name', 'tp_ip', 'op_price', 'op_proto_tp_proto', 'tp_tarif_id', 'prefix_in', 'tp_price', 'client_name', 'op_client_id', 'op_name', 'op_currency_id', 'tp_name', 'nas', 'tp_codec', 'tp_id', 'margin', 'tp_balance_id', 'is_loop_call']	True	List([Enum('op_ip', 'tp_currency_id', 'op_balance_id', 'tp_client_id', 'prefix_out', 'ani', 'number', 'duration_short', 'op_id', 'dialpeer_name', 'op_codec', 'op_tarif_id', 'dialpeer_name_context', 'tp_client_name', 'transcoding', 'destination_name', 'tp_ip', 'op_price', 'op_proto_tp_proto', 'tp_tarif_id', 'prefix_in', 'tp_price', 'client_name', 'op_client_id', 'op_name', 'op_currency_id', 'tp_name', 'nas', 'tp_codec', 'tp_id', 'margin', 'tp_balance_id', 'is_loop_call')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
duration_max	N/A	True	Integer(gte: 0)	filter by call duration. Returns calls with duration shorter than given
context_id	N/A	True	Integer(gt: 0)	filter by context
sort_by	{}	True	{Enum('op_ip', 'tp_currency_id', 'op_balance_id', 'tp_client_id', 'prefix_out', 'ani', 'number', 'duration_short', 'op_id', 'dialpeer_name', 'op_codec', 'op_tarif_id', 'dialpeer_name_context',	field name on which result will be sorted

			'tp_client_name', 'transcoding', 'destination_name', 'tp_ip', 'op_price', 'op_proto_tp_proto', 'tp_tarif_id', 'prefix_in', 'tp_price', 'client_name', 'op_client_id', 'op_name', 'op_currency_id', 'tp_name', 'nas', 'tp_codec', 'tp_id', 'margin', 'tp_balance_id', 'is_loop_call'): Enum('asc', 'desc', ")}	
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
duration_min	N/A	True	Integer(gte: 0)	filter by call duration. Returns calls with duration longer than given
dialpeer_id	N/A	True	Integer(gt: 0)	filter by dialpeer id
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
codes_inc	N/A	True	List([String(allow_blank: False, regex: "^\\s*[0-9#]**\$"), min_length: 0)	filter by destination codes. Defines call direction or call directions to be displayed in response
tp_ids	N/A	True	List([Integer(gt: 0)], min_length: 1)	Filter by listed TPs
currency_id	N/A	True	Integer(gt: 0)	currency to which op/tp prices will be converted
session_id	N/A	True	String(allow_blank: False)	not in use
op_ids	N/A	True	List([Integer(gt: 0)], min_length: 1)	Filter by listed

			1)	OPs
hosts	N/A	True	List([String(allow_blank: False)], min_length: 1)	filter by RTP host IP, if not specified returns active calls from all hosts
codes_exc	N/A	True	List([String(allow_blank: False, regex: "^s*[0-9#]**\$")], min_length: 0)	filter by destination codes. Defines destination or group of destinations to be excluded
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 1)	Filter report by points that belong to listed clients
detect_loop_calls	False	True	Bool	set flag 'is_loop_call' = True if there are a few (2 or more) calls to the same destination number with difference in duration 2 seconds at most

Default fields:

Name	Type & Comments
op_ip	String
tp_currency_id	Int
op_balance_id	Int
tp_client_id	Int
prefix_out	String - TP prefix

ani	String - caller number
number	String - called number without prefix
duration_short	Int - call duration in seconds
op_id	Int
dialpeer_name	String
op_codec	String
op_tarif_id	Int
dialpeer_name_context	String
tp_client_name	String
transcoding	Boolean
destination_name	String - Destination name from destination codes
tp_ip	String
op_price	String
op_proto_tp_proto	String - OP and TP protocols (h323 or sip) separated by "/"
tp_tarif_id	Int
prefix_in	String - OP prefix
tp_price	String
client_name	String
op_client_id	Int
op_name	String
op_currency_id	Int

tp_name	String
nas	String - RTP IP
tp_codec	String
tp_id	Int
margin	numeric(16.5) - difference between OP and TP prices in the default currency
tp_balance_id	Int
is_loop_call	Bool - True suppose that this is a loop call. In case 'detect_loop_calls'=False then will be False too.

2.11 active_calls_points_get

Docstring:

Function similar to sum_active_calls_points_get but returns data by hosts. For period "day" it doesn't return ASR stat.

"data" and "connected_calls" have following format:

```
[[host_ip, [list of 288 values]], [host_ip, [list of 288 values]] ... ]
```

Example:

- Client Request

```
{
  "session_id": "d41f1261a9e96522a143564063347a4b",
  "data": {"period": "month"}
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": [
    ["193.161.85.79", [2.1003333333, 0.4002222222, 0.0, 0.0, 1.8328888
889, 2.0665555556,
    ...
    0.4664444444, 0.0, 1.433, 2.0666666667, 2.2332222222, 2.0, 2.26655
5556, 2.9]
  ],
  "connected_calls" : [
    ["193.161.85.79", [3.4321111111, 5.7995555556, 4.1332222222, 5.5613
```

```

333333, 0.0,
                ...
                2.4344444444, 1.8663333333, 4.6333333333, 5.0994444444, 5.2668888888
9]
    ],
    "start_time": 1459345800
}

```

Request structure and fields description:

```

{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')"},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.

2.12 active_sessions_get

Docstring:

Function returns data about current active sessions.

Example:

- Client Request

```

{
    "session_id": "cacd0254320957af40463e9f15d1b910",
    "data": {
        "sort_by": [{"time_in_system": "desc"}],
        "fields": ["login", "role_name", "ip", "country", "time_in_system",
"session_id"],
        "limit": 3
    }
}

```

- Server response

```

{
    "status": 200,
    "fields": ["login", "role_name", "ip", "country", "time_in_system", "se
ssion_id"],
    "message": "OK",
}

```

```

        "data": [
            ["ert", "Owner", "188.230.128.172", "Ukraine", "0:08:14", "fc9cdd6c3e9da090d2108427162efe3d"],
            ["imax", "Owner", "193.104.120.116", "United Kingdom", "0:12:10", "cacd0254320957af40463e9f15d1b910"],
            ["imax", "Owner", "89.209.77.18", "Ukraine", "0:57:44", "15a77f3f134c3373d9388a5e78300f8e"]
        ],
        "time_elapsed": 0.041938
    }

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('time_in_system', 'user_id', 'ip', 'session_id', 'role_name', 'country', 'login')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('time_in_system', 'user_id', 'ip', 'session_id', 'role_name', 'country', 'login')": "Enum('asc', 'desc', '')"}]},
          'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
limit	0	True	Integer(gte: 0)	rows quantity per one request
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
sort_by	[]	True	List([{"Enum('time_in_system', 'user_id', 'ip', 'session_id', 'role_name', 'country', 'login')": Enum('asc', 'desc', '')}], min_length: 0)	sorting by field(s) and ordering
fields	['time_in_system', 'user_id', 'ip', 'session_id', 'role_name', 'country', 'login']	True	List([Enum('time_in_system', 'user_id', 'ip', 'session_id', 'role_name', 'country', 'login')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
time_in_system	String
user_id	Int
ip	String
session_id	String
role_name	String
country	String (User's location by IP)
login	String

2.13 active_smpp_sessions_get

Docstring:

Returns list of active SMPP sessions.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "is_inbound": true,
    "fields": ["host", "type", "login"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["host", "type", "login"],
  "message": "OK",
  "data": [
    ["1.2.3.4", "trx", "user123"],
    ["1.2.3.5", "trx", "user123"]
  ],
  "time_elapsed": 0.008065
}
```


Request structure and fields description:

```
{'data': {'fields': ["Enum('point_id', 'host', 'smpp_session_id', 'login', 'password',  
'type', 'point_name')"],  
          'is_inbound': 'Bool',  
          'point_id': 'Integer',  
          'sort_by': [{"Enum('point_id', 'host', 'smpp_session_id', 'login', 'password',  
'type', 'point_name')": "Enum('asc', 'desc', '')"}]},  
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[{'type': 'asc'}]	True	List([Enum('point_id', 'host', 'smpp_session_id', 'login', 'password', 'type', 'point_name'): Enum('asc', 'desc', "")]), min_length: 0)	Field on which list will be sorted
point_id	N/A	True	Integer(gt: 0)	Point ID
is_inbound	N/A	False	Bool	Define which sessions would be returned - inbound or outbound
fields	['point_id', 'host', 'smpp_session_id', 'login', 'password', 'type', 'point_name']	True	List([Enum('point_id', 'host', 'smpp_session_id', 'login', 'password', 'type', 'point_name')]), min_length: 0)	Fields to return

Default fields:

Name	Type & Comments
point_id	Int
host	String
smpp_session_id	String
login	String

password	String
type	String
point_name	String

2.14 actual_session_get

Docstring:

Returns user session if this session active and updates it's ttl.
Creates new user session if old session already expired.

Example:

```
{
  "login": "test",
  "password": "test",
  "ip": "193.127.12.4",
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
}
```

Request structure and fields description:

```
{
  'description': 'String',
  'ip': 'String',
  'login': 'String',
  'password': 'String',
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
ip	N/A	False	String(allow_blank: False)	user IP address
description	""	True	String(allow_blank: True)	
password	N/A	False	String(allow_blank: False)	user password

			False)	
session_id	N/A	False	String(allow_blank: True)	old session_id
login	N/A	False	String(allow_blank: False)	user login

2.15 agg_data_today

Docstring:

Return all existed combinations of client, destination and price from today aggregation data.

Function always returns fixed fields set: ['company_name', 'price', 'destination'].

Example:

- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {
    "destinations": ["380", "38066"],
    "limit": 100,
    "offset": 20
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["company_name", "destination", "price"],
  "message": "OK",
  "data": [
    ["Some client", "0.00125", "380"]
  ],
  "time_elapsed": 0.167038
}
```

Request structure and fields description:

```
{
  'data': {
    'destinations': ['String'],
    'limit': 'Integer',
    'offset': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
destination s	N/A	True	List([String(allow_blank: False)], min_length: 0)	filter by destination codes
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
price	String
destination	String
company_name	String

2.16 allowed_ip_add

Docstring:

Function adds single IP address or subnet to the list of IP addresses that allowed to access the web panel. List of allowed IPs affects all web users.

To add whole subnet need specify IP and mask, for example '194.120.101.0/24' allows to access the web panel from IP's range from 193.104.107.1 - 193.104.107.254 with mask 255.255.255.0.

Function returns 'id'(type - Int) of new entry in database.

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {"ip": "1.1.1.2"}}
```

- Server Response

```
{"status": 200,
  "message": "OK",
  "id": 5,
```

```
"time_elapsed": 0.041132}
```

Request structure and fields description:

```
{'data': {'comment': 'String', 'ip': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
ip	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d{3}[0-2])?(0\.){3}0(/0)?)\$")	single IP or subnet with mask. To allow access from any IP specify 0.0.0.0/0 here.
comment	""	True	String(allow_blank: True)	some comment for the IP address

2.17 allowed_ip_check

Docstring:

Checks if origin user IP address in the list of allowed IPs or allowed subnets to access the web interface.
Function calls when somebody tries to login in web panel. At first checks if origin IP in web.users table, then in web.allowed_ip.
If IP wasn't found neither in the first nor in the second table - sends message with 'auth code' to the user's email address and redirects browser to page where this code need insert to unblock web panel.

The 'auth code' is valid during 5 minutes and may be used once.

Function returns Boolean flag 'allowed' (True - given IP is in the list of allowed IPs, False - not).

Allowed IP check sequence:

- * check specified IP address from allowed user ip (Maintenance->Users->[username]->Allowed Ip)
- * check specified IP address in allowed_ips list (Maintenance->Allowed IPs)

Example:

- Client Request

```
{"session_id": "88fa42f23bea3330780e802a44f53864",  
  "data": {"ip": "183.232.125.170"}}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "time_elapsed": 0.003935,  
  "allowed": true}
```

Request structure and fields description:

```
{'data': {'ip': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
ip	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ((([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?)\$")	IP to check

2.18 allowed_ip_del

Docstring:

Removes IP address or subnet from list of IP addresses that allowed to access the web panel. This action affects all users.

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",  
  "data": {"id": 5}}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.036697}
```

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	single IP address or subnet and mask

2.19 allowed_ips_get

Docstring:

Returns list of IP addresses that allowed to access the web panel for all users.

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",  
  "data": {"limit": 100,  
           "fields": ["id", "ip"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "ip"],  
  "message": "OK",  
  "data": [[4, "1.1.1.1/32"], [5, "1.1.1.2/32"]],  
  "time_elapsed": 0.004284}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('comment', 'ip', 'id')"],  
          'id': 'Integer',  
          'ip': 'String',  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('comment', 'ip', 'id')": "Enum('asc', 'desc', '')"}]},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['comment', 'ip', 'id']	True	List([Enum('comment', 'ip', 'id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('comment', 'ip', 'id'): Enum('asc', 'desc', "")}], min_length: 0)	field name on which result will be sorted
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
id	N/A	True	Integer(gt: 0)	allowed IP's record id
ip	N/A	True	String(allow_blank: True)	filter by exact IP address or by first digits of IP address
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
comment	String
ip	String
id	Int

2.20 asr_acd_get

Docstring:

Returns statistical data (ASR, ACD, Attempts quantity) combined by termination point and destination code for calls that passed through suppliers points during given time period.

Function takes data from aggregated statistic, thats why data for last 3 hours may differentiate from actual.

Example:

- Client Request

```
{
  "session_id": "88fa42f23bea3330780e802a44f53864",
  "data": {
    "limit": 3,
    "end_date": "2014-01-12T23:59:59",
    "offset": 0,
    "start_date": "2014-01-12T00:00:00",
    "fields": ["is_closed", "tp_name", "tp_destination",
              "attempts", "asr", "acd", "sa"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["is_closed", "tp_name", "tp_destination", "attempts",
            "asr", "acd", "sa"],
  "message": "OK",
  "data": [[false, "H_to", "9230", "3933", "14.06", "03:04", 553],
           [false, "H_to", "9231", "858", "21.91", "03:54", 188],
           [false, "H_to", "9232", "1601", "11.43", "03:16", 183]],
  "time_elapsed": 0.057094
}
```

Request structure and fields description:

```
{
  'data': {
    'acd_less_than': 'Integer',
    'acd_more_than': 'Integer',
    'asr_less_than': 'Float',
    'asr_more_than': 'Float',
    'attempts_less_than': 'Integer',
    'attempts_more_than': 'Integer',
    'end_date': 'String',
    'exact_code': 'Bool',
    'fields': ["Enum('asr', 'attempts', 'tp_name', 'tp_id', 'sa', 'tp_destination', 'is_closed', 'acd')"],
    'from_cdr': 'Bool',
    'limit': 'Integer',
    'offset': 'Integer',
    'sort_by': [{"Enum('asr', 'attempts', 'tp_name', 'tp_id', 'sa', 'tp_destination', 'is_closed', 'acd')": "Enum('asc', 'desc', '')"}],
    'start_date': 'String',
    'time_zone': 'String',
    'tp_destination': 'String',
    'session_id': 'String'
  }
}
```

data validation description:

Name	Default value	Optional	Type	Description
attempts_more_than	N/A	True	Integer(gte: 0)	filter by attempts quantity. Displays only rows with attempts more than given
attempts_less_than	N/A	True	Integer(gte: 0)	filter by attempts quantity. Displays only rows with attempts less than given
end_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2}):(\d{2})\$")	Defines data end date and time of the report being generated.
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\d{1,2}):(\d{2})\$")	"start_date" and "end_date" timezone
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['asr', 'attempts', 'tp_name', 'tp_id', 'sa', 'tp_destination', 'is_closed', 'acd']	True	List([Enum('asr', 'attempts', 'tp_name', 'tp_id', 'sa', 'tp_destination', 'is_closed', 'acd')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
asr_less_than	N/A	True	Float(gte: 0.0)	filter by asr, displays in report only rows with asr less than given
acd_more_than	N/A	True	Integer(gte: 0)	filter by asr, displays in report only rows with acd more than given
asr_more_than	N/A	True	Float(gte: 0.0)	filter by asr, displays in report only rows with asr more than given
tp_destination	N/A	True	String(allow_blank: True, regex: "^(^d+(, ?d+)*\$) (^\$)")	destination code or first digits of code
from_cdr	False	False	Bool	Getting data from CDR

				instead of aggregated data.
sort_by	[]	True	List([Enum('asr', 'attempts', 'tp_name', 'tp_id', 'sa', 'tp_destination', 'is_closed', 'acd'): Enum('asc', 'desc', ")"}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
acd_less_than	N/A	True	Integer(gte: 0)	filter by asr, displays in report only rows with acd less than given
start_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2}):(?\\d{2}):(?\\d{2})\$")	Defines data start date and time of the report being generated.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
exact_code	N/A	True	Bool	if True - searches by exact code specified in field "tp_destination", else - searches by given code and all available subcodes

Default fields:

Name	Type & Comments
asr	String
attempts	String
tp_name	String
tp_id	Int
sa	Int - quantity of successful attempts
tp_destination	String

is_closed	Boolean
acd	String

2.21 asr_acd_get_ac

Docstring:

Function for accucore.

Request structure and fields description:

```
{'data': {'clear_number': 'String',
          'client_id': 'Integer',
          'direction': 'String',
          'end_date': 'String',
          'fields': ["Enum('asr', 'acd')"],
          'pt_id': 'Integer',
          'start_date': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
clear_number	N/A	True	String(allow_blank: False)	filter by clear destination number(number without OP or TP prefix) or first digits of number.
direction	N/A	True	String(allow_blank: False, regex: "^(incoming) (outgoing)\$")	filter defines call direction (incoming for incoming traffic, outgoing - for outgoing traffic)
fields	['asr', 'acd']	True	List([Enum('asr', 'acd')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
pt_id	N/A	True	Integer(gt: 0)	filter by Point id

client_id	N/A	True	Integer(gt: 0)	filter by client id
end_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T)((\\d{2}):\\d{2}):\\d{2})(\\.\\d{6}){0,1}\$")	end date and time for data gathering. Function considers calls with setup_time not more than given.
start_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T)((\\d{2}):\\d{2}):\\d{2})(\\.\\d{6}){0,1}\$")	start date and time for data gathering. Function considers calls with setup_time not less than given.

Default fields:

Name	Type & Comments
asr	Numeric(15, 2)
acd	Int

2.22 audit_tables_names_get

Docstring:

Function retrieves names from audit table field with table name, which was modified.

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "date_from": "2014-01-17T10:38:00",
    "date_to": "2014-01-17T23:59:59",
    "fields": ["name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["name"],
  "message": "OK",
  "data": [["allowed_ips"]],
  "time_elapsed": 0.004875
}
```

Request structure and fields description:

```
{'data': {'date_from': 'String',  
          'date_to': 'String',  
          'fields': ["Enum('name',)"],  
          'user_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
user_id	N/A	True	Integer(gt: 0)	user_id for tables filtering
date_to	N/A	True	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	defines end date for request to filter by adding date
date_from	N/A	True	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	defines start date to filter result by adding date
fields	['name']	True	List([Enum('name',)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
name	String

2.23 audites_get

Docstring:

Displays changes log for specified time interval in chosen DB table space. Function retrieves old and new field values, user name who made changes, what kind of actions were made, field of table that has been edited, which IP address were actions done from, date and time of action.

Possible values of key "table" for necessary time period you can get using API function "audit_tables_names_get".

Example:

- Client Request

```
{ "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": { "limit": 101,
            "date_from": "2014-01-17T10:38:00",
            "date_to": "2014-01-17T23:59:59",
            "offset": 0,
            "fields": [ "row_id", "user", "date", "tables", "table_name",
                        "action", "field_name", "old_value", "new_value" ] }
}
```

- Server Response

```
{ "status": 200,
  "fields": [ "row_id", "user", "date", "tables", "table_name",
              "action", "field_name", "old_value", "new_value" ],
  "message": "OK",
  "data": [ [ "", "User1", "2014-01-17T10:38:01.195483+00:00",
              "allowed_ips", "allowed_ips", "delete", "comment",
              "45", null ],
            [ "", "User1", "2014-01-17T10:38:01.195483+00:00",
              "allowed_ips", "allowed_ips", "delete", "ip",
              "45.45.45.45/32", null ] ],
  "time_elapsed": 0.009779 }
```

Request structure and fields description:

```
{ 'data': { 'action': 'String',
            'date_from': 'String',
            'date_to': 'String',
            'fields': [ "Enum('log_user_id', 'user', 'date', 'new_value', 'ip_address', 'tables', 'field_name', 'old_value', 'session_id', 'table_name', 'action', 'row_id')"],
            'limit': 'Integer',
            'offset': 'Integer',
            'old_new_value': 'String',
            'sort_by': [ "Enum('log_user_id', 'user', 'date', 'new_value', 'ip_address', 'tables', 'field_name', 'old_value', 'session_id', 'table_name', 'action', 'row_id')": "Enum('asc', 'desc', '')" ] ],
            'table': 'String',
            'time_zone': 'String',
            'user_id': 'Integer' },
  'session_id': 'String' }
```

data validation description:

Name	Default value	Optiona	Type	Description
------	---------------	---------	------	-------------

		1		
old_new_value	N/A	True	String(allow_blank: False)	search by old or new value of column of changed/added/deleted entry in DB.
table	N/A	True	String(allow_blank: False)	filter by database tables
user_id	N/A	True	Integer(gt: 0)	filter by user id
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?d{1,2}):(\d{2})\$")	"start_date" and "end_date" timezone
action	N/A	True	String(allow_blank: False)	filter by actions: insert, update, login, logout or delete
date_to	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})((T) (O)(\\d{2}):(\d{2}):(\d{2}))\$")	filter by action date. Defines upper boundary of the action's date. If absent returns action logs until last.
sort_by	[{'date': 'desc'}]	True	List([Enum('log_user_id', 'user', 'date', 'new_value', 'ip_address', 'tables', 'field_name', 'old_value', 'session_id', 'table_name', 'action', 'row_id'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
date_from	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})((T) (O)(\\d{2}):(\d{2}):(\d{2}))\$")	filter by action date. Defines lower boundary of the action's date. If absent this function returns action logs beginning from the first.
fields	{'log_user_id': 'Int', 'user': 'String', 'date': 'String', 'new_value': 'String', 'ip_address': 'String', 'tables':	True	List([Enum('log_user_id', 'user', 'date', 'new_value', 'ip_address', 'tables', 'field_name', 'old_value',	required fields. If absent - takes fields specified in "default fields". If

	'String', 'field_name': 'String', 'old_value': 'String', 'session_id': 'String', 'table_name': 'String', 'action': 'String', 'row_id': 'String'}		'session_id', 'table_name', 'action', 'row_id']], min_length: 0)	empty - returns error.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
log_user_id	Int
user	String
date	String
new_value	String
ip_address	String
tables	String
field_name	String
old_value	String
session_id	String
table_name	String
action	String
row_id	String

2.24 auto_test_add

Docstring:

Function creates request for destination (aka route) quality testing and returns created test group id.

Test automatically obtains status "processing" (value "4" in database).

Allows to specify test numbers manually (in param "numbers") or get them from CDRs.

Search conditions in CDRs are following:

- time period: last 7 days
- NAS hosts (or in another words RTP hosts): all
- termination points: all
- duration: at least 60 seconds
- called numbers are unique
- quantity of unique numbers the same as specified in "number_of_calls_per_dest

" parameter

Example:

- Client Request

```
{
  "session_id": "680121f94e87e540415c22d931a58800",
  "data": {
    "host": "8.8.8.4",
    "number_of_calls_per_dest": 6,
    "duration": 30,
    "numbers_from_cdr": False,
    "send_email_report": True,
    "tps_dests_data": [[4692, "38050"]],
    "numbers": ["380501111111", "380507777777", "380508888888", "380509999999"],
    "ani_number": "+01234567"
  }
}
```

- Server Response

```
{
  "id": 203,
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.006361
}
```

Request structure and fields description:

```
{'data': {'ani_number': 'String',
  'block_destination': 'Bool',
  'copy_to': ['String'],
  'duration': 'Integer',
  'host': 'String',
  'number_of_calls_per_dest': 'Integer',
  'numbers': ['String'],
  'numbers_from_cdr': 'Bool',
  'send_email_report': 'Bool',
  'test_type': 'Integer',
```

```
'tps_dests_data': [['Integer', 'String']],
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
duration	30	True	Integer()	duration of each test call in seconds
number_of_calls_per_dest	5	True	Integer(gt: 0, lt: 101)	quantity of test calls (and numbers from CDR) per destination code in case 'numbers_from_cdr' is enabled
send_email_report	True	True	Bool	send report to the current user's email
numbers	[]	True	List([Destination(allow_blank: False, _dest_pattern: ^\d*\$)], min_length: 0)	manual list of numbers for test
tps_dests_data	N/A	False	List([List([Integer(gt: 0), Destination(allow_blank: False, _dest_pattern: ^\d*\$)], 2)], min_length: 1, max_length: 10)	array of arrays where first element is TP id and the second one - respective destination code
test_type	2	True	Integer(gt: 0, lt: 3)	1 - test destination, 2 - test supplier
host	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?)\$")	RTP host that will process test calls
block_destination	False	True	Bool	close destination in case all calls

				were unsuccessful
copy_to	N/A	True	List([String(allow_blank: False, regex: "(?P^[-#\$%&'*/+=?^_`{} ~0-9A-Z]+(\\[-#\$%&'*/+=?^_`{} ~0-9A-Z]+)* ^"([\\001-\\010\\013\\014\\016-\\037!#-\\[\\]-\\177])\\ \\001-011\\013\\014\\016-\\177))*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\\. (?:[A-Z0-9-]{2,}\\. ?))\$)\\ [(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\. (25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)){3}\\ "\$", max_length: 254)], min_length: 1)	given emails will be added to copy in case 'send_email_report' is enabled
ani_number	N/A	True	String(allow_blank: True, regex: "^\\+?\\d*\$")	ANI number for checking routing before test execution. Max length is 20 digits with "+".
numbers_from_cdr	True	True	Bool	says to take numbers for tests from CDR (last 7 days)

2.25 auto_test_del

Docstring:

Deletes chosen test by test group id.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {"id": 6},
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.067966,
  "id": 6
}
```

Request structure and fields description:

```
{'data': {'id': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	String(allow_blank: False)	test group id to be deleted

2.26 auto_test_email_send

Docstring:

Sends test results to the email of user who make request.
Returns test group id within key id if request was processed successfully.

Example:

- Client Request:

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {"id": 10},
}
```

- Server Response:

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.067936,
  "id": 10,
}
```

Request structure and fields description:

```
{'data': {'id': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	String(allow_blank: False)	test group id to send email

2.27 auto_test_get

Docstring:

Returns list of existed tests.

Example:

- Client Request

```
{
  "fields": ["id", "date_created", "tp_names", "destinations", "calls_count", "host", "user", "status"],
  "sort_by": [{"date_created": "asc"}],
  "limit": 10,
  "tp_name": "Arbim_"
}
```

- Server Response

```
{
  "data": [
    [1000, "2015-02-02 08:58:34", "Arbim_H323_out, Arbim_H323_out", "38050,38044", 6, "8.8.8.4/32", "dev", "Complete"],
    [1001, "2015-02-02 09:00:00", "Arbim_H323_out", "8801", 6, "8.8.8.4/32", "dev", "Complete"]
  ],
  "fields": ["id", "date_created", "tp_names", "destinations", "calls_count", "host", "user", "status"],
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.006361
}
```

Request structure and fields description:

```
{'data': {'company': 'String',
          'destination': 'String',
          'fields': ["Enum('status', 'tp_names', 'host', 'link', 'user', 'calls_count', 'date_created', 'status_date', 'id', 'destinations')"],
```

```

        'limit': 'Integer',
        'offset': 'Integer',
        'sort_by': [{"Enum('status', 'tp_names', 'host', 'link', 'user', 'calls_count', 'date_created', 'status_date', 'id', 'destinations')": "Enum('asc', 'desc', '')"}],
        'tp_name': 'String'},
        'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['status', 'tp_names', 'host', 'link', 'user', 'calls_count', 'date_created', 'status_date', 'id', 'destinations']	True	List([Enum('status', 'tp_names', 'host', 'link', 'user', 'calls_count', 'date_created', 'status_date', 'id', 'destinations')], min_length: 0)	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
destination	N/A	True	String(allow_blank: False)	filter by destination code
limit	0	True	Integer(gte: 0)	rows quantity per one request
company	N/A	True	String(allow_blank: False)	filter by company name
sort_by	[]	True	List([Enum('status', 'tp_names', 'host', 'link', 'user', 'calls_count', 'date_created', 'status_date', 'id', 'destinations'): Enum('asc', 'desc', '')}], min_length: 0)	field name on which result will be sorted
tp_name	N/A	True	String(allow_blank: False)	filter by tp name

Default fields:

Name	Type & Comments
status	Int (test status name: 4 - "Processing", 3 - "Error", 2 - "Complete" or 1 - "In progress")
tp_names	String (comma separated TP names)
host	String (RTP host IP address)
link	String (link to see tests calls result)
user	String (name of user who added test request)
calls_count	calls quantity per destination
date_created	String (date and time of test creation)
status_date	String (date and time of test status update)
id	String (test group code in database)
destinations	String (comma separated destination codes)

2.28 auto_test_result_get

Docstring:

Returns result of chosen test and test calls data.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  {
    "id": 'test_user_2015421001123',
    "fields": [
      "voiper_test_id", "destination", "asr", "acd", "date", "status"
, "tp_name", "test_calls"
    ]
  },
}
```

- Server Response


```

{
  "status": 200, "message": "OK", "time_elapsed": 0.067966,
  "data":
    [
      759, u"8801 - Bangladesh mobile", Decimal("40.00"), 18, u'2015-
05-01T11:20:16+00:00',
      u'Finished', 'Safa_out',
      [
        [u'2015-05-01T11:20:16+00:00', u'88013420943', 5, 0, 0,
          u'21/Call rejected', u'5481028270', '', '', False],
        [u'2015-05-01T11:20:23+00:00', u'88013420944', 0, 0, 0,
          u'41/Normal temporary failure', u'8990588407', '', '',
False],
        [u'2015-05-01T11:20:26+00:00', u'88013420945', 2, 10, 25,
          u'16/Normal call clearing', u'8990588406',
          u'http://1.1.1.1/record/playFile?uuid=0b38e90d-5ee1-
4671-86ee-546c9af3f722_rbt',
          u'http://1.1.1.1/record/playFile?uuid=0b38e90d-5ee1-
4671-86ee-546c9af3f722_call', True],
        [u'2015-05-01T11:20:29+00:00', u'88013420946', 0, 0, 0,
          u'17/User busy', u'8990588407', '', '', False],
        [u'2015-05-01T11:20:35+00:00', u'88013420955', 3, 5, 11,
          u'16/Normal call clearing', u'8990588408',
          u'http://1.1.1.1/record/playFile?uuid=d8d3bf31-a95b-
4201-9744-d5a277bf3ee6_rbt',
          u'http://1.1.1.1/record/playFile?uuid=d8d3bf31-a95b-
4201-9744-d5a277bf3ee6_call', True],
      ]
    ]
  "fields": ["voiper_test_id", "destination", "asr", "acd", "date", "stat
us", "tp_name", "test_calls"]
}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('status', 'asr', 'destination', 'status_code', 'tp_client_id',
'calls_prefix', 'voiper_test_id', 'op_id', 'date', 'acd', 'op_client_id', 'test_calls',
'tp_name', 'tp_id')"],
'id': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	String(allow_blank: False)	Test group id to get tests result
fields	['status', 'asr', 'destination',	True	List([Enum('status', 'asr',	Required fields. If

	'status_code', 'tp_client_id', 'calls_prefix', 'voiper_test_id', 'op_id', 'date', 'acd', 'op_client_id', 'test_calls', 'tp_name', 'tp_id']		'destination', 'status_code', 'tp_client_id', 'calls_prefix', 'voiper_test_id', 'op_id', 'date', 'acd', 'op_client_id', 'test_calls', 'tp_name', 'tp_id']], min_length: 0)	absent - takes fields specified in 'default fields'. If empty - returns error.
--	--	--	--	--

Default fields:

Name	Type & Comments
status	String - test status in VoiperCan
asr	String - asr for given calls
destination	String - test destination
status_code	Int - test group status in Mediacore: 1 - in progress, 2 - completed, 3 - error, 4 - processing
tp_client_id	Int - TP client id
calls_prefix	String - call prefix for filter incoming/outgoing calls
voiper_test_id	Int - test id in VoiperCan DB
op_id	Int
date	String - date and time when test were added
acd	Int - ACD for tested calls
op_client_id	Int - OP client id
test_calls	list of test calls with following fields: start time(String), called number(String), PDD(Int, in seconds), RBT(Int, in seconds), call duration(Int), hangup cause(String), caller number(String), links for RBT and call audio records(both String), good_call(Boolean)
tp_name	String
tp_id	Int

2.29 available_routes_by_providers_get

Docstring:

Function returns list of current actual TP prices, which fit given destination.
Also function provides ASR, ACD and attempts statistic per TP destination for current day from 00:00:00 by GMT+0 to the last hour with aggregation.

Prices values in report display in currency of their tariff.

Example:

- Client Request

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
  "data":{"limit":101,
        "common_codes":false,
        "offset":0,
        "show_subcodes": false,
        "fields":["tp_id",
                  "tp_name",
                  "destination",
                  "price_priority",
                  "price",
                  "date_start",
                  "attempts",
                  "asr",
                  "acd",
                  "country"],
        "destination":"23"}
```

Request structure and fields description:

```
{'data': {'common_codes': 'Bool',
          'destination': 'String',
          'fields': ["Enum('price_priority', 'asr', 'attempts', 'date_start', 'country'
, 'tp_name', 'tp_id', 'price', 'destination', 'acd')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'session_id': 'String',
          'show_subcodes': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
show_subcodes	False	True	Bool	If "True" returns sub codes
session_id	N/A	True	String(allow_blank: False)	by default from user session
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
destination	N/A	False	String(allow_blank: False)	destination code
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
common_codes	N/A	False	Bool	If "true" returns prices for codes which are sub codes for given "destination"
fields	['price_priority', 'asr', 'attempts', 'date_start', 'country', 'tp_name', 'tp_id', 'price', 'destination', 'acd']	True	List([Enum('price_priority', 'asr', 'attempts', 'date_start', 'country', 'tp_name', 'tp_id', 'price', 'destination', 'acd')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error

Default fields:

Name	Type & Comments
price_priority	Int
asr	String (ASR for current day in percent)
attempts	Int
date_start	timestamp

country	String
tp_name	String
tp_id	Int
price	numeric
destination	String
acd	Int (ACD for current day in seconds)

2.30 average_stat_get

Docstring:

Returns calls statistic data to draw graphs for calls with duration by chosen date, point and destination (optionally).

Key "data" contains 288 lists if user requests data for previous day. Each list - is data for 5 minute interval (first list - is the first 5 minute interval of day, the second - is the second 5 minute interval and so on).

For today function returns as many nested lists as 5 minutes intervals passed for today.

The first value in each nested list is value in x-axis(milliseconds from the Epoch - 01.01.1970), second value - y-axis.

Value for y-axis counts in the following way:

- 1) if call lasted all five minute interval then value is increased by 1,
- 2) if call lasted only 60 seconds of the interval then value is increased by 0.2.

Example:

- Client Request

```
{
  "session_id": "e52185466e19827155f598fcaf4faf41",
  "data": {
    "pt_id": 7,
    "date": "2014-01-13T00:00:00"
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": [
    [1389571350000, 3.8966666666666665],
    [1389571650000, 2.0333333333333333],
    [1389571950000, 2.2533333333333334],
  ]
}
```

```

[1389572250000, 2.2766666666666667],
[1389572550000, 2.3266666666666667],
[1389572850000, 2.9433333333333334],
.... ],
"time_elapsed": 1.41132"}

```

Request structure and fields description:

```

{'data': {'date': 'String',
          'dest': 'String',
          'exact_code': 'Bool',
          'pt_id': 'Integer',
          'time_zone': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2}):(\d{2})\$")	date for data gathering.
time_zone	+0:00	True	String(allow_blank: False, regex: "^([+-]?\d{1,2}):(\d{2})\$")	
exact_code	False	True	Bool	exact dest or like
pt_id	N/A	False	Integer(gt: 0)	filter by PT id
dest	N/A	True	String(allow_blank: False, regex: "^(\d+(, \s)?)*\$")	filter by destination codes separated by comma

2.31 balance_add

Docstring:

```

Creates new balance for given client with
given currency, balance parameters and notification options.
Function returns 'id'(type - Int) of new balance.

```

Example:

- Client Request

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
  "data":{"limit":0.0,
    "currency_id":86,
    "notify_flag":false,
    "name":"new_balance",
    "alarm_limit":0.0,
    "alarm_limit_unit": 2,
    "alarm_flag":false,
    "client_id":511,
    "limit_flag":true}
}
```

Request structure and fields description:

```
{'data': {'alarm_flag': 'Bool',
  'alarm_limit': 'String',
  'alarm_limit_unit': 'Integer',
  'client_id': 'Integer',
  'currency_id': 'Integer',
  'limit': 'String',
  'limit_flag': 'Bool',
  'name': 'String',
  'notify_flag': 'Bool',
  'tp_limit': 'String',
  'tp_limit_flag': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_limit	N/A	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (may be as well positive as negative). If "alarm_flag" is True client will get notification about balance state. Notification for the same company may be send only one time per day.
alarm_limit_unit	N/A	False	Integer(gt: 0)	alarm limit unit type: 1 - absolute value in balance's currency, 2 - %
currency_id	N/A	False	Integer()	id of the balance currency
alarm_limit	N/A	False	Numeric(precision: 21,	difference between current balance and value specified in "limit" when alarm about balance state will be generated.

			scale: 5, gte: 0)	If "alarm_limit" is "0" notification about balance state won't be send neither to client nor to email address specified in the appropriate template.
client_id	N/A	False	Integer()	id of the client, to which the balance will be added
limit	N/A	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (may be as well positive as negative). If "alarm_flag" is True client will get notification about balance state. Notification for the same company may be send only one time per day.
limit_flag	N/A	False	Bool	option that enables/disables closing origination points when value specified in "limit" has been reached
notify_flag	N/A	False	Bool	option that enables/disables sending "alarm_limit", "limit" and notification about closure of origination points to the client. If "notify_flag" is "False" these notifications won't be sent to the client. If "alarm_flag" is "False" "alarm_limit" and "limit" notification won't be sent even if "notify_flag" is "True".
name	N/A	False	String(allow_blank: False)	balance name
alarm_flag	N/A	False	Bool	option that enables/disables sending "alarm_limit" and "limit" notifications.
tp_limit_flag	N/A	False	Bool	option that enables/disables closing termination points when value specified in "tp_limit" has been reached

2.32 balance_get

Docstring:

Returns history of changes for chosen balance during specified time period. History includes positive("pay_to_us") and negative("we_pay") payments, customer's and supplier's invoices.

Excluding data mentioned in 'fields' function also returns "owner_email"(email of owner assigned to the client owner of given balance; type String) and "client_email"(billing email of client which owns given balance; type "String").

Key "check_balance_exception" contains details about didn't created invoices for chosen time period.

Example:

- Client Request:

```
{"session_id": "0f875930826540ca9f9285263e0eee08",  
  "data": {"end_date": "2013-11-03T00:00:00",  
           "balance_id": 172,  
           "start_date": "2013-11-01T00:00:00",  
           "fields": ["client_document",  
                     "begin_balance",  
                     "we_invoice",  
                     "invoice_to_us",  
                     "we_pay",  
                     "pay_to_us",  
                     "end_balance"]}  
}
```

- Server Response

```
{"status": 200,  
  "time_elapsed": 0.258547,  
  "check_balance_exception": "Incoming Invoice for period 2013-06-24 00:00:00+00 - 2014-04-25 23:59:59 is absent. Sum for this period is : 9708.70443USD. \nIncoming Invoice for period 2014-05-16 00:00:00 - 2014-06-15 23:59:59 is absent. Sum for this period is : 2858.53139USD.",  
  "client_email": "luda@dev.speedflow.com",  
  "fields": ["client_document", "begin_balance", "we_invoice",  
            "invoice_to_us", "we_pay", "pay_to_us", "end_balance"],  
  "message": "OK",  
  "owner_email": "cz5080@speedflow.com",  
  "data": [[ "2013-11-02 payment for traffic (1)", "0.00000", "0.00000",  
            "0.00000", "0.00000", "1.00000", "1.00000"]]  
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',  
          'client_id': 'Integer',  
          'end_date': 'String',  
          'fields': ["Enum('invoice_to_us', 'we_pay', 'gen_date', 'date', 'begin_balance', 'id', 'we_invoice', 'client_document', 'pay_to_us', 'type', 'end_balance')"],  
          'include_vat': 'Bool',  
          'session_id': 'String',  
          'start_date': 'String',  
          'use_current_traffic': 'Bool'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
use_curre	False	True	Bool	if "True", then hasn't created yet

nt_traffic				auto invoice for current traffic will be added
end_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})((T)())(\\d{2}):\\d{2}):\\d{2})\$")	end date in format "YYYY-MM-DDTHH:MM:SS" or "YYYY-MM-DD HH:MM:SS"
balance_id	N/A	True	Integer(gt: 0)	
client_id	N/A	True	Integer(gt: 0)	filter by client id, which owns balance
fields	['invoice_to_us', 'we_pay', 'gen_date', 'date', 'begin_balance', 'id', 'we_invoice', 'client_document', 'pay_to_us', 'type', 'end_balance']	True	List([Enum('invoice_to_us', 'we_pay', 'gen_date', 'date', 'begin_balance', 'id', 'we_invoice', 'client_document', 'pay_to_us', 'type', 'end_balance')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
session_id	N/A	True	String(allow_blank: False)	need for checking user permissions to see history of balance with specific id. By default from current user session.
start_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})((T)())(\\d{2}):\\d{2}):\\d{2})\$")	start date in format "YYYY-MM-DDTHH:MM:SS" or "YYYY-MM-DD HH:MM:SS"
include_vat	False	True	Bool	include VAT or not

Default fields:

Name	Type & Comments
invoice_to_us	String
we_pay	String
gen_date	String

date	String
begin_balance	String
id	Int
we_invoice	String
client_document	String
pay_to_us	String
type	String
end_balance	String

2.33 balance_get_on_date

Docstring:

Function for ACCUCORE.

Returns approximate balance values (in balance currencies) on selected date.

Example:

```
- Client request
{
  "session_id":"ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "date": "2020-01-14",
    "balance_id": 494,
  }
}
```

```
- Server Response
{
  "status": 200,
  "fields": [
    "approximate_value",
    "balance_id",
    "client_id"
  ],
  "message": "OK",
}
```

```

    "data": [
      [
        "-62886.45436",
        144,
        12
      ]
    ],
    "time_elapsed": 0.028509
  }

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
  }
}

```

2.34 balance_report

Docstring:

Function calls engine that creates balance change report file given balance and period. Then it generates download link and returns it.

- Client Request:

```

{
  "session_id": "fe90713461732f5d8b518298bc8378cf",
  "data": {
    "balance_id": 271,
    "start_date": "2014-01-01T00:00:00",
    "end_date": "2014-06-20T00:00:00",
  }
}

```

- Server Response:

```

{
  "status": 200,

```

```

    "fields": ["link"],
    "message": "OK",
    "data": [{"http://193.161.85.80:8070/cdn/download/2712014-e01-0g120er14-
re06-"}],
    "time_elapsed": 0.33943
}

```

Request structure and fields description:

```

{'data': {'balance_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('link',)"],
          'include_vat': 'Bool',
          'start_date': 'String',
          'use_current_traffic': 'Bool'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	False	Integer(gte: 0)	id of balance for which report will be created
end_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T)(\\d{2}):\\d{2}:\\d{2})\$")	end date for the report
use_current_traffic	False	True	Bool	this option determines whether to include current traffic to the report or not
start_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T)(\\d{2}):\\d{2}:\\d{2})\$")	start date for the report
include_vat	False	True	Bool	this option determines whether to include VAT to the report or not
fields	['link']	True	List([Enum('link',)], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

Default fields:

Name	Type & Comments
link	String

2.35 balance_set

Docstring:

Changes parameters of the chosen balance:name, limits and notification options.

Function returns 'id'(type - Int) of balance that was processed.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "limit": 0.00,
    "id": 756,
    "notify_flag": false,
    "name": "xdf",
    "alarm_limit_unit": 1,
    "alarm_limit": 0.00,
    "alarm_flag": false,
    "limit_flag": true
  }
}
```

Request structure and fields description:

```
{
  'data': {
    'alarm_flag': 'Bool',
    'alarm_limit': 'String',
    'alarm_limit_unit': 'Integer',
    'id': 'Integer',
    'limit': 'String',
    'limit_flag': 'Bool',
    'name': 'String',
    'notify_flag': 'Bool',
    'tp_limit': 'String',
    'tp_limit_flag': 'Bool'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
name	N/A	False	String(allow_blank: False)	new name of the balance
limit	N/A	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (may be as well positive as negative). If "alarm_flag" is True client will get notification about balance state. Notification for the same company may be send only one time per day.
alarm_flag	N/A	False	Bool	option that enables/disables sending "alarm_limit" and "limit" notifications.
alarm_limit_unit	N/A	False	Integer(gt: 0)	alarm limit unit type: 1 - absolute value in balances currency, 2 - %
notify_flag	N/A	True	Bool	option that enables/disables sending "alarm_limit", "limit" and notification about closure of origination points to the client. If "notify_flag" is "False" these notifications won't be sent to the client. If "alarm_flag" is "False" "alarm_limit" and "limit" notification won't be sent even if "notify_flag" is "True".
alarm_limit	N/A	False	Numeric(precision: 21, scale: 5, gte: 0)	difference between current balance and value specified in "limit" when alarm about balance state will be generated. If "alarm_limit" is "0" notification about balance state won't be send neither to client nor to email address specified in the appropriate template.
tp_limit	0	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (may be as well positive as negative). If "tp_alarm_flag" is True client will get notification about balance state. Notification for the same company may be send only one time per day.
id	N/A	False	Integer()	balance id
limit_flag	N/A	False	Bool	option that enables/disables closing origination points when value specified in "limit" has been reached
tp_limit_flag	False	False	Bool	option that enables/disables closing termination points when value specified in "tp_limit" has been reached

2.36 balancer_relation_set

Docstring:

Changes weight coefficient or disables chosen host <=> balancer relation.

Example:

- Client Request:

```
{"session_id": "3fbcfc27845ec9a10490288b9ed89b73",  
  "data": {"balancer_id": 12,  
           "host_id": 5,  
           "weight": 15}  
}
```

Request structure and fields description:

```
{'data': {'active': 'Bool',  
          'balancer_id': 'Integer',  
          'host_id': 'Integer',  
          'weight': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balancer_id	N/A	False	Integer(gt: 0)	id of balancer which relation should be changed
active	N/A	True	Bool	relation status
weight	N/A	True	Integer(gt: 0)	new weight of relation
host_id	N/A	False	Integer(gt: 0)	id of host which relation should be changed

2.37 balancer_relations_get**Docstring:**

Returns hosts assigned to the given balancer or all relations
host <=> balancer.

Example:

- Client Request:

```
{
  "session_id": "3fbcfc27845ec9a10490288b9ed89b73",
  "data": {
    "balancer_id": 12,
    "limit": 1,
    "host_id": 5,
    "sort_by": [{"balancer_id": ""}],
    "fields": ["balancer_id", "host_id", "weight", "active"]}
}
```

- Server Response:

```
{
  "data": [[12, 5, 10, true]],
  "fields": ["balancer_id", "host_id", "weight", "active"],
  "time_elapsed": 0.005129,
  "status": 200}
}
```

Request structure and fields description:

```
{
  'data': {
    'balancer_id': 'Integer',
    'fields': ["Enum('active', 'host_id', 'host', 'balancer_id', 'weight')"],
    'host_id': 'Integer',
    'limit': 'Integer',
    'offset': 'Integer',
    'sort_by': [{"Enum('active', 'host_id', 'host', 'balancer_id', 'weight')": "Enum('asc', 'desc', '')"}]},
    'session_id': 'String'
  }
}
```

data validation description:

Name	Default value	Optional	Type	Description
limit	0	True	Integer(gte: 0)	says how many rows will be in response
fields	['active', 'host_id', 'host', 'balancer_id', 'weight']	True	List([Enum('active', 'host_id', 'host', 'balancer_id', 'weight')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
host_id	N/A	True	Integer(gt: 0)	filter by host id

sort_by	[]	True	List([{Enum('active', 'host_id', 'host', 'balancer_id', 'weight'): Enum('asc', 'desc', "")}}, min_length: 0)	field name on which result will be sorted
balancer_id	N/A	True	Integer(gt: 0)	filter by balancer id
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
active	Bool
host_id	Int
host	String - host IP address
balancer_id	Int
weight	Int

2.38 balancer_rtp_relation_set

Docstring:

Request structure and fields description:

```
{'data': {'balancer': 'String',
          'host': 'String',
          'protocol': 'Integer',
          'weight': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balancer	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12\d 3[0-2]))?(0\.){3}0(0)?)\$")	balancer which relation should be changed
host	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12\d 3[0-2]))?(0\.){3}0(0)?)\$")	host which relation should be changed
weight	N/A	False	Integer(gte: 0, lte: 100)	new weight of relation
protocol	N/A	False	Integer(gte: 1, lte: 2)	SIP or H.323 protocol of relation

2.39 balancer_rtp_relations_get

Docstring:

Request structure and fields description:

```
{'data': {'balancer': 'String',
          'fields': ["Enum('host', 'weight', 'balancer')"],
          'protocol': 'Integer',
          'sort_by': [{"Enum('host', 'weight', 'balancer')": "Enum('asc', 'desc', '')"}]
},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
protocol	N/A	False	Integer(gte: 1, lte: 2)	SIP or H.323 protocol of relation

balancer	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?\$")	address where to take relations
fields	['host', 'weight', 'balancer']	True	List([Enum('host', 'weight', 'balancer')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{'Enum('host', 'weight', 'balancer'): Enum('asc', 'desc', '')}], min_length: 0)	field name on which result will be sorted

Default fields:

Name	Type & Comments
host	String - host IP address
weight	Int
balancer	String

2.40 balancers_get

Docstring:

Get balancer hosts from API config

Request structure and fields description:

```
{'data': {'fields': ["Enum('host',)"],
              'sort_by': [{"Enum('host',)": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['host']	True	List([Enum('host'),], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('host'): Enum('asc', 'desc', ")"}], min_length: 0)	field name on which result will be sorted

Default fields:

Name	Type & Comments
host	String

2.41 balances_get

Docstring:

Returns financial settings information of the given balance(s).

Example:

- Client Request

```
{
  "session_id": "d8f085383f4f04fd1d6b30fe537282cd",
  "data": {
    "id": 472,
    "sort_by": [{"client_name": "", "name": ""}],
    "fields": ["id", "name", "client_name"],
    "show_currency": true
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name", "client_name"],
  "message": "OK",
  "data": [[472,
    "test 15",
    "test 15"]],
  "time_elapsed": 0.006361
}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer or Null',
          'fields': ["Enum('limit_flag', 'currency_name', 'name', 'tp_limit_flag', 'currency_id', 'alarm_limit_unit', 'alarm_flag', 'include_vat', 'limit', 'actual_vat_balance', 'client_id', 'client_name', 'tp_limit', 'balance', 'alarm_limit', 'id', 'notify_flag')"],
          'id': 'Integer or Null',
          'limit': 'Integer',
          'offset': 'Integer',
          'show_currency': 'Bool',
          'show_invisible': 'Bool',
          'sort_by': [{"Enum('limit_flag', 'currency_name', 'name', 'tp_limit_flag', 'currency_id', 'alarm_limit_unit', 'alarm_flag', 'include_vat', 'limit', 'actual_vat_balance', 'client_id', 'client_name', 'tp_limit', 'balance', 'alarm_limit', 'id', 'notify_flag')": "Enum('asc', 'desc', '')"}]},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	True	Or(Integer(gt: 0), Null)	client's id that owns the balance
fields	['limit_flag', 'currency_name', 'name', 'tp_limit_flag', 'currency_id', 'alarm_limit_unit', 'alarm_flag', 'include_vat', 'limit', 'actual_vat_balance', 'client_id', 'client_name', 'tp_limit', 'balance', 'alarm_limit', 'id', 'notify_flag']	True	List([Enum('limit_flag', 'currency_name', 'name', 'tp_limit_flag', 'currency_id', 'alarm_limit_unit', 'alarm_flag', 'include_vat', 'limit', 'actual_vat_balance', 'client_id', 'client_name', 'tp_limit', 'balance', 'alarm_limit', 'id', 'notify_flag')], min_length: 0)	defines required fields. If absent - takes fields specified in "default fields". If empty - returns error.
show_currency	True	True	Bool	if "True" then function returns name with currency, else returns just name of the balance
show_invisible	False	True	Bool	if True then function returns balances of invisible clients
sort_by	[]	True	List([Enum('limit_flag',	field on which

			'currency_name', 'name', 'tp_limit_flag', 'currency_id', 'alarm_limit_unit', 'alarm_flag', 'include_vat', 'limit', 'actual_vat_balance', 'client_id', 'client_name', 'tp_limit', 'balance', 'alarm_limit', 'id', 'notify_flag'): Enum('asc', 'desc', ""), min_length: 0)	result sorts
id	N/A	True	Or(Integer(gt: 0), Null)	balance id. If absent in request returns information about all balances.
limit	0	True	Integer(gte: 0)	rows quantity per one request
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
limit_flag	Boolean
currency_name	String(3) - name_short
name	String
tp_limit_flag	Boolean
currency_id	Int
alarm_limit_unit	Int
alarm_flag	Boolean
include_vat	Boolean

limit	String
actual_vat_balance	String
client_id	Int
client_name	String
tp_limit	String
balance	String
alarm_limit	String
id	Int
notify_flag	Boolean

2.42 bind_smpp_session

Docstring:

Bind SMPP session.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "ip": "125.13.80.201",
    "port": "8070",
    "username": "dev",
    "password": "qwerty",
    "protocol": "trx",
    "system_type": "system_type",
    "tp_id": 144
  }
}
```

- Server Response

Request structure and fields description:

```
{'data': {'ip': 'String',
```



```

'password': 'String',
'port': 'String',
'protocol': 'String',
'system_type': 'String or Null',
'tp_id': 'Integer',
'username': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
username	N/A	False	String(allow_blank: False)	Username
port	N/A	False	String(allow_blank: False)	Port
system_type	N/A	False	Or(String(allow_blank: False), Null)	System type
ip	N/A	False	String(allow_blank: False)	IP address
password	N/A	False	String(allow_blank: False)	Password
tp_id	N/A	False	Integer(gt: 0)	Termination point ID
protocol	N/A	False	String(allow_blank: False, regex: <code>"^(trx)\$ ^(txrx)\$ ^(tx)\$ ^(rx)\$"</code>)	Protocol - trx or tx or rx. txrx - Deprecated, use tx

2.43 call_chart_diagram_points_get

Docstring:

Returns number of average calls and attempts for the last day (144 values for each one) to draw graphics by client's incoming/outgoing traffic.

Takes data from cache for better performance.

Example:

```

- Client Request

{

```

```

    "session_id":"33de45c5498772061e43b301da278e04",
    "data":{
        "direction":"in","client_id":252
    }
}

```

- Server Response

```

{
    "status": 200,
    "attempts": [
        3314, 3512, 3744, 2593, 3514, 3316, 2865, 2257, 1502, 999, 766, 714
, 535, 611, 451, 385, 345,
        290, 237, 251, 218, 187, ...],
    "calls": [
        0.0, 0.0, 0.0, 1.5776816666666664, 2.1820933333333334, 2.4447616666
666665, 3.1814533333333337,
        2.9261783333333335, 2.9837249999999997, 2.2631433333333333, 3.009978
3333333333, 3.2525783333333333,
        2.939235, 3.3242916666666667, 2.87151, 1.7224716666666664, 3.003864
9999999993, 2.1292349999999995,
        3.1707449999999997, ...],
    "time_elapsed": 0.436724,
    "message": "OK"
}

```

Request structure and fields description:

```

{'data': {'client_id': 'Integer', 'direction': "Enum('in', 'out')"},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	filter by client
direction	N/A	False	Enum('in', 'out')	filter by traffic direction relatively to the softswitch

2.44 call_chart_diagram_points_get_old

Docstring:

Request structure and fields description:

```
{'data': {'client_id': 'Integer', 'direction': "Enum('in', 'out')"},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	filter by client
direction	N/A	False	Enum('in', 'out')	filter by traffic direction relatively to the softswitch

2.45 call_trace_get

Docstring:

Returns list of possible routes (combination of IP address, prefix and port) for call from chosen OP, ANI number to the given destination number via specific RADIUS (authorization server).

API sends request to the chosen RADIUS server, which returns either currently available routes if any or RADIUS error code.

Prices are being displayed in the tariff currency.

Example:

- Client Request

```
{"session_id": "1149e8a40f1ec8812f7a4349020c7259",  
  "data": {"limit": 101,  
            "op_id": 7,  
            "a_number": "123",  
            "radius_id": 1,  
            "contexts": [1],  
            "number": "9230000000000",  
            "offset": 0,  
            "op_fields": ["op_id", "op_price_id", "op_price", "op_name",  
                          "op_destination", "op_country", "op_currency",  
                          "work_time"],  
            "fields": ["tp_name", "route", "tp_dest", "tp_country",  
                       "tp_price", "tp_capacity", "tp_currency"]}  
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.024503,
  "op_data": [7, 13, 0.04, "H_in", "9230", "Pakistan Mobile Mobilink",
    "USD", "238"],
  "fields": ["tp_name", "route", "tp_dest", "tp_country",
    "tp_price", "tp_capacity", "tp_currency"],
  "message": "OK",
  "data": [{"H_to", "h323/4009230000000000@198.21.162.12:1720",
    "9230", "Pakistan Mobile Mobilink", "10.3500",
    500, "USD"}]
}
```

Request structure and fields description:

```
{'data': {'a_number': 'String',
  'contexts': ['Integer'],
  'fields': ["Enum('tp_lata', '#', 'tp_ocn', 'tp_capacity', 'tp_currency', 'tp_
dialpeer_name', 'tp_name', 'tp_price', 'work_time', 'tp_country', 'tp_dialpeer_id', 'tp
_mc', 'tp_dest', 'tp_ft', 'tp_id', 'position', 'tp_ci', 'route')"],
  'lata': 'String',
  'limit': 'Integer',
  'nar': 'Bool',
  'number': 'String',
  'ocn': 'String',
  'offset': 'Integer',
  'op_fields': ["Enum('op_id', 'op_ocn', 'op_destination', 'op_country', 'op_pr
ice_id', 'op_lata', 'work_time', 'op_currency', 'op_name', 'op_price')"],
  'op_id': 'Integer',
  'radius_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	False	Integer(gt: 0)	id of Origination Point
number	N/A	False	String(allow_blank: False, regex: "^[0-9#*]+\$")	called number. May contain: digits, "#" and "*" symbols
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows

				will be in response
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	ids context, in which request will be processed
op_fields	['op_id', 'op_ocn', 'op_destination', 'op_country', 'op_price_id', 'op_lata', 'work_time', 'op_currency', 'op_name', 'op_price']	True	List([Enum('op_id', 'op_ocn', 'op_destination', 'op_country', 'op_price_id', 'op_lata', 'work_time', 'op_currency', 'op_name', 'op_price')], min_length: 0)	OP information which will be displayed in response. If absent - returns fields specified in "default_fields_op" section. If empty - returns error.
a_number	N/A	True	String(allow_blank: True)	caller number. May contain any symbols
fields	['tp_lata', '#', 'tp_ocn', 'tp_capacity', 'tp_currency', 'tp_dialpeer_name', 'tp_name', 'tp_price', 'work_time', 'tp_country', 'tp_dialpeer_id', 'tp_mc', 'tp_dest', 'tp_ft', 'tp_id', 'position', 'tp_ci', 'route']	True	List([Enum('tp_lata', '#', 'tp_ocn', 'tp_capacity', 'tp_currency', 'tp_dialpeer_name', 'tp_name', 'tp_price', 'work_time', 'tp_country', 'tp_dialpeer_id', 'tp_mc', 'tp_dest', 'tp_ft', 'tp_id', 'position', 'tp_ci', 'route')], min_length: 0)	TP information which will be displayed in response. If absent - returns fields specified in "default_fields" section. If empty - returns error.
ocn	N/A	True	String(allow_blank: True, regex: "^%?[0-9A-Fa-f]{0,4}")	
lata	N/A	True	String(allow_blank: True, regex: "^%?[0-9]{0,5}")	
nar	False	True	Bool	
radius_id	N/A	False	Integer(gt: 0)	id of radius, which will process request. What RADIUS available depends from chosen context.

Default fields:

Name	Type & Comments
tp_lata	String

#	deprecated
tp_ocn	String
tp_capacity	Int
tp_currency	String
tp_dialpeer_name	String
tp_name	String
tp_price	Numeric(16,5)
work_time	String - time of request processing on Radius in microseconds
tp_country	String
tp_dialpeer_id	Int
tp_mc	Int
tp_dest	String
tp_ft	Int
tp_id	Int
position	Int
tp_ci	Int
route	String

2.46 cdr_classes_get

Docstring:

Returns table of accordance call classes with their ids. Call class is determined by protocol, disconnect cause code and call duration.

Example:

- Client Request

```
{"session_id": "1cc0a21e2ee4b723115b5719dfa92df6",  
  "data": {"sort_by": [{"name": ""}],  
          "fields": ["id", "name"]}}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[2, "Completed, successful"],  
           [1, "Completed, unsuccessful"],  
           [7, "Completed, zero"], [3, "Non-routable"],  
           [11, "One second call"], [0, "Other successful"],  
           [4, "Other unsuccessful"], [5, "Routed, misconfiguration possible"],  
           [6, "Routed, uncompleted"], [10, "Wrong Numbers"],  
           [9, "Undesirable calls"], [8, "Unrouted"]],  
  "time_elapsed": 0.00533}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([{"Enum('id', 'name')": Enum('asc', 'desc', '')}], min_length: 0)	defines field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
fields	['id',	True	List([Enum('id',	required fields. If absent - takes fields specified in

	'name']		'name')]], min_length: 0)	"default fields". If empty - returns error.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
id	Int
name	String

2.47 cdr_daily_gen_for_accucore

Docstring:

Creates new request for regeneration CDR's for accucore for specific date and returns request id. You can check request status with help of function 'cdr_daily_gen_get_status'.

User can run this function only from 00:00:00 to 05:59:59 in server's timezone (UTC).

Example:

- Client Request:

```
{
  "session_id": "d12008c5d5684ede64a8f180febba5b5",
  "data": {
    "date": "2013-10-05"
  }
}
```

- Server Response:

```
{
  "status": 200,
  "time_elapsed": 0.047496,
  "message": "OK",
}
```

Request structure and fields description:

```
{'data': {'date': 'String'}, 'session_id': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})\$")	date for daily cdr generation

2.48 cdr_daily_gen_get_status

Docstring:

Function checks CDR generation request status by request id. Request status is available during 12 hours after last status update.

Possible statuses:

- 1) COMPLETED - request completed
- 2) IN PROCESS - request is being processed

Example:

- Client Request:

```
{
  'data': {'request_id': '123'}, 'session_id': 'ewfwm4flwngqlwnf5lnf'
}
```

- Server Response:

Response doesn't contains "data", only "job_status".

```
{
  "job_status": "COMPLETED",
  "status": 200,
  "time_elapsed": 0.047496,
  "message": "OK",
}
```

Request structure and fields description:

```
{'data': {'request_id': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
request_id	N/A	False	String(allow_blank: False)	unique id of gearman submitted job

2.49 cdr_request_add

Docstring:

Creates new CDR request for chosen company, point(s) or spam calls for specific time period.

Function returns id of created request in database.

Example:

- Client Request:

```
{"session_id": "3195e5c6c7fa3490d96dbd26ae523099",  
  "data": {"time_zone": "0:00",  
           "end_date": "2013-10-30T23:59:59",  
           "start_date": "2013-10-30T00:00:00",  
           "point_id": 4519,  
           "fields": ["tp_ip",  
                      "op_name",  
                      "tp_name",  
                      "op_called_number",  
                      "called_number",  
                      "tp_called_number",  
                      "connect_time",  
                      "duration",  
                      "disconnect_time",  
                      "calc_duration_in",  
                      "calc_duration_out",  
                      "disconnect_cause",  
                      "opprice",  
                      "tpprice",  
                      "cost_in",  
                      "cost_out",  
                      "op_ip",  
                      "nas",  
                      "op_dest",  
                      "tp_dest",  
                      "call_id",  
                      "op_protocol",  
                      "tp_protocol",  
                      "src_codec",  
                      "dst_codec",  
                      "dialpeer",  
                      "pdd",  
                      "setup_time",
```

```

        "disconnect_side",
        "clear_number",
        "op_caller_number",
        "termcause",
        "op_traffic_group",
        "tp_caller_number",
        "tp_traffic_group",
        "op_currency",
        "tp_currency",
        "op_origin_country",
        "tp_origin_country"]}]
    }

```

- Server Response:

```

{
    "status": 200,
    "time_elapsed": 0.034496,
    "message": "OK",
}

```

Request structure and fields description:

```

{'data': {'client_id': 'Integer',
          'destination': 'String',
          'email': 'String or String',
          'end_date': 'String',
          'fields': ["Enum('tp_ip', 'op_name', 'tp_name', 'op_called_number', 'called_n
umber', 'tp_called_number', 'connect_time', 'duration', 'disconnect_time', 'calc_durati
on_in', 'calc_duration_out', 'disconnect_cause', 'opprice', 'tpprice', 'cost_in', 'cost
_out', 'op_ip', 'nas', 'op_dest', 'tp_dest', 'call_id', 'op_protocol', 'tp_protocol', '
src_codec', 'dst_codec', 'dialpeer', 'pdd', 'setup_time', 'disconnect_side', 'clear_num
ber', 'op_caller_number', 'termcause', 'op_traffic_group', 'tp_caller_number', 'tp_traf
fic_group', 'op_currency', 'tp_currency', 'op_origin_country', 'tp_origin_country', 'op
_media_ip', 'tp_media_ip')"],
          'include_subcodes': 'Bool',
          'longer_than': 'Integer',
          'point_id': 'Integer',
          'shorter_than': 'Integer',
          'start_date': 'String',
          'time_zone': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optiona l	Type	Description
longer_tha n	N/A	True	Integer(gte: 0)	Uses to export calls data with duration "longer_than" in CDR

				report being generated
client_id	N/A	True	Integer(gt: 0)	company id. If it defined and "point_id" is not - then generates CDR's for all points of given client. For each point will be created individual file.
shorter_than	N/A	True	Integer(gt: 0)	Uses to export calls data with duration "shorter_than" in CDR report being generated
destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	Allows to generate CDR with single destination or destinations included only. Codes should be separated by comma.
include_subcodes	False	False	Bool	Use this option in conjunction with "destination" key. If True CDRs will be generated for "destination" and its subcodes.
point_id	N/A	True	Integer(gte: -2)	point ID. Specify here '-1' if you want to generate cdrs for all origination points of chosen company (if 'client_id' not specified in request then will be generated cdrs for all OPs), '-2' if you want to generate cdrs for all termination points of chosen company (if 'client_id' not specified in request then will be generated cdrs for all TPs), '0' if you want to get cdrs of not authorized calls (in another words "spam

				calls"
fields	['tp_ip', 'op_name', 'tp_name', 'op_called_number', 'called_number', 'tp_called_number', 'connect_time', 'duration', 'disconnect_time', 'calc_duration_in', 'calc_duration_out', 'disconnect_cause', 'opprice', 'tpprice', 'cost_in', 'cost_out', 'op_ip', 'nas', 'op_dest', 'tp_dest', 'call_id', 'op_protocol', 'tp_protocol', 'src_codec', 'dst_codec', 'dialpeer', 'pdd', 'setup_time', 'disconnect_side', 'clear_number', 'op_caller_number', 'termcause', 'op_traffic_group', 'tp_caller_number', 'tp_traffic_group', 'op_currency', 'tp_currency', 'op_origin_country', 'tp_origin_country', 'op_media_ip', 'tp_media_ip']	False	List([Enum('tp_ip', 'op_name', 'tp_name', 'op_called_number', 'called_number', 'tp_called_number', 'connect_time', 'duration', 'disconnect_time', 'calc_duration_in', 'calc_duration_out', 'disconnect_cause', 'opprice', 'tpprice', 'cost_in', 'cost_out', 'op_ip', 'nas', 'op_dest', 'tp_dest', 'call_id', 'op_protocol', 'tp_protocol', 'src_codec', 'dst_codec', 'dialpeer', 'pdd', 'setup_time', 'disconnect_side', 'clear_number', 'op_caller_number', 'termcause', 'op_traffic_group', 'tp_caller_number', 'tp_traffic_group', 'op_currency', 'tp_currency', 'op_origin_country', 'tp_origin_country', 'op_media_ip', 'tp_media_ip')], min_length: 1)	Defines fields to be exported in the cdrs report.
start_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	request start date in format "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-ddhh:mm:ss"
end_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	request end date in format "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-ddhh:mm:ss"

data validation description:

Name	Default value	Optional	Type	Description
request_id	N/A	False	Integer(gte: 0)	id of request that need to remove

2.51 cdr_request_ftp_get

Docstring:

```
This function not in use.
```

Request structure and fields description:

2.52 cdr_request_ftp_set

Docstring:

```
This function not in use.
```

Request structure and fields description:

2.53 cdr_request_get

Docstring:

```
Returns list of created cdr and xdr requests with additional data including files download links for already processed requests if such exist.
```

```
Examples:
```

- Client's request

```
{"session_id": "d12008c5d5684ede64a8f180febbab6",
  "data": {"limit":1,
    "sort_by":[{"request_id":"desc"}],
    "offset":0,
    "fields":["request_id","user_name","email",
      "task_type","request_date",
      "request_status_date","point_id",
      "point_name","longer_than",
      "shorter_than","destination",
      "start_date","end_date","files",
      "cdr_fields","time_zone"]}
}
```

- Server response:

```
{'status': 200,
  'fields': ['request_id', 'user_name', 'email', 'task_type',
    'request_date', 'request_status_date', 'point_id',
    'point_name', 'longer_than', 'shorter_than',
    'destination', 'start_date', 'end_date', 'files',
    'cdr_fields', 'time_zone'],
  'message': 'OK',
  'data': [
    [400, 'dev', 'test@mail.com', 'Generate CDR',
      '2013-10-30T12:23:58.035486+00:00',
      '2013-10-30T12:26:11.764025+00:00', 0, None, None, None,
      None, '2013-10-30T00:00:00', '2013-10-30T23:59:59',
      [['cdr_spam_calls.csv.gz',
        'http://193.165.81.80:8070/cdn/download/cdr/400/
cdr_spam_calls.csv.gz',
        5579]],
      [['tp_ip', 'op_name', 'tp_name', 'op_called_number',
        'called_number', 'tp_called_number', 'connect_time',
        'duration', 'disconnect_time', 'calc_duration_in',
        'calc_duration_out', 'disconnect_cause', 'opprice',
        'tpprice', 'cost_in', 'cost_out', 'op_ip', 'nas',
        'op_dest', 'tp_dest', 'call_id', 'op_protocol', 'tp_protocol',
        'src_codec', 'dst_codec', 'dialpeer', 'pdd',
        'setup_time', 'disconnect_side', 'clear_number',
        'op_caller_number', 'termcause', 'op_traffic_group',
        'tp_caller_number', 'tp_traffic_group', 'op_currency',
        'tp_currency']],
        '+0:00']
    ],
  'time_elapsed': 0.010401}
```

Request structure and fields description:

```
{'data': {'destination': 'String',
  'fields': ["Enum('files', 'task_type', 'end_date', 'request_date', 'type', 'p
oint_id', 'destination', 'request_status_date', 'time_zone', 'start_date', 'companies_l
ist', 'request_id', 'point_name', 'user_name', 'email', 'cdr_fields')"],
  'limit': 'Integer',
  'offset': 'Integer',
  'pt_name': 'String',
  'request_id': 'Integer',
  'service_id': 'Integer',
  'sort_by': [{"Enum('files', 'task_type', 'end_date', 'request_date', 'type',
```



```
'point_id', 'destination', 'request_status_date', 'time_zone', 'start_date', 'companies_list', 'request_id', 'point_name', 'user_name', 'email', 'cdr_fields')": "Enum('asc', 'desc', '')"}]]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
request_id	N/A	True	Integer(gt: 0)	ID of CDR request
sort_by	[]	True	List([Enum('files', 'task_type', 'end_date', 'request_date', 'type', 'point_id', 'destination', 'request_status_date', 'time_zone', 'start_date', 'companies_list', 'request_id', 'point_name', 'user_name', 'email', 'cdr_fields'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted
service_id	N/A	True	Integer(gt: 0)	filter by service type(e.g. voice, sms)
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
pt_name	N/A	True	String(allow_blank: True)	not in use
fields	['files', 'task_type', 'end_date', 'request_date', 'type', 'point_id', 'destination', 'request_status_date', 'time_zone', 'start_date', 'companies_list', 'request_id', 'point_name', 'user_name', 'email', 'cdr_fields']	True	List([Enum('files', 'task_type', 'end_date', 'request_date', 'type', 'point_id', 'destination', 'request_status_date', 'time_zone', 'start_date', 'companies_list', 'request_id', 'point_name', 'user_name', 'email', 'cdr_fields')], min_length: 0)	fields, which will be present in response. If empty - returns fields specified in "default_fields" section
limit	0	True	Integer(gte: 0)	says how many rows from database will be returned in

				response
destination	N/A	True	String(allow_blank: True)	not in use

Default fields:

Name	Type & Comments
files	List(List(String(), String(), Int())) - List that contains file name, download link and file size in bytes.
task_type	String
end_date	String
request_date	String
type	String - voice service or SMS service
point_id	Int (from -2 to +infinity)
destination	String or null
request_status_date	String or Null
time_zone	String
start_date	String
companies_list	List(String(),...)
request_id	Int
point_name	String or null
user_name	String
email	String
cdr_fields	List(List(String))

2.54 change_dests_min_max

Docstring:

Function adds new request 'change min/max digits' to the queue to change 'min_len/max_len' in prices with given destination code (or given code with subcodes) for all OP/TP tariffs.

Example:

- Client Request

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
  "data":{"sub_dest":false,
         "is_tp":true,
         "destination":"1",
         "min_len":11,
         "max_len":12
        }
}
```

Request structure and fields description:

```
{'data': {'destination': 'String',
          'is_tp': 'Bool',
          'max_len': 'Integer',
          'min_len': 'Integer',
          'session_id': 'String',
          'sub_dest': 'Bool'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
max_len	24	True	Integer(gte: 1, lte: 100)	Defines the maximum length of destination code for current price.
sub_dest	False	True	Bool	if True changes 'min_len'/'max_len' digits prices for given destination with sub codes
is_tp	True	True	Bool	if True(is default value) changes 'min_len'/'max_len' only in prices of TP tariffs,

				else - only in prices of OP tariffs
destination	N/A	False	String(allow_blank: False, regex: "^[0-9]**\$")	destination code
min_len	1	True	Integer(gte: 1, lte: 100)	Defines the minimal length of destination code for current price.
session_id	N/A	True	String(allow_blank: False)	session id for logging. By defaults current user session_id will be used.

2.55 check_is_password_expired

Docstring:

Checks user password expiration.

If no user_id in session - returns 400(Bad Request).

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {}
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.56 client_add

Docstring:

Creates new client and balance for this client in the chosen currency.

Function returns id of client that has been added.

Example:

- Client Request

```
{"session_id": "ef46ffbe3042f5d4cfe18db7c2fca805",
 "data": {"email_tech": "st@test.com",
          "email": "st@test.com",
          "invisible": false,
          "name": "H_test",
          "owner_id": 1,
          "balance_currency_id": 1,
          "multicurrency": false,
          "send_traffic_report": true,
          "service_restore_flag": false}}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 11, "time_elapsed": 0.052673}
```

Request structure and fields description:

```
{'data': {'address': 'String',
          'balance_currency_id': 'Integer',
          'description': 'String',
          'email': 'String',
          'email_tech': 'String',
          'ext_id': 'Integer',
          'include_vat': 'Bool',
          'invisible': 'Bool',
          'manager_id': 'Integer',
          'multicurrency': 'Bool',
          'name': 'Name(at least one printable character). max_length=None',
          'owner_id': 'Integer',
          'phone': 'String',
          'routes_changes_report': 'Bool',
          'send_traffic_report': 'Bool',
          'service_restore_flag': 'Bool',
          'services_ids': ['Integer'],
          'vat_number': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description

invisible	False	True	Bool	option that makes company and all points assigned to the company invisible and disabled
multicurrency	False	True	Bool	If 'True' allows company to have several balances in different currencies
ext_id	N/A	True	Integer()	not in use
email	N/A	False	String(allow_blank: True, regex: ".+@.+\.[a-z]+")	billing email address for sending invoices, notification about closure points of the company and balance notifications. A few addresses need separate by comma.
email_tech	""	False	String(allow_blank: False, regex: ".+@.+\.[a-z]+")	technical email address of the company for Guardian Trouble Tickets and also notification about closure points of the company and balance notifications in case if billing email not specified. A few addresses need separate by comma.
services_ids	[1]	True	List([Integer(gt: 0)], min_length: 0)	list of services(e.g. voice, sms)
vat_number	""	True	String(allow_blank: True)	
phone	N/A	True	String(allow_blank: True)	phone number of the company
address	N/A	True	String(allow_blank: True)	legal or physical address of the company
owner_id	N/A	False	Integer(gt: 0)	id of the owner
send_traffic_report	False	True	Bool	If True allows to send short traffic report email
balance_currency_id	N/A	False	Integer()	id of the balance currency
description	N/A	True	String(allow_blank: True)	some information about company
name	N/A	False	Name(at least one printable character).	company name

			max_length=None	
include_vat	False	True	Bool	if "true" VAT will be added to the amount
routes_changes_report	False	True	Bool	if "true" Routes Changes Report will be sent to Client manager
manager_id	N/A	True	Integer(gt: 0)	user's id who is manager of the company
service_restore_flag	False	True	Bool	enables service restore functionality(reopens negative balances when they become positive)

2.57 client_get

Docstring:

Shows information about given companies. Function returns companies locked by chosen user if 'user_id' specified in request.

Example:

- Client Request

```
{
  "session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {
    "id": 1,
    "fields": [
      "name", "description", "invisible", "email",
      "phone", "address", "email_tech", "manager_id",
      "multicurrency", "owner_id", "balance_id",
      "send_traffic_report", "service_restore_flag"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "name", "description", "invisible", "email", "phone",
    "address", "email_tech", "manager_id", "multicurrency",
    "owner_id", "balance_id", "send_traffic_report",
    "service_restore_flag"
  ],
  "message": "OK",
  "data": [
    [
      "Test_client", "Main Test Company", false,
      "test@test.com", "test1@test.com",
      "065-82-79-481", "Ukraine.
      Kharkov city.",
      "tech_email@test.com", 1, false, 1, 1, false,
      false
    ]
  ],
  "time_elapsed": 0.006684
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'currency_id': 'Integer',
          'ext_id': 'Integer',
          'fields': ["Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag',
'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'i
nvisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currenc
y_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id',
'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_i
d', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')"],
          'id': 'Integer',
          'is_accucore': 'Bool',
          'limit': 'Integer',
          'manager_id': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'send_traffic_report': 'Bool',
          'services_ids': ['Integer'],
          'show_invisible': 'Bool or Null',
          'sort_by': [{"Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag'
, 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address',
'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'curre
ncy_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id
', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manage
r_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')": "Enum('asc', 'desc
', '')"}]},
          'user_id': 'Integer',
          'vat_number': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optiona l	Type	Description
sort_by	[]	True	List([{'Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note',	field on which result sorts

			'owner_id'): Enum('asc', 'desc', ")}], min_length: 0)	
id	N/A	True	Integer(gt: 0)	id of company
balance_id	N/A	True	Integer(gt: 0)	balance id to get client's id if "id" not provided
fields	['email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id']	True	List([Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
currency_id	N/A	True	Integer(gt: 0)	Allows to see companies with specific currency
user_id	N/A	True	Integer(gt: 0)	this filter allows to show companies locked to the given user.
name	N/A	True	String(allow_blank: True)	company name or first chars of name
ext_id	N/A	True	Integer(gt: 0)	client id for external billing.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip

				that many rows before beginning to return rows.
manager_id	N/A	True	Integer(gt: 0)	filter by manager
is_accucore	False	True	Bool	special parameter for accucore, to show all clients (visible and invisible)
show_invisible	False	True	Or(Bool, Null)	Allows to display only visible (if False) or only invisible (if True) companies in report. If this option is absent in request returns visible and invisible companies. In conjunction with company id this option doesn't have influence with response.
send_traffic_report	N/A	True	Bool	If True allows to send short traffic report email
owner_id	N/A	True	Integer(gt: 0)	filter by owner
services_ids	[1]	True	List([Integer(gt: 0)], min_length: 0)	filter by service type, by default service type is 1(voice)
vat_number	N/A	True	String(allow_blank: True)	client VAT number

Default fields:

Name	Type & Comments
------	-----------------

email_tech	String
owner_name	String
description	String
tp_limit_flag	Boolean
multicurrency	Boolean
tp_limit	String
phone	String
actual_vat_balance	String
balance_id	Int (the id of company's balance)
address	String
invisible	Boolean
vat_number	String
short_currency	String
alarm_limit	String
id	Int (company id)
notify_flag	Boolean
currency_id	Int (id of company's currency)
send_traffic_report	Boolean
service_restore_flag	Boolean
name	String (company name)
services_ids	Array of integers
ext_id	Int (external id of company for synchronization with external billing,

	deprecated)
routes_changes_report	Boolean
limit_flag	Boolean
alarm_flag	Boolean
include_vat	Boolean
limit	String
manager_id	Int or null
alarm_limit_unit	Int (1-default currency, 2-%)
balance	String
email	String (billing emails of company)
note	String
owner_id	Int (id of owner assigned to the company)

2.58 client_note_add

Docstring:

Adds note record for given Client.

Example:

- Client Request

```
{
  "session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {"client_id": 385, "note": "Test Client Note"}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 11, "time_elapsed": 0.006684}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer', 'note': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
note	""	True	String(allow_blank: True, max_length: 70)	Client note text
client_id	N/A	False	Integer(gt: 0)	ID of client

2.59 client_notes_history_get

Docstring:

Returns list of notes for given Client.

Example:

- Client Request

```
{
  "session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {
    "client_id": 391,
    "fields": ["note", "date_of_change"],
    "limit": 11,
    "offset": 0,
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "fields": ["note", "date_of_change"],
  "data": [
    [u'Note number 49', u'2018-12-31T00:49:00.797731'],
    [u'Note number 48', u'2018-12-31T00:48:00.797731'],
    [u'Note number 47', u'2018-12-31T00:47:00.797731'],
    [u'Note number 46', u'2018-12-31T00:46:00.797731'],
    [u'Note number 45', u'2018-12-31T00:45:00.797731'],
    [u'Note number 44', u'2018-12-31T00:44:00.797731'],
  ]
}
```

```

        [u'Note number 43', u'2018-12-31T00:43:00.797731'],
        [u'Note number 42', u'2018-12-31T00:42:00.797731'],
        [u'Note number 41', u'2018-12-31T00:41:00.797731'],
        [u'Note number 40', u'2018-12-31T00:40:00.797731'],
        [u'Note number 39', u'2018-12-31T00:39:00.797731']
    ],
    "time_elapsed": 0.006684
}

```

Request structure and fields description:

```

{'data': {'client_id': 'Integer',
          'fields': ["Enum('note', 'date_of_change')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('note', 'date_of_change')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	ID of client
fields	['enabled', 'date_of_change']	True	List([Enum('note', 'date_of_change')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([{"Enum('note', 'date_of_change')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result sorts

Default fields:

Name	Type & Comments
------	-----------------

note	String (70)
date_of_change	String

2.60 client_point_get

Docstring:

Function returns data set about points assigned to the chosen client(s). Besides the data mentioned in the 'default fields' this function always returns for each requested company:

- * list of OPs assigned to the client and their ids. Example: [[4361, "first_op"], [4387, "second_op"]]
- * list of TPs assigned to the client with their ids. Example: [[4774, "first_tp"], [4668, "second_tp"]]
- * list of client's balances and their ids. Example: [[535, "second_balance (EUR)"], [538, "first_balance (USD)"]]

Example:

- Client Request

```
{"session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {"limit": 101,
           "show_invisible_points": false,
           "show_visible_points": true,
           "show_invisible_companies": false,
           "sort_by": [{"name": ""}],
           "name": "",
           "offset": 0,
           "show_visible_companies": true,
           "fields": ["id", "name", "account_manager"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name", "account_manager"],
  "message": "OK",
  "data": [[1, "Test_client", "dev", [[3, "Test_1_OP"]], [],
           [[1, "Test_balance (USD)"]]],
           [[2, "Test_supplier", "dev", [], [[4, "Test_1_TP"]],
           [[2, "Test_supplier_balance (USD)"]]]],
  "time_elapsed": 0.006684}
```

Request structure and fields description:

```
{'data': {'currency_id': 'Integer',
          'fields': ["Enum('account_manager', 'owner_name', 'id', 'name')"],
          'ids': ['Integer'],
          'limit': 'Integer',
          'name': 'String',
```

```

'offset': 'Integer',
'show_invisible_companies': 'Bool',
'show_invisible_points': 'Bool',
'show_sms_points': 'Bool',
'show_visible_companies': 'Bool',
'show_visible_points': 'Bool',
'show_voice_points': 'Bool',
'sort_by': [{"Enum('account_manager', 'owner_name', 'id', 'name')": "Enum('as
c', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
limit	0	True	Integer(gte: 0)	says how many rows will be in response
name	N/A	True	String(allow_blank: True)	company name or first chars of company name
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
currency_id	N/A	True	Integer(gt: 0)	Allows to returns in report only companies with specific currency
show_sms_points	True	True	Bool	if True - sms points will be in th result
ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	companies ids
show_visible_points	True	True	Bool	
show_invisible_points	False	True	Bool	
show_voice_points	True	True	Bool	if True - voice points will be in th result

sort_by	[]	True	List([Enum('account_manage r', 'owner_name', 'id', 'name'): Enum('asc', 'desc', ")"}], min_length: 0)	field on which result sorts
fields	['account_manage r', 'owner_name', 'id', 'name']	True	List([Enum('account_manager' , 'owner_name', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
show_visible_ companies	True	True	Bool	
show_invisibl e_companies	False	True	Bool	

Default fields:

Name	Type & Comments
account_manager	String or null
owner_name	String - name of owner which assigned to the client
id	Int - Client's id
name	String - Client's name

2.61 client_set

Docstring:

If 'limit' is absent in request then calls function "_client_set_general_settings" which allows to change general settings of the given client.
Otherwise calls function "_client_set_financial_settings" that changes financial settings of the client.
To make changes user should have permissions to edit company parameters i.e. client have to be present in the list of allowed companies of the current user.

Parameter 'id' should be present in query - otherwise error will be raised.

Function returns id of client that was modified.

`_client_set_general_settings` ([data_validation](#)) ([default_fields](#))

Allows to change general settings of the given company: emails, address and so on

Example:

- Client Request

```
{
  "session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {
    "id": 1,
    "phone": "067-81-79-481",
    "email_tech": "tech_email@speedflow.com",
    "address": "Ukraine.
Kharkov city.",
    "email": "test@test.com",
    "manager_id": 1,
    "invisible": false,
    "description": "Main Test Company",
    "name": "Test_client",
    "owner_id": 1,
    "multicurrency": false,
    "send_traffic_report": true
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 1,
  "time_elapsed": 0.060193
}
```

`_client_set_financial_settings` ([data_validation](#)) ([default_fields](#))

Allows to change financial settings of the given company such as balance limits and balance notification settings.

If maximum negative balance limit ("limit") specified in query and it different from previous value was then creates email notification. Function fills mail fields 'to' and 'copy' with emails specified in the template "email_minimum_balance_change" if any or with client's billing email. If the template's subject is empty or template is turned off then notifications are disabled.

Example:

- Client Request

```
{
  "session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {
    "limit": -10000.0,
    "id": 1,
    "currency_id": 1,
    "notify_flag": true,
    "alarm_limit": 10.0,
  }
}
```

```
"alarm_flag":true,
"limit_flag":true,
"service_restore_flag":false}}
```

- Server response

```
{"status": 200, "message": "OK", "id": 1, "time_elapsed": 0.042481}
```

Request structure and fields description:

_client_set_general_settings

```
{'data': {'address': 'String',
'description': 'String',
'email': 'String',
'email_tech': 'String',
'id': 'Integer',
'include_vat': 'Bool',
'invisible': 'Bool',
'manager_id': 'Integer',
'multicurrency': 'Bool',
'name': 'Name(at least one printable character). max_length=None',
'owner_id': 'Integer',
'phone': 'String',
'routes_changes_report': 'Bool',
'send_traffic_report': 'Bool',
'services_ids': ['Integer'],
'vat_number': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
email	N/A	True	String(allow_blank: True, regex: ".+@.+\\.[a-z]+")	company billing department email(s) for sending notifications about balance state and points closure. More than one email should separate by comma.
multicurrency	N/A	True	Bool	if 'True' this option allows company to have several balances in different currencies
manager_id	N/A	True	Integer(gt: 0)	id of user which assigned to the company as manager
id	N/A	False	Integer(gt: 0)	company id

email_tech	N/A	True	String(allow_blank: True, regex: ".+@.+\\.[a-z]+")	company technical department email(s) for sending guardian notification. More than one email should separate by comma.
name	N/A	True	Name(at least one printable character). max_length=None	filter by company name
send_traffic_report	N/A	True	Bool	If True allows to send short traffic report email
vat_number	N/A	True	String(allow_blank: True)	
owner_id	N/A	True	Integer(gt: 0)	
routes_changes_report	N/A	True	Bool	If True allows to send Routes Changes Report email to Client manager
description	N/A	True	String(allow_blank: True)	any information about company
phone	N/A	True	String(allow_blank: True)	
services_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	list of services(e.g. voice, sms)
include_vat	N/A	True	Bool	if "true" VAT will be added to the amount
invisible	N/A	True	Bool	If value of this option is 'True' then sets company invisible. All points assigned to the company automatically become invisible and disabled.
address	N/A	True	String(allow_blank: True)	

_client_set_financial_settings

```
{'data': {'alarm_flag': 'Bool',
          'alarm_limit': 'String',
          'alarm_limit_unit': 'Integer',
          'currency_id': 'Integer',
```

```

'id': 'Integer',
'limit': 'String',
'limit_flag': 'Bool',
'notify_flag': 'Bool',
'service_restore_flag': 'Bool',
'tp_limit': 'String',
'tp_limit_flag': 'Bool'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
alarm_limit	N/A	False	Numeric(precision: 21, scale: 5, gte: 0)	difference between current balance and value specified in "limit" when alarm about balance state will be generated
alarm_limit_unit	1	True	Integer(gt: 0)	alarm limit unit type (1-default currency, 2-%)
limit_flag	N/A	False	Bool	if True - company origination points will be closed if balance reaches "limit" value
alarm_flag	N/A	False	Bool	if True this option enables sending "alarm_limit" notification to the client
service_restore_flag	False	True	Bool	enables service restore functionality(reopens negative balances when they become positive)
id	N/A	False	Integer(gt: 0)	company id
limit	N/A	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (positive values allowed)
tp_limit_flag	False	False	Bool	if True - company termination points will be closed if balance reaches "tp_limit" value
tp_limit	0	False	Numeric(precision: 21, scale: 5)	maximum negative balance limit (positive values allowed)
currency_id	N/A	False	Integer()	company currency id for "alarm_limit", "limit" values and currency in which company balance will be shown

notify_flag	N/A	True	Bool	enables sending notification to client about closure of points when balance reached "limit" value
-------------	-----	------	------	---

2.62 client_vat_history_get

Docstring:

Shows information about VAT change history for given company.

Example:

- Client Request

```
{"session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {"client_id": 385, "fields": ["enabled", "date_of_change"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["enabled", "date_of_change"],
  "message": "OK",
  "data": [[false, '2017-10-27T05:08:14.115171'],
            [true, '2017-10-27T09:10:21.125230']],
  "time_elapsed": 0.006684}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer',
          'fields': ["Enum('enabled', 'date_of_change')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('enabled', 'date_of_change')": "Enum('asc', 'desc', '')"}]},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

fields	['enabled', 'date_of_change']	True	List([Enum('enabled', 'date_of_change')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([Enum('enabled', 'date_of_change'): Enum('asc', 'desc', "")], min_length: 0)	field on which result sorts
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
client_id	N/A	False	Integer(gt: 0)	id of company

Default fields:

Name	Type & Comments
enabled	Boolean (vat flag state for date)
date_of_change	String

2.63 codecs_get

Docstring:

Returns table of accordance voip codecs presented in mediacore database and the ir ids.

Example:

- Client Request

```
{
  "session_id": "058091ccbed1662aa1dfb32af40b539d",
  "data": {
    "sort_by": [{"name": ""}],
    "fields": ["id", "name"]}
}
```

- Server response

```
{
  "status": 200,
  "fields": ["id", "name"],
}
```

```

"message": "OK",
"data": [[7, "G.711-ALaw"], [8, "G.711-uLaw"], [2, "G.723.1"],
        [6, "G.723.1C"], [1, "G.729"], [3, "G.729A"],
        [5, "G.729A/B"], [4, "G.729B"], [9, "GSM FR"]],
"time_elapsed": 0.005184}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('spec_name', 'id', 'name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'protocol': 'Integer',
          'sort_by': [{"Enum('spec_name', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
protocol	N/A	True	Integer(gt: 0)	filter by protocol id, 1-sip, 2-h323
fields	['spec_name', 'id', 'name']	True	List([Enum('spec_name', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('spec_name', 'id', 'name')": Enum('asc', 'desc', '')}], min_length: 0)	field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
spec_name	String

id	Int
name	String

2.64 config_get

Docstring:

Returns current config data: connection price for whole system, accucore status (present or not in configuration of the system) and current routing mode. Routing mode affects to all users of the system.

Example:

- Client Request

```
{
  "session_id": "2d5272e49ca2c24ed18e8f9b00aa9e86",
  "data": {
    "fields": [
      "simplified",
      "dest_deny_limit"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "simplified": false,
    "dest_deny_limit": 125
  },
  "time_elapsed": 0.00362
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': [
      'Enum('link_template', 'start_page_top_targets_email_from', 'fin_sum_period_dest_and_price', 'tp_dest_capacity_count_limit', 'dumpers_signaling', 'a_numbers_modification_by_password_enabled', 'dumpers_media', 'simplified', 'loop_detection', 'fin_sum_period_total_points', 'start_page_top_targets_email_cc', 'fin_sum_period_managers', 'function_default_timeout', 'interval_from_cdr', 'tp_dest_capacity_zero', 'start_page_sms_top_offers_email_from', 'fin_sum_period_countries', 'service_dp_id', 'start_page_top_offers_email_from', 'aggregation_cron_time', 'tp_deviation', 'wrong_login_attempts', 'fin_sum_max_period', 'sms_stat_individual', 'connection_price', 'start_page_sms_top_offers_email_to', 'a_number_replacement', 'start_page_sms_top_targets_email_cc', 'start_page_sms_top_targets_email_to', 'start_page_sms_top_offers_email_subject', 'session_ttl', 'sms_stat_from_aggregation', 'start_page_top_targets_email_to', 'fin_sum_period_individual', 'start_page_sms_top_targets_email_subject', 'fin_sum_period_companies', 'capacity', 'accucore', 'max_call_time', 'dst_deny_limit', 'sms_stat_from_cdr', 'anumbers_limit', 'service_sms_dp_id', 'start_page_top_offers_email_cc', 'start_page_top_targets_email_subject', 'start_page_top_offers_email_to', 'start_page_sms_top_targets_email_from', 'start_page_top_offers_email_subject', 'guardian_advanced_mode', 'fin_sum_period_dest', 'fin_sum_period_mcc_mnc', 'start_page_sms_top_offers_email_cc', 'dp_capacity_default', 'function_timeouts', 'ab_routing_enabled', 'use_fw_whitelist')"]
    ]
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['link_template', 'start_page_top_targets_email_from', 'fin_sum_period_dest_and_price', 'tp_dest_capacity_count_limit', 'dumpers_signaling', 'a_numbers_modification_by_pass_word_enabled', 'dumpers_media', 'simplified', 'loop_detection', 'fin_sum_period_total_points', 'start_page_top_targets_email_cc', 'fin_sum_period_managers', 'function_default_timeout', 'interval_from_cdr', 'tp_dest_capacity_zero', 'start_page_sms_top_offers_email_from', 'fin_sum_period_countries', 'service_dp_id', 'start_page_top_offers_email_from', 'aggregation_cron_time', 'tp_deviation', 'wrong_login_attempts', 'fin_sum_max_period', 'sms_stat_individual', 'connection_price', 'start_page_sms_top_offers_email_to', 'a_number_replacement', 'start_page_sms_top_targets_email_cc', 'start_page_sms_top_targets_email_to', 'start_page_sms_top_offers_email_subject', 'session_ttl', 'sms_stat_from_aggregation', 'start_page_top_targets_email_to', 'fin_sum_period_individual', 'start_page_sms_top_targets_email_subject', 'fin_sum_period_companies', 'capacity', 'accucore', 'max_call_time', 'dst_deny_limit', 'sms_stat_from_cdr', 'anumbers_limit', 'service_sms_dp_id', 'start_page_top_offers_email_cc', 'start_page_top_targets_email_subject',	True	List([Enum('link_template', 'start_page_top_targets_email_from', 'fin_sum_period_dest_and_price', 'tp_dest_capacity_count_limit', 'dumpers_signaling', 'a_numbers_modification_by_pass_word_enabled', 'dumpers_media', 'simplified', 'loop_detection', 'fin_sum_period_total_points', 'start_page_top_targets_email_cc', 'fin_sum_period_managers', 'function_default_timeout', 'interval_from_cdr', 'tp_dest_capacity_zero', 'start_page_sms_top_offers_email_from', 'fin_sum_period_countries', 'service_dp_id', 'start_page_top_offers_email_from', 'aggregation_cron_time', 'tp_deviation', 'wrong_login_attempts', 'fin_sum_max_period', 'sms_stat_individual', 'connection_price', 'start_page_sms_top_offers_email_to', 'a_number_replacement', 'start_page_sms_top_targets_email_cc', 'start_page_sms_top_targets_email_to', 'start_page_sms_top_offers_email_subject', 'session_ttl', 'sms_stat_from_aggregation', 'start_page_top_targets_email_to', 'fin_sum_period_individual', 'start_page_sms_top_targets_email_subject', 'fin_sum_period_companies', 'capacity', 'accucore', 'max_call_time', 'dst_deny_limit', 'sms_stat_from_cdr', 'anumbers_limit', 'service_sms_dp_id', 'start_page_top_offers_email_cc', 'start_page_top_targets_email_subject',	required fields. If absent - takes fieldsspecified in "default fields". If empty - returns error.

ect', 'start_page_top_offers_email_to', 'start_page_sms_top_targets_email_from', 'start_page_top_offers_email_subject', 'guardian_advanced_mode', 'fin_sum_period_dest', 'fin_sum_period_mcc_mnc', 'start_page_sms_top_offers_email_cc', 'dp_capacity_default', 'function_timeouts', 'ab_routing_enabled', 'use_fw_whitelist']	'start_page_top_offers_email_to', 'start_page_sms_top_targets_email_from', 'start_page_top_offers_email_subject', 'guardian_advanced_mode', 'fin_sum_period_dest', 'fin_sum_period_mcc_mnc', 'start_page_sms_top_offers_email_cc', 'dp_capacity_default', 'function_timeouts', 'ab_routing_enabled', 'use_fw_whitelist']], min_length: 0)
---	---

Default fields:

Name	Type & Comments
link_template	String - template of link to download
start_page_top_targets_email_from	String - Start Page top targets email from address
fin_sum_period_dest_and_price	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
tp_dest_capacity_count_limit	Integer - max amount of TP dest capacity rules
dumpers_signaling	List of Strings- IP address of dumper for calls by host (dumps with RTP packets). If not defined - empty String. (aka dumper_one_host)
a_numbers_modification_by_password_enabled	Boolean - True if numbersDB modification password is set
dumpers_media	List of Strings - IP address of dumper where all calls dumps store. If not defined - empty String. (aka dumper_all)
simplified	Boolean - state of simple/advanced mode toggle (for web interface)
loop_detection	Boolean - is loop detection enabled
fin_sum_period_total_points	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client

start_page_top_targets_email_cc	List of Strings - Start Page top targets email copy addresses
fin_sum_period_managers	String - if "fin_sum_period_managers" haven't been exceeded and this period is exceeded returns fin summary report only for the first manager and managers list
function_default_timeout	Float - default timeout for API functions (seconds)
interval_from_cdr	String - maximal allowed interval in minutes for incoming/outgoing calls requests with enabled option "by_cdr"
tp_dest_capacity_zero	Boolean - is zero capacity value allowed
start_page_sms_top_offers_email_from	String - Start Page SMS top offers email from address
fin_sum_period_countries	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
service_dp_id	id of service dialpeer (for all contexts)
start_page_top_offers_email_from	String - Start Page top offers email from address
aggregation_cron_time	String - minute(s) of hour when cron starts calculation of hourly calls aggregation
tp_deviation	Boolean - is TP max_call_time_deviation enabled
wrong_login_attempts	Int - number of unsuccessful login attempts after which user will be banned
fin_sum_max_period	String - maximal allowed interval for all types of fin summary report in days
sms_stat_individual	String - max allowed interval for individual incoming/outgoing SMS report in days
connection_price	String - is connection price in use ("true" or "false")
start_page_sms_top_offers_email_to	List of Strings - Start Page SMS top offers email to addresses
a_number_replacement	Boolean - is a-number replacement enabled

start_page_sms_top_targets_email_cc	List of Strings - Start Page SMS top targets email copy addresses
start_page_sms_top_targets_email_to	List of Strings - Start Page SMS top targets email to addresses
start_page_sms_top_offers_email_subject	String - Start Page SMS top offers email subject
session_ttl	Int - API session TTL (after last user activity) in seconds. By default is 7200 seconds.
sms_stat_from_aggregation	String - max allowed interval for aggregated SMS incoming/outgoing report (takes data from aggregation) in days
start_page_top_targets_email_to	List of Strings - Start Page top targets email to addresses
fin_sum_period_individual	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
start_page_sms_top_targets_email_subject	String - Start Page SMS top targets email subject
fin_sum_period_companies	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
capacity	String - default value of point "capacity" when point is being created
accucore	String - is accucore in use ("yes" or "no")
max_call_time	String - default value of "max_call_time" in seconds when point is being created
dst_deny_limit	Int - max length for 'dst_regexp', 'deny_regexp' fields in dialpeer (default value=128)
sms_stat_from_cdr	String - max allowed interval for aggregated SMS incoming/outgoing report (takes data from CDR's) in minutes
anumbers_limit	Int - is a-numbers upper threshold
service_sms_dp_id	id of SMS service dialpeer (for all contexts)
start_page_top_offers_email	List of Strings - Start Page top offers email copy addresses

_cc	
start_page_top_targets_email_subject	String - Start Page top targets email subject
start_page_top_offers_email_to	List of Strings - Start Page top offers email to addresses
start_page_sms_top_targets_email_from	String - Start Page SMS top targets email from address
start_page_top_offers_email_subject	String - Start Page top offers email subject
guardian_advanced_mode	Boolean - display guardian advanced option on web
fin_sum_period_dest	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
fin_sum_period_mcc_mnc	String - if "fin_sum_max_period" haven't been exceeded and this period is exceeded returns fin summary report only for first client
start_page_sms_top_offers_email_cc	List of Strings - Start Page SMS top offers email copy addresses
dp_capacity_default	String - default value of dp "capacity" when dp is being created
function_timeouts	Dict - specific timeout for API functions
ab_routing_enabled	Boolean - is AB Routing enabled
use_fw_whitelist	Boolean - use firewall whitelist instead of blacklist

2.65 config_set

Docstring:

Enables or disables simplified routing mode. You should remember that changing routing mode will change it for all users of web panel.

Example:

- Client Request

```
    {"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",  
      "data":{"simplified":true}  
    }  
  }
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.024296}
```

Request structure and fields description:

```
{'data': {'simplified': 'Bool'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
simplified	N/A	True	Bool	if "true" - sets "Simple Routing" mode,"false" - "Advanced Routing" mode

2.66 contexts_get

Docstring:

Returns information about contexts existent in the database: context namespace, status.

Example:

- Client Request

```
{"session_id":"058091ccbed1662aa1dfb32af40b539d",  
  "data":{"sort_by":[{"name":""}],  
          "fields":["id","name"]}}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[1, "routing_cz"]],  
  "time_elapsed": 0.005381}
```

Request structure and fields description:

```
{'data': {'enabled': 'Bool',
          'fields': ["Enum('description', 'short_name', 'namespace', 'enabled', 'id', 'name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('description', 'short_name', 'namespace', 'enabled', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
enabled	True	True	Bool	filter by context status. If "True" - returns data only for enabled contexts
fields	['description', 'short_name', 'namespace', 'enabled', 'id', 'name']	True	List([Enum('description', 'short_name', 'namespace', 'enabled', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('description', 'short_name', 'namespace', 'enabled', 'id', 'name'): Enum('asc', 'desc', '')]), min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
description	String
short_name	String

namespace	String
enabled	Boolean
id	Int
name	String

2.67 correct_balance_by_paypal

Docstring:

This function designed to add funds to the balance of chosen client using paypal.
 Call this function after you make payment to the mediacore owner paypal account.
 Specified in request amount will be added to the company balance.

Example:

- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {
    "payment_fee": 0.33,
    "balance_id": 545,
    "payment_amount": 1.0,
    "transaction_id": "11111",
    "user_id": 7
  }
}
```

Request structure and fields description:

```
{
  'data': {
    'balance_id': 'Integer',
    'client_id': 'Integer',
    'payment_amount': 'Float',
    'payment_fee': 'Float',
    'session_id': 'String',
    'transaction_id': 'String',
    'user_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optiona	Type	Description
------	---------------	---------	------	-------------

		I		
balance_id	N/A	True	Integer(gt: 0)	
payment_amount	N/A	False	Float(gt: 0)	
user_id	N/A	False	Integer(gt: 0)	
session_id	N/A	True	String(allow_blank: False)	By defaults from user's session
transaction_id	N/A	False	String(allow_blank: False)	paypal transaction id
client_id	N/A	True	Integer(gt: 0)	
payment_fee	N/A	False	Float(gt: 0)	paypal fee for transaction

2.68 countries_get

Docstring:

Returns country names by first digits of country code or first symbols of country name.

Example:

1) get country names by first symbols of country name

-Client Request

```
{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "limit": 11,
    "country_or_dstcode": "Uk",
    "sort_by": [{"country": "asc"}],
    "offset": 0,
    "fields": ["country"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["country"],
  "message": "OK",
  "data": [
```

```

["Ukraine Cherkasy H3g"], ["Ukraine"], ["UKRAINE"], ["Ukraine Cherkasy"],
["Ukraine Cherkasy Zone 1"],
["Ukraine Chernigov"], ["Ukraine Chernigov Zone 2"], ["Ukraine Cher
novtsy"],
["Ukraine Dnepropetrovsk"], ["Ukraine Dnepropetrovsk Zone 1"],
["Ukraine Dnepropetrovsk Zone 2"]
],
"time_elapsed": 0.449245
}

```

2) get codes by first digits of destination code

- Client Request

```

{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "limit": 11,
    "country_or_dstcode": "13",
    "sort_by": [{"country": "asc"}],
    "offset": 0,
    "fields": ["country"]
  }
}

```

- Server Response

```

{
  "status": 200,
  "fields": ["country"],
  "message": "OK",
  "data": [
    ["American Virgin Islands"], ["Canada"], ["Canada Ontario"], ["Cana
da Saskatchewan"],
    ["Canada Zone 1"], ["Cayman Islands"], ["Cayman Islands mobile"],
    ["Cayman Islands mobile Digicel"], ["CAYMAN ISLANDS MOBILE DIGICEL"
], ["Usa"], ["Usa Alabama"]
  ],
  "time_elapsed": 0.443707
}

```

Request structure and fields description:

```

{'data': {'country_or_dstcode': 'String',
          'fields': ["Enum('country', 'network_code', 'dstcode')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('country', 'network_code', 'dstcode')": "Enum('asc', 'desc
', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
country_or_dstcode	N/A	True	String(allow_blank: True)	first digits of country code or first symbols of country name for service_id = 1, first symbols of country name or first digits of network code for service_id = 2
service_id	1	True	Integer(gt: 0, lt: 3)	filter by service id. 1 - Voice service, 2 - SMS service
sort_by	[]	True	List([Enum('country', 'network_code', 'dstcode'): Enum('asc', 'desc', ")]), min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"country": "asc"}] - sorts result ascending by field "country".
limit	0	True	Integer(gte: 0)	says how many rows will be in response
fields	['country', 'network_code', 'dstcode']	True	List([Enum('country', 'network_code', 'dstcode')]), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
country	String
network_code	String
dstcode	String

2.69 countries_regex_get

Docstring:

Function returns accordance of counties and allow/deny regexp.

These regular expressions can be used in dialpeer settings for "dstregexp" and "denyregexp" fields respectively.

Example:

- Client Request:

```
{
  "session_id": "ac2d9bc02a820409db8f0de6e9c8c812",
  "data": {"fields": ["country", "allow_regexp", "deny_regexp"]}
}
```

- Server Response

```
{
  "data": [
    ["Ukraine", "^(380).*", ""],
    ["Canada", "^(1873|1236|1250|1365|1418|1431|1437|1450|1519|1639|1705|1807|1819).*", ""],
    ...
  ],
  "fields": ["country", "allow_regexp", "deny_regexp"]
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('country', 'deny_regexp', 'allowed_regexp')"]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['country', 'deny_regexp', 'allowed_regexp']	True	List([Enum('country', 'deny_regexp', 'allowed_regexp')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
country	String

deny_regexp	String
allowed_regexp	String

2.70 country_code_add

Docstring:

Add a single country code record to the database (or updates existing one).

Example:

-Client Request

```
{
  "session_id": "ac2d9bc02a820409db8f0de6e9c8c812",
  "data": {
    "dstcode": "8801",
    "country": "Argentina San Juan"
  }
}
```

Request structure and fields description:

```
{'data': {'country': 'String', 'dstcode': 'String', 'is_mobile': 'Bool'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
country	N/A	False	String(allow_blank: False)	country name corresponding to the destination code
is_mobile	False	True	Bool	currently not in use
dstcode	N/A	False	String(allow_blank: False, regex: "[0-9#\\[*]*\$")	destination code

2.71 country_code_bulk_add

Docstring:

Adds a batch of new country codes records to the database (or updates existing ones)

Use this method when you need to add a large amount of records at once.

If the same destination code present at more than once - function will return error.

If "replace_all_codes" and "renew_all_names" are False - then function only adds new codes.

Only "replace_all_codes" or "renew_all_names" should be True.

Example:

-Client Request - add only new codes

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
  "data":{"country_codes_data":
    [[["919764","India mobile Vodafone "],
      ["919765","India mobile Vodafone"],
      ["919769","India mobile Vodafone"],
      ["919772","India mobile Vodafone"],
      ["6286","Indonesia mobile 1"],
      ["6288","Indonesia mobile 2"],
      ["6285","Indonesia mobile 3"]],
    "replace_all_codes":false,
    "renew_all_names":false}
}
```

Request structure and fields description:

```
{'data': {'country_codes_data': [['String', 'String']],
  'renew_all_names': 'Bool',
  'replace_all_codes': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
replace_all	N/A	False	Bool	if True - deletes all existed entries

_codes				from the database and uploads does not existent ones from the "country_codes_data"
country_codes_data	N/A	False	List([List([Destination(allow_blank: False, _dest_pattern: ^\d*\$), String(allow_blank: False)], 2)], min_length: 0)	List of country codes and corresponding country names
renew_all_names	N/A	False	Bool	if True updates existed codes and adds new from the "country_codes_data"

2.72 country_code_bulk_delete

Docstring:

Removes all country codes from list.
Country code name shouldn't obligatory to match the name of corresponding code in the database.

Example:

-Client Request

```
{
  "session_id": "ac2d9bc02a820409db8f0de6e9c8c812",
  "data": {
    "country_codes_data": [
      ["8801", "q"],
      ["542644", "Argentina San Juan"]
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "cnt": 2,
  "time_elapsed": 0.060193
}
```

Request structure and fields description:

```
{
  'data': {
    'country_codes_data': [['String', 'String']],
    'session_id': 'String'
  }
}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
country_codes_data	N/A	False	List(List(Destination(allow_blank: False, _dest_pattern: ^\d*\$), String(allow_blank: False)), 2), min_length: 0)	List of country codes and corresponding country names

2.73 country_codes_del

Docstring:

Deletes specified destination codes records from database.

Example:

-Client Request

```
{"session_id": "ac2d9bc02a820409db8f0de6e9c8c812",
  "data": {"dstcode": ["8801", "8802"]}}
}
```

Request structure and fields description:

```
{'data': {'dstcodes': ['String']}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
dstcodes	N/A	False	List(String(allow_blank: False, regex: "^[0-9#]**\$"), min_length: 0)	destination code

2.74 country_codes_get

Docstring:

Returns list of country codes and corresponding names by first digits of destination code or first symbols of country name

Example:

-Client Request

```
{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "limit": 101,
    "country_or_dstcode": "Cuba",
    "sort_by": [{"dstcode": "asc"}],
    "offset": 0,
    "fields": ["dstcode", "country", "is_mobile"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["dstcode", "country", "is_mobile"],
  "message": "OK",
  "data": [
    ["53", "Cuba", false], ["5321", "Cuba", false], ["5399", "Cuba Guan
tanamo Bay", false]
  ],
  "time_elapsed": 0.46834
}
```

Request structure and fields description:

```
{'data': {'country': 'String',
  'country_or_dstcode': 'String',
  'dstcode': 'String',
  'fields': ["Enum('dstcode', 'is_mobile', 'country')"],
  'insensitive': 'Bool',
  'is_mobile': 'Bool',
  'limit': 'Integer',
  'offset': 'Integer',
  'sort_by': [{"Enum('dstcode', 'is_mobile', 'country')": "Enum('asc', 'desc',
  '')"}]},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
country_or_dstcode	N/A	True	String(allow_blank: True)	first digits of country code or first symbols of country name

insensitive	False	True	Bool	flag is used for insensitive sorting
is_mobile	N/A	True	Bool	not in use
dstcode	N/A	True	String(allow_blank: True)	destination code or first digits of destination code
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['dstcode', 'is_mobile', 'country']	True	List([Enum('dstcode', 'is_mobile', 'country')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
country	N/A	True	String(allow_blank: True)	country name or first symbols of country name
sort_by	[]	True	List([{Enum('dstcode', 'is_mobile', 'country'): Enum('asc', 'desc', '')}], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"country": "asc"}] - sorts result ascending by field "country"

Default fields:

Name	Type & Comments
dstcode	String
is_mobile	Boolean
country	String

2.75 country_codes_get_grouped

Docstring:

! function is for new web 3.10 with startswith logic for 'country'

Returns list of country codes and corresponding names(for details of filtering - see fields description)

Example:

-Client Request - country names for codes which starts with '88'

```
{"session_id":"ac2d9bc02a820409db8f0de6e9c8c812",
  "data":{"dstcode":"88",
        "limit":1001,
        "sort_by":[{"country":""}],
        "offset":0,
        "fields":["country","dstcode_group"]}
}
```

Request structure and fields description:

```
{'data': {'country': 'String',
          'country_or_dstcode': 'String',
          'dstcode': 'String',
          'fields': ["Enum('country', 'dstcode_group')"],
          'is_mobile': 'Bool',
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('country', 'dstcode_group')": "Enum('asc', 'desc', '')"}]}
, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
country_or_dstcode	N/A	True	String(allow_blank: True)	first digits of country code or first symbols of country name
dstcode	N/A	True	String(allow_blank: True)	destination code or first digits of destination code. If in request present only this filter function returns region name and all codes corresponding to this region.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['country', 'dstcode_group']	True	List([Enum('country', 'dstcode_group')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

limit	0	True	Integer(gte: 0)	says how many rows will be in response
country	N/A	True	String(allow_blank: True)	country name or first symbols of country name
is_mobile	N/A	True	Bool	not in use
sort_by	[]	True	List([Enum('country', 'dstcode_group'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"country": "asc"}] - sorts result ascending by field "country"

Default fields:

Name	Type & Comments
country	String
dstcode_group	List(String())

2.76 country_codes_get_grouped_equals

Docstring:

Returns list of country codes and corresponding names(for details of filtering - see fields description).

Difference between this function and country_codes_get_grouped that it filters by "country" using equality comparison (not like).

Example:

-Client Request - country names for codes which starts with '88'

```
{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "fields": ["dstcode_group"],
    "country": "Ukraine"
  }
}
```

- Server Response

```
{
  "status": 200,
```

```

    "fields": ["dstcode_group"],
    "message": "OK",
    "data": [
        ["380401", "380407", "380408", "3804", "38040"]]
    ],
    "time_elapsed": 0.086499
}

```

Request structure and fields description:

```

{'data': {'country': 'String',
          'country_or_dstcode': 'String',
          'dstcode': 'String',
          'fields': ["Enum('country', 'dstcode_group')"],
          'is_mobile': 'Bool',
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('country', 'dstcode_group')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
country	N/A	True	String(allow_blank: True)	exact country name
is_mobile	N/A	True	Bool	not in use
dstcode	N/A	True	String(allow_blank: True)	destination code or first digits of destination code. If in request present only this filter function returns region name and all codes corresponding to this region.
fields	['country', 'dstcode_group']	True	List([Enum('country', 'dstcode_group')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('country', 'dstcode_group')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"country": "asc"}] - sorts result ascending by field "country"
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before

				beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
country_or_dstcode	N/A	True	String(allow_blank: True)	first digits of country code or first symbols of country name

Default fields:

Name	Type & Comments
country	String
dstcode_group	List(String())

2.77 cpm_points_get

Docstring:

Returns data to draw summary attempts charts (separately for each RTP hosts of the system) for the given period (288 values for each period). "start_time" field contains start time of the period as POSIX timestamp (int).

"day" period contains data 2 last days.

Example:

- Client Request

```
{
  "session_id": "1d81ff9b02a2ea9f55df0cf4327f3124",
  "data": {
    "period": "month"
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": [
    ["194.145.39.19", [426.819576966, 219.767931342, 9.5516722038, 1.85424038046, 1.03966881114, ...]]
  ],
  "time_elapsed": 0.226724,
  "start_time": 1459345800
}
```

```
}
```

Request structure and fields description:

```
{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')"},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.

2.78 currency_add

Docstring:

Add new currency with given unique international short name, long name, symbol and international code to the database.

Function returns "id" of new currency in the database.

Example:

- Client Request

```
{  
  "session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",  
  "data": {  
    "symbol": "\u20AE",  
    "prefix": "",  
    "name_short": "MNT",  
    "code": 496,  
    "suffix": "",  
    "name_long": "Tugric"  
  }  
}
```

- Server response

```
{  
  "status": 200,  
  "message": "OK",  
  "id": 4,  
  "time_elapsed": 0.059638  
}
```

Request structure and fields description:

```
{'data': {'code': 'Integer',
```



```

'name_long': 'String',
'name_short': 'String',
'prefix': 'String',
'suffix': 'String',
'symbol': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
name_long	N/A	False	String(allow_blank: False)	full name of the currency, should be unique.
code	N/A	False	Integer()	international code of currency, should be unique.
suffix	N/A	True	String(allow_blank: True)	currency suffix, not in use.
symbol	N/A	False	String(allow_blank: False)	currency sign, for example '\$'. Symbol should be unique.
prefix	N/A	True	String(allow_blank: True)	currency prefix, not in use.
name_short	N/A	False	String(allow_blank: False)	international short name of currency. Consists of 3 chars (USD, UAH etc.).

2.79 currency_course_add

Docstring:

Allows to change OP and TP course for given currency. OP course("value_op") should be less than or equal TP value("value_tp").

Function returns currency id in database.

Example:

- Client Request

```

{"session_id": "ba6c2e2f97ebdbdf6910369b2e48b19a1",
 "data": {"currency_id": 4,

```

```

        "value_op":1.2E-6,
        "value_tp":1.3E-6}
    }

```

- Server Response

```

{"status": 200, "message": "OK", "id": 4, "time_elapsed": 0.068219}

```

Request structure and fields description:

```

{'data': {'currency_id': 'Integer', 'value_op': 'Float', 'value_tp': 'Float'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
value_op	N/A	False	Float()	currency course for OP with 5 decimal digits. This value defines cost of a unit of the given currency relative to the default currency. For example default currency is dollar. 1\$ costs 10 units of the given currency, then unit of the given currency costs (1/10)\$ or 0,10000\$.
currency_id	N/A	False	Integer(gt: 0)	
value_tp	N/A	False	Float()	currency course for TP with 5 decimal digits. This value defines cost of a unit of the given currency relative to the default currency. For example default currency is dollar. 1\$ costs 10 units of the given currency, then unit of the given currency costs (1/10)\$ or 0,10000\$.

2.80 currency_course_get

Docstring:

Returns history of changes OP and TP courses for chosen currency.

Example:

- Client Request

```

{"session_id":"ba6c2e2f97ebddf6910369b2e48b19a1",
 "data":{"limit":101,
        "currency_id":4,
        "sort_by":[{"currency_name": ""}, {"date_start":"desc"}],
        "start_date":"2013-01-15T10:42:35",
        "end_date":"2014-01-15T10:42:35",
        "offset":0,
        "fields":["id", "currency_id", "value_op","value_tp","date_start"]}
}

```

- Server Response

```

{"status": 200,
 "fields": ["id", "value_op", "value_tp", "date_start"],
 "message": "OK",
 "data": [[1, "0.00000120", "0.00000130", "2014-01-15T10:42:35.565128+00:00"]],
 "time_elapsed": 0.004649}

```

Request structure and fields description:

```

{'data': {'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('currency_name', 'value_op', 'date_start', 'currency_id', 'value_tp', 'id')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('currency_name', 'value_op', 'date_start', 'currency_id', 'value_tp', 'id')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'with_prev_change': 'Bool'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
currency_id	N/A	True	Integer(gt: 0)	filter by currency id
start_date	N/A	True	String(allow_blank: False)	start date and time for course history filtering
sort_by	[]	True	List([{"Enum('currency_name', 'value_op', 'date_start', 'currency_id', 'value_tp', 'id')": "Enum('asc', 'desc', '')"}]),	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"currency_id":"asc"}] - sorts result ascending by field

			min_length: 0)	"currency_id"
with_prev_change	N/A	True	Bool	show last currency change before given "start_date". In case "start_date" didn't provide value of the "with_prev_change" will be ignored.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
end_date	N/A	True	String(allow_blank: False)	end date and time for course history filtering
fields	['currency_name', 'value_op', 'date_start', 'currency_id', 'value_tp', 'id']	True	List([Enum('currency_name', 'value_op', 'date_start', 'currency_id', 'value_tp', 'id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
currency_name	String(3) - name_short
value_op	numeric(16,8)
date_start	String
currency_id	Int
value_tp	numeric(16,8)
id	Int

2.81 currency_del

Docstring:

Removes currency with given id from database. Make sure before removing that there is no balance with given currency.

Function returns deleted currency id.

Example:

- Client Request

```
{"session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",  
  "data": {"id": 5}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 5, "time_elapsed": 0.053819}
```

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	currency id

2.82 currency_get

Docstring:

Returns list of currencies that exist in database with current currency courses.

Example:

- Client Request

```
{
  "session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",
  "data": {
    "limit": 101,
    "sort_by": [{"name_short": ""}],
    "name_short": "",
    "offset": 0,
    "fields": ["id", "name_short", "name_long", "symbol", "prefix",
               "suffix", "code", "value_op", "value_tp"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name_short", "name_long", "symbol", "prefix",
             "suffix", "code", "value_op", "value_tp"],
  "message": "OK",
  "data": [
    [2, "EUR", "Euro", "\u20ac", "", "", 978, "1.33000000", "1.33000000"],
    [4, "MNT", "Tugric", "\u20ae", "", "", 496, "1.00000000", "1.00000000"],
    [1, "USD", "US Dollar", "$", "", "", 840, "1.00000000", "1.00000000"]
  ],
  "time_elapsed": 0.004684
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': ["Enum('name_short', 'prefix', 'code', 'bank_requisite', 'suffix',
                    'value_op', 'symbol', 'value_tp', 'id', 'name_long')"],
    'id': 'Integer',
    'limit': 'Integer',
    'name_long': 'String',
    'name_short': 'String',
    'offset': 'Integer',
    'sort_by': [{"Enum('name_short', 'prefix', 'code', 'bank_requisite', 'suffix',
                      'value_op', 'symbol', 'value_tp', 'id', 'name_long')": "Enum('asc', 'desc', '')"}]},
    'session_id': 'String'
  }
}
```

data validation description:

Name	Default value	Optional	Type	Description
name_long	N/A	True	String(allow_blank: True)	filter by full name of currency or first chars of full name.

sort_by	[]	True	List([Enum('name_short', 'prefix', 'code', 'bank_requisite', 'suffix', 'value_op', 'symbol', 'value_tp', 'id', 'name_long'): Enum('asc', 'desc', ")]), min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"currency_id": "asc"}] - sorts result ascending by field "currency_id".
name_short	N/A	True	String(allow_blank: True)	filter by short name of currency. Short name is international name consists of 3 chars(example: USD,UAH).
limit	0	True	Integer(gte: 0)	says how many rows will be in response.
id	N/A	True	Integer(gt: 0)	currency id
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['name_short', 'prefix', 'code', 'bank_requisite', 'suffix', 'value_op', 'symbol', 'value_tp', 'id', 'name_long']	True	List([Enum('name_short', 'prefix', 'code', 'bank_requisite', 'suffix', 'value_op', 'symbol', 'value_tp', 'id', 'name_long')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
name_short	String(3)
prefix	String
code	Int
bank_requisite	String (WILL BE REMOVED. SOON.)
suffix	String
value_op	numeric(16,8)

symbol	String(16)
value_tp	numeric(1,8)
id	Int
name_long	String(24)

2.83 currency_history_get

Docstring:

Returns full history of OP and TP courses for given period. First row in result ed data contains initial currency courses before given time period, where "date" is date and time when last currency course change has been made.

Response also contains list of provided in the request "currencies_ids".

"data" contains lists in the following format:
{"date": "", "courses": [["", "", ""], [...]...]}

Currencies order in the "courses" is the same as in "currencies_ids".

Example

- Client Request

```
{
  "session_id": "45948d5cbce137710f5e2b2ab665bf58",
  "data": {
    "end_date": "2014-11-13T10:59:59",
    "currencies_ids": [1, 35, 31, 29, 44, 30, 2, 4, 24, 9, 43, 54, 23],
    "start_date": "2014-11-11T10:00:00"
  }
}
```

- Server Response

```
{
  "status": 200,
  "currencies_ids": [1, 35, 31, 29, 44, 30, 2, 4, 24, 9, 43, 54, 23],
  "message": "OK",
  "data": [
    {
      "date": "2014-11-07T10:23:32.506745+00:00",
      "courses": [
        ["1.00000000", "1.00000000", false], ["2.00000000", "2.00000000", false],
        ["1.00000000", "1.00000000", false], ["5656.00000000", "5656.00000000", false],
        ["1.00000000", "1.00000000", false], [null, null, false],
        ["1.32000000", "1.34300000", false], ["0.23400000", "0.23400000", false]
      ]
    }
  ]
}
```



```

    ["5.000000000", "8.000000000", false], ["0.020700000", "0.020800000", true],
    [null, null, false], [null, null, false], ["0.100000000", "0.200000000", false]
  ]},
  {"date": "2014-11-11T10:21:09.968736+00:00",
   "courses": [
     ["1.000000000", "1.000000000", false], ["2.000000000", "2.000000000", false],
     ["1.000000000", "1.000000000", false], ["5656.000000000", "5656.000000000", false],
     ["1.000000000", "1.000000000", false], [null, null, false], ["1.440000000", "1.444000000", true],
     ["0.234000000", "0.234000000", false], ["5.000000000", "8.000000000", false],
     ["0.020700000", "0.020800000", false], [null, null, false], [null, null, false],
     ["0.100000000", "0.200000000", false]
   ]}
],
"time_elapsed": 0.050829
}

```

Request structure and fields description:

```

{'data': {'currencies_ids': ['Integer'],
          'end_date': 'String',
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('date_start', 'currency_id', 'value_op', 'value_tp')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String'},
         'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([{"Enum('date_start', 'currency_id', 'value_op', 'value_tp'): Enum('asc', 'desc', '')"}], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"currency_id": "asc"}] - sorts result ascending by field "currency_id"
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
start_date	N/A	False	String(allow_blank: False)	start date and time for course history

				filtering
end_date	N/A	False	String(allow_blank: False)	end date and time for course history filtering
currencies_ids	N/A	False	List([Integer()], min_length: 0)	list of currencies which courses will be returned
limit	0	True	Integer(gte: 0)	says how many rows will be in response

2.84 currency_requisite_by_owner_get

Docstring:

Returns bank requisites by currency id and owner id.

Example:

- Client Request

```
{
  "session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",
  "data": {
    "currency_id": 2,
    "owners_id": 1,
    "fields": ["bank_requisite"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["bank_requisite"],
  "message": "OK",
  "data": [[]],
  "time_elapsed": 0.011828
}
```

Request structure and fields description:

```
{'data': {'currency_id': 'Integer',
          'fields': ["Enum('currency_id', 'bank_requisite', 'owners_id')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'owners_id': 'Integer',
          'sort_by': [{"Enum('currency_id', 'bank_requisite', 'owners_id')": "Enum('asc', 'desc', '')"}]},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
owners_id	N/A	True	Integer(gt: 0)	owner id
currency_id	N/A	True	Integer(gt: 0)	
fields	['currency_id', 'bank_requisite', 'owners_id']	True	List([Enum('currency_id', 'bank_requisite', 'owners_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('currency_id', 'bank_requisite', 'owners_id'): Enum('asc', 'desc', ")]], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"currency_id": "asc"}] - sorts result ascending by field "currency_id".
limit	0	True	Integer(gte: 0)	says how many rows will be in response.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
currency_id	Int
bank_requisite	String
owners_id	Int

2.85 currency_requisite_by_owner_set

Docstring:

Changes requisites by owner_id and currency id in currency_owners.

Function returns currency id in database.

Example:

- Client Request

```
{"session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",  
  "data": {"currency_id": 4,  
           "owners_id": 1,  
           "bank_requisite": "bank"}  
}
```

- Server response

```
{"status": 200,  
  "currency_id": 4,  
  "message": "OK",  
  "time_elapsed": 0.038057}
```

Request structure and fields description:

```
{'data': {'bank_requisite': 'String',  
          'currency_id': 'Integer',  
          'owners_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
bank_requisite	N/A	True	String(allow_blank: True)	mediacore owner's bank account for given currency
currency_id	N/A	False	Integer(gt: 0)	
owners_id	N/A	False	Integer(gt: 0)	

2.86 currency_set

Docstring:

Changes parameters of the currency with given id excluding currency course.

Function returns currency id in database.

Example:

- Client Request

```
{"session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",  
  "data": {"id": 4,  
            "symbol": "\u20AE",  
            "prefix": "tug",  
            "name_short": "MNT",  
            "code": 496,  
            "suffix": "tug",  
            "name_long": "Tugric"}}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 4, "time_elapsed": 0.036685}
```

Request structure and fields description:

```
{'data': {'bank_requisite': 'String',  
          'code': 'Integer',  
          'id': 'Integer',  
          'name_long': 'String',  
          'name_short': 'String',  
          'prefix': 'String',  
          'suffix': 'String',  
          'symbol': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name_short	N/A	True	String(allow_blank: False)	international short name of currency. Consists of 3 chars (e.g. USD, UAH).
prefix	N/A	True	String(allow_blank: True)	currency prefix, not in use.
name_long	N/A	True	String(allow_blank: False)	full name of currency, should be unique.
id	N/A	False	Integer(gt: 0)	id of currency
suffix	N/A	True	String(allow_blank: True)	currency suffix, not in use.

			k: True)	
symbol	N/A	True	String(allow_blank: False)	currency symbol, for example '\$', should be unique.
code	N/A	True	Integer()	currency international code, should be unique.
bank_requirement	N/A	True	String(allow_blank: True)	mediacore owner's bank account for given currency

2.87 customized_report_get

Docstring:

Returns customized report (for voice traffic) by chosen parameters with statistical information
 (common duration of calls, duration with considering prices cost counting parameters,
 asr, acd, attempts statistic, cost in/out, profit, margin values combining by destination codes, points,
 dates, managers or companies).

Example:

- Client Request

```
{
  "session_id": "90cc89d231aec5174431a630bd3e6a5f",
  "data": {
    "currency_id": 1,
    "start_date": "2017-07-24T00:00:00",
    "end_date": "2017-07-30T23:59:59",
    "time_zone": "0:00",
    "fields": [
      "company", "point_name", "destination_code", "country", "origin_country",
      "duration", "calculated_duration", "op_price", "tp_price", "attempts",
      "successful_attempts",
      "asr", "acd", "pdd", "short_calls", "cost_in", "cost_out", "margin", "profit", "manager", "ner"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "company", "point_name", "destination_code", "country", "origin_country",
    "duration", "calculated_duration", "op_price", "tp_price", "attempts", "successful_attempts",
    "asr", "acd", "pdd", "short_calls", "cost_in", "cost_out", "margin", "profit"
  ]
}
```

```
it","manager","ner"
    ],
    "message": "OK",
    "data": {
        "inbound": [
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6651",null,null,0,0,null,null
,3,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6671",null,null,0,0,null,null
,9,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6648",null,null,0,0,null,null
,1,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6655",null,null,0,0,null,null
,12,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6674",null,null,0,0,null,null
,4,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6673",null,null,0,0,null,null
,1,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6656",null,null,0,0,null,null
,3,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6647",null,null,0,0,null,null
,1,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6678",null,null,0,0,null,null
,1,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6650",null,null,0,0,null,null
,2,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6645",null,null,0,0,null,null
,36,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6646",null,null,0,0,null,null
,1,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6641",null,null,0,0,null,null
,5,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
            ["Company #2_VAT","OP_ᲚᲗᲕ_Client","6643",null,null,0,0,null,null
,17,0,"0.00",0,
            "0.0","0.0","0.00000","0.00000","0.0","0.0","Bogdan Sergey","0
.00"],
```

```

,1,0,"0.00",0,      ["Company #2_VAT","OP_»»_Client","6659",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,13,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6677",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,85,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6653",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,1,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6675",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,3,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6654",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,2,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6676",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,70,0,"0.00",0,    ["Company #2_VAT","OP_»»_Client","6644",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
0,"0.000000",22257,0,"0.00",0,    ["Call Path","Call Path OP","380","Ukraine all","",0,0,"11.0000
85"],              "0.0","0.0","0.000000","0.000000","0.000000","0.0","phoenix","12.
,4,0,"0.00",0,     ["Company #2_VAT","OP_»»_Client","6672",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
,96,0,"0.00",0,    ["Company #2_VAT","OP_»»_Client","6642",null,null,0,0,null,null
,0.00"],            "0.0","0.0","0.000000","0.000000","0.0","0.0","Bogdan Sergey","0
],
"outbound": [
, "0.000000","10.000000",    ["1234d_www","Call Path TP","38050","UKRAINE MOBILE MTC","",0,0
, "0.0","ID","100.00"],      "2860","0","0.00",0,"0.0","0.0","0.000000","0.000000","0.000000",
],
"inbound_total": [
0.00,"0.000000",            null,null,null,null,null,0,0,null,null,22628,0,"0.00",0,"0.0","
, "0.000000","0.00","0.000000",null,"0.00"
],
"outbound_total": [
, "0.00","0.000000",        null,null,null,null,null,0,0,null,null,"2860","0","0.00",0,"0.0
, "0.000000","0.00","0.000000",null,"0.00"
]
},
"time_elapsed": 18.878173
}

```


Request structure and fields description:

```
{'data': {'acd_less_than': 'Integer',
          'acd_more_than': 'Integer',
          'asr_less_than': 'Float',
          'asr_more_than': 'Float',
          'currency_id': 'Integer',
          'destinations': ['String'],
          'end_date': 'String',
          'exclude_codes': ['String'],
          'fields': ["Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin')"],
          'group_by': ["Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin')"],
          'min_attempts': 'Integer',
          'op_company_ids': ['Integer'],
          'op_ids': ['Integer'],
          'op_manager_ids': ['Integer'],
          'profit_less_than': 'Float',
          'profit_more_than': 'Float',
          'session_id': 'String',
          'sort_by': [{"Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'time_zone': 'String',
          'tp_company_ids': ['Integer'],
          'tp_ids': ['Integer'],
          'tp_manager_ids': ['Integer']},
        'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd',	True	List([Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin')], min_length:	required fields. If absent - takes fields specified in "default_fields". If empty - returns error.

	'margin']		0)	
start_date	N/A	False	String(allow_blank: False)	
end_date	N/A	False	String(allow_blank: False)	
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2})\$")	selects data in specified timezone
profit_more_than	N/A	True	Float()	filter by profit min threshold
destinations	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that should be present in reply
min_attempts	0	True	Integer(gte: 0)	display rows where attempts more than "min_attempts"
op_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	Filter by Origination Point(s)
op_manager_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	Filter by Origination Account Manager(s)
currency_id	1	True	Integer(gt: 0)	display prices and cost in chosen currency. By defaults from user's session
group_by	[]	True	List([Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin')], min_length: 0)	defines field(s) on which result will be grouped.
tp_company_ids	N/A	True	List([Integer(gt: 0)],	Filter by Termination

			min_length: 0)	Company(ies)
asr_more_than	N/A	True	Float(gte: 0.0)	filter by asr, returns rows with asr more than specified
acd_more_than	N/A	True	Integer(gte: 0)	filter by acd, returns rows with acd more than specified
exclude_codes	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that shouldn't be present in reply
asr_less_than	N/A	True	Float(gt: 0.0)	filter by asr, returns rows with asr less than specified
session_id	N/A	True	String(allow_blank: False)	by defaults from user's session
tp_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	Filter by Termination Point(s)
sort_by	[]	True	List([{Enum('destination_code', 'successful_attempts', 'manager', 'duration', 'cost_in', 'calculated_duration', 'profit', 'point_name', 'op_price', 'ner', 'cost_out', 'asr', 'origin_country', 'company', 'date', 'short_calls', 'acd', 'country', 'tp_price', 'attempts', 'pdd', 'margin'): Enum('asc', 'desc', '')}], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"manager": "asc"}] - sorts result ascending by field "manager"
acd_less_than	N/A	True	Integer(gt: 0)	filter by acd, returns rows with acd less than specified
profit_less_than	N/A	True	Float()	filter by profit max threshold
tp_manager_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	Filter by Termination Account Manager(s)

op_compan y_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	Filter by Origination Company(ies)
--------------------	-----	------	--	---------------------------------------

Default fields:

Name	Type & Comments
destination_code	String
successful_attempts	Int - quantity of successful attempts
manager	String
duration	Int - calls duration in seconds
cost_in	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)
calculated_duration	Int - calculated calls duration in seconds
profit	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)
point_name	String
op_price	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)
ner	String
cost_out	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)
asr	String
origin_country	String
company	String
date	String

short_calls	String
acd	Int
country	String
tp_price	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)
attempts	Int
pdd	Numeric - post dial delay in seconds
margin	String or Null (Null if "hide_owners_profit" flag is enabled and different manager)

2.88 destinations_by_tp_get

Docstring:

Function returns list of currently actual destinations for given termination point within chosen context.

Actual means that price for the destination works at the moment.

Example:

- Client Request

```
{
  {
    "tp_id": 4692,
    "context_id": 1,
    "fields": ['destination', 'country', 'price', 'allowed']
  }
  "session_id": "ca84cd3eaa596f2e3f7836db065777da"
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.010708,
  "message": "OK",
  "data": [
    ['38050', None, Decimal('0.00710'), True],
    ['8801', 'BANGLADESH MOBILE', Decimal('0.01380'), True],
  ],
  "fields": ['tp_name', 'destination', 'country', 'price', 'allowed']
}
```

}

Request structure and fields description:

```
{'data': {'context_id': 'Integer',
          'country_or_code': 'String',
          'fields': ["Enum('ci', 'ft', 'mc', 'country', 'price', 'destination', 'allowed')"],
          'hide_without_country': 'Bool',
          'limit': 'Integer',
          'offset': 'Integer',
          'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	False	Integer(gt: 0)	exact destination code
hide_without_country	N/A	True	Bool	show only records with non-empty country
context_id	N/A	False	Integer(gt: 0)	filter by context
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
country_or_code	N/A	True	String(allow_blank: False)	filter by country name or destination code
fields	['ci', 'ft', 'mc', 'country', 'price', 'destination', 'allowed']	True	List([Enum('ci', 'ft', 'mc', 'country', 'price', 'destination', 'allowed')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
ci	Int
ft	Int
mc	Int
country	String or NULL
price	Numeric(16, 5)
destination	String
allowed	Boolean

2.89 dialpeer_add

Docstring:

Creates a new dialpeer. This function also used for copying already existent dialpeer to given context and it's parameters:

- dialpeer -> TP relations
 - dialpeer -> Group relations
- and so on.

Function returns id of created dialpeer. If dialpeer was added to several contexts their id will be the same.

Examples:

- add simple dialpeer:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "name": "test_dp",
    "dstregex": ".*",
    "enabled": True,
    "contexts": [1],
  }
}
```

- add dialpeer with predefined Groups and TPs relations:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "name": "test_dp",
    "dstregex": ".*",
    "enabled": True,
    "contexts": [1],
    "groups_ids": [[935], [855]],
  }
}
```

```

    "tp_ids": [[4536, 0, 0], [4730, 0, 0]],
    "current_context": 2,
}

```

where:

"tp_ids" - is a lists which contain following data:
 TP id linked with the dialpeer,
 dialpeer -> TP relation priority,
 dialpeer -> TP relation weight;

"group_ids" - list of group ids which have to be connected with
 chosen dialpeer;

Request structure and fields description:

```

{
  'data': {
    'ani_allow': 'String',
    'ani_block': 'String',
    'ani_regexp': 'String',
    'capacity': 'Integer',
    'contexts': ['Integer'],
    'current_context': 'Integer',
    'delta_price': 'Float',
    'delta_price_unit': 'Enum(1, 2)',
    'denyregexp': 'String',
    'dnis_regexp': 'String',
    'dstregexp': 'String',
    'enabled': 'Bool',
    'groups_ids': [['Integer']],
    'invisible': 'Bool',
    'name': 'Name(at least one printable character). max_length=None',
    'priority': 'Integer',
    'profit': 'Float or Null',
    'profit_control': 'Bool',
    'service_id': 'Integer',
    'stop': 'Bool',
    'tp_ids': [['Integer', 'Integer', 'Integer']],
  },
  'session_id': 'String'
}

```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	True	List([Integer(gt: 0)], min_length: 0)	context ids to which dialpeer will be added
ani_regexp	""	True	String(allow_blank: True, max_length: 64)	string variable which defines conversion of the caller number with help of regular expression
groups_ids	N/A	True	List([List([Integer(gt:	list of groups existed in the chosen context, to

			0)], min_length: 0)], min_length: 0)	which given dialpeer will be added. Use this parameter in conjunction with "current_context" option.
current_context	N/A	True	Integer(gt: 0)	context id to which groups relations will be added
dnis_regexp	""	True	String(allow_blank: True, max_length: 64)	string variable which defines conversion of the called number with help of regular expressions
profit_control	False	True	Bool	toggle to enable/disable dialpeers dynamic routing (for voip only)
delta_price_unit	1	True	Enum(1, 2)	delta price unit type (1-value, 2-percent)
tp_ids	N/A	True	List([List([Integer(gt: 0), Integer(gte: -100, lte: 100), Integer(gte: 0, lte: 100)], 3)], min_length: 0)	nested lists contains TP -> Dialpeer relation data which can be copied to another context(s) in the following order: TP id to link it with the dialpeer, dialpeer -> TP relation priority and dialpeer -> TP relation weight
name	N/A	False	Name(at least one printable character). max_length=None	name of dialpeer or new name for dialpeer's copy. Name should be unique. Dialpeer's name can't be "Service_dialpeer_" because this name reserved for configuration in Simple Routing mode.
profit	1.0	True	Or(Float(), Null)	threshold value of dialpeer profit for dynamic routing (for voip only)
ani_block	""	True	String(allow_blank: True)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
delta_price	0	True	Float()	this floating value allows to permit only routes with necessary profitability. If distinction between OP price and TP price in the default currency less than this value then given TP won't participate in routing. This value specifies in default currency.
priority	0	True	Integer(gte: -100, lte: 100)	integer number, dialpeer's priority in range [-100:+100]
invisible	False	True	Bool	option that makes dialpeer invisible. relations between dp and tp, dp and group will be

				removed.
dstregex	N/A	False	String(allow_blank: False)	a regular expressions which defines allowed destinations. For example ".*" - describes all possible destinations, a few expressions should be separated by semicolon
ani_allow	""	True	String(allow_blank: True, max_length: 255)	exact allowed numbers or regular expressions for group of numbers separated by semicolon
capacity	500	True	Integer(gte: 0, lte: 1000)	integer value from 0 to 1000, which means amount of concurrent calls and call attempts through the dialpeer
stop	True	True	Bool	boolean option, if "True" then all dialpeers from global dialpeers tree that shorter than current won't participate in the routing.
denyregex	""	True	String(allow_blank: True, max_length: 128)	regular expression which defines denied destinations for given dialpeer. For example "^3.*\$" block calls to destinations which begins with 3, a few expressions should be separated by semicolon.
service_id	1	True	Integer(gt: 0)	service type(1 - voice, 2 - sms)
enabled	False	True	Bool	state of dialpeer, if "True" then dialpeer is enabled

2.90 dialpeer_copy

Docstring:

Deprecated. Currently not in use.

Request structure and fields description:

```
{'data': {'contexts': ['Integer'], 'id': 'Integer', 'name': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	
name	N/A	False	String(allow_blank: False)	

2.91 dialpeer_groups_get

Docstring:

Returns list of groups connected with chosen dialpeer.

Example:

- Client Request

```
{
  "session_id": "593a9260cb1406179301fe2c2680e0fd",
  "data": {
    "sort_by": [{"name": ""}],
    "name": "",
    "contexts": [1],
    "dialpeer_id": 4,
    "fields": ["group_id", "name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["group_id", "name"],
  "message": "OK",
  "data": [[3, "h_p (CZ)"]],
  "time elapsed": 0.007634
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'dialpeer_id': 'Integer',
          'fields': ["Enum('group_id', 'dialpeer_id', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('group_id', 'dialpeer_id', 'name')": "Enum('asc', 'desc',
          '')"}]},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name
contexts	[]	False	List([Integer()], min_length: 0)	contexts id where need search relations
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['group_id', 'dialpeer_id', 'name']	True	List([Enum('group_id', 'dialpeer_id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
dialpeer_id	N/A	False	Integer(gt: 0)	
sort_by	[]	True	List([Enum('group_id', 'dialpeer_id', 'name'): Enum('asc', 'desc', ")]], min_length: 0)	field name on which result will be sorted.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
group_id	Int
dialpeer_id	Int
name	String

2.92 dialpeer_set

Docstring:

Updates general settings of dialpeer with given id within given context(s) and in schema routing.

This function disallows to edit service dialpeers.
Function returns id of dialpeer that was edited.

Example:

- Client Request

```
{"session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",
  "data": {"id": 6,
           "stop": true,
           "enabled": true,
           "denyregexp": "^123.*$",
           "priority": 0,
           "delta_price": -100.0,
           "name": "Test_DP",
           "ani_regexp": "",
           "contexts": [1],
           "dstregexp": "^.*",
           "dnis_regexp": ""}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 6, "time_elapsed": 0.039199}
```

Request structure and fields description:

```
{'data': {'ani_allow': 'String',
          'ani_block': 'String',
          'ani_regexp': 'String',
          'capacity': 'Integer',
          'contexts': ['Integer'],
          'delta_price': 'Float',
          'delta_price_unit': 'Enum(1, 2)',
          'denyregexp': 'String',
          'dnis_regexp': 'String',
          'dstregexp': 'String',
          'enabled': 'Bool',
          'id': 'Integer',
          'invisible': 'Bool',
          'name': 'Name(at least one printable character). max_length=None',
          'priority': 'Integer',
          'profit': 'Float or Null',
          'profit_control': 'Bool',
          'service_id': 'Integer',
          'stop': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
service_id	1	True	Integer(gte: 1, lte: 1)	service type(1 - voice)
delta_price_unit	N/A	True	Enum(1, 2)	delta price unit type (1-value, 2-percent)
capacity	N/A	True	Integer(gte: 0, lte: 1000)	integer value from 0 to 1000 which means amount of concurrent calls and call attempts through the dialpeer
name	N/A	True	Name(at least one printable character). max_length=None	dialpeer name
dstregex	N/A	True	String(allow_blank: False)	a regular expressions which defines allowed destinations. For example ".*" - describes all possible destinations,a few expressions should be separated by comma
invisible	N/A	True	Bool	if dialpeer becomes invisible all its relation with tps and groups will be removed.
stop	N/A	True	Bool	boolean option, if "True" then all dialpeers from global dialpeers tree that shorter than current won't participate in the routing
enabled	N/A	True	Bool	dialpeer's state, if True - dialpeer is enabled
priority	N/A	True	Integer(gte: -100, lte: 100)	dialpeer's priority in range from -100 to +100
denyregex	N/A	True	String(allow_blank: True, max_length: 128)	regular expression which defines denied destinations for given dialpeer. For example "^3.*\$" block calls to destinations which begins with 3, a few expressions should be separated by comma.
profit_control	N/A	True	Bool	toggle to enable/disable dialpeers dynamic routing (for voip only)

profit	N/A	True	Or(Float(), Null)	threshold value of dialpeer profit for dynamic routing (for voip only)
ani_regexp	N/A	True	String(allow_blank: True, max_length: 64)	string variable which defines conversion of the caller number with help of regular expression; max length of regex is 64
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	contexts ids where chosen dialpeer will be changed
ani_block	N/A	True	String(allow_blank: True)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
ani_allow	""	True	String(allow_blank: True, max_length: 255)	exact allowed numbers or regular expressions for group of numbers separated by semicolon
delta_price	N/A	True	Float()	this floating value allows to permit only routes with necessary profitability. If distinction between OP price and TP price in the default currency less than this value then given TP won't participate in routing. This value specifies in default currency.
dnis_regexp	N/A	True	String(allow_blank: True, max_length: 64)	string variable which defines conversion of the called number with help of regular expressions; max length of regex is 64
id	N/A	False	Integer(gt: 0)	id of the dialpeer in database

2.93 dialpeer_tp_set

Docstring:

Changes priority and weight of given TP <--> Dialpeer relations within chosen context(s). Function doesn't allow to change relations of service dialpeer.

Example:

- Client Request

```
{"session_id":"773b6b683024ac6b6cfd95c8b034e659",
  "data":{"weight":0,
          "priority":3,
          "contexts":[1],
          "dialpeer_id":6,"tp_id":8}}
```

```
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.030787}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'dialpeer_id': 'Integer',  
          'priority': 'Integer',  
          'tp_id': 'Integer',  
          'weight': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	False	Integer(gt: 0)	termination point id in database
dialpeer_id	N/A	False	Integer(gt: 0)	id of dialpeer in database
priority	N/A	True	Integer(gte: -100, lte: 100)	relation priority in range from -100 to 100 inclusive
weight	N/A	True	Integer(gte: 0, lte: 100)	weight of the relation in range from 0 to 100 inclusive
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	contexts ids where need change given relation

2.94 dialpeer_tps_op_get

Docstring:

Returns termination points connected with chosen OP via chosen group and dialpeer in the given context(s).

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"limit": 101,  
           "op_id": 3,  
           "sort_by": [{"tp_name": ""}],  
           "group_id": 6,  
           "contexts": [1],  
           "offset": 0,  
           "dialpeer_id": 6,  
           "fields": ["dialpeer_id", "tp_id", "tp_name", "priority", "weight"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["dialpeer_id", "tp_id", "tp_name", "priority", "weight"],  
  "message": "OK",  
  "data": [[6, 4, "Test_1_TP", 0, 0]],  
  "time_elapsed": 0.010412}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'dialpeer_id': 'Integer',  
          'fields': ["Enum('priority', 'tp_id', 'dialpeer_id', 'weight', 'tp_name')"],  
          'group_id': 'Integer',  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'op_id': 'Integer',  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('priority', 'tp_id', 'dialpeer_id', 'weight', 'tp_name')":  
"Enum('asc', 'desc', '')"}],  
          'tp_name': 'String'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
dialpeer_id	N/A	False	Integer(gt: 0)	filter by certain DP
group_id	N/A	False	Integer(gt: 0)	filter by certain group
tp_name	N/A	True	String(allow_blank: True)	filter by first symbols of termination point name

contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	required contexts ids
fields	['priority', 'tp_id', 'dialpeer_id', 'weight', 'tp_name']	True	List([Enum('priority', 'tp_id', 'dialpeer_id', 'weight', 'tp_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('priority', 'tp_id', 'dialpeer_id', 'weight', 'tp_name'): Enum('asc', 'desc', "")}], min_length: 0)	field on which list of groups and dialpeers will be sorted
limit	0	True	Integer(gte: 0)	rows quantity per one request
op_id	N/A	False	Integer(gt: 0)	filter by certain TP
service_id	N/A	True	Integer(gt: 0)	service type filter(1 - voice, 2 - sms)
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
priority	Int
tp_id	Int
dialpeer_id	Int
weight	Int
tp_name	String

2.95 dialpeers_free_group_get

Docstring:

Returns list of dialpeers which don't related with chosen group.

Example:

- Client Request

```
{"session_id": "677232544e5f894e13cc8a3f906504bd",  
  "data": {"sort_by": [{"name": ""}],  
           "name": "",  
           "group_id": 3,  
           "contexts": [1],  
           "fields": ["id", "name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[3, "st-dialpeer2 (CZ)"], [2, "st-dialpeer (CZ)"]],  
  "time_elapsed": 0.007284}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'fields': ["Enum('id', 'name')"],  
          'group_id': 'Integer',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
group_id	N/A	True	Integer(gt: 0)	filter by group
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
name	N/A	True	String(allow_blank: True)	filter by dialpeer name
contexts	[]	False	List([Integer()], min_length: 0)	contexts id where need search relations

fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('id', 'name'): Enum('asc', 'desc', ")]), min_length: 0)	field name(s) on which result will be sorted.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows

Default fields:

Name	Type & Comments
id	Int
name	String

2.96 dialpeers_from_groups_del

Docstring:

Removes relations between chosen dialpeers and groups. Function doesn't allow to remove relations between groups and service dialpeers and raises error if user tries to do it.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "dialpeers_ids": [6],
    "contexts": [1],
    "groups_ids": [1]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.041017
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'dialpeers_ids': ['Integer'],
          'groups_ids': ['Integer']},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
dialpeers_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of dialpeer ids
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of group ids
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	contexts ids where need remove specified relations

2.97 dialpeers_get

Docstring:

Returns dialpeers data by chosen criteria.

If dialpeer's id don't set explicitly in query function returns parameters of dialpeers within given context(s) which satisfy 'name' filter.

If dialpeer id specified - function returns settings of chosen dialpeer within given context(s).

Example:

- Client Request

```
{"session_id": "4dcfe8adb532e2b3856ada8caea50506",
  "data": {"id": 2,
           "contexts": [1],
           "fields": ["id", "name", "dstregexp", "priority", "enabled",
                     "stop", "denyregexp", "ani_regexp", "dnis_regexp",
                     "delta_price", "name_context"]}}
```

- Server response

```
{
  "status": 200,
  "fields": [
    "id", "name", "dstregexp", "priority", "enabled",
    "stop", "denyregexp", "ani_regexp", "dnis_regexp",
    "delta_price", "name_context"
  ],
  "message": "OK",
  "data": [
    [2, "ikex.dialpeer", "^.*", 0, true, false, "", "",
    "^(.*)$/0001\\1", "0.00000", "ikex.dialpeer (CZ)"]
  ],
  "time_elapsed": 0.008622
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'dstregexp': 'String',
    'fields': [
      "Enum('capacity', 'ani_regexp', 'name', 'ani_allow', 'profit', 'stop', 'delta_price', 'enabled', 'priority', 'name_context', 'context_id', 'dstregexp', 'dnis_regexp', 'last_modified', 'denyregexp', 'invisible', 'ani_block', 'delta_price_unit', 'id', 'profit_control', 'service_id')",
      'id': 'Integer',
      'invisible': 'Bool or Null',
      'limit': 'Integer',
      'name': 'String',
      'offset': 'Integer',
      'service_id': 'Integer',
      'sort_by': [
        "{Enum('capacity', 'ani_regexp', 'name', 'ani_allow', 'profit', 'stop', 'delta_price', 'enabled', 'priority', 'name_context', 'context_id', 'dstregexp', 'dnis_regexp', 'last_modified', 'denyregexp', 'invisible', 'ani_block', 'delta_price_unit', 'id', 'profit_control', 'service_id')": "Enum('asc', 'desc', '')"}"
      ],
      'session_id': 'String'
    ]
  }
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['capacity', 'ani_regexp', 'name', 'ani_allow', 'profit', 'stop', 'delta_price', 'enabled', 'priority', 'name_context', 'context_id', 'dstregexp', 'dnis_regexp', 'last_modified', 'denyregexp', 'invisible', 'ani_block', 'delta_price_unit', 'id', 'profit_control', 'service_id']	True	List([Enum('capacity', 'ani_regexp', 'name', 'ani_allow', 'profit', 'stop', 'delta_price', 'enabled', 'priority', 'name_context', 'context_id', 'dstregexp', 'dnis_regexp', 'last_modified', 'denyregexp', 'invisible', 'ani_block', 'delta_price_unit', 'id', 'profit_control', 'service_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('capacity', 'ani_regexp', 'name', 'ani_allow', 'profit', 'stop', 'delta_price',	field name on which result will

			'enabled', 'priority', 'name_context', 'context_id', 'dstregexp', 'dnis_regexp', 'last_modified', 'denyregexp', 'invisible', 'ani_block', 'delta_price_unit', 'id', 'profit_control', 'service_id': Enum('asc', 'desc', ''), min_length: 0)	be sorted
contexts	[]	False	List([Integer(gt: 0)], min_length: 0)	required context ids
invisible	False	True	Or(Bool, Null)	filter by visible/invisible
id	N/A	True	Integer(gt: 0)	dialpeer id
limit	0	True	Integer(gte: 0)	rows quantity per one request
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
dstregexp	N/A	True	String(allow_blank: True, max_length: 128)	
service_id	1	True	Integer(gt: 0)	service type filter(1 - voice, 2 - sms)
name	N/A	True	String(allow_blank: True)	first symbols of dialpeer name

Default fields:

Name	Type & Comments
capacity	String
ani_regexp	String

name	String
ani_allow	String
profit	Numeric or Null - threshold value of profit for dynamic routing (for voip only)
stop	Boolean
delta_price	numeric(16,5)
enabled	Boolean
priority	Int
name_context	String (dialpeer name with context name)
context_id	Int
dstregexp	String
dnis_regexp	String
last_modified	String
denyregexp	String
invisible	Boolean
ani_block	String
delta_price_unit	Int
id	Int
profit_control	Boolean - toggle to enable/disable dynamic routing in dialpeer (for voip only)
service_id	Int

2.98 dialpeers_tp_get

Docstring:

Returns information about dialpeers('dstregexp', 'denyregexp', 'ani_regexp', dialpeer status and so on) that connected with chosen termination point and chosen group.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {"limit": 101,
           "sort_by": [{"name": ""}],
           "group_id": 6,
           "contexts": [1],
           "offset": 0,
           "tp_id": 4,
           "fields": ["id", "name", "dstregexp", "priority", "enabled", "stop"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name", "dstregexp", "priority", "enabled", "stop"],
  "message": "OK",
  "data": [[6, "Test_DP", "^.*", 0, true, true]],
  "time_elapsed": 0.008475}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'dialpeer_name': 'String',
          'fields': ["Enum('priority', 'dstregexp', 'ani_regexp', 'name', 'denyregexp',
                           'stop', 'delta_price', 'enabled', 'id', 'dnis_regexp')"],
          'group_id': 'Integer',
          'limit': 'Integer',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('priority', 'dstregexp', 'ani_regexp', 'name', 'denyregexp',
                              'stop', 'delta_price', 'enabled', 'id', 'dnis_regexp')": "Enum('asc', 'desc', '')"}]
          },
          'tp_id': 'Integer'},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
		1		

service_id	N/A	True	Integer(gt: 0)	service type filter(1 - voice, 2 - sms)
group_id	N/A	False	Integer(gt: 0)	filter by certain group
fields	['priority', 'dstregex', 'ani_regex', 'name', 'denyregex', 'stop', 'delta_price', 'enabled', 'id', 'dnis_regex']	True	List([Enum('priority', 'dstregex', 'ani_regex', 'name', 'denyregex', 'stop', 'delta_price', 'enabled', 'id', 'dnis_regex')], min_length: 0)	required fields
tp_id	N/A	False	Integer(gt: 0)	filter by TP
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	rows quantity per one request
sort_by	[]	True	List([Enum('priority', 'dstregex', 'ani_regex', 'name', 'denyregex', 'stop', 'delta_price', 'enabled', 'id', 'dnis_regex'): Enum('asc', 'desc', '')}], min_length: 0)	column name on which result sorts
contexts	[]	False	List([Integer(gt: 0)], min_length: 0)	required contexts ids
dialpeer_name	N/A	True	String(allow_blank: True)	filter by first symbols of dialpeer name

Default fields:

Name	Type & Comments
priority	Int
dstregex	String
ani_regex	String

name	String - dialpeer's name
denyregexp	String
stop	Boolean
delta_price	numeric(16,5)
enabled	Boolean
id	Int
dnis_regexp	String

2.99 dialpeers_tps_get

Docstring:

Returns list of dialpeers binded with chosen TP or TPS binded with chosen dialpeer and parameters (priority and weight) of their relation.

Function can filter output by TP's name or dialpeer's name. Filter by TP name should be used in conjunction with dialpeer's id, filter by dialpeer name - in conjunction with TP's id.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "sort_by": [{"dialpeer_name": ""}],
    "contexts": [1],
    "tp_id": 4,
    "dialpeer_name": "",
    "fields": ["dialpeer_id", "dialpeer_name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["dialpeer_id", "dialpeer_name"],
  "message": "OK",
  "data": [[6, "Test_DP (CZ)"]],
  "time_elapsed": 0.008504
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'dialpeer_id': 'Integer',
          'dialpeer_name': 'String',
          'fields': ["Enum('weight', 'tp_id', 'priority', 'dialpeer_name', 'dialpeer_id', 'tp_name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('weight', 'tp_id', 'priority', 'dialpeer_name', 'dialpeer_id', 'tp_name')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer',
          'tp_name': 'String'},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	True	Integer(gt: 0)	filter by certain TP
sort_by	[]	True	List([{'Enum('weight', 'tp_id', 'priority', 'dialpeer_name', 'dialpeer_id', 'tp_name')': Enum('asc', 'desc', '')}], min_length: 0)	field on which list will be sorted
dialpeer_id	N/A	True	Integer(gt: 0)	filter by certain dialpeer
tp_name	N/A	True	String(allow_blank: True)	filter by first symbols of TP name
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	required contexts ids
fields	['weight', 'tp_id', 'priority', 'dialpeer_name', 'dialpeer_id', 'tp_name']	True	List([Enum('weight', 'tp_id', 'priority', 'dialpeer_name', 'dialpeer_id', 'tp_name')], min_length: 0)	requested fields
service_id	N/A	True	Integer(gt: 0)	service type filter(1 - voice, 2 - sms)

dialpeer_name	N/A	True	String(allow_blank: True)	filter by first symbols of dialpeer name
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	rows quantity per one request

Default fields:

Name	Type & Comments
weight	Int
tp_id	Int
priority	Int
dialpeer_name	String
dialpeer_id	Int
tp_name	String

2.100 email_list_get

Docstring:

Returns information about general email templates (owner independent email templates). In general this these templates are designed for notifications for owner of the system.

Possible types:

- "email_swap_report"
- "email_close_point_hand"
- "email_auto_invoice_create"
- "email_after_not_authorised"
- "email_guardian_blocked_destinations_report"
- "email_guardian_TT"
- "email_payment_del"

- "email_price_log_default"
- "email_wrong_credentials"
- "email_minimum_balance_change"
- "email_change_point_ip"

Example:

- Client Request:

```
{
  "session_id": "0796687af37294445fc61ab124668b00",
  "data": {
    "type": "email_after_not_authorised",
    "fields": [
      "type", "from", "to", "description", "use_template", "subject_template",
      "body_template", "copy", "enabled"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "type", "from", "to", "description", "use_template", "subject_template",
    "body_template",
    "copy", "enabled"],
  "message": "OK",
  "time_elapsed": 0.004426,
  "data": [
    ["email_after_not_authorised", "auth@speedflow.com", "gostar@speedflow.com, sergey@speedflow.com",
      "[NOTIFY] from, to, copy after not authorised login; {user} - user name, {ip_adress} - ip address, {date} - date and time.",
      true, "Not Authorised Login", "There was a successful login from not authorised IP address {ip_adress} by user {user} at {date}."],
    ["agostars@gmail.com", true]
  ]
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('to', 'from', 'description', 'use_template', 'body_template', 'subject_template', 'copy', 'type', 'enabled', 'email_type')"],
  'limit': 'Integer',
  'offset': 'Integer',
  'sort_by': [{"Enum('to', 'from', 'description', 'use_template', 'body_template', 'subject_template', 'copy', 'type', 'enabled', 'email_type')": "Enum('asc', 'desc', '')"}],
  'type': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
type	N/A	True	String(allow_blank: True)	filter by notification type. If absent in request then returns all available types.
fields	['to', 'from', 'description', 'use_template', 'body_template', 'subject_template', 'copy', 'type', 'enabled', 'email_type']	True	List([Enum('to', 'from', 'description', 'use_template', 'body_template', 'subject_template', 'copy', 'type', 'enabled', 'email_type')), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[{'type': ""}]	True	List([Enum('to', 'from', 'description', 'use_template', 'body_template', 'subject_template', 'copy', 'type', 'enabled', 'email_type'): Enum('asc', 'desc', "")]), min_length: 0)	field name on which result will be sorted

Default fields:

Name	Type & Comments
to	String
from	String
description	String
use_template	Boolean

body_template	String
subject_template	String
copy	String
type	String
enabled	Boolean
email_type	String

2.101 email_list_owners_get

Docstring:

Returns templates of the email notifications for owner with given id or for all owners if 'owner_id' not specified.
Those templates used by mediacore for sending email notifications which depends on the balance or company settings. In general this these templates are designed to inform clients about events which depends on the balance state, invoices etc.

Possible types:

- "email_close_point"
- "email_negative_limit"
- "email_negative_threshold"
- "email_financial_department"
- "email_traffic_report"
- "email_account_notification" - SoA report

Example:

- Client's Request:

```
{'data': {'fields': ['type',
                    'from',
                    'to',
                    'description',
                    'use_template',
                    'subject_template',
                    'body_template',
                    'copy',
                    'enabled',
                    'send_manager',
                    'send_billing',
                    'send_tech']},
```



```

    'limit': 1,
    'owner_id': 3},
    'session_id': u'fcde6e1a7894c3f24d2a6bcfe8cbfd0'
}

```

-Server Response:

```

{'data': [['email_close_point',
            'noc@test.com',
            'mail@test.com',
            '[NOTIFY] from, to, copy when point closed;',
            True,
            '(Lemgrove) Payment Request @client_name',
            'Dear @client_name.

```

Your account has been blocked due to insufficient funds.',

```

            True,
            False,
            False,
            False]],
    'fields': ['type',
               'from',
               'to',
               'description',
               'use_template',
               'subject_template',
               'body_template',
               'copy',
               'enabled',
               'send_manager',
               'send_billing',
               'send_tech'],
    'message': 'OK',
    'status': 200,
    'time_elapsed': 0.004426
}

```

Request structure and fields description:

```

{'data': {'balance_id': 'Integer',
          'fields': ["Enum('description', 'body_template', 'use_template', 'copy', 'send_billing', 'from', 'email_type', 'send_tech', 'enabled', 'to', 'subject_template', 'send_manager', 'type', 'owner_id')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'sort_by': [{"Enum('description', 'body_template', 'use_template', 'copy', 'send_billing', 'from', 'email_type', 'send_tech', 'enabled', 'to', 'subject_template', 'send_manager', 'type', 'owner_id')": "Enum('asc', 'desc', '')"}],
          'type': 'String'},
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
owner_id	N/A	True	Integer(gt: 0)	filter by owner id which templates are required to get
balance_id	N/A	True	Integer(gt: 0)	filter that allows to get owner's templates by balance id
type	N/A	True	String(allow_blank: True)	Type of notification email template.
fields	['description', 'body_template', 'use_template', 'copy', 'send_billing', 'from', 'email_type', 'send_tech', 'enabled', 'to', 'subject_template', 'send_manager', 'type', 'owner_id']	True	List([Enum('description', 'body_template', 'use_template', 'copy', 'send_billing', 'from', 'email_type', 'send_tech', 'enabled', 'to', 'subject_template', 'send_manager', 'type', 'owner_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'type': ""}]	True	List([Enum('description', 'body_template', 'use_template', 'copy', 'send_billing', 'from', 'email_type', 'send_tech', 'enabled', 'to', 'subject_template', 'send_manager', 'type', 'owner_id'): Enum('asc', 'desc', "")]), min_length: 0)	field name on which result will be sorted
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
------	-----------------

description	String
body_template	String
use_template	Boolean
copy	String
send_billing	Boolean
from	String (if field from in template is empty or absent then returns email from main settings/Owner Info)
email_type	String
send_tech	Boolean
enabled	Boolean
to	String
subject_template	String
send_manager	Boolean
type	String
owner_id	Int

2.102 email_list_set

Docstring:

Allows to change template parameters of the given email notification, which system sends automatically to client or mediacore owner after specific event.

Example:

- Client Request:

```
{"session_id":"1f5adcacaa938949b74af1e14b8a0aab",
  "data":{"to":"ikex@speedflow.com",
```


				"email_month_duration"
from	N/A	False	String(allow_blank: False, regex: "(?P^[-#\$%&'*/+=?^_`{} ~0-9A-Z]+(\.[-#\$%&'*/+=?^_`{} ~0-9A-Z]+)* ^"([\001-\010\013\014\016-\037#-\\[\]-\177] \[\001-011\013\014\016-\177])*)"@(?P(?:[A-Z0-9](?:[A-Z0-9]{0,61}[A-Z0-9])?\.)+(?:[A-Z]{2,6}\.? [A-Z0-9]{2,\.})\$) ((25[0-5] 2[0-4]\d [0-1]? \d? \d)(\. (25[0-5] 2[0-4]\d [0-1]? \d? \d)){3})\\$)", max_length: 254)	sender email address or addresses separated by comma
copy	N/A	True	EmailCSV(blank allowed, regex: "^(^[-#\$%&'*/+=?^_`{} ~0-9A-Z]+(\.[-#\$%&'*/+=?^_`{} ~0-9A-Z]+)* ^"([\001-\010\013\014\016-\037#-\\[\]-\177] \[\001-011\013\014\016-\177])*)"@(((?:[A-Z0-9](?:[A-Z0-9]{0,61}[A-Z0-9])?\.)+(?:[A-Z]{2,6}\.? [A-Z0-9]{2,\.})\$) ((25[0-5] 2[0-4]\d [0-1]? \d? \d)(\. (25[0-5] 2[0-4]\d [0-1]? \d? \d)){3})\\$")	email address to which message copy will be sent
body_template	N/A	False	String(allow_blank: False)	defines template body with help of html tags and special variables. Which variables available is depend from chosen "type" of notification
subject_template	N/A	False	String(allow_blank: False)	defines template subject with help of special variables. Which variables available is depend from chosen "type" of notification
to	N/A	False	EmailCSV(regex: "^(^[-#\$%&'*/+=?^_`{} ~0-9A-Z]+(\.[-#\$%&'*/+=?^_`{} ~0-9A-Z]+)* ^"([\001-\010\013\014\016-\037!#-\\[\]-\177] \[\001-011\013\014\016-\177])*)"@(((?:[A-Z0-9](?:[A-Z0-9]{0,61}[A-Z0-9])?\.)+(?:[A-Z]{2,6}\.? [A-Z0-9]{2,\.})\$) ((25[0-5] 2[0-4]\d [0-1]? \d? \d)(\. (25[0-5] 2[0-4]\d [0-1]? \d? \d)){3})\\$")	recipients email address or addresses separated by comma
enabled	N/A	True	Bool	template status

Name	Default value	Optional	Type	Description
type	N/A	False	String(allow_blank: False)	Type of notification. There is available following notifications: 'email_close_point', 'email_negative_limit' and 'email_negative_threshold'.
enabled	N/A	True	Bool	template status used or not
to	N/A	False	EmailCSV(regex: "(^[-#%&'*/+=?^_`{}] ~0-9A-Z]+(\\.[-#%&'*/+=?^_`{}] ~0-9A-Z]+)* ^"([\\001-\\010\\013\\014\\016-\\037#-\\[\\]-\\177])\\[\\001-011\\013\\014\\016-\\177])*")@((?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\\.[A-Z0-9]{2,}\\. (?\$)\\[(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)\\.((25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)){3}\\]\$")	recipients email address or email addresses separated by comma
send_billing	N/A	True	Bool	if True then billing email addresses of company will be added in "to" field
copy	N/A	True	EmailCSV(blank allowed, regex: "(^[-#%&'*/+=?^_`{}] ~0-9A-Z]+(\\.[-#%&'*/+=?^_`{}] ~0-9A-Z]+)* ^"([\\001-\\010\\013\\014\\016-\\037#-\\[\\]-\\177])\\[\\001-011\\013\\014\\016-\\177])*")@((?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\\.[A-Z0-9]{2,}\\. (?\$)\\[(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)\\.((25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)){3}\\]\$")	email address to which message copy will be sent
owner_id	N/A	False	Integer(gt: 0)	filter by owner id which templates are required to change
body_template	N/A	False	String(allow_blank: False)	defines template body with help of html tags and special variables. Which variables available is depend from chosen "type" of notification

send_manager	N/A	True	Bool	if True then user's email address (who is assigned to company as manager) will be added in "to" field
subject_template	N/A	False	String(allow_blank: False)	defines template subject with help of special variables. Which variables available is depend from chosen "type" of notification
from	N/A	False	String(allow_blank: False, regex: "(?P^[-#\$\$%&'*/+=?^_`{}] ~0-9A-Z]+(\.[^-#\$\$%&'*/+=?^_`{}] ~0-9A-Z]+)* ^"([\\001-\\010\\013\\014\\016-\\037!#-\\\\\\-\\177]) \\[\\001-011\\013\\014\\016-177])*")@(?P(?:[A-Z0-9])(?:[A-Z0-9-]{0,61}[A-Z0-9])?.)+(?:[A-Z]{2,6}\\.[A-Z0-9-]{2,}\\.)?\$) \\[(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\. (25[0-5] 2[0-4]\\d [0-1]?\\d?\\d))){3}\\]" , max_length: 254)	sender email address or email addresses separated by comma
send_tech	N/A	True	Bool	if True then tech emails of company will be added in "to" field

2.104 email send

Docstring:

```
If "type" is "invoice" then calls API function 'invoice_email_send'.
If "type" is "balance" then add balance report email to the table api.web mail.
```

In other cases creates connection to SMTP server specified in the owner setting and send message at once it get SMTP connection.

Example:

- Client's Request:

```
{
  "session_id": "fcde6e1a7894c3f24d2a6bcfe8cbfd0",
  "data": {
    "to": ["test@gmail.com"],
    "body": "Please, find attached file.",
    "subject": "Balance",
    "attach": ["172"],
    "from": "test@test.com",
    "type": "balance"
  }
}
```


`_balance_email_send` ([data validation](#)) ([default fields](#))

Request structure and fields description:

```
{'data': {'attach': ['String'],
          'balance_id': 'Integer',
          'bcc': ['String'],
          'body': 'String',
          'client_id': 'Integer',
          'copy': ['String'],
          'email_id': 'Integer',
          'from': 'String',
          'mime_type': 'String',
          'owner_id': 'Integer',
          'subject': 'String',
          'to': ['String'],
          'type': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
from	N/A	False	String(allow_blank: False)	email sender address
to	[]	True	List([String(allow_blank: True)], min_length: 0)	list that contains recipients email addresses
attach	N/A	True	List([String(allow_blank: False)], min_length: 0)	list that contains files that should be attached
subject	N/A	False	String(allow_blank: True)	message subject
copy	[]	True	List([String(allow_blank: True)], min_length: 0)	list that contains email addresses of copy recipients

body	N/A	False	String(allow_blank: True)	message body
balance_id	N/A	True	Integer(gt: 0)	used to define client_id and then owner_id, from which SMTP settings should taken
type	MC_notification	True	String(allow_blank: False)	if "invoice" - 'invoice_email_send' will be called. if "balance" - 'balance_email_send'
client_id	N/A	True	Integer(gt: 0)	used to define owner_id, from which SMTP settings should be taken. If "client_id" and "balance_id" are absent in query and current user isn't client function takes SMTP settings of owner with 'id'=1
mime_type	html	True	String(allow_blank: False)	content type("html" or "plain") for "body" and "subject". "html" allows to use HTML tags in template for text formatting, "plain" allows to use escape sequences(without quotes) ' ', ' ' etc.
email_id	0	True	Integer(gte: 0)	email id in DB table (for update dt_sent field in DB)
bcc	[]	True	List([String(allow_blank: True)], min_length: 0)	list that contains email addresses of BCC (Blind Carbon Copy) recipients, this mail field is invisible for other recipients
owner_id	N/A	True	Integer(gt: 0)	to get SMTP connection of chosen owner

_balance_email_send

```
{'data': {'attach': ['String'],
           'balance_id': 'Integer',
           'body': 'String',
           'client_id': 'Integer',
           'copy': ['String'],
           'email_id': 'Integer',
           'from': 'String',
           'mime_type': 'String',
           'owner_id': 'Integer',
           'subject': 'String',
           'to': ['String'],
           'type': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
copy	[]	True	List([String(allow_blank: True)], min_length: 0)	list that contains email addresses of copy recipients
subject	N/A	False	String(allow_blank: True)	
body	N/A	False	String(allow_blank: True)	
type	MC_notification	True	String(allow_blank: False)	if 'invoice' - 'invoice_email_send' will be called. if 'balance' - 'balance_email_send'
attach	N/A	True	List([String(allow_blank: False)], min_length: 0)	list that contains files that should be attached
balance_id	N/A	False	Integer(gt: 0)	used to define client_id and then owner_id, from which SMTP settings should taken
client_id	N/A	True	Integer(gt: 0)	used to define owner_id, from which SMTP settings should be taken. If "client_id" and "balance_id" are absent in query and current user isn't client function takes SMTP settings of owner with "id"="1"
mime_type	html	True	String(allow_blank: False)	content type("html" or "plain") for "body" and "subject". "html" allows to use HTML tags in template for text formatting, "plain" allows to use escape sequences (without quotes) " ", " " etc.
from	N/A	False	String(allow_blank: False)	email sender address
email_id	0	True	Integer(gte: 0)	email id in DB table (for update dt_sent field in DB)
to	[]	True	List([String(allow_blank: True)], min_length: 0)	list that contains recipients email addresses
owner_id	N/A	True	Integer(gt: 0)	to get SMTP connection of chosen owner

2.105 email_settings_get

Docstring:

Returns current email server options. By defaults - mail server is localhost.

Example:

- Client Request:

```
{"session_id": "1f5adcacaa938949b74af1e14b8a0aab",
 "data": {"fields": ["box_host",
                    "box_port",
                    "box_user",
                    "box_password",
                    "box_ssl",
                    "email"]}}
```

- Server Response:

```
{"status": 200,
 "fields": ["box_host",
            "box_port",
            "box_user",
            "box_password",
            "box_ssl",
            "email"],
 "message": "OK",
 "data": [{"mailserv.speedflow.com",
            465,
            "billing-kh",
            "Chedyn-56",
            true,
            "cz5080@speedflow.com"}],
 "time_elapsed": 0.003367}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('email', 'support_email', 'box_port', 'box_password', 'guardian_email', 'box_ssl', 'box_host', 'owner_id', 'box_user')"],
              'limit': 'Integer',
              'offset': 'Integer',
              'owner_id': 'Integer',
              'sort_by': [{"Enum('email', 'support_email', 'box_port', 'box_password', 'guardian_email', 'box_ssl', 'box_host', 'owner_id', 'box_user')": "Enum('asc', 'desc', '')"}]},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['email', 'support_email', 'box_port', 'box_password', 'guardian_email', 'box_ssl', 'box_host', 'owner_id', 'box_user']	True	List([Enum('email', 'support_email', 'box_port', 'box_password', 'guardian_email', 'box_ssl', 'box_host', 'owner_id', 'box_user']), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('email', 'support_email', 'box_port', 'box_password', 'guardian_email', 'box_ssl', 'box_host', 'owner_id', 'box_user'): Enum('asc', 'desc', "")}], min_length: 0)	field name on which result will be sorted
owner_id	1	True	Integer(gt: 0)	id of owner which settings is required to get. Default id is 1.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
email	String, owner's default email
support_email	String, support email
box_port	Int
box_password	String
guardian_email	String, owner's guardian email
box_ssl	Boolean

box_host	String
owner_id	Int
box_user	String

2.106 email_settings_set

Docstring:

Allows to change mail server SMTP settings of chosen owner for sending emails from mediacore.

Example:

- Client Request:

```
{
  "session_id": "1f5adcacaa938949b74af1e14b8a0aab",
  "data": {
    "box_host": "check.lemgrove.com",
    "guardian_email": "guardian@test.com",
    "box_user": "billing@test.com",
    "box_port": 465,
    "email": "billing@test.com",
    "owner_id": 3,
    "box_ssl": true
  }
}
```

Request structure and fields description:

```
{'data': {'box_host': 'String',
          'box_password': 'String',
          'box_port': 'Integer',
          'box_ssl': 'Bool',
          'box_user': 'String',
          'email': 'String',
          'guardian_email': 'String',
          'owner_id': 'Integer',
          'support_email': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
box_host	N/A	False	String(allow_blank: False)	mail server domain address
box_user	N/A	False	String(allow_blank: False)	user name for SMTP authorization
support_email	N/A	True	String(allow_blank: False, regex: "(?P^[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+(\\[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+)* ^"([001-010013014016-037#-\\[-177])\\[\\001-011013014016-177])*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?.\\.)+(?:[A-Z]{2,6}\\.[A-Z0-9-]{2,}\\.)?\$) \\((25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\. (25[0-5] 2[0-4]\\d [0-1]?\\d?\\d))){3}\\.\$", max_length: 254)	Owner's support email for copy in TT.
box_ssl	N/A	False	Bool	use SSL or not
guardian_email	N/A	True	String(allow_blank: False, regex: "(?P^[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+(\\[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+)* ^"([001-010013014016-037#-\\[-177])\\[\\001-011013014016-177])*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?.\\.)+(?:[A-Z]{2,6}\\.[A-Z0-9-]{2,}\\.)?\$) \\((25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\. (25[0-5] 2[0-4]\\d [0-1]?\\d?\\d))){3}\\.\$", max_length: 254)	Owner's guardian email for TT emails.
box_port	N/A	False	Integer()	SMTP port
email	N/A	True	String(allow_blank: False, regex: "(?P^[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+(\\[-#\$\$%&'*/+/?^_`{} ~0-9A-Z]+)* ^"([001-010013014016-037!#-\\[-177])\\[\\001-011013014016-177])*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?.\\.)+(?:[A-Z]{2,6}\\.[A-Z0-9-]{2,}\\.)?\$) \\((25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\. (25[0-5] 2[0-4]\\d [0-1]?\\d?\\d))){3}\\.\$", max_length: 254)	Owner's default email. This address is inserted into 'from' field of new email.
box_password	N/A	True	String(allow_blank: False)	user password for SMTP authorization
owner_id	N/A	False	Integer(gt: 0)	owner which settings require to change

2.107 export_price_list

Docstring:

Returns prices in specified format belonging to chosen tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Use only single of types - "current", "actual_prices" or "future", not all together.

Example:

- Client Request

```
{
  "session_id": "f28a3d96a0ecf3b1bbf44feed8b0c952",
  "data": {
    "sort_by": [
      { "destination": "" },
      { "added": "desc" }
    ],
    "tariff_id": 4201,
    "is_tp": false,
    "format": "csv",
    "delimiter": ",",
    "export_fields": [
      "enabled", "allowed", "description", "destination", "price", "priority",
      "date_start", "date_end", "time_start", "time_end", "sun",
      "mon", "tue", "wed", "thu", "fri", "sat", "min_len", "max_len", "tariff_id", "country", "actual_price_id", "mcc", "mnc",
      "destination": "",
      "fields": [ "link", "filename", "size" ]
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [ "link", "filename", "size" ],
  "message": "OK",
  "data": [ [ "http://API_IP:8070/cdn/download/tariff_export/
filename.csv", "filename.csv", 0 ] ],
  "time_elapsed": 26.300286
}
```

Request structure and fields description:

```
{
  'data': {
    'actual_prices': 'Bool',
    'allowed': 'Bool',
    'current': 'Bool',
    'delimiter': "Enum(',', ' ;')",
    'destination': 'String',
    'enabled': 'Bool',
    'export_fields': [
      "Enum('ci', 'date_start', 'date_end', 'time_end', 'max_len',
      'origin_country', 'allowed', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri',
      'destination', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len',
      'price', 'mon', 'tue', 'tariff_id', 'mc', 'country', 'thu', 'enabled', 'connection_price')"],
    'fields': [ "Enum('size', 'link', 'filename')"],
    'format': "Enum('csv', 'xlsx')",
  }
}
```



```

        'is_tp': 'Bool',
        'latest': 'Bool',
        'sort_by': [{"Enum('ci', 'date_start', 'date_end', 'time_end', 'max_len', 'origin_country', 'allowed', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'mon', 'tue', 'tariff_id', 'mc', 'country', 'thu', 'enabled', 'connection_price')": "Enum('asc', 'desc', '')"}],
        'tariff_id': 'Integer'},
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
format	N/A	False	Enum('csv', 'xlsx')	format in which result will be saved
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current, date_start=
allowed	N/A	True	Bool	if "true" returns allowed prices
tariff_id	N/A	True	Integer(gt: 0)	tariff id
actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current, date_start=
is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
delimiter	;	False	Enum(',', ';')	delimiter for csv format. Ignored if

				format is xlsx
enabled	N/A	True	Bool	if "true" returns enabled prices
latest	False	True	Bool	If "true" returns last uploaded prices by destination. May cause timeout if tariff_id is not passed!
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
export_fields	['ci', 'date_start', 'date_end', 'time_end', 'max_len', 'origin_country', 'allowed', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'mon', 'tue', 'tariff_id', 'mc', 'country', 'thu', 'enabled', 'connection_price']	True	List([Enum('ci', 'date_start', 'date_end', 'time_end', 'max_len', 'origin_country', 'allowed', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'mon', 'tue', 'tariff_id', 'mc', 'country', 'thu', 'enabled', 'connection_price')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	N/A	True	List([Enum('ci', 'date_start', 'date_end', 'time_end', 'max_len', 'origin_country', 'allowed', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'mon', 'tue', 'tariff_id', 'mc', 'country', 'thu', 'enabled', 'connection_price'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
fields	['size', 'link', 'filename']	True	List([Enum('size', 'link', 'filename')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
size	Int
link	String
filename	String

2.108 export_price_list_jurisdictional

Docstring:

Returns prices in specified format belonging to chosen tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Use only single of types - "current", "actual_prices" or "future", not all together.

Example:

- Client Request

```
{
  "session_id": "f28a3d96a0ecf3b1bbf44feed8b0c952",
  "data": {
    "sort_by": [{"destination": ""}, {"added": "desc"}],
    "tariff_id": 4201,
    "is_tp": false,
    "format": "csv",
    "delimiter": ",",
    "export_fields": [
      "enabled", "allowed", "description", "destination", "price", "priority",
      "date_start", "date_end", "time_start", "time_end", "sun",
      "mon", "tue", "wed", "thu", "fri", "sat", "min_len", "max_len", "tariff_id", "country", "actual_price_id", "mcc", "mnc"],
    "destination": "",
    "fields": ["link", "filename", "size"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["link", "filename", "size"],
  "message": "OK",
  "data": [
    ["http://API_IP:8070/cdn/download/tariff_export/filename.csv", "filename.csv", 0]
  ],
  "time_elapsed": 26.300286
}
```

Request structure and fields description:

```
{'data': {'actual_prices': 'Bool',
          'allowed': 'Bool',
          'current': 'Bool',
          'delimiter': "Enum(',', ';;')",
          'destination': 'String',
          'enabled': 'Bool',
          'export_fields': ["Enum('ci', 'added', 'ft', 'price_intrastate', 'date_end',
'price_interstate', 'time_end', 'tariff_id', 'mon', 'allowed', 'destination', 'tue', 's
at', 'description', 'actual_price_id', 'time_start', 'mc', 'wed', 'price_indeterminate'
, 'sun', 'fri', 'date_start', 'enabled', 'connection_price', 'priority', 'lata', 'count
ry', 'ocn', 'thu')"],
          'fields': ["Enum('size', 'link', 'filename')"],
          'format': "Enum('csv', 'xlsx')",
          'is_tp': 'Bool',
          'lata': 'String',
          'ocn': 'String',
          'sort_by': [{"Enum('ci', 'added', 'ft', 'price_intrastate', 'date_end', 'pric
e_interstate', 'time_end', 'tariff_id', 'mon', 'allowed', 'destination', 'tue', 'sat',
'description', 'actual_price_id', 'time_start', 'mc', 'wed', 'price_indeterminate', 'su
n', 'fri', 'date_start', 'enabled', 'connection_price', 'priority', 'lata', 'country',
'ocn', 'thu')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optiona l	Type	Description
format	N/A	False	Enum('csv', 'xlsx')	format in which result will be saved
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current , date_start=
allowed	N/A	True	Bool	if "true" returns allowed prices
tariff_id	N/A	True	Integer(gt: 0)	tariff id

actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current , date_start=
is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
delimiter	;	False	Enum(',', ';')	delimiter for csv format. Ignored if format is xlsx
enabled	N/A	True	Bool	if "true" returns enabled prices
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
ocn	N/A	True	String(allow_blank: True, regex: "^%?[0-9A-Fa-f]{0,4}")	
lata	N/A	True	String(allow_blank: True, regex: "^%?[0-9]{0,5}")	
fields	['size', 'link', 'filename']	True	List([Enum('size', 'link', 'filename')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	N/A	True	List([Enum('ci', 'added', 'ft', 'price_intrastate', 'date_end', 'price_interstate', 'time_end', 'tariff_id', 'mon', 'allowed', 'destination', 'tue', 'sat', 'description', 'actual_price_id', 'time_start', 'mc', 'wed', 'price_indeterminate', 'sun', 'fri', 'date_start', 'enabled',	field on which result will be sorted. Result can be sorted ascending or descending.

			'connection_price', 'priority', 'lata', 'country', 'ocn', 'thu'): Enum('asc', 'desc', ")}], min_length: 0)	
export_fields	['ci', 'added', 'ft', 'price_intrastate', 'date_end', 'price_interstate', 'time_end', 'tariff_id', 'mon', 'allowed', 'destination', 'tue', 'sat', 'description', 'actual_price_id', 'time_start', 'mc', 'wed', 'price_indeterminate', 'sun', 'fri', 'date_start', 'enabled', 'connection_price', 'priority', 'lata', 'country', 'ocn', 'thu']	True	List([Enum('ci', 'added', 'ft', 'price_intrastate', 'date_end', 'price_interstate', 'time_end', 'tariff_id', 'mon', 'allowed', 'destination', 'tue', 'sat', 'description', 'actual_price_id', 'time_start', 'mc', 'wed', 'price_indeterminate', 'sun', 'fri', 'date_start', 'enabled', 'connection_price', 'priority', 'lata', 'country', 'ocn', 'thu')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
size	Int
link	String
filename	String

2.109 export_price_list_sms

Docstring:

Returns prices in specified format belonging to chosen tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Use only single of types - "current", "actual_prices" or "future", not all together.

Example:

- Client Request

```
{"session_id": "f28a3d96a0ecf3b1bbf44feed8b0c952",
  "data": {"sort_by": [{"destination": ""}, {"added": "desc"}],
```

```

        "tariff_id":4201,
        "is_tp":false,
        "format":"csv",
        "delimiter":",",
        "export_fields":
["enabled","allowed","description","destination","price","priority",
        "date_start","date_end","time_start","time_end","sun
", "mon", "tue", "wed", "thu", "fri",
        "sat", "min_len", "max_len", "tariff_id", "country", "act
ual_price_id", "mcc", "mnc"],
        "destination": "",
        "fields":["link", "filename", "size"]}
    }

```

- Server Response

```

{"status": 200,
 "fields": ["link", "filename", "size"],
 "message": "OK",
 "data": [["http://API_IP:8070/cdn/download/tariff_export/
filename.csv", "filename.csv", 0]],
 "time_elapsed": 26.300286}

```

Request structure and fields description:

```

{'data': {'actual_prices': 'Bool',
 'allowed': 'Bool',
 'current': 'Bool',
 'delimiter': "Enum(',', ' ;')",
 'destination': 'String',
 'enabled': 'Bool',
 'export_fields': ["Enum('date_start', 'date_end', 'time_end', 'max_len', 'all
owed', 'mnc', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination
', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'mon', 'tue', 'tariff
_id', 'country', 'thu', 'enabled')"],
 'fields': ["Enum('size', 'link', 'filename')"],
 'format': "Enum('csv', 'xlsx')",
 'is_tp': 'Bool',
 'network_code': 'String',
 'sort_by': [{"Enum('date_start', 'date_end', 'time_end', 'max_len', 'allowed
', 'mnc', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'p
riority', 'added', 'description', 'min_len', 'price', 'mcc', 'mon', 'tue', 'tariff_id',
'country', 'thu', 'enabled')": "Enum('asc', 'desc', '')"}],
 'tariff_id': 'Integer'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optiona l	Type	Description
format	N/A	False	Enum('csv', 'xlsx')	format in which

				result will be saved
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current, date_start=
allowed	N/A	True	Bool	if "true" returns allowed prices
tariff_id	N/A	True	Integer(gt: 0)	tariff id
actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current, date_start=
is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
delimiter	;	False	Enum(',', ';')	delimiter for csv format. Ignored if format is xlsx
enabled	N/A	True	Bool	if "true" returns enabled prices
network_code	N/A	True	String(allow_blank: False, min_length: 3, max_length: 6)	filter by network code(mcc+mnc)
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
sort_by	N/A	True	List([Enum('date_start', 'date_end', 'time_end', 'max_len', 'allowed', 'mnc', 'sat', 'actual_price_id', 'time_start',	field on which result will be sorted. Result can be sorted ascending or

			'wed', 'sun', 'fri', 'destination', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'mon', 'tue', 'tariff_id', 'country', 'thu', 'enabled'): Enum('asc', 'desc', ")}], min_length: 0)	descending.
fields	['size', 'link', 'filename']	True	List([Enum('size', 'link', 'filename')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
export_fields	['date_start', 'date_end', 'time_end', 'max_len', 'allowed', 'mnc', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'mon', 'tue', 'tariff_id', 'country', 'thu', 'enabled']	True	List([Enum('date_start', 'date_end', 'time_end', 'max_len', 'allowed', 'mnc', 'sat', 'actual_price_id', 'time_start', 'wed', 'sun', 'fri', 'destination', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'mon', 'tue', 'tariff_id', 'country', 'thu', 'enabled')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
size	Int
link	String
filename	String

2.110 fin_summary_get

Docstring:

Returns financial report (for voice traffic) by chosen time period and company(ies) with statistical information (common duration of calls, duration with considering prices cost counting parameters, asr, acd, attempts statistic, cost in/out, profit, delta price values combining by destination codes, points or

companies).

Report results are ordered by "company", "type" "point_name", "country" columns

For each client, type (inbound or outbound) and point (this is correct for all report types excluding type 5)

row with "Total" result (column "tot") is being added after each group of "company" and "type".

For the report type 5 only rows with "tot" = "Total" will be returned.

For type 6 and 7 rows with "Total" won't be returned by functions because each row already contains total result.

There are time interval limitations for each "type" in the API configuration.

If "type"=6 function returns list of manager ids and names("managers" param) for which financial

data is available in case of interval has been exceeded.

For other types function returns list of companies ids and names("clients" param) for which financial

data is available in case of interval has been exceeded.

If there are no clients or managers available then returns empty data and required fields list.

By defaults limits are following:

1) for "type" 1, 2 and 3 - 2 days

2) for "type" 4, 5, 6, 7 and 8 - 7 days

Function provides for each grouped data "inbound" and "outbound" parts if any ("type" field in resulted data)

and "direction":0.

"inbound" - OP traffic

"outbound" - TP traffic

NOTE! "agg_last_time" field - defines aggregation last time.

Example:

1) for type "4" - group by point

- Client Request

```
{"session_id":"f345e24573e77cadffbc28a11ae64af4",
  "data":{"time_zone":"0:00",
    "session_id":"f345e24573e77cadffbc28a11ae64af4",
    "currency_id":1,
    "end_date":"2014-01-13T23:59:59",
    "codes_exc":[""],
    "direction":0,
    "codes_inc":[""],
    "companies":[5],
    "type":4,
    "start_date":"2014-01-01T00:00:00",
    "fields":["is_closed","point_name","duration",
      "calc_duration","attempts","sa","asr","acd",
      "pdd","sc","op_sum","tp_sum","delta_price",
      "delta_profit","company","type","tot"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["is_closed", "point_name", "duration", "calc_duration",
    "attempts", "sa", "asr", "acd", "pdd", "sc", "op_sum",
    "tp_sum", "delta_price", "delta_profit", "company",
```

```

        "type", "tot"],
    "message": "OK",
    "data": [[false, "H_in", 735396, 735396, 559118, 3828, "0.68%",
        192, "2.5", "27.22%", "490.26382", "429.00555",
        "61.25827", "14.28%", "H_From", "inbound", ""],
        [false, "", 735396, 735396, 559118, 3828, "0.68%", 192,
        "2.5", "27.22%", "490.26382", "429.00555", "61.25827",
        "14.28%", "H_From", "inbound", "Total"]],
    "time_elapsed": 0.489811
}

```

Request structure and fields description:

```

{'data': {'acd_less_than': 'Integer',
    'acd_more_than': 'Integer',
    'asr_less_than': 'Float',
    'asr_more_than': 'Float',
    'codes_exc': ['String'],
    'codes_inc': ['String'],
    'companies': ['Integer'],
    'context_id': 'Integer',
    'currency_id': 'Integer',
    'direction': 'Integer',
    'end_date': 'String',
    'fields': ["Enum('op_sum', 'tp_sum', 'delta_price', 'duration', 'total', 'is_
closed', 'country_group', 'destination', 'tot', 'traffic_share', 'point_name', 'ner', '
asr', 'direction', 'origin_country', 'company', 'calc_duration', 'op_price', 'manager',
'acd', 'country', 'tp_price', 'attempts', 'delta_profit', 'sc', 'sa', 'pdd')"],
    'groups': ['Integer'],
    'managers': ['Integer'],
    'min_attempts': 'Integer',
    'owner_id': 'Integer',
    'profit_less_than': 'Float',
    'profit_more_than': 'Float',
    'session_id': 'String',
    'sort_by_quality': 'Bool',
    'start_date': 'String',
    'time_zone': 'String',
    'type': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
asr_less_than	N/A	True	Float(gt: 0.0)	filter by asr, returns rows with asr less than specified
owner_id	0	True	Integer(gte: 0)	filter by owner. For 0 - all

				owners
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2}\$")	requires if company works in timezone different from GMT+0
profit_less_than	N/A	True	Float()	filter by profit max threshold
acd_less_than	N/A	True	Integer(gt: 0)	filter by acd, returns rows with acd less than specified
fields	['op_sum', 'tp_sum', 'delta_price', 'duration', 'total', 'is_closed', 'country_group', 'destination', 'tot', 'traffic_share', 'point_name', 'ner', 'asr', 'direction', 'origin_country', 'company', 'calc_duration', 'op_price', 'manager', 'acd', 'country', 'tp_price', 'attempts', 'delta_profit', 'sc', 'sa', 'pdd']	True	List([Enum('op_sum', 'tp_sum', 'delta_price', 'duration', 'total', 'is_closed', 'country_group', 'destination', 'tot', 'traffic_share', 'point_name', 'ner', 'asr', 'direction', 'origin_country', 'company', 'calc_duration', 'op_price', 'manager', 'acd', 'country', 'tp_price', 'attempts', 'delta_profit', 'sc', 'sa', 'pdd')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
asr_more_than	N/A	True	Float(gte: 0.0)	filter by asr, returns rows with asr more than specified
profit_more_than	N/A	True	Float()	filter by profit min threshold
direction	0	True	Integer(gte: 0)	display only incoming(by OPs, 1) or outgoing(TPs, 2) traffic, For 0 - both directions
groups	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by group id. Currently only first groupid from the list is being taken into account.
min_attempts	0	True	Integer(gte: 0)	display rows where attempts more than

				"min_attempts"
session_id	N/A	True	String(allow_blank: False)	by defaults from user's session
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
type	N/A	False	Integer(gte: 1, lte: 9)	report type id. 1 - Individual report for each destination, price, point and company (inbound and outbound), 2 - by destination name, 3 - by destination name and prices, 4 - by points, 5 - total for all companies, 6 - by manager, 7 - by companies, 8 - by country, 9 - by company by country
codes_inc	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that should be present in reply
codes_exc	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that shouldn't be present in reply
sort_by_quality	False	True	Bool	sort by total quality and company name for types 1, 2, 3, 4, 7
end_date	N/A	False	String(allow_blank: False)	
companies	[0]	True	List([Integer(gte: 0)], min_length: 0)	filter by company ids. [0] means all companies. Currently only first company id from the list isbeing taken into account.
acd_more_than	N/A	True	Integer(gte: 0)	filter by acd, returns rows with acd more than specified
context_id	N/A	True	Integer(gt: 0)	filter by context

currency_id	N/A	True	Integer(gt: 0)	display prices and cost in chosen currency. 'By defaults from user's session
start_date	N/A	False	String(allow_blank: False)	

Default fields:

Name	Type & Comments
op_sum	String or Null (Null if hide_owners_profit enabled and different manager)
tp_sum	String or Null (Null if hide_owners_profit enabled and different manager)
delta_price	String or Null (Null if hide_owners_profit enabled and different manager)
duration	Int
total	String ('Total', 'TotalCountry' for type 9 grouped by country within single client or " for not total)
is_closed	Boolean (if True - dest is closed, unavailable for 'type' 5)
country_group	String
destination	String
tot	String - like as 'total' field (DEPRECATED, backport)
traffic_share	String
point_name	String
ner	String
asr	String
direction	Int (1 - inbound, 2 - outbound)

origin_country	String
company	String
calc_duration	Int
op_price	String or Null (Null if hide_owners_profit enabled and different manager)
manager	String
acd	Int
country	String
tp_price	String or Null (Null if hide_owners_profit enabled and different manager)
attempts	Int
delta_profit	String or Null (Null if hide_owners_profit enabled and different manager)
sc	String
sa	Int - quantity of successful attempts
pdd	numeric - post dial delay in seconds

2.111 firewall_control_add

Docstring:

Adds chosen IP address or subnet to the firewall blocking rules on given RTP server or on all enabled media servers to block given IP address or whole subnet.

Such rules allow to prevent any kind of traffic from undesirable IP addresses.

Example:

- Client Request

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "hosts": ["194.165.82.80"],
    "ips": ["1.2.3.4"]
  }
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.271727}
```

Request structure and fields description:

```
{'data': {'description': 'String', 'hosts': ['String'], 'ips': ['String']},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
ips	N/A	False	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?\$"), min_length: 0)	IP addresses list or whole network(in format IP/mask) that need to block. Example: [193.100.24.13/24] that need to block. IP address can't belongs to internal network(networks like 192.168.0.0/16, 10.0.0.0/8 or 172.16.0.0/12), localhost(127.x.x.x), 0.0.0.0 or 255.255.255.255.
description	N/A	True	String(allow_blank: False, max_length: 60)	
hosts	N/A	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?\$"), min_length: 0)	IP address of RTP server to which need add rule, if absent in request - adds rule to all RTP hosts.

2.112 firewall_control_del

Docstring:

Deletes blocking rule from firewall on specified RTP server or in all enabled RTP servers.

If rule cannot be deleted on any host (host in unavailable or IP is already abs

ent)

returns 304 code with a corresponding message, containing host's address and reason.

Example:

- Client Request

```
{"session_id": "7a1fe5eb3650e7f8908569c676fdec59",  
  "data": {"host": "194.165.82.80",  
           "ips": ["1.2.3.4"]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.275096}
```

Request structure and fields description:

```
{'data': {'description': 'String', 'hosts': ['String'], 'ips': ['String']},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
ips	N/A	False	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/((([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?)\$")], min_length: 0)	IP addresses list or whole network(in format IP/mask) that need to block. Example: [193.100.24.13/24] that need to block. IP address can't belongs to internal network(networks like 192.168.0.0/16, 10.0.0.0/8 or 172.16.0.0/12), localhost(127.x.x.x), 0.0.0.0 or 255.255.255.255.
description	N/A	True	String(allow_blank: True)	description if any. If description added to the rule then it should be provided.
hosts	[]	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/((([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?)\$")], min_length: 0)	RTP server address in which need to delete rule, if absent - removes IP from all RTP hosts.

2.113 firewall_control_get

Docstring:

Returns list of blocked IP addresses on specified RTP host or on all enabled servers of the system.

For each IP address added to the firewall function also returns description () and list of points (point id, name and is_op (True/False)) to which given IP address assigned if any.

If host unavailable "is_available" flag set to False.

Data format for each host:

```
[
    ,
    [ , [[ , ], ...]],
]
```

Key "fields" value doesn't matter at the moment.

Key "errors" - contains list of host's IPs to which connection is absent.

Example:

- Client Request:

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "host": "all",
    "fields": ["host", "ip", "is_available"]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": ["host", "ip"],
  "message": "OK",
  "data": [
    ["193.164.84.81", [
      ["178.255.225.129", "", [[11, "OP New", true]]],
      ["37.59.56.147", "", []],
      ["137.117.184.199", "Banned IP", []],
      ["204.12.242.186", "", []],
      ["162.210.199.20", "", []],
      ["54.215.217.134", "", []],
      ["193.104.107.111", "", []],
      ["77.246.22.103", "", []],
      ["142.0.41.57", "", []],
      ["193.164.84.80", [
        ["142.0.41.57", "", []],
        True
      ]],
      True
    ]],
    "time_elapsed": 0.884717
  ]
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': ['String'],
    'host': 'String',
    'ip': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['ip', 'host', 'is_available', 'description']	True	List([String(allow_blank: False)], min_length: 0)	Not in use at the moment ! required fields. If absent - takes fields specified in "default fields". If empty - returns error.
ip	N/A	True	String(allow_blank: False, regex: "^(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?\\.){0,3}(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)"	filter by IP address to search in firewall rules
host	all	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(0)?)(all)"	IP address of RTP server, if absent - returns blocked IPs on all RTP servers.

Default fields:

Name	Type & Comments
ip	String
host	String
is_available	Boolean
description	String

2.114 get_half_hour_stat_ac

Docstring:

Function for accucore.

Request structure and fields description:

```
{'data': {'client_id': 'Integer',  
          'direction': 'String',  
          'fields': ["Enum('success', 'price', 'destination', 'attempts', 'cost', 'duration')"],  
          'start_date': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	True	Integer(gt: 0)	filter by client
direction	N/A	True	String(allow_blank: False, regex: "^(incoming) (outgoing)\$")	filter defines call direction (incoming for incoming traffic, outgoing - for outgoing traffic)
fields	['success', 'price', 'destination', 'attempts', 'cost', 'duration']	True	List([Enum('success', 'price', 'destination', 'attempts', 'cost', 'duration')]), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
start_date	N/A	False	String(allow_blank: False)	filter defines request start date and time. Function will take into account calls which setup_time not less than given and not more than 'start_date' + 30 min.

Default fields:

Name	Type & Comments
success	Int

price	numeric(16,5)
destination	String
attempts	Int
cost	numeric(16,5)
duration	Int

2.115 group_add

Docstring:

Creates new group in the given context(s).
Function returns id of the created group.

Example:

- Client Request

```
{"session_id": "4595aceb14c686aa9752bb9860462bf2",
  "data": {"name": "1212121",
           "contexts": [1],
           "internal": false}
}
```

- Server Response

```
{"status": 200,
  "message": "OK",
  "id": 1398,
  "time_elapsed": 0.081872}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'internal': 'Bool',
          'name': 'Name(at least one printable character). max_length=80',
          'service_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
internal	False	True	Bool	option that makes group internal (if this flag is "true") or not (if "false"). Point, which is added to the internal group can not be added or belong to another group. Thus, adding point to internal group you at the same time remove existant group relations of given point.
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of contexts id to which group will be added
name	N/A	False	Name(at least one printable character). max_length=80	group name
service_id	1	True	Integer(gt: 0)	defines service id (1 - voice service, 2 - SMS service)

2.116 group_copy

Docstring:

Creates new group with same parameters as given one, including group relations with points and dialpeers.
New group will be created in the same context as existent one but with another name.
Function returns id of created group.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "id": 6,
    "name": "test_group",
    "contexts": [1],
    "internal": false
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 8,
  "time_elapsed": 0.062069
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'], 'id': 'Integer', 'name': 'String'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	id of existent group
name	N/A	False	String(allow_blank: False)	name of new group, should be unique
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	context id of parental group, on which new group will be created.

2.117 group_del

Docstring:

Deletes chosen group from the given context(s).
Function returns group id.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",  
  "data": {"id": 7,  
           "contexts": [1]}}  
}
```

- Server response

```
{"status": 200, "message": "OK", "id": 7, "time_elapsed": 0.07674}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'], 'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	group id
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	context(s) where need to delete group with given id.

2.118 group_dialpeers_get

Docstring:

Returns list of dialpeers that assigned to chosen group.

Example:

- Client Request

```
{"session_id": "677232544e5f894e13cc8a3f906504bd",
  "data": {"dp_name": "",
           "sort_by": [{"dp_name": ""}],
           "group_id": 1,
           "contexts": [1],
           "fields": ["dialpeer_id", "dp_name"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["dialpeer_id", "dp_name"],
  "message": "OK",
  "data": [[2, "Test_DP (CZ)"]],
  "time_elapsed": 0.006144}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
           'dialpeer_id': 'Integer',
           'dp_name': 'String',
           'fields': ["Enum('priority', 'dp_name_short', 'stop', 'dstregex', 'dialpeer_id', 'dp_name', 'group_id', 'enabled', 'group_name')"],
           'group_id': 'Integer',
           'limit': 'Integer',
           'offset': 'Integer',
           'sort_by': [{"Enum('priority', 'dp_name_short', 'stop', 'dstregex', 'dialpeer_id', 'dp_name', 'group_id', 'enabled', 'group_name')": "Enum('asc', 'desc', '')"}]},
  'session_id': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
contexts	[]	False	List([Integer()], min_length: 0)	contexts id where should search relations
dialpeer_id	N/A	True	Integer(gt: 0)	filter by dialpeer id
fields	['priority', 'dp_name_short', 'stop', 'dstregexp', 'dialpeer_id', 'dp_name', 'group_id', 'enabled', 'group_name']	True	List([Enum('priority', 'dp_name_short', 'stop', 'dstregexp', 'dialpeer_id', 'dp_name', 'group_id', 'enabled', 'group_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([Enum('priority', 'dp_name_short', 'stop', 'dstregexp', 'dialpeer_id', 'dp_name', 'group_id', 'enabled', 'group_name'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"dp_name": "asc"}] - sorts result ascending by field "dp_name"
dp_name	N/A	True	String(allow_blank: True)	filter by dialpeer name or first symbols of dialpeer name
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
group_id	N/A	True	Integer(gt: 0)	filter by group id

Default fields:

Name	Type & Comments
------	-----------------

priority	Int
dp_name_short	String
stop	Boolean
dstregex	String
dialpeer_id	Int
dp_name	String
group_id	Int
enabled	Boolean
group_name	String

2.119 group_dp_tp_dnis_get

Docstring:

Returns list of Dialpeers, TP's and Dnis_regex that assigned to chosen group.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "group_id": 11,
    "contexts": [1]
    "fields": [
      "group_id",
      "group_name",
      "dialpeer_id",
      "dialpeer_name",
      "dnis_regex",
      "tp_id",
      "tp_name"
    ]
  }
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
```

```

'dialpeer_id': 'Integer',
'dialpeer_name': 'String',
'dnis_regexp': 'String',
'fields': ["Enum('tp_id', 'group_name', 'dialpeer_name', 'dnis_regexp', 'tp_name', 'group_id', 'dialpeer_id')"],
'group_id': 'Integer',
'group_name': 'String',
'limit': 'Integer',
'offset': 'Integer',
'sort_by': [{"Enum('tp_id', 'group_name', 'dialpeer_name', 'dnis_regexp', 'tp_name', 'group_id', 'dialpeer_id')": "Enum('asc', 'desc', '')"}],
'tp_id': 'Integer',
'tp_name': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['tp_id', 'group_name', 'dialpeer_name', 'dnis_regexp', 'tp_name', 'group_id', 'dialpeer_id']	True	List([Enum('tp_id', 'group_name', 'dialpeer_name', 'dnis_regexp', 'tp_name', 'group_id', 'dialpeer_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error
group_id	N/A	True	Integer(gt: 0)	filter by particular group
offset	0	True	Integer(gte: 0)	offset says to skip that many rows before beginning to return rows
tp_id	N/A	True	Integer(gt: 0)	filter by TP
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	required contexts
dnis_regexp	N/A	True	String(allow_blank: True)	filter by Dnis_regexp
tp_name	N/A	True	String(allow_blank: True)	filter by TP name
group_name	N/A	True	String(allow_blank: True)	filter by group name
limit	0	True	Integer(gte: 0)	says how many rows will

				be in response
sort_by	[]	True	List([Enum('tp_id', 'group_name', 'dialpeer_name', 'dnis_regexp', 'tp_name', 'group_id', 'dialpeer_id'): Enum('asc', 'desc', ")}], min_length: 0)	field(s) on which resulted list be sorted
dialpeer_name	N/A	True	String(allow_blank: True)	filter by dialpeer name
dialpeer_id	N/A	True	Integer(gt: 0)	filter by certain dialpeer

Default fields:

Name	Type & Comments
tp_id	Int
group_name	String
dialpeer_name	String
dnis_regexp	String
tp_name	String
group_id	Int
dialpeer_id	Int

2.120 group_set

Docstring:

Allows to rename or(and) set chosen group internal.
Function returns id of group which was modified.

Example:

- Client Request

```
{"session_id": "4595aceb14c686aa9752bb9860462bf2",  
  "data": {"id": 1399,  
            "name": "121211",  
            "contexts": [1]}  
}
```

- Server response

```
{"status": 200,  
  "message": "OK",  
  "id": 1399,  
  "time_elapsed": 0.088758}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'id': 'Integer',  
          'internal': 'Bool',  
          'name': 'Name(at least one printable character). max_length=None'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	group id
name	N/A	True	Name(at least one printable character). max_length=None	new name of group. Group name shouldn't ends with '_group_simple' because this suffix is designed for service groups in Simple Routing.
internal	N/A	True	Bool	Deprecated
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	context where group parameters will be changed

2.121 groups_free_dialpeer_get

Docstring:

Returns list of groups which are not assigned with chosen dialpeer.

Example:

- Client Request

```
{"session_id": "593a9260cb1406179301fe2c2680e0fd",  
  "data": {"group_name": "",  
           "sort_by": [{"group_name": ""}],  
           "contexts": [1],  
           "dialpeer_id": 4,  
           "fields": ["group_id", "group_name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["group_id", "group_name"],  
  "message": "OK",  
  "data": [[2, "H_P_in_group_simple (CZ)"], [1, "st.group (CZ)"]],  
  "time_elapsed": 0.008413}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'dialpeer_id': 'Integer',  
          'fields': ["Enum('group_id', 'group_name')"],  
          'group_name': 'String',  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('group_id', 'group_name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	[]	False	List([Integer()], min_length: 0)	contexts id where search relations
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
fields	['group_id', 'group_name']	True	List([Enum('group_id', 'group_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('group_id', 'group_name')": Enum('asc',	field name on which result will be

			'desc', ")]], min_length: 0)	sorted.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
dialpeer_id	0	True	Integer(gte: 0)	
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
group_name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name

Default fields:

Name	Type & Comments
group_id	Int
group_name	String

2.122 groups_free_op_get

Docstring:

Returns list of groups aren't connected with chosen OP.

Example:

- Client Request

```
{
  "session_id": "91af15315af776928e1abfa0e88b6641",
  "data": {
    "op_id": 19,
    "sort_by": [{"name": ""}],
    "name": "",
    "contexts": [1],
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[6, "Alexander_test_group (CZ)"], [3, "h_pak (CZ)"],
```

```
[2, "H_Pak_in_group_simple (CZ)", [1, "ikex.group (CZ)",
[8, "test_group (CZ)"]],
"time_elapsed": 0.000902}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
'fields': ["Enum('id', 'name')"],
'limit': 'Integer',
'name': 'String',
'offset': 'Integer',
'op_id': 'Integer',
'service_id': 'Integer',
'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
op_id	N/A	True	Integer()	
name	N/A	True	String(allow_blank: True)	filter results by group name or first symbols of group name
contexts	N/A	False	List([Integer()], min_length: 0)	context(s) where need to search relations
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('id', 'name')": Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"name": "asc"}] - sorts result ascending by field "name"
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before

				beginning to return rows.
--	--	--	--	---------------------------

Default fields:

Name	Type & Comments
id	Int
name	String

2.123 groups_free_tp_get

Docstring:

Returns list of groups are not related with chosen Termination point.

Example:

- Client Request

```
{
  "session_id": "593a9260cb1406179301fe2c2680e0fd",
  "data": {
    "sort_by": [{"name": ""}],
    "name": "",
    "contexts": [1],
    "tp_id": 8,
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[2, "H_P_in_group_simple (CZ)"], [1, "st.group (CZ)"]],
  "time_elapsed": 0.009848
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ["Enum('id', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
tp_id	N/A	True	Integer(gt: 0)	filter by termination point id
contexts	[]	False	List([Integer()], min_length: 0)	contexts id where need search relations
name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name
sort_by	[]	True	List([Enum('id', 'name'): Enum('asc', 'desc', ")]], min_length: 0)	field name on which result will be sorted.
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows

Default fields:

Name	Type & Comments
id	Int
name	String

2.124 groups_get

Docstring:

Returns list of groups existent in the chosen context(s) and information about them.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fcla089d8c8f",
  "data": {"limit": 101,
           "sort_by": [{"name": ""}],
           "contexts": [1],
           "offset": 0,
           "fields": ["id", "name", "internal"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name", "internal"],
  "message": "OK",
  "data": [[6, "Test_group (CZ)", false],
           [3, "h_p (CZ)", false],
           [2, "H_in_group_simple (CZ)", false],
           [1, "st.group (CZ)", false],
           [9, "pablo_gr (CZ)", false],
           [8, "test_group1 (CZ)", false]],
  "time_elapsed": 0.008566}
```

Request structure and fields description:

```
{'data': {'companies_id': ['Integer'],
          'contexts': ['Integer'],
          'fields': ["Enum('context_id', 'name_context', 'internal', 'id', 'name')"],
          'id': 'Integer',
          'limit': 'Integer',
          'manager_id': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('context_id', 'name_context', 'internal', 'id', 'name')":
"Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([Enum('context_id',	field on which result will be sorted.

			'name_context', 'internal', 'id', 'name'): Enum('asc', 'desc', ''), min_length: 0)	Result can be sorted ascending, descending or . For example [{"name": "asc"}] - sorts result ascending by field "name"
name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name
contexts	N/A	True	List([Integer(gt: 0)], min_length: 0)	context(s) id where search groups
id	N/A	True	Integer(gt: 0)	group id
companies_id	N/A	True	List([Integer(gte: 0)], min_length: 0)	filtration by companies. It fills up automatically when manager id is given.
service_id	1	True	Integer(gt: 0)	filter by service id: 1 - voice service, 2 - SMS service
manager_id	N/A	True	Integer(gte: 0)	filtration by manager.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	N/A	True	Integer(gte: 0)	says how many rows will be in response.
fields	['context_id', 'name_context', 'internal', 'id', 'name']	True	List([Enum('context_id', 'name_context', 'internal', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
context_id	Int
name_context	String
internal	Boolean
id	Int

name	String

2.125 groups_to_dialpeers_add

Docstring:

Creates relations between chosen dialpeer and groups or chosen group and dialpeers.
This function disallows to add relation between groups and service dialpeer.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "dialpeers_ids": [6],
    "contexts": [1],
    "groups_ids": [1]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.039321
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'dialpeers_ids': ['Integer'],
    'groups_ids': ['Integer']
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of groups ids
dialpeers_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of dialpeers ids which shouldn't contain

			min_length: 0)	ids of service dialpeers.
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	contexts ids where need add relations will be added

2.126 guardian_blocked_dest_get

Docstring:

Returns list of destinations blocked by Guardian

-Client Request:

```
{ "data": { "destination": "",
  "fields": ["id", "pt_id", "pt_name", "event_id", "destination", "time_autoclosed",
    "reopens_count", "reopens_remaining", "pending_time", "remaining_time",
    "state_changes", "decisions_in_advance", "date_from", "date_to"
  ]
},
"session_id": "4690905d4417a7e81a4ee6972eece6f1"
}
```

-Server Response:

```
{ "status": 200,
  "fields": ["id", "pt_id", "pt_name", "event_id", "destination", "time_autoclosed",
    "reopens_count", "reopens_remaining", "pending_time", "remaining_time", "state_changes",
    "decisions_in_advance", "date_from", "date_to"
  ],
  "message": "OK",
  "data": [
    [2316, 697, "Auto test OP", "3e56c3b5acec7c1c2db5be0a0580a725", "0", "2017-11-14T16:00:00+00:00", 0, 0,
      "1:00:00", "0:35:44", "Auto", ["open_now"], "2017-11-14T15:00:00+00:00", "2017-11-14T15:59:59.999999+00:00"
    ],
    [2317, 149, "Call OP", "3e56c3b5acec7c1c2db5be0a0580a725", "380", "2017-11-14T16:19:36.024930+00:00", 0, 0,
      "1:00:00", "0:55:20", "Auto", ["open_now"], "2017-11-14T15:19:36.024930+00:00", "2017-11-14T16:19:36.024930+00:00"
    ]
  ],
  "time_elapsed": 0.024977
}
```

Request structure and fields description:

Request:

```
{
```

```

"session_id": String(),
"data": {
  "id": Int(gt=0),
  "pt_name": String(allow_blank=True),
  "destination": String(allow_blank=True),
  "manager_id": Int(gt=0),
  "fields": List(String()),
  "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
  "limit": Int(gte=0),
  "offset": Int(gte=0),
}
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	True	Int(gt=0)	entry id
"pt_name"	N/A	True	String(allow_blank=True)	filter by first symbols of point name
"destination"	N/A	True	String(allow_blank=True)	filter by first digits of destination code
"manager_id"	N/A	True	Int(gt=0)	filter by company manager
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field name on which result will be sorted
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'id'	Int
'pt_id'	Int
'pt_name'	String
'event_id'	String
'destination'	String
'time_autoclosed'	String
'reopens_count'	Int
'reopens_remaining'	Int - value of 'reopens_remaining' counter
'pending_time'	String
'remaining_time'	String - how much time remains to dest auto open
'state_changes'	String - route state: "Opened" (opened manually), "Closed" (closed manually), "Auto" (if route is under Guardian control)
'decisions_in_advance'	List(String()) - available actions('open_now', 'back_to_auto' or 'leave_closed')
'date_from'	String
'date_to'	String

2.127 guardian_blocked_dest_set**Docstring:**

Makes chosen action with given destination blocked by Guardian. Each action leads to uploading of appropriate price with suitable "description", "allowed" flag:

Action 'open_now' - uploads price with 'Guardian: open now' and 'allowed' == True, reset counter 'reopens_count' to the action's 'reopens_remain';

Action 'leave_closed' - uploads price with 'description' 'Guardian: leave closed' and 'allowed' == False;

Action 'back_to_auto' - uploads price with 'description' 'Guardian: back to auto' and with 'allowed' the same as was in AUTO mode before human intervention,

returns value of counter 'reopens_count' to the value which was in AUTO mode.

-Client Request:

-Server Response:

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "action": String(regex=BDEST_ACTIONS_REGEX),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	entry id in the table guardian.blocked_destination
"action"	N/A	False	String(regex=BDEST_ACTIONS_REGEX)	necessary action

2.128 guardian_conditions_get

Docstring:

Returns all possible names and corresponding to them ids for event's conditions.

-Client Request:

```
{"data":{"fields": ["id", "name"]},
  "session_id": "03fd203bd20481a8892c59487cf29f5a"
}
```

-Server Response:

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [
    [18, "Origination Point Name"],
    [19, "Termination Point Name"],
    [2, "Destination"],
    [14, "Calls quantity"],
    [15, "ASR"],
    [16, "ACD"],
    [17, "TP Price"],
    [21, "Profit"],
    [1002, "Completed, successful"],
    [1001, "Completed, unsuccessful"],
    [1007, "Completed, zero"],
    [1015, "Long calls"],
    [1003, "Non-routable"],
    [1016, "OP RTP timeout"],
    [1000, "Other successful"],
    [1004, "Other unsuccessful"],
    [1005, "Routed, misconfiguration possible"],
    [1006, "Routed, uncompleted"],
    [1018, "Short calls"],
    [1013, "Stuck calls"],
    [1017, "TP RTP timeout"],
    [1009, "Undesirable calls"],
    [1008, "Unrouted"],
    [1010, "Wrong Numbers"],
    [2000, "FAS possible"],
    "time_elapsed": 0.030231
  ]
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
```

```

    "fields": List(String()),
  }
}

```

Name	Default value	Optional	Type	Description
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
"id"	Int
"name"	String
"general"	Bool
"units" # List measurement units	

2.129 guardian_event_add

Docstring:

Creates new Guardian event with specific conditions and actions.

param_id may take following values:

- 1 - 'Point name',
- 2 - 'Destination',
- 14 - 'Calls quantity',
- 15 - 'ASR',
- 16 - 'ACD',
- 17 - 'TP Price',
- 18 - 'Origination Point Name',

19 - 'Termination Point Name',
Returns id of added event if error wasn't occurred.

Example:

- Client Request:

```
{
  "session_id":"4907dc561da853507fd0380381eb7f3d",
  "data":
  {
    "name":"OP ALL All2",
    "conditions":
    [
      {"value":"0","operation":"=","param_id":18,"relation":"AND"},
      {"value":"*","operation":"=","param_id":2,"relation":"AND"},
      {"preset":1,"value":"500,100000","operation":"range","param_id":14,"relation":"AND"},
      {"preset":1,"value":"1.0,20.0","value_units":"%","operation":"range","param_id":15,"relation":"AND"},
      {"value":"*","operation":"=","param_id":16,"relation":null}
    ],
    "enabled":true,
    "group_by_region":false
  }
}
```

- Server Response:

```
{
  "status": 200,
  "message": "OK",
  "intersected_events": ["OP ALL All", "Short Calls OP", "OP ALL All1"],
  "id": "a045520a1bf566d8fe313571e5444624",
  "time_elapsed": 0.228357
}
```

"intersected_events" (optional key) : the list of names of events,
points/directions which mention or coincide with the points/directions
from the current event

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "name": String(),
    "email_list_default_send_to": Bool(),
    "email_list_default_send_copy": Bool(),
    "enabled": Bool(),
    "group_by_region": Bool(),
    "conditions": List(condition_validation),
    "actions": List(action_validation),
  }
}
```

```

    "working_time": working_time_validation,
  }
}

```

Name	Default value	Optional	Type	Description
"name"	N/A	False	String()	event name
"email_list_default_send_to"	False	True	Bool()	default send_to flag for email list in new event
"email_list_default_send_copy"	True	True	Bool()	default send_copy flag for email list in new event
"enabled"	N/A	True	Bool()	event status. Status "enabled" = False turns off handling of given event by Guardian.
"group_by_region"	N/A	True	Bool()	group traffic by region
"conditions"	N/A	False	List(condition_validation)	List which contains conditions specified in 'condition_validation'. Each condition presents in the form of dictionary
"actions"	N/A	True	List(action_validation)	List which contains actions specified in 'action_validation'. Each action presents in the form of dictionary.
"working_time"	N/A	True	working_time_validation	event work time restrictions

condition_validation - is a list that contains parameters of each condition in the form of dictionary.

```

"conditions": [
  {
    "param_name": String(regex="[a-zA-Z0-9_]+"),
    "param_id": Int(gt=0),
    "value": String(regex=VALUE_REGEXP),
    "value_units": String(regex=VALUE_UNITS_REGEXP),
    "operation": String(regex=OPERATION_REGEXP),
  }
]

```

"relation": (String(regex=RELATION_REGEX) | Null),
 "sub_relation": String(regex=RELATION_REGEX),
 "conditions": List(pre_condition_validation),
 "preset": Int(gte=0),

}, {...},{...}, ...

]

Name	Default value	Optional	Type	Description
"param_name"	N/A	True	String(regex="[a-zA-Z0-9_]+")	parameter name
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX) Null)	logical operator that defines relation of the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.
"sub_relation"	N/A	True	String(regex=RELATION_REGEX)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"conditions"	N/A	True	List(pre_condition_validation)	parameters of sub conditions from pre_condition_validation in for of dictionary also known as object type in JSON
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

pre_condition_validation - is a list that contains parameters of each subcondition in the form of dictionary.

```
"conditions": [
  {
    "param_name": String(regex="[a-zA-Z0-9_]+"),
    "param_id": Int(gt=0),
    "value": String(regex=VALUE_REGEX),
    "value_units": String(regex=VALUE_UNITS_REGEX),
    "operation": String(regex=OPERATION_REGEX),
    "relation": (String(regex=RELATION_REGEX) | Null),
    "sub_relation": String(regex=RELATION_REGEX),
    "preset": Int(gte=0),

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

Name	Default value	Optional	Type	Description
"param_name"	N/A	True	String(regex="[a-zA-Z0-9_]+")	parameter name
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX) Null)	logical operator that defines relation of the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.
"sub_relation"	N/A	True	String(regex=RELATION_REGEX)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

--	--	--	--	--

action_validation - is a dictionary that contains settings of each action("tt_send", "generate_warning" and "close_dest_supplier") in the form of dictionary.

```
"actions": {
    "tt_send":
    {
        "enabled": Bool(),
        "email_copy": (String(allow_blank=True) | Email()),
        "email_list": List(TTSendEmailList(allow_blank=True)),
        "template": String(),
        "copy_to_owner_support_email": Bool(),
        "copy_to_company_manager": Bool(),
    },
    "generate_warning":
    {
        "enabled": Bool(),
        "show_traffic_light": Bool(),
        "warning_type": List(Int()),
    },
    "close_dest_supplier":
    {
        "enabled": Bool(),
        "open_time_out": Int(gt=0, lte=999),
        "reopen_count": Int(gte=0, lte=999),
        "keep_alive": Bool(),
    },
}
```

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	TT sending action status(true or false)
"email_copy"	""	True	(String(allow_blank=True) Email())	additional email addresses of TT copy recipients
"email_list"	[]	True	List(TTSendEmailList(allow_blank=True))	list which consists of following lists: ['company_id','company_name','company_tech_emails','send_to_flag','send_copy_flag']
"template"	N/A	False	String()	TT template id

"copy_to_owner_support_email"	N/A	True	Bool()	add owner support email to copy_to
"copy_to_company_manager"	N/A	True	Bool()	add company manager email to copy_to

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	warning status flag(true or false)
"show_traffic_light"	N/A	True	Bool()	not in use
"warning_type"	N/A	False	List(Int())	List of warning ids

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	action status(true or false)
"open_timeout"	N/A	False	Int(gt=0, lte=999)	timeout for auto opening destination after it has been closed by Guardian
"reopen_count"	N/A	False	Int(gte=0, lte=999)	positive integer value which defines how many times given route(s) may be automatically reopened by Guardian
"keep_alive"	N/A	True	Bool()	not in use; this flag turns on/off "keep_alive" timer

2.130 guardian_event_copy

Docstring:

Function allows to copy existing event.

New event will have the same conditions and actions as parental event but another name.

Function returns id of new event if error wasn't occurred.

-Client Request:

```
{"data":{"id": "341951f8af3588ece34048715d81e48c",
"name": "OP ALL 100%"}}
```

```
    },
    "session_id": "4907dc561da853507fd0380381eb7f3d"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "id": "3b426435331060f2d116954358cc4130",
  "time_elapsed": 0.238209
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": String(),
    "name": String(),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	String()	id of the parental event
"name"	N/A	False	String()	name of created event

2.131 guardian_event_del

Docstring:

Removes event with given id from events database.

-Client Request:

```
{"data":{"ids":["541cf27445a6d45003833754fcb07a4"]},
"session_id": "4907dc561da853507fd0380381eb7f3d"}
```

-Server Response:

```
{"status": 200,
"deleted": 1,
"message": "OK",
"time_elapsed": 0.104803}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "ids": List(String()),
  }
}
```

Name	Default value	Optional	Type	Description
"ids"	N/A	False	List(String())	events id list

2.132 guardian_event_get

Docstring:

Function returns current event list

- Client Request:

```
{"session_id":"fcd527739bc6babf53729608c3c6790b",
"data":{"sort_by":[{"name":"asc"}],
"fields":["id","name","enabled"]}
}
```

- Server Response:

```
{
  "status": 200,
  "message": "OK",
  "fields": ["id", "name", "enabled"],
  "data": [
    ["369f67c79784506b6bf27f2ecf2d50a0", "OP ALL All Spam Traffic", false],
    ["850d6a12901736ffe2ab5a24c05dfc38", "Short Calls OP", true],
    ["e1e26ae540cbade1fe606f0b71beafb5", "Short Calls TP", true],
    ["918120c142403c2be91a67ec3614a2c9", "TP ALL All 0% ASR", false],
    ["90ef92121c346cda596f9ca03a554c47", "TP ALL All 0% ASR 11", false],
    ["62215a54ea3864b919ed410f3b2db039", "TP ALL All FAS Possible", false],
    ["500dabd948ff471bf15c8354ac6d6845", "TP ALL All Low ACD", false],
    ["044b2cc7136b2ee1194a6dd8c9b54f46", "TP ALL All Low ASR", false],
    ["94023b69075c0e7e457a53a6c3782371", "TP ALL All ttia", false]
  ],
  "time_elapsed": 0.001168
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": String(),
    "name": String(allow_blank=True),
    "pt_name": String(allow_blank=True),
    "destination": String(),
    "enabled": Bool(),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	True	String()	event id. If present field 'last_modify' is allowed.
"name"	N/A	True	String(allow_blank=True)	event name
"pt_name"	N/A	True	String(allow_blank=True)	filter by point name mentioned in the event conditions

"destination"	N/A	True	String()	filter by destination code mentioned in the event conditions
"enabled"	N/A	True	Bool()	return only "enabled" events
"fields"	default_fields	True	List(String())	fields that will be present in reply, if empty returns fields specified in default fields
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field name on which result will be sorted
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'id'	String
'name'	String
'conditions'	List(Dict())
'actions'	List(Dict()) - Dict contains keys: "tt_send"(has key "email_list" which contains Lists with following data - [Int(company id), String(company name), String(company tech email), Bool(send to), Bool(send copy), "close_dest_supplier", "generate_warning").
'last_modify'	String - last modification date for chosen event

'group_by_region'	Boolean - enable traffic aggregation by region
'enabled'	Boolean
"working_time"	Dict() - settings for event working time

2.133 guardian_event_set

Docstring:

Allows to change conditions and actions of the given event.

param_id may take following values:

- 1 - 'Point name',
- 2 - 'Destination',
- 14 - 'Calls quantity',
- 15 - 'ASR',
- 16 - 'ACD',
- 17 - 'TP Price',
- 18 - 'Origination Point Name',
- 19 - 'Termination Point Name',

Function returns id of event that was changed if error wasn't occurred.

Example:

- Client Request:

```
{
  "data":{
    "name":"OP ALL 100%",
    "id":"eac4e3d0b5cf94f4fc9186beadf5191b",
    "conditions":[
      {"value":"995", "operation":"=", "param_id":18, "relation":"AND"},
      {"value":"3811", "operation":"=", "param_id":2, "relation":"AND"},
      {"value":"120", "operation": ">", "param_id":14, "relation":"AND"},
      {"value":"20", "value_units":"%", "operation": "<", "param_id":15, "relation":"AND"},
      {"value":"30", "operation": "<", "param_id":16, "relation":null}
    ],
    "enabled":true,
  },
  "session_id":"54c3d994a2bc2b519e396be78aa52a67"
}
```

- Server Response:

```
{
  "status":200,
```

```

    "message": "OK",
    "id": "eac4e3d0b5cf94f4fc9186beadf5191b",
    "time_elapsed": 0.174015,
    "intersected_events": ["Joe event", "Bob event"]
}

```

"intersected_events" (optional key) : the list of names of events, points/directions which mention or coincide with the points/directions from the current event

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "id": String(),
    "name": String(),
    "enabled": Bool(),
    "group_by_region": Bool(),
    "conditions": List(condition_validation),
    "actions": List(action_validation),
    "working_time": working_time_validation,
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	False	String()	event id
"name"	N/A	True	String()	filter by event name
"enabled"	N/A	True	Bool()	event status. Status "enabled" = False turns off handling event by Guardian.
"group_by_region"	N/A	True	Bool()	group traffic by region
"conditions"	N/A	True	List(condition_validation)	List which contains conditions specified in 'condition_validation'. Each condition presents in the form of dictionary.
"actions"	N/A	True	List(action_validation)	List which contains actions specified in 'action_validation'. Each action presents in the form of dictionary.
"working_time"	N/A	True	working_time_validation	event work time restrictions

			ation	
--	--	--	-------	--

condition_validation - is a list that contains parameters of each condition in the form of dictionary.

```
"conditions": [
  {
    "param_id": Int(gt=0),
    "value": String(regex=VALUE_REGEX),
    "value_units": String(regex=VALUE_UNITS_REGEX),
    "operation": String(regex=OPERATION_REGEX),
    "relation": (String(regex=RELATION_REGEX) | Null),
    "sub_relation": String(regex=RELATION_REGEX),
    "conditions": List(pre_condition_validation),
    "preset": Int(gte=0),

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

Name	Default value	Optional	Type	Description
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX) Null)	logical operator that defines relation of the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.

"sub_relation"	N/A	True	String(regex=RELATION_REGEX)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"conditions"	N/A	True	List(pre_condition_validation)	parameters of sub conditions from pre_condition_validation in for of dictionary also known as object type in JSON
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

pre_condition_validation - is a list that contains parameters of each subcondition in the form of dictionary.

```
"conditions": [
  {
    "param_id": Int(gt=0),
    "value": String(regex=VALUE_REGEX),
    "value_units": String(regex=VALUE_UNITS_REGEX),
    "operation": String(regex=OPERATION_REGEX),
    "relation": (String(regex=RELATION_REGEX) | Null),
    "sub_relation": String(regex=RELATION_REGEX),
    "preset": Int(gte=0),

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

Name	Default value	Optional	Type	Description
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX)	logical operator that defines relation of

			EGEXP) Null)	the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.
"sub_relation"	N/A	True	String(regex=RELATION_R EGEXP)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

action_validation - is a dictionary that contains settings of each action("tt_send", "generate_warning" and "close_dest_supplier") in the form of dictionary.

```
"actions": {
  "tt_send":
  {
    "enabled": Bool(),
    "email_copy": String(allow_blank=True),
    "email_list": List(TTSendEmailList(allow_blank=True)),
    "template": String(),
    "copy_to_owner_support_email": Bool(),
    "copy_to_company_manager": Bool(),
  },
  "generate_warning":
  {
    "enabled": Bool(),
    "show_traffic_light": Bool(),
    "warning_type": List(Int()),
  },
  "close_dest_supplier":
  {
    "enabled": Bool(),
    "open_time_out": Int(gt=0, lte=999),
    "reopen_count": Int(gte=0, lte=999),
    "keep_alive": Bool(),
  },
}
```

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	TT sending action status
"email_copy"	N/A	True	String(allow_blank=True)	additional email addresses of TT copy recipients

"email_list"	N/A	True	List(TTSendEmailList(allow_blank=True))	list which consists of following lists: ['company_id','company_name','company_tech_emails','send_to_flag','send_copy_flag']
"template"	N/A	False	String()	TT template id
"copy_to_owner_support_email"	N/A	True	Bool()	add owner support email to copy_to
"copy_to_company_manager"	N/A	True	Bool()	add company manager email to copy_to

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	action status
"show_traffic_light"	N/A	True	Bool()	not in use
"warning_type"	N/A	False	List(Int())	list of warnings binded with given event

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	action status
"open_timeout"	N/A	False	Int(gt=0, lte=999)	timeout for auto opening destination after it has been closed by Guardian
"reopen_count"	N/A	False	Int(gte=0, lte=999)	defines how many times route(s) may be auto opened by Guardian.
"keep_alive"	N/A	True	Bool()	this flag turns on/off "keep_alive" timer

2.134 guardian_event_trace

Docstring:

check all or given event matching for fake traffic

-Client Request:

```
{
  "data": {
    "only_active_event": true,
    "asr": 100.0, "event_id": "b4351f8b028dfcbe7692cc6309455490",
    "acd": 15, "pt_id": 1227,
    "calls_quantity": 100,
    "destination": "38057",
    "fields": ["event_id", "event_name", "proceed_by_region"]
  },
  "session_id": "7bd269d0e553d93d7a9020e67e09ef05"
}
```

-Server Response:

```
{
  "status": 200,
  "fields": ["event_id", "event_name", "proceed_by_region"],
  "message": "OK",
  "data": [
    ["b4351f8b028dfcbe7692cc6309455490", "OP ALL All", false]
  ],
  "time_elapsed": 0.018136
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "pt_id": Int(),
    "destination": String(),
    "calls_quantity": Int(gte=1, lt=10000000000),
    "asr": Float(gte=0, lte=100),
    "acd": Int(gte=0, lt=100000),
    "event_id": String(),
    "only_active_event": Bool(),
  }
}
```

Name	Default value	Optional	Type	Description
"pt_id"	N/A	False	Int()	OP/TP id
"destination"	N/A	False	String()	destination string
"calls_quantity"	N/A	False	Int(gte=1, lt=10000000000)	calls quantity
"asr"	N/A	False	Float(gte=0, lte=100)	ASR

"acd"	N/A	False	Int(gte=0, lt=100000)	ACD
"event_id"	N/A	True	String()	show results only for selected event
"only_active_event"	N/A	True	Bool()	show

2.135 guardian_global_settings_get

Docstring:

returns configuration data for guardian global settings

-Client Request:

```
{
  "data": {
    "fields": [
      "tt_sending_timeout",
      "tt_sending_delay",
      "tt_expire",
      "tt_close",
      "tt_keep",
      "event_presets",
      "operation_threshold",
      "operating_mode",
      "operating_modes"
    ]
  },
  "session_id": "3a626c2ba2e46ad72381e90834c7603b"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "fields": [
    "tt_sending_timeout",
    "tt_sending_delay",
    "tt_expire",
    "tt_close",
    "tt_keep",
    "event_presets",
    "operation_threshold",
  ]
}
```

```

        "operating_mode",
        "operating_modes"
    ],
    "data": [[24, 30, 7, 7, 7, {"asr": [[0.0, 0.0], [1.0, 20.0], [21.0, 40.0]],
        "calls_quantity": [[100, 100000], [500, 100000], [1000, 100000]],
        "acd": [[1, 30], [31, 60], [61, 120]]
    },
        {"medium": 500, "minimum": 100, "maximum": 1000},
        "manual",
        {"manual": {"detected_problems": ["zero_asr",
            "low_asr",
            "low_acd",
            "fas_possible",
            "spam_traffic"
        ],
        "ops": [],
        "operation_threshold": 500,
        "tps": [],
        "action_settings": {"close_destination": {"enabled": false,
            "reopen_count": 3,
            "open_timeout": 12
        },
        "tt_send": "to_partner_and_owner",
        "warning": true
    },
        "destinations": []
    },
        "automatic": {"action": "tt_to_owner", "operation_threshold": 500},
        "semi_automatic": {"detected_problems": ["zero_asr",
            "low_asr",
            "low_acd",
            "fas_possible",
            "spam_traffic"],
        "ops": [],
        "operation_threshold": 500,
        "tps": [],
        "action_settings": {"close_destination": {"enabled": true,
            "reopen_count": 3,
            "open_timeout": 12
        },
        "tt_send": "to_owner",
        "warning": true
    },
        "destinations": []
    }
    ],
    "time_elapsed": 0.229467
}

```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "fields": List(String()),
  }
}
```

Name	Default value	Optional	Type	Description
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
"tt_sending_delay"	Int
"tt_sending_timeout"	Int, hours
"tt_expire"	Int, days
"tt_close"	Int, days
"tt_keep"	Int, days
"event_presets"	Dict, event presets
"operating_mode"	String, guardian operating mode
"operation_threshold"	Dict, custom settings for web (it reads from conf.ini)
"operating_modes" # Dict, event	

presets	
---------	--

2.136 guardian_global_settings_set

Docstring:

set configuration data for guardian global settings

-Client Request:

```
{ "data": { "tt_sending_delay": 30,
            "tt_close": 7,
            "tt_keep": 7,
            "tt_sending_timeout": 24,
            "event_presets": { "asr": [[0.0, 0.0], [1.0, 20.0], [21.0, 40.0]],
                              "acd": [[1, 30], [31, 60], [61, 120]],
                              "calls_quantity": [[100, 100000], [500, 100000], [1000, 100000]]},
            "tt_expire": 7
          },
  "session_id": "3a626c2ba2e46ad72381e90834c7603b"
}
```

-Server Response:

```
{ "status": 200, "message": "OK", "time_elapsed": 1.378297 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tt_sending_delay": Int(gt=0, lte=TT_SENDING_DELAY_MAX),
    "tt_sending_timeout": Int(gt=0, lte=TT_SENDING_TIMEOUT_MAX),
    "tt_expire": Int(gt=0, lte=999),
    "tt_close": Int(gt=0, lte=999),
    "tt_keep": Int(gt=0, lte=365),
    "event_presets": preset_validation,
    "operating_mode": String(regex="(automatic)|(manual)|(semi_automatic)"),
    "operating_modes": operating_modes_validation,
  }
}
```

Name	Default value	Optional	Type	Description
"tt_sending_de	N/A	True	Int(gt=0,	

lay"			lte=TT_SENDING_DELAY_MAX)	
"tt_sending_timeout"	N/A	True	Int(gt=0, lte=TT_SENDING_TIMEOUT_MAX)	
"tt_expire"	N/A	True	Int(gt=0, lte=999)	
"tt_close"	N/A	True	Int(gt=0, lte=999)	
"tt_keep"	N/A	True	Int(gt=0, lte=365)	
"event_presets"	N/A	True	preset_validation	
"operating_mode"	N/A	True	String(regex="(automatic) (manual) (semi_automatic)")	
"operating_modes"	N/A	True	operating_modes_validation	

2.137 guardian_report_send

Docstring:

Function allows to send report that contains destinations currently closed by Guardian with Reopens Remaining = 0.

Fills email template 'email_guardian_blocked_destinations_report' with following data:

- 1) "to" - E-mail address specified in request. If it absent - email of user who made request.
- 2) attaches file with list of destinations blocked by Guardian.

Then function places created message to the appropriate database table and fills field "dt_created" with current date.

-Client Request:

```

{"data":{}},
"session_id":"f77a3980f65d239d5cd8f4e0353ba48a"
}

```

-Server Response:

```

{"status": 200,
"message": "OK",
"time_elapsed": 0.092342
}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "user_email": String(allow_blank=True),
    "email_copy": String(allow_blank=True),
  }
}

```

Name	Default value	Optional	Type	Description
"user_email"	session_info.get("user_email", "")	True	String(allow_blank=True)	email address of message recipient. If absent - function takes email address of user who made request.
"email_copy"	"	True	String(allow_blank=True)	email address of message copy recipients

2.138 guardian_template_add

Docstring:

Creates new TT template with given sender email address.

Checks validity of special variables in the template body and subject.

-Client Request:

```
{ "data": { "subject": "test",
           "name": "test_guard",
           "body": "Test",
           "enabled": false },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "id": 1186,
  "time_elapsed": 0.049356
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "name": String(),
    "enabled": Bool(),
    "subject": String(allow_blank=True),
    "body": String(allow_blank=True),
    "description": String(allow_blank=True),
  }
}
```

Name	Default value	Optional	Type	Description
"name"	N/A	False	String()	Template name
"enabled"	False	True	Bool()	Template status
"subject"	N/A	True	String(allow_blank=True)	Template subject. There is possible to use special variables in the subject.
"body"	N/A	True	String(allow_blank=True)	Template body. There is possible to use special variables in the subject.
"description"	N/A	True	String(allow_blank=True)	Some notes or explanations

2.139 guardian_template_del

Docstring:

Removes chosen TT template from database.

-Client Request:

```
{ "data": {"id": 1186},  
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"  
}
```

-Server Response:

```
{ "status": 200,  
  "message": "OK",  
  "time_elapsed": 0.058276  
}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "id": Int(gt=0),  
  }  
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	TT template id

2.140 guardian_template_preview

Docstring:

Preview Trouble Ticket without saving.

-Client Request:

```
{"data":{"subject":"qwerty","body":"qwerty"},
  "session_id":"3a626c2ba2e46ad72381e90834c7603b"
}
```

-Server Response:

```
{"status": 200, "body": "qwerty",
  "message": "OK",
  "time_elapsed": 0.000539,
  "subject": "qwerty"
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "subject": String(),
    "body": String(),
  }
}
```

Name	Default value	Optional	Type	Description
"subject"	N/A	True	String()	TT subject
"body"	N/A	True	String()	TT body

2.141 guardian_template_set

Docstring:

Changes data of chosen TT template. Checks validity of special variables in the template body and subject.

-Client Request:

```
{ "data": { "subject": "test",
           "name": "test_guard",
           "id": 1186,
           "body": "Test",
           "enabled": true
         },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "id": 1186,
  "time_elapsed": 0.029315
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "enabled": Bool(),
    "subject": String(allow_blank=True),
    "body": String(allow_blank=True),
    "description": String(allow_blank=True),
  }
}
```

Name	Default value	Optional	Type	Description

"id"	N/A	False	Int(gt=0)	template id
"name"	N/A	True	String()	new template name
"enabled"	N/A	False	Bool()	template status
"subject"	N/A	True	String(allow_blank=True)	Template subject. There is possible to use special variables in the subject.
"body"	N/A	True	String(allow_blank=True)	Template body. There is possible to use special variables in the subject.
"description"	N/A	True	String(allow_blank=True)	Some notes or explanations

2.142 guardian_templates_get

Docstring: Returns data of TT templates, which existent in the database.

-Client Request:

```
{
  "data": {
    "sort_by": [{"name": "asc"}],
    "fields": ["id", "name", "subject", "enabled", "read_only"]
  },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "fields": ["id", "name", "subject", "enabled", "read_only"],
  "data": [
    [1001, "0% ASR", "TT #%(tt_number)s (%(client_name)s) - 0%% ASR on destination(s) %", true, true],
    [1181, "0% a Test", "Test", false, false],
    [1174, "ikex2", "qwerty", true, false],
  ]
}
```

```

[1175, "ikex3", "qwerty", true, false],
[1003, "Low ACD", "TT #%(tt_number)s %(client_name)s - low ACD on destination(s) %(destination)s", true, true],
[1004, "Low ASR", "TT #%(tt_number)s %(client_name)s - low ASR on destination(s) %(destination)s", true, true],
[1002, "Low Quality", "TT #%(tt_number)s %(client_name)s - low quality on destination(s) %(destination)s", true, true],
[1005, "Spam Traffic", "TT #%(tt_number)s %(client_name)s - Spam Traffic on destination(s) %(destination)s", true, true]
],
"time_elapsed": 0.020476}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "enabled": Bool(),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	True	Int(gt=0)	Filter by template id
"name"	N/A	True	String()	Filter by first symbols of template name
"enabled"	N/A	True	Bool()	Filter by status
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[{"name": ""}]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field name on which result will be sorted
"limit"	0	True	Int(gte=0)	says how many rows will be in

				response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'id'	Int
'name'	String
'email_from'	String
'email_to'	String
'email_copy'	String
'subject'	String
'body'	String
'description'	String
'preview'	String
"enabled"	Boolean
"read_only"	Boolean

2.143 guardian_tt_add

Docstring:

Add Trouble Ticket manually or automatically by Guardian system.

-Client Request:

```
{
  "data": {
    "copy_to": "",
    "tt_template": 1001,
    "include_subcodes": false,
    "send_to_me_only": false,
    "destination": "380",
    "date_to": "2017-11-10T13:33:12",
    "time_zone": "0:00",
    "point": 149,
    "date_from": "2017-11-10T12:33:12"
  },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "status": "pending",
    "body": "",
    "date_update": "2017-11-10T13:33:12+00:00",
    "client_id": 27,
    "email_from": "bochka@speedflow.com",
    "date_insert": "2017-11-10T13:33:12+00:00",
    "id": 7881,
    "subject": "TT #16842 (Call Company) - 0% ASR on destination(s) 380",
    "task": "0% ASR",
    "email_copy": "",
    "tt_number": 16842,
    "destination": "380",
    "template_id": 1001,
    "pt_id": 149,
    "processed_by_event": "ffffffffffffffffffffffffffff",
    "email": "alduin@speedflow.com"
  },
  "time_elapsed": 0.472752
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "point": Int(gt=0),
```

```

    'destination': String(regex="^[0-9, ]*$"),
    "include_subcodes": Bool(),
    "date_from": String(regex=DATE_REGEXP),
    "date_to": String(regex=DATE_REGEXP),
    "time_zone": String(regex="^[+-]?d{1,2}):(\d{2})$"),
    "tt_template": Int(gt=0),
    "copy_to": String(allow_blank=True),
    "send_to_me_only": Bool(),
}
}

```

Name	Default value	Optional	Type	Description
"point"	N/A	False	Int(gt=0)	OP/TP id
'destination'	N/A	False	String(regex="^[0-9,]*\$")	destination code(s), which need show in report, separated by comma
"include_subcodes"	False	True	Bool()	
"date_from"	N/A	False	String(regex=DATE_REGEXP)	start date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
"date_to"	N/A	False	String(regex=DATE_REGEXP)	end date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
"time_zone"	'+00:00'	True	String(regex="^[+-]?d{1,2}):(\d{2})\$")	"date_from" and "date_to" timezone
"tt_template"	N/A	False	Int(gt=0)	TT template id
"copy_to"	""	True	String(allow_blank=True)	email address for get a copy of sent emails
"send_to_me_only"	N/A	True	Bool()	send email only to user email from user settings page

2.144 guardian_tt_cancel

Docstring:

Cancels sending of TT if 5 minutes timeout hasn't passed yet after function guardian_tt_send() was invoked.

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	TT id in database

2.145 guardian_tt_get

Docstring:

Returns entries of TTs generated by Guardian from the database.

-Client Request:

```
{"data":{"offset":0,
          "limit":11,
          "sort_by":[{"tt_number":"desc"}]}
```

```

    "fields":["id","date",
        "company","sent_to",
        "template","subject",
        "body","status","description",
        "is_waiting","is_mailed",
        "sending_allowed","time_remaining","tt_number"
    ]
},
"session_id":"f77a3980f65d239d5cd8f4e0353ba48a"
}

```

-Server Response:

```

{"status": 200,
 "fields": ["id", "date", "company", "sent_to", "template", "subject", "body", "status",
    "description", "is_waiting", "is_mailed", "sending_allowed", "time_remaining", "tt_number"],
 "message": "OK",
 "data": [[7884, "2017-11-14T16:00:00+00:00", "Auto Test", "alduin@speedflow.com", "Spam Traffic",
    "TT #16845 (Auto Test) - Spam Traffic on destination(s) 0", " ",
    "pending", null, false, true, false, 0, 16845]],
 "time_elapsed": 0.027988
}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "company": String(),
    "status": String(regex=TT_STATUS_REGEX),
    "id": Int(gt=0),
    "tt_number": Int(),
    'destination': String(regex="^[0-9, ]*$"),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"company"	N/A	True	String()	filter by client
"status"	N/A	True	String(regex=TT_STATUS_REGEX)	filter by TT status. Possible statuses are: 'answered', 'expired', 'closed',

				'canceled', 'pending'
"id"	N/A	True	Int(gt=0)	TT id in database
"tt_number"	N/A	True	Int()	TT number in database
'destination'	N/A	True	String(regex="^[0-9,]*\$")	destination code(s), which need show in report, separated by comma
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[{"date": ""}]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field name on which result will be sorted
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'id'	Int (TT id in the database)
'date'	String date when TT was generated or updated by another one
'company'	String

'sent_to'	String (emails 'to'; email 'copy')
'template'	String
'subject'	String
'body'	String
'status'	String
'description'	String
'destination'	String
'is_waiting'	Boolean (if True then TT hasn't been sent already)
'is_mailed'	Boolean (if True then TT has been sent already)
'time_remaining'	Int - time in seconds remaining to the TT sending
'tt_number'	bigserial (int number for TT grouping by region)
"sending_allowed"	Boolean, allow "Send" link on web

2.146 guardian_tt_preview

Docstring:

Preview Trouble Ticket without saving.

-Client Request:

```
{ "data": { "copy_to": "",
  "tt_template": 1001,
  "include_subcodes": false,
  "send_to_me_only": false,
  "destination": "380",
  "date_to": "2017-11-10T13:33:12",
  "fields": ["body"],
  "time_zone": "0:00",
  "point": 149,
  "date_from": "2017-11-10T12:33:12"
},
```

```
"session_id":"248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "status": "pending",
    "body": "",
    "date_update": "2017-11-10T13:33:12+00:00",
    "client_id": 27,
    "email_from": "bochka@speedflow.com",
    "date_insert": "2017-11-10T13:33:12+00:00",
    "id": 7881,
    "subject": "TT #16842 (Call Company) - 0% ASR on destination(s) 380",
    "task": "0% ASR",
    "email_copy": "",
    "tt_number": 16842,
    "destination": "380",
    "template_id": 1001,
    "pt_id": 149,
    "processed_by_event": "ffffffffffffffffffffffffffff",
    "email": "alduin@speedflow.com"
  },
  "time_elapsed": 0.472752
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "point": Int(gt=0),
    'destination': String(regex="^[0-9, ]*$"),
    "include_subcodes": Bool(),
    "date_from": String(regex=DATE_REGEX),
    "date_to": String(regex=DATE_REGEX),
    "time_zone": String(regex="^[+-]?\\d{1,2}:\\d{2}$"),
    "tt_template": Int(gt=0),
    "copy_to": String(allow_blank=True),
    "send_to_me_only": Bool(),
    "fields": List(String()),
  }
}
```

Name	Default value	Optional	Type	Description
"point"	N/A	False	Int(gt=0)	OP/TP id

'destination'	N/A	False	String(regex="^[0-9,]*\$")	destination code(s), which need show in report, separated by comma
"include_subcodes"	False	True	Bool()	
"date_from"	N/A	False	String(regex=DATE_REGEX)	start date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
"date_to"	N/A	False	String(regex=DATE_REGEX)	end date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
"time_zone"	'+00:00'	True	String(regex="^([+-]?\\d{1,2}):(?\\d{2})\$")	"date_from" and "date_to" timezone
"tt_template"	N/A	False	Int(gt=0)	TT template id
"copy_to"	""	True	String(allow_blank=True)	email address for get a copy of sent emails
"send_to_me_only"	N/A	True	Bool()	send email only to user email from user settings page
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
"email_from"	String - TT email from address

"email"	String - TT email to address
"email_copy"	String - email copy address
"subject"	String - email subject
"body"	String - email body text
"status"	String - TT status
"template_id"	Int - TT template id
"destination"	String - TT destinations
"pt_id"	Int - point id
"client_id"	Int - client_id for selected point

2.147 guardian_tt_send

Docstring:

Function has been designed to send Trouble Tickets manually. It places TT data to the appropriate key in the cache.

Function has 5 minutes timeout to cancel query(to do it use function guardian_tt_cancel()).

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	TT id in the database

2.148 guardian_tt_set

Docstring:

Sets status of TT with given id and updates "date_update" in the database with current date even if new TT status the same as old one. Returns id of TT which status was changed.

-Client Request:

```
{ "data": { "description": "",
            "id": 7881,
            "status": "pending"
          },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "id": 7881,
  "time_elapsed": 0.02301
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "status": String(regex=TT_STATUS_REGEX),
    "description": String(allow_blank=True),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	TT id

"status"	N/A	True	String(regex=TT_STATU S_REGEX)	status name
"description "	N/A	True	String(allow_blank=True)	

2.149 guardian_warning_add

Docstring:

Creates new warning with given name and color.
Returns id of created warning in database.

-Client Request:

```
{ "data": { "name": "Unrouted traffic",
           "color": "#9e84bb"
         },
  "session_id": "4907dc561da853507fd0380381eb7f3d"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "id": 1377,
  "time_elapsed": 0.370323
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
```

```

    "name": String(),
    "color": String(),
    "description": String(allow_blank=True),
  }
}

```

Name	Default value	Optional	Type	Description
"name"	N/A	False	String()	warning name, should be unique.
"color"	N/A	False	String()	warning color for showing.
"description"	"	True	String(allow_blank=True)	warning description.

2.150 guardian_warning_del

Docstring:

Removes warning with given id from database.
Returns id of the deleted warning.

-Client Request:

```

{"data":{"id": "1377"},
 "session_id": "4907dc561da853507fd0380381eb7f3d"
}

```

-Server Response:

```

{"status": 200,
 "message": "OK",
 "id": 1377,
 "time_elapsed": 0.048924
}

```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	warning id

2.151 guardian_warning_set

Docstring:

Changes name or(and) color of warning with given id.
Returns warning id.

-Client Request:

```
{ "data": { "id": 1377,
            "name": "Unrouted traffic",
            "color": "#9e84bb"
          },
  "session_id": "4907dc561da853507fd0380381eb7f3d"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "id": 1377,
  "time_elapsed": 0.218551
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "color": String(),
    "description": String(allow_blank=True),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	warning id
"name"	N/A	True	String()	new warning name.
"color"	N/A	True	String()	new color of the warning.
"description"	N/A	True	String(allow_blank=True)	warning description.

2.152 guardian_warnings_get

Docstring:

Returns information about warning existent in the database.

-Client Request:

```
{"data":{"sort_by":[{"name":"asc"}],
```

```

        "fields":["id","name","color","description","read_only"]
    },
    "session_id": "4907dc561da853507fd0380381eb7f3d"
}

```

-Server Response:

```

{"status": 200,
 "message": "OK",
 "fields": ["id", "name", "color", "description", "read_only"],
 "data": [
     [1001, "0% ASR", "#FF5B5B", null, true],
     [1004, "Low ACD", "#C34242", null, true],
     [1003, "Low ASR", "#C34242", null, true],
     [1002, "Low Quality", "#ED9D47", null, true],
     [1005, "Other problem", "#34683C", null, true]],
 "time_elapsed": 0.032561
}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('read_only', 'description', 'color', 'position', 'id', 'name')"],
          'id': 'Integer',
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('read_only', 'description', 'color', 'position', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	True	String(allow_blank: False)	filter by warning name
sort_by	[{'position': 'asc'}]	True	List([{'Enum('read_only', 'description', 'color', 'position', 'id', 'name')': Enum('asc', 'desc', '')}], min_length: 0)	field name on which result will be sorted
fields	['read_only', 'description', 'color', 'position', 'id', 'name']	True	List([Enum('read_only', 'description', 'color', 'position', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

id	N/A	True	Integer(gt: 0)	filter by warning id
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
read_only	Bool
description	String
color	String
position	Int
id	Int
name	String

2.153 guardian_warnings_replace

Docstring: Defines warnings order for displaying in the web interface.

-Client Request:

```
{"data":{"warnings_ids": [1001,1002,1003,1004]},
"session_id": "f280ead069f44c536888d3837bb14682"
}
```

-Server Response:

```
{"status": 200, "message": "OK", "time_elapsed": 0.783691}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "warnings_ids": List(Int(gt=0)),
  }
}
```

Name	Default value	Optional	Type	Description
"warnings_ids"	N/A	False	List(Int(gt=0))	list of warnings ids in necessary order

2.154 guardian_warnings_report_get

Docstring:

Returns warnings created by Guardian for given time period and point's destinations

NOTE: guardian stores warnings data only for last 24 hours.

-Client Request:

```
{
  "data": {
    "sort_by": [{"name": "asc"}],
    "fields": ["id", "name", "color", "description", "read_only"]
  },
  "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "fields": ["id", "name", "color", "description", "read_only"],
  "data": [
    [1001, "0% ASR", "#FF5B5B", null, true],
    [1004, "Low ACD", "#C34242", null, true],
    [1003, "Low ASR", "#C34242", null, true],
    [1002, "Low Quality", "#ED9D47", null, true],
    [1005, "Other problem", "#34683C", null, true]
  ],
  "time_elapsed": 0.042853
}
```

}

Request structure and fields description:

Default fields:

Name	Type & Comments
pt_name	String
warnings	List(Int())
client_id	String - company id for point
tendency	String (currently not in use)
is_closed	Boolean - means that given destination price of given point was blocked by Guardian
calls	Integer, calls count
event_id	String
destination	String
date_insert	String
pt_id	Int
company_name	String - company name for point

2.155 guardian_wizard_countries_get

Docstring:

Specialized function which get short countries and destination list for country code selection in guardian wizard.

-Client Request:

```
{"data": {}, "session_id": "f280ead069f44c536888d3837bb14682"}
```

-Server Response:

```
{
  "status": 200,
  "message": "OK",
  "fields": ["code", "name"],
  "data": [
    ["7840,7940", "Abkhazia"],
    ["93", "Afghanistan"],
    ["355", "Albania"],
    ["213", "Algeria"],
    ["376", "Andorra"],
    ["244", "Angola"],
    ["1264", "Anguilla"],
    ["1268", "Antigua and Barbuda"],
    ["54", "Argentina"],
    ["374", "Armenia"],
    ["297", "Aruba"],
    ["247", "Ascension Island"],
    ["61", "Australia"],
    ["672", "Australian External Territories"],
    ["43", "Austria"],
    ["994", "Azerbaijan"],
    ["1242", "Bahamas"],
    ["973", "Bahrain"],
    ["880", "Bangladesh"],
    ["1246", "Barbados"],
    ["375", "Belarus"],
    ["32", "Belgium"],
    ["501", "Belize"],
    ["229", "Benin"],
    ["263", "Zimbabwe"]
  ],
  "time_elapsed": 0.03146
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

Default fields:

Name	Type & Comments
'code'	String
'name'	String

2.156 guardian_wizard_events_add

Docstring:

function which implement one-user blocking bulk event add

-Client Request:

```
{
  "data": {
    "events": [
      {
        "email_list_default_send_to": true,
        "email_list_default_send_copy": true,
        "name": "OP ALL All Spam Traffic",
        "conditions": [
          {
            "value": "0",
            "operation": "=",
            "param_id": 18,
            "relation": "AND"
          },
          {
            "value": "*",
            "operation": "=",
            "param_id": 2,
            "relation": "AND"
          },
          {
            "preset": 1,
            "value": "500,100000",
            "operation": "range",
            "param_id": 14,
            "relation": "AND"
          },
          {
            "preset": 0,
            "value": "0,0",
            "operation": "range",
            "param_id": 15,
            "relation": "AND"
          },
          {
            "value": "*",
            "operation": "=",
            "param_id": 16,
            "relation": null
          }
        ],
        "actions": [
          {
            "tt_send": {
              "template": "1005",
              "email_list": [],
              "copy_to_owner_support_email": true,
              "email_copy": "",
              "enabled": true,
              "copy_to_company_manager": true
            }
          },
          {
            "close_dest_supplier": {
              "open_time_out": 12,
              "reopen_count": 3,
              "enabled": false
            }
          },
          {
            "generate_warning": {
              "warning_type": [1005],
              "enabled": true
            }
          }
        ],
        "enabled": true,
        "group_by_region": true
      }
    ]
  }
}
```

```

    }],
    },
    "session_id": "248dc7fedec32b0b7eff9170c038b31e"
}

```

-Server Response:

```

{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.173821
}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "events": List(event_validation) # events list,
  }
}

```

Name	Default value	Optional	Type	Description
"events"	N/A	False	List(event_validation) # events list	

condition_validation - is a list that contains parameters of each condition in the form of dictionary.

```

"conditions": [
  {
    "param_name": String(regex="[a-zA-Z0-9_]+"),
    "param_id": Int(gt=0),
    "value": String(regex=VALUE_REGEXP),
    "value_units": String(regex=VALUE_UNITS_REGEXP),
    "operation": String(regex=OPERATION_REGEXP),
    "relation": (String(regex=RELATION_REGEXP) | Null),
    "sub_relation": String(regex=RELATION_REGEXP),
    "conditions": List(pre_condition_validation),
    "preset": Int(gte=0),
  }, {...}, {...}, ...
]

```

Name	Default value	Optional	Type	Description
"param_name"	N/A	True	String(regex="[a-zA-Z0-9_]+")	parameter name
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX) Null)	logical operator that defines relation of the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.
"sub_relation"	N/A	True	String(regex=RELATION_REGEX)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"conditions"	N/A	True	List(pre_condition_validation)	parameters of sub conditions from pre_condition_validation in for of dictionary also known as object type in JSON
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

pre_condition_validation - is a list that contains parameters of each subcondition in the form of dictionary.

```
"conditions": [
  {
    "param_name": String(regex="[a-zA-Z0-9_]+"),
    "param_id": Int(gt=0),
```

"value": String(regex=VALUE_REGEX),
 "value_units": String(regex=VALUE_UNITS_REGEX),
 "operation": String(regex=OPERATION_REGEX),
 "relation": (String(regex=RELATION_REGEX) | Null),
 "sub_relation": String(regex=RELATION_REGEX),
 "preset": Int(gte=0),

}, {...},{...}, ...

]

Name	Default value	Optional	Type	Description
"param_name"	N/A	True	String(regex="[a-zA-Z0-9_]+")	parameter name
"param_id"	N/A	False	Int(gt=0)	parameter id
"value"	N/A	False	String(regex=VALUE_REGEX)	parameter value. Point name, first chars of point name, calls quantity with chosen CDR class, ASR, ACD, price value and so on. To specify all OP or all TP '0' is used in conjunction with 'param_id' '19' or '18'.
"value_units"	N/A	True	String(regex=VALUE_UNITS_REGEX)	measurement units (pcs. or %)
"operation"	N/A	False	String(regex=OPERATION_REGEX)	Defines if value is exact('='), in some range(sign '>' or '<') or begins with given characters('like').
"relation"	N/A	False	(String(regex=RELATION_REGEX) Null)	logical operator that defines relation of the given condition with other conditions in the same level. Possible values: 'OR', 'AND', 'OR NOT', 'AND NOT'.
"sub_relation"	N/A	True	String(regex=RELATION_REGEX)	logical operator that defines relation with sub conditions ('OR', 'AND', 'OR NOT', 'AND NOT')
"preset"	N/A	True	Int(gte=0)	guardian global settings range preset

action_validation - is a dictionary that contains settings of each action("tt_send", "generate_warning" and "close_dest_supplier") in the form of dictionary.


```

"actions": {
  "tt_send":
  {
    "enabled": Bool(),
    "email_copy": (String(allow_blank=True) | Email()),
    "email_list": List(TTSendEmailList(allow_blank=True)),
    "template": String(),
    "copy_to_owner_support_email": Bool(),
    "copy_to_company_manager": Bool(),
  },
  "generate_warning":
  {
    "enabled": Bool(),
    "show_traffic_light": Bool(),
    "warning_type": List(Int()),
  },
  "close_dest_supplier":
  {
    "enabled": Bool(),
    "open_time_out": Int(gt=0, lte=999),
    "reopen_count": Int(gte=0, lte=999),
    "keep_alive": Bool(),
  },
}

```

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	TT sending action status(true or false)
"email_copy"	""	True	(String(allow_blank=True) Email())	additional email addresses of TT copy recipients
"email_list"	[]	True	List(TTSendEmailList(allow_blank=True))	list which consists of following lists: ['company_id','company_name','company_tech_emails','send_to_flag','send_copy_flag']
"template"	N/A	False	String()	TT template id
"copy_to_owner_support_email"	N/A	True	Bool()	add owner support email to copy_to
"copy_to_company_manager"	N/A	True	Bool()	add company manager email to copy_to

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	warning status flag(true or false)
"show_traffic_light"	N/A	True	Bool()	not in use
"warning_type"	N/A	False	List(Int())	List of warning ids

Name	Default value	Optional	Type	Description
"enabled"	N/A	False	Bool()	action status(true or false)
"open_time_out"	N/A	False	Int(gt=0, lte=999)	timeout for auto opening destination after it has been closed by Guardian
"reopen_count"	N/A	False	Int(gte=0, lte=999)	positive integer value which defines how many times given route(s) may be automatically reopened by Guardian
"keep_alive"	N/A	True	Bool()	this flag turns on/off "keep_alive" timer

2.157 guardian_wizard_status_get

Docstring:

get GUI single user blocking flag for guardian wizard

-Client Request:

```
{ "data": {},
  "session_id": "3a626c2ba2e46ad72381e90834c7603b"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "wizard_active": false,
  "time_elapsed": 0.032097
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.158 guardian_wizard_status_set

Docstring:

set GUI single user blocking flag for guardian wizard

-Client Request:

```
{ "data": {"wizard_active": true},
  "session_id": "3a626c2ba2e46ad72381e90834c7603b"
}
```

-Server Response:

```
{ "status": 200,
  "message": "OK",
  "time_elapsed": 0.04961
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.159 gw_auth_add

Docstring:

Adds GSM GW auth for given TP id and context.

Example:

- Client Request:

```
{"session_id": "3fbcfc27845ec9a10490288b9ed89b73",
 "data": {
     "tp_id": 3,
     "login": 'gw_test_login',
     "password": 'gw_test_password',
     "contexts": [1, 2],
 }
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ["Enum('tp_id', 'login', 'password', 'id', 'prefix')"],
          'login': 'String',
          'password': 'String',
          'prefix': 'String',
          'tp_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	False	Integer(gt: 0)	Termination point id.
password	N/A	False	String(allow_blank: False,	GSM GW password.

			max_length: 32)	
contexts	[]	True	List([Integer()], min_length: 0)	Filter by contexts. If is absent function searches GW Auths in all contexts.
login	N/A	False	String(allow_blank: False, min_length: 3, max_length: 24)	GSM GW login.
fields	['tp_id', 'login', 'password', 'id', 'prefix']	True	List([Enum('tp_id', 'login', 'password', 'id', 'prefix')], min_length: 0)	Required fields. If absent - takes fields specified in "default fields". If empty - returns error.
prefix	""	True	String(allow_blank: True, regex: "^[\\d#+]*\$")	GSM GW prefix.

2.160 gw_auth_del

Docstring:

Removes GSM GW auth by given GSM GW record id and context.

Example:

- Client Request:

```
{
  "session_id": "3fbcfc27845ec9a10490288b9ed89b73",
  "data": {
    "id": 3,
    "contexts": [1, 2],
  }
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ["Enum('tp_id', 'login', 'password', 'id', 'prefix')"],
          'id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['tp_id', 'login', 'password', 'id', 'prefix']	True	List([Enum('tp_id', 'login', 'password', 'id', 'prefix')], min_length: 0)	Required fields. If absent - takes fields specified in "default fields". If empty - returns error.
id	N/A	False	Integer(gt: 0)	GSM GW auth record id.
contexts	[]	True	List([Integer()], min_length: 0)	Filter by contexts. If is absent function searches GW Auths in all contexts.

2.161 gw_auth_set

Docstring:

Changes GSM GW auth by given GSM GW record id and context.

Example:

- Client Request:

```
{
  "session_id": "3fbcfc27845ec9a10490288b9ed89b73",
  "data": {
    "id": 3,
    "login": "new_gw_test_login",
    "password": "new_gw_test_password",
    "contexts": [1],
  }
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
  'fields': ["Enum('tp_id', 'login', 'password', 'id', 'prefix')"],
  'id': 'Integer',
  'login': 'String',
  'password': 'String',
  'prefix': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
prefix	""	True	String(allow_blank: True, regex: "^[\\d#+]*\$")	GSM GW prefix.
id	N/A	False	Integer(gt: 0)	GSM GW auth record id.
contexts	[]	True	List([Integer()], min_length: 0)	Filter by contexts. If is absent function searches GW Auths in all contexts.
fields	['tp_id', 'login', 'password', 'id', 'prefix']	True	List([Enum('tp_id', 'login', 'password', 'id', 'prefix')], min_length: 0)	Required fields. If absent - takes fields specified in "default fields". If empty - returns error.
login	N/A	False	String(allow_blank: False, min_length: 3, max_length: 24)	New GSM GW login.
password	N/A	False	String(allow_blank: False, max_length: 32)	New GSM GW password.

2.162 gw_auths_get

Docstring:

Returns GSM GW auths for given TP id and context.

Example:

- Client Request:

```
{
  "session_id": "3fbcfc27845ec9a10490288b9ed89b73",
  "data": {
    "tp_id": 4,
    "contexts": [1],
  }
}
```

- Response:

```
{
  "fields": ["id", "tp_id", "login", "password", "prefix"],
  "data": [[1, 4, 'gw_test_login', 'gw_test_password', '1#']]
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'fields': ["Enum('tp_id', 'login', 'password', 'id', 'prefix')"],  
          'login': 'String',  
          'tp_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	True	Integer(gt: 0)	Filter by termination point id.
contexts	[]	True	List([Integer()], min_length: 0)	Filter by contexts. If is absent function searches GW Auths in all contexts.
fields	['tp_id', 'login', 'password', 'id', 'prefix']	True	List([Enum('tp_id', 'login', 'password', 'id', 'prefix')], min_length: 0)	Required fields. If absent - takes fields specified in "default fields". If empty - returns error.
login	N/A	True	String(allow_blank: False, max_length: 24)	Filter by GSM GW login.

Default fields:

Name	Type & Comments
tp_id	Int
login	String(24)
password	String(32)
id	Int
prefix	String

2.163 host_add

Docstring:

Adds new RTP host to the database and assign it to the chosen context.

Only one host may be "balancer". If balancer already exists - function returns status 302.

Function returns id of new host in the database.

Example:

- Client Request

```
{"session_id": "8efce2f78c7087eafbf88bb8c58390b9",  
  "data": {"enabled": true,  
            "host": "194.102.107.2",  
            "context_id": 1,  
            "type_id": 1}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 134, "time_elapsed": 0.238676}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',  
          'enabled': 'Bool',  
          'host': 'String',  
          'type_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	context id to which host will be added
enabled	N/A	False	Bool	host status. If "false" then host disabled.
host	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]).(\d [1-9]\d 1\d{2} 2[0-4]\d	RTP server IP

			25[0-5])\.\){2}(\d{1}[1-9]\d{1}\d{2} 2[0-4]\d{25[0-5])/(([1-9] [12]\d 3[0-2]))?(0\.\){3}0(/0)?)\$")	
type_id	N/A	False	Integer(gt: 0, lt: 3)	defines host type by id: Balancer, SMS or RTP

2.164 host_del

Docstring:

Removes given host from system.

This function doesn't allow to remove balancer(host with type_id=3) or remove host(RTP host with type_id=1) if it has any relation with balancer.

Example:

- Client Request

```
{"session_id": "8efce2f78c7087eafbf88bb8c58390b9",  
  "data": {"id": 134}}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.178371}
```

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	entry id in database

2.165 host_dumper_relations_get

Docstring:

Returns list of relations between RTP-hosts and dumper-hosts.

Example:

- Client Request

```
{
  "session_id": "59e93af682809527eabdf1d03b20b551",
  "data": {
    "fields": ["rtp_host", "dumper_host", "dumper_type"]
  }
}
```

- Server Response

```
{
  "status": "200",
  "fields": ["rtp_host", "dumper_host", "dumper_type"],
  "message": "OK",
  "data": [
    [
      ["1.1.1.1", "2.2.2.2", "dumper_media"],
      ["1.1.1.2", "2.2.2.2", "dumper_media"],
      ["1.1.1.1", "2.2.2.3", "dumper_signaling"],
      ["1.1.1.3", "2.2.2.4", "dumper_signaling"],
      ...
    ]
  ]
  "time_elapsed": 1.047421
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.166 host_from_balancer_del

Docstring:

Removes chosen host from the balancer.

Example:

- Client Request:

```
{"session_id": "3fbcfc27845ec9a10490288b9ed89b73",  
  "data": {"balancer_id": 12,  
           "host_id": 5}  
}
```

Request structure and fields description:

```
{'data': {'balancer_id': 'Integer', 'host_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
host_id	N/A	False	Integer(gt: 0)	host's id that should be removed from balancer
balancer_id	N/A	False	Integer(gt: 0)	balancer from which host should be removed

2.167 host_set

Docstring:

This function changes parameters of the given host: context or status.

Function returns host id in the database.

If given RTP host(type_id=1) has relation with balancer or given host is balancer function raises error.

Example:

- Client Request

```
{"session_id": "8efce2f78c7087eafbf88bb8c58390b9",  
  "data": {"id": 103,  
            "enabled": true,  
            "context_id": 1}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 103, "time_elapsed": 0.041597}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer', 'enabled': 'Bool', 'id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	host id
enabled	N/A	True	Bool	host status. If "enabled" = False it will not be shown in the web pages.
context_id	N/A	True	Integer(gt: 0)	new context id

2.168 host_to_balancer_add

Docstring:

Adds given host to the traffic balancer.

Example:

- Client Request:

```
{"session_id": "3fbcfc27845ec9a10490288b9ed89b73",  
  "data": {"balancer_id": 12,  
           "host_id": 5,  
           "weight": 10}  
}
```

Request structure and fields description:

```
{'data': {'balancer_id': 'Integer', 'host_id': 'Integer', 'weight': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balancer_id	N/A	False	Integer(gt: 0)	balancer to which host should be added
host_id	N/A	False	Integer(gt: 0)	host's id that should be added to balancer
weight	N/A	False	Integer(gt: 0)	weight coefficient for given relation

2.169 host_types_get

Docstring:

Returns possible host types.

Example:

- Client Request

```
{"session_id": "1671fafc97f35236904a5d706f0bbf18",  
  "data": {"sort_by": [{"id": ""}],  
           "offset": 0,  
           "fields": ["id", "name", "port"]}  
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name", "port"],
  "message": "OK",
  "data": [[1, "RTP", 1234],
            [2, "SMS", null],
            [3, "Balancer", null]],
  "time_elapsed": 0.005385}

```

Request structure and fields description:

```
{'data': {'fields': ["Enum('port', 'id', 'name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('port', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([{"Enum('port', 'id', 'name')": "Enum('asc', 'desc', '')"}], min_length: 0)	field name on which result will be sorted
fields	['port', 'id', 'name']	True	List([Enum('port', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
port	Int or Null
id	Int

name	String
------	--------

2.170 hosts_get

Docstring:

Returns hosts which exists in the database. Allows filter them by type id, context id, context name or status.

Example:

- Client Request

```
{
  "session_id": "1671fafc97f35236904a5d706f0bbf18",
  "data": {
    "limit": 101,
    "sort_by": [{"host": ""}],
    "offset": 0,
    "fields": ["id", "host", "enabled", "context_id"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "host", "enabled", "context_id"],
  "message": "OK",
  "data": [
    [132, "10.10.153.30", false, 1],
    [103, "194.105.112.190", true, 1]
  ],
  "time_elapsed": 0.005385
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',
          'enabled': 'Bool',
          'fields': ["Enum('type_id', 'context_id', 'enabled', 'host', 'context_name',
'id')"],
          'id': 'Integer',
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('type_id', 'context_id', 'enabled', 'host', 'context_name',
'id')": "Enum('asc', 'desc', '')"}],
          'type_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optiona	Type	Description
------	---------------	---------	------	-------------

		1		
limit	0	True	Integer(gte: 0)	says how many rows will be in response
enabled	N/A	True	Bool	returns only hosts with chosen status
context_id	N/A	True	Integer(gt: 0)	filter by contexts id. Returns only hosts from chosen context.
sort_by	[]	True	List([Enum('type_id', 'context_id', 'enabled', 'host', 'context_name', 'id'): Enum('asc', 'desc', ")]), min_length: 0)	field name on which result will be sorted
id	N/A	True	Integer(gt: 0)	host id in database
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
type_id	N/A	True	Integer(gt: 0, lt: 4)	filter by type
fields	['type_id', 'context_id', 'enabled', 'host', 'context_name', 'id']	True	List([Enum('type_id', 'context_id', 'enabled', 'host', 'context_name', 'id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
type_id	String
context_id	Int
enabled	Boolean
host	String
context_name	String

id	Int
----	-----

2.171 incoming_call_pathes_get

Docstring:

Returns list of Termination Points to which given OP have route for specific destination at the given time ordered according to the routing algorithm:

- 1) by priority of OP->Group relation
- 2) by dialpeer priority
- 3) by priority of dialpeer->TP relation
- 4) by weight of relation between dialpeer and TP if priority of dialpeer->TP relation is the same
- 5) by TP price priority
- 6) LCR (more profitable routes will be higher in list)

Function returns also boolean parameter - "dest_changed". It determines if tested number will be changed by some conversion defined in any route (only B number conversion in OP and dialpeer is taken into account).

Function returns parameters of the targeted OP within key 'op_data'.

'op_data' fields:

```
[
    'op_name',
    'op_capacity',
    'op_country',
    'op_price_id',
    'op_id',
    'op_enabled',
    'op_destination',
    'op_price_default',
    'op_mc',
    'op_ci',
    'op_ft',
    'op_currency_id',
    'op_cid',
    'op_tarif_name',
    'op_price_tarif',
    'op_currency_sign',
    'op_currency_name',
    'routing_algorithm',
    'op_tariff_invisible',
    'op_origin_country'
]
```

Example:

- Client Request

```
{"session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {"time_zone": "0:00",
           "time_period": "2014-01-13T15:32:50",
```

```

"op_id":7,
"contexts":[1],
"number":"9230000000000",
"fields":["tp_enabled","delta_price","dif_price","tp_id",
"tp_price_id","group_id","dialpeer_id",
"allow_internal","tp_without_ip","allowed_price",
"tp_name","tp_destination","tp_country",
"tp_price_default","tp_price_tarif",
"tp_currency_name","tp_mc","tp_ci","tp_ft",
"tp_capacity","hunt_stop","dp_name",
"dp_tp_priority","dp_priority","weight",
"group_name","op_group_priority","price_priority",
"profit","client_id"]}
}

```

- Server Response

```

{"status": 200,
"time_elapsed": 0.028728,
"dest_changed": false,
"fields": ["tp_enabled", "delta_price", "dif_price", "tp_id",
"tp_price_id", "group_id", "dialpeer_id",
"allow_internal", "tp_without_ip", "allowed_price",
"tp_name", "tp_destination", "tp_country",
"tp_price_default", "tp_price_tarif", "tp_currency_name",
"tp_mc", "tp_ci", "tp_ft", "tp_capacity", "hunt_stop",
"dp_name", "dp_tp_priority", "dp_priority", "weight",
"group_name", "op_group_priority", "price_priority",
"profit", "client_id"],
"message": "OK",
"op_data": ["H_in", 5, "Pakistan Mobile Mobilink", 13, 7, true,
"9230", "10.4000", 1, 1, 0, 1, 5, "T_H_in1",
"10.4000", "$", "USD", "QUALITY", false, "92"],
"data": [[true, "0.00000", "0.00500", 8, 20, 3, 4, false, false,
true, "H_to", "9230", "Pakistan Mobile Mobilink",
"10.3500", "10.3500", "USD", 1, 1, 0, 500, false,
"H_p(CZ)", 0, 0, 0, "h_p (CZ)", 0, 0, "14.29", 6]]}

```

Request structure and fields description:

```

{'data': {'caller_number': 'String',
'contexts': ['Integer'],
'currency_id': 'Integer',
'fields': ["Enum('dp_tp_priority', 'price_priority', 'op_destination', 'tp_de
stination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'o
p_tariff_invisible', 'tp_currency_sign', 'tp_capacity', 'dif_price', 'op_cid', 'op_id',
'tp_tariff_ac_upload_date', 'client_id', 'op_mc', 'tp_price_default', 'op_name', 'dp_ca
pacity', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_gro
up_priority', 'op_price_tarif', 'op_capacity', 'tp_gw_authorization', 'op_origin_countr
y', 'weight', 'tp_name', 'op_currency_name', 'tp_without_ip', 'delta_price', 'tp_price
tarif', 'hunt_stop', 'profit', 'op_enabled', 'dp_priority', 'tp_tariff_invisible', 'op
price_default', 'dp_name', 'op_ci', 'asr', 'tp_enabled', 'op_currency_sign', 'tp_id',
'allow_internal', 'tp_mc', 'tp_origin_country', 'dialpeer_id', 'routing_algorithm', 'op
price_id', 'allowed_price', 'acd', 'op_ft', 'tp_country', 'tp_price_id', 'op_currency_i
d')"],
'limit': 'Integer',
'number': 'String',
'offset': 'Integer',
'one_route': 'Bool',
'op_id': 'Integer',

```

```

'session_id': 'String',
'show_disabled_pt': 'Bool',
'show_stat': 'Bool',
'time_period': 'String',
'time_zone': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['dp_tp_priority', 'price_priority', 'op_destination', 'tp_destination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'op_tariff_invisible', 'tp_currency_sign', 'tp_capacity', 'dif_price', 'op_cid', 'op_id', 'tp_tariff_ac_upload_date', 'client_id', 'op_mc', 'tp_price_default', 'op_name', 'dp_capacity', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_group_priority', 'op_price_tarif', 'op_capacity', 'tp_gw_authorization', 'op_origin_country', 'weight', 'tp_name', 'op_currency_name', 'tp_without_ip', 'delta_price', 'tp_price_tarif', 'hunt_stop', 'profit', 'op_enabled', 'dp_priority', 'tp_tariff_invisible', 'op_price_default', 'dp_name', 'op_ci', 'asr', 'tp_enabled', 'op_currency_sign', 'tp_id',	True	List([Enum('dp_tp_priority', 'price_priority', 'op_destination', 'tp_destination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'op_tariff_invisible', 'tp_currency_sign', 'tp_capacity', 'dif_price', 'op_cid', 'op_id', 'tp_tariff_ac_upload_date', 'client_id', 'op_mc', 'tp_price_default', 'op_name', 'dp_capacity', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_group_priority', 'op_price_tarif', 'op_capacity', 'tp_gw_authorization', 'op_origin_country', 'weight', 'tp_name', 'op_currency_name', 'tp_without_ip', 'delta_price', 'tp_price_tarif', 'hunt_stop', 'profit', 'op_enabled', 'dp_priority', 'tp_tariff_invisible', 'op_price_default', 'dp_name', 'op_ci', 'asr', 'tp_enabled', 'op_currency_sign', 'tp_id',	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.

	'allow_internal', 'tp_mc', 'tp_origin_country', 'dialpeer_id', 'routing_algorithm', 'op_price_id', 'allowed_price', 'acd', 'op_ft', 'tp_country', 'tp_price_id', 'op_currency_id']		'allow_internal', 'tp_mc', 'tp_origin_country', 'dialpeer_id', 'routing_algorithm', 'op_price_id', 'allowed_price', 'acd', 'op_ft', 'tp_country', 'tp_price_id', 'op_currency_id')], min_length: 0)	
session_id	N/A	True	String(allow_blank: False)	user session id, need for checking user permissions. By defaults from user session. Currently not in use.
number	N/A	False	String(allow_blank: False, regex: "[0-9#*]+\$")	called number
caller_number	""	True	String(allow_blank: True)	caller number
time_period	N/A	False	String(allow_blank: False)	request time in format yyyy-MM-dd'T'HH:mm:ss
currency_id	N/A	True	Integer(gt: 0)	currency in which price will display, by defaults from user session
op_id	N/A	False	Integer(gt: 0)	id of chosen Origination Point
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?d{1,2}):(\d{2})\$")	time zone for 'time_period'
show_disabled_pt	False	True	Bool	if "True" then disabled TPs also will be shown
contexts	N/A	False	List([Integer()], min_length: 0)	ids context, in which request will be processed
one_route	False	True	Bool	if "True" then report

				shows only one route per one Termination point.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
show_stat	False	True	Bool	if "True" then ACD,ASR for last 4 hours will returns

Default fields:

Name	Type & Comments
dp_tp_priority	Int
price_priority	Int
op_destination	String
tp_destination	String
tp_currency_id	Int
group_name	String
attempts	Int
tp_ft	Int
tp_currency_name	String(3) - name_short
op_tariff_invisible	Bool
tp_currency_sign	String
tp_capacity	Int
dif_price	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default

op_cid	Int - OP client id
op_id	Int
tp_tariff_ac_upload_date	String
client_id	Int
op_mc	Int
tp_price_default	numeric(16,5)
op_name	String
dp_capacity	numeric(15,2)
delta_profit	numeric(15,2)
delta_price_unit	Int
tp_ci	Int
group_id	Int
op_country	String
op_group_priority	Int
op_price_tarif	numeric(16,5) - op price in the tariff's currency
op_capacity	Int
tp_gw_authorization	Boolean
op_origin_country	String
weight	Int
tp_name	String
op_currency_name	String(3) - name_short

tp_without_ip	Boolean
delta_price	numeric(16,5)
tp_price_tarif	numeric(16,5) - tp price in the tariff's currency
hunt_stop	Boolean
profit	numeric(15,2) = (dif_price/tp_price_default) * 100% if tp_price_default not equal 0, else 0
op_enabled	Boolean
dp_priority	Int
tp_tariff_invisible	Bool
op_price_default	numeric(16,5)
dp_name	String
op_ci	Int
asr	numeric or null
tp_enabled	Boolean
op_currency_sign	String
tp_id	Int
allow_internal	Boolean
tp_mc	Int
tp_origin_country	String
dialpeer_id	Int
routing_algorithm	String
op_price_id	Int

allowed_price	Bool
acd	Int or null
op_ft	Int
tp_country	String
tp_price_id	Int
op_currency_id	Int

2.172 incoming_calls_get

Docstring:

Returns calls data of authorized incoming calls which passed through mediacore and has been initiated by originator during given time period.

To define from where to get data (from cdrs or aggregation) options "type" and "by_cdr" has been designed. To get each call separately chose type "individual". With type "individual" this function invokes function "_incoming_calls_get_individual" which returns data from cdrs ordered by 'setup_time' descending.

Function stores responses in cache during 5 minutes. That's why user should wait 5 minutes to get new calls data for the same request or change request params.

To get calls data combined in some way use necessary "type" ("combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" or "combine_disconnect_code") in conjunction with boolean option "by_cdr".

If "by_cdr" is "true" - it calls function "_incoming_calls_get_by_cdr" that returns data from cdrs, otherwise - calls function "_incoming_calls_get_by_agg" which takes data from aggregation.

In the second case ("type" = "combine_dest", "month" etc.) function may return less precise result for last several hours.

If type not individual key 'by_cdr' should be defined.

Note, that this function has limitation of maximal time period which depends on chosen "type":

- 1) 7 days for "type" "individual";
- 2) 31 day for "type": "combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" and "combine_disconnect_code";
- 3) 12 hours for "type" listed in 2) if flag "by_cdr" is "true".

`_incoming_calls_get_individual` ([data_validation](#)) ([default_fields](#))

Returns authorized calls records for each call individually ("type":"individual") for given time period ordered by 'setup_time' descending.

Example:

- Client Request

```
{
  "session_id": "152bab1c9d436bad4fb64435b9398e08",
  "data": {
    "limit": 1,
    "time_zone": "0:00",
    "currency_id": 1,
    "end_date": "2013-12-20T11:59:59",
    "not_use_like": false,
    "offset": 0,
    "type": "individual",
    "start_date": "2013-12-20T00:22:04",
    "fields": [
      "bill_id", "nas", "op_name", "setup_time",
      "country", "called_number_in", "clear_number",
      "op_caller_number", "tp_caller_number",
      "called_number_out", "duration", "connect_time",
      "disconnect_time", "disconnect_cause_name",
      "disconnect_cause", "local_disconnect_cause_name",
      "local_disconnect_cause", "op_price", "tp_price",
      "op_cost", "tp_cost", "op_ip", "op_protocol",
      "tp_ip", "tp_protocol", "nas", "op_codecs",
      "tp_codecs", "dialpeer", "pdd", "op_class_name",
      "tp_class_name", "ds", "scd"
    ]
  }
}
```

- Server response

```
{
  "status": 200,
  "fields": [
    "bill_id", "nas", "op_name", "setup_time", "country",
    "called_number_in", "clear_number", "op_caller_number",
    "tp_caller_number", "called_number_out", "duration",
    "connect_time", "disconnect_time",
    "disconnect_cause_name", "disconnect_cause",
    "local_disconnect_cause_name", "local_disconnect_cause",
    "op_price", "tp_price", "op_cost", "tp_cost", "op_ip",
    "op_protocol", "tp_ip", "tp_protocol", "nas",
    "op_codecs", "tp_codecs", "dialpeer", "pdd",
    "op_class_name", "tp_class_name", "ds", "scd"
  ],
  "message": "OK",
  "data": [
    [
      "1385380414-1505058", "193.161.85.79", "Calls_from_57",
      "2013-12-20T11:59:48.826000+00:00", "PAKISTAN-M-MILINK",
      "200923032333018", "923032333018", "966598351346",
      "966598351346", "400923032333018", 0,
      "2013-12-20T12:00:16.450000+00:00",
      "2013-12-20T12:00:16.450000+00:00",
      "Normal call cleared (CANCEL or BYE received)", 0,
      "CANCEL received from origination side", 12000,
      "0.04000", "0.01000", "0.00000", "0.00000",
      "193.104.107.13", "sip", "193.104.107.10", "sip",
      "193.161.85.79", "ALAW,MULAW,G729,G723", "",
      "SF_HA0S_AZ", 6, "Completed, zero", "Completed, zero",
      1, 27
    ]
  ],
  "time_elapsed": 1.394601
}
```

`_incoming_calls_get_by_cdr` ([data_validation](#)) ([default_fields](#))

Returns incoming calls statistic combined by given criteria. This function returns more precise data then request from aggregation because it accounts for current traffic.

Example:

- Client Request

```
{"session_id":"152bab1c9d436bad4fb64435b9398e08",
  "data":{"limit":1,
    "time_zone":"0:00",
    "currency_id":1,
    "end_date":"2013-12-25T10:59:59",
    "not_use_like":false,
    "offset":0,
    "type":"day",
    "start_date":"2013-12-25T00:00:00",
    "by_cdr":true,
    "fields":["some_xy","tr_date","asr","acd","pdd","sc",
      "duration","attempts","sa","op_sum","tp_sum",
      "delta_price"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["some_xy", "tr_date", "asr", "acd", "pdd", "sc",
    "duration", "attempts", "sa", "op_sum", "tp_sum",
    "delta_price"],
  "message": "OK",
  "data": [[true, "2013-12-25T00:00:00+00:00", "0.10", 161, "3.5",
    "28.29", 33082, 213143, "205",
    "22.0546300000000000000000000000",
    "13.6601200000000000000000000000", "8.39451"]],
  "time_elapsed": 26.300286}
```

`_incoming_calls_get_by_agg` ([data_validation](#)) ([default_fields](#))

Function returns incoming calls statistic combined by chosen criteria from aggregation and thats why it doesn't account for traffic during last hour.

Example:

- Client Request

```
{"session_id":"152bab1c9d436bad4fb64435b9398e08",
  "data":{"limit":1,
    "time_zone":"0:00",
    "currency_id":1,
    "end_date":"2013-12-25T23:59:59",
    "not_use_like":false,
    "offset":0,
    "type":"day",
    "start_date":"2013-12-25T00:00:00",
```

```

        "by_cdr":false,
        "fields":["some_xy","tr_date","asr","acd","pdd","sc",
            "duration","attempts","sa","op_sum","tp_sum",
            "delta_price"]}
    }

```

- Server Response

```

{"status": 200,
 "fields": ["some_xy", "tr_date", "asr", "acd", "pdd", "sc",
            "duration", "attempts", "sa", "op_sum", "tp_sum",
            "delta_price"],
 "message": "OK",
 "data": [[true, "2013-12-25T00:00:00+00:00", "0.48", 162, "3.2",
            "34.26", 169747, 217810, "1045",
            "113.1645800000000000000000000000",
            "75.3931100000000000000000000000", "37.77147"]],
 "time_elapsed": 0.23369}

```

Request structure and fields description:

_incoming_calls_get_individual

```

{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fas_possible': 'Bool',
          'fields': ["Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name',
          'op_caller_number', 'clear_number', 'jurisdict', 'tp_protocol', 'op_price', 'tp_client_
          id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_class_name', 'op_i
          d', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'l
          ocal_disconnect_cause', 'op_media_ip', 'op_name', 'op_client_name', 'outgoing_ani', 'tp
          _caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconnect
          _time', 'tp_id', 'setup_time', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', '
          tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'd
          isconnect_cause', 'bill_id', 'pdd', 'called_number_in')"],
          'hosts': "['String'] or String",
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_class_id': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'outgoing_ip': 'String',
          'owner_id': 'Integer or Null',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'show_stat': 'Bool',
          'sort_by': [{"Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name
          ', 'op_caller_number', 'clear_number', 'jurisdict', 'tp_protocol', 'op_price', 'tp_clie
          nt_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_class_name', 'o
          p_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name',

```

```

'local_disconnect_cause', 'op_media_ip', 'op_name', 'op_client_name', 'outgoing_ani', '
tp_caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconne
ct_time', 'tp_id', 'setup_time', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs',
'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', '
disconnect_cause', 'bill_id', 'pdd', 'called_number_in')": "Enum('asc', 'desc', '')"}],
    'start_date': 'String',
    'termcause': 'String',
    'time_zone': 'String',
    'type': 'String',
    'use_old': 'Bool'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optiona l	Type	Description
fields	['op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'jurisdict', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_class_name', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_na me', 'tp_class_name', 'local_disconnect_cause', 'op_media_ip', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'bill_id', 'pdd', 'called_number_in']	True	List([Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'jurisdict', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_class_name', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_na me', 'tp_class_name', 'local_disconnect_cause', 'op_media_ip', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'bill_id', 'pdd', 'called_number_in')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
time_zone	+00:00	True	String(allow_blank: False,	"start_date" and

			regex: "^[+-]?\\d{1,2}):\\d{2})\$")	"end_date" timezone
not_use_like	False	True	Bool	if "true" searches only calls by exact destination codes without subcodes
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
op_destination	N/A	True	String(allow_blank: False, regex: "^(%?\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
use_old	False	False	Bool	Flag for unoptimised DB function usage. MCD-3434
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults from user session
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults -

				from user session.
termcause	N/A	True	String(allow_blank: False)	filter by disconnect cause code
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9] [12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$")	filter by IP address of call originator
hosts	N/A	True	Or(List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9] [12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$")), min_length: 0), String(allow_blank: False, regex: "^(([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9] [12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$"))	filter by RTP server IP
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
outgoing_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9] [12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$")	filter by IP address of the termination side

type	N/A	False	String(allow_blank: False, regex: "^individual\$")	statistical resolution type
fas_possible	False	True	Bool	filter calls for signs of FAS (SCD < 1 sec, SCD is Setup Connect Delay: connect_time - setup_time)
owner_id	None	True	Or(Integer(gt: 0), Null)	filter by owner
op_id	N/A	True	Integer(gt: 0)	filter by OP
clients_ids	[]	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
client_id	N/A	True	Integer(gt: 0)	filter by OP company.
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
sort_by	[]	True	List([{'Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'jurisdict', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_class_name', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op_media_ip', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price',	field on which result will be sorted. Result can be sorted ascending or descending.

			'op_client_id', 'disconnect_cause', 'bill_id', 'pdd', 'called_number_in'): Enum('asc', 'desc', ")}}, min_length: 0)	
show_stat	False	True	Bool	not in use
op_class_id	N/A	True	Integer(gte: 0)	Shows in report only calls with given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed wrong".
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger

_incoming_calls_get_by_cdr

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ['String'],
          'hosts': ['String'],
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'min_attempts': 'Integer',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_class_id': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'outgoing_ip': 'String',
          'owner_id': 'Integer or Null',
          'session_id': 'String',
          'shorter_than': 'Integer',
```

```

'sort_by': [{"Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration',
'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date',
', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd')": "Enum('asc', 'desc', '')"}],
'start_date': 'String',
'termcause': 'String',
'time_zone': 'String',
'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_latest_stat', 'combine_point_hours')",
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
op_destination	N/A	True	String(allow_blank: False, regex: "^(\d+(, \s)?)*\$")	destination code(s), which need show in report, separated by comma
op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\d+(, \s)?)*\$")	destination code(s), which should not show in report, separated by comma
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ((([1-9] [12]\d 3[0-2]))? (0\.){3}0(/0)?)\$")	IP address of call originator
outgoing_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ((([1-9] [12]\d 3[0-2]))? (0\.){3}0(/0)?)\$")	IP address of the termination side

			2[0-4]\d 25[0-5])\.\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.)?{3}0(/0)?\$")	
hosts	N/A	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.)?{3}0(/0)?\$"))], min_length: 0)	filter by RTP server IP. A few addresses need separate by comma.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes
type	N/A	True	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_latest_stat', 'combine_point_hours')	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults - from user session
termcause	N/A	True	String(allow_blank: False)	filter by disconnect cause code
start_date	N/A	False	String(allow_blank:	start date and time for data

			False)	gathering. Report returns calls which setup_time not less than given.
op_class_id	N/A	True	Integer(gte: 0)	Shows in report only calls with given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed wrong".
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
sort_by	[]	True	List([Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by companies

min_attempts	0	True	Integer(gte: 0)	filter by attempts quantity
client_id	N/A	True	Integer(gt: 0)	not in use
owner_id	None	True	Or(Integer(gt: 0), Null)	filter by owner
fields	['tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd']	True	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
op_id	N/A	True	Integer(gt: 0)	filter by OP
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.

_incoming_calls_get_by_agg

```
{'data': {'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ['String'],
          'hosts': "[ 'String' ] or String",
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'min_attempts': 'Integer',
          'offset': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'owner_id': 'Integer or Null',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'sort_by': [{"Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration',
'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date',
'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd')": "Enum('asc', 'desc', '')"}]},
          'start_date': 'String',
          'time_zone': 'String',
          'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_a
```

```
nd_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destina
tion', 'combine_point_nas_ip', 'combine_point_hours')"},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optiona l	Type	Description
session_id	N/A	True	String(allow_blank: False)	not in use, by default - from user session
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given. End date will be rounded up to an hour.
op_id	N/A	True	Integer(gt: 0)	filter by OP
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
op_destinati on	N/A	True	String(allow_blank: False, regex: " [^] (\d+(, \s)?)*\$")	destination code(s), which need show in report, separated by comma

op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
incoming_ip	N/A	True	String(allow_blank: False, regex: "^[([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\/([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(\\/0)?\$")	IP address of call originator
hosts	N/A	True	Or(List([String(allow_blank: False, regex: "^[([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\/([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(\\/0)?\$"), min_length: 0), String(allow_blank: False, regex: "^[([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\/([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(\\/0)?\$"))	filter by RTP servers IP address. A few addresses need separate by comma.
type	N/A	False	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_point_hours')	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^[\\+\\-]?\\d{1,2}:\\d{2}\$")	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids

client_id	N/A	True	Integer(gt: 0)	not in use
fields	['tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd']	True	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger
min_attempts	0	True	Integer(gte: 0)	filter by attempts quantity
sort_by	[]	True	List([Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'attempts_hunt', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
owner_id	None	True	Or(Integer(gt: 0), Null)	filter by owner

Default fields:

_incoming_calls_get_individual

Name	Type & Comments
op_destination	String
tp_destination	String
scd	Int (setup connect delay, connect_time - setup_time in seconds)
tp_client_name	String
op_caller_number	String
clear_number	String
jurisdict	Bool
tp_protocol	String
op_price	numeric(16,5)
tp_client_id	Int
op_protocol	String
tp_class_id	Int
called_number_out	String
op_ip	String
op_class_name	String
op_id	Int
op_class_id	Int
ds	Int (call disconnect side)
dialpeer	String

local_disconnect_cause_name	String
tp_class_name	String
local_disconnect_cause	Int
op_media_ip	String
op_name	String
op_client_name	String
outgoing_ani	String
tp_caller_number	String
tp_name	String
connect_time	String
tp_codecs	String
duration	Int (call duration in seconds)
nas	String (RTP IP address)
disconnect_time	String
tp_id	Int
setup_time	String
calc_duration	Int
tp_ip	String
incoming_ani	String
op_codecs	String
tp_cost	numeric(16,6)

disconnect_cause_name	String
op_cost	numeric(16,5)
country	String
tp_price	numeric(16,5)
op_client_id	Int
disconnect_cause	Int
bill_id	String
pdd	Int - post dial delay(seconds)
called_number_in	String

_incoming_calls_get_by_cdr

Name	Type & Comments
tp_name	String
op_sum	numeric
tp_sum	numeric
delta_price	numeric - difference between op_sum and tp_sum
duration	Int
destination	String
nas	String (RTP IP address)
percentage	String - rate of calls per disconnect code (available only for types "combine_disconnect_code" and "combine_disconnect_code_and_destination")
ner	String

termcode	Int
attempts_hunt	Int
asr	String
tr_date	String - start date of group period (begin of the month for 'type' - "month" and so on)
calc_duration	Int
name_termcode	String
acd	numeric(15,2) (ACD in seconds) - takes into account only successful calls
country	String
attempts	Int
op_name	String
callback_rate	numeric(10,2)
sc	numeric(10,2) or numeric(15,2) or numeric - short calls percentage (N_short/N_success)*100%
some_xy	Bool - "True" if ASR <= 5% or ACD <= 60 seconds
sa	numeric - number of successful attempts
pdd	Int - post dial delay in seconds

_incoming_calls_get_by_agg

Name	Type & Comments
tp_name	String
op_sum	numeric
tp_sum	numeric

delta_price	numeric(16,5) - difference between op_sum and tp_sum
duration	Int (total duration in seconds)
destination	String
nas	String (RTP IP address)
percentage	String - rate of calls per disconnect code (only for types "combine_disconnect_code" and "combine_disconnect_code_and_destination")
ner	String
termcode	Int
attempts_hunt	Int
asr	String
tr_date	String - start date of group period (begin of the month for 'type' - "month" and so on)
calc_duration	Int
name_termcode	String
acd	Int (ACD in seconds) - takes into account only successful calls
country	String
attempts	Int
op_name	String
sc	numeric - short calls percentage
some_xy	Bool - "True" if ASR <= 5% or ACD <= 60 seconds
sa	numeric - number of successful attempts
pdd	Int - post dial delay

--	--

2.173 incoming_calls_get_old

Docstring:

Returns calls data of authorized incoming calls which passed through mediacore and has been initiated by originator during given time period.

To define from where to get data (from cdrs or aggregation) options "type" and "by_cdr" has been designed. To get each call separately chose type "individual". With type "individual" this function invokes function "_incoming_calls_get_individual" which returns data from cdrs ordered by 'setup_time' descending.

Function stores responses in cache during 5 minutes. That's why user should wait 5 minutes to get new calls data for the same request or change request params.

To get calls data combined in some way use necessary "type" ("combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" or "combine_disconnect_code") in conjunction with boolean option "by_cdr".

If "by_cdr" is "true" - it calls function "_incoming_calls_get_by_cdr_old" that returns data from cdrs, otherwise - calls function "_incoming_calls_get_by_agg" which takes data from aggregation.

In the second case ("type" = "combine_dest", "month" etc.) function may return less precise result for last several hours.

If type not individual key 'by_cdr' should be defined.

Note, that this function has limitation of maximal time period which depends on chosen "type":

- 1) 7 days for "type" "individual";
- 2) 31 day for "type": "combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" and "combine_disconnect_code";
- 3) 12 hours for "type" listed in 2) if flag "by_cdr" is "true".

_incoming_calls_get_by_agg

Function returns incoming calls statistic combined by chosen criteria from aggregation and thats why it doesn't account for traffic during last hour.

Example:

- Client Request

```
{"session_id":"152bab1c9d436bad4fb64435b9398e08",
  "data":{"limit":1,
    "time_zone":"0:00",
    "currency_id":1,
    "end_date":"2013-12-25T23:59:59",
    "not_use_like":false,
    "offset":0,
    "type":"day",
    "start_date":"2013-12-25T00:00:00",
    "by_cdr":false,
    "fields":["some_xy","tr_date","asr","acd","pdd","sc",
      "duration","attempts","sa","op_sum","tp_sum",
      "delta_price"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["some_xy", "tr_date", "asr", "acd", "pdd", "sc",
    "duration", "attempts", "sa", "op_sum", "tp_sum",
    "delta_price"],
  "message": "OK",
  "data": [[true, "2013-12-25T00:00:00+00:00", "0.48", 162, "3.2",
    "34.26", 169747, 217810, "1045",
    "113.164580000000000000000000000000",
    "75.393110000000000000000000000000", "37.77147"]],
  "time_elapsed": 0.23369}
```

Request structure and fields description:

_incoming_calls_get_by_agg

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.174 incoming_invoice_add

Docstring:

Creates new invoice for customer with given balance and given time period considering VAT if needed.

Invoice won't be created in the following cases:

- 1) If invoice for given period already exists.
- 2) If OP cost for invoicing period equal to zero.
- 3) If session id incorrect.
- 4) If "date_to" or "date_from" more than current and appropriate error message will be returned.

If time period in query covers several month function creates invoice for each one.

Function returns message within key "error".

In case function returned status 305 - then it is just informational message. In other cases key "error" contains message which describes error has been occurred.

Example:

- Client Request

```
{"session_id": "7b97510df478c3cf9f61c42a2629e6d5",
  "data": {"time_zone": "0:00",
           "date_from": "2013-11-01T00:00:00",
           "include_vat": false,
           "email": "test@test.com",
           "date_to": "2013-12-10T23:59:59",
           "due_date": "2013-12-13T23:59:59",
           "balance_id": 765,
           "notes": ""}
}
```

- Server Response

```
{"status": 305,
  "message": "Informal",
  "details": "Invoice for the period from 2013-11-01 00:00:00 to 2013-12-10
23:59:59 successfully created!"
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'due_date': 'String',
          'email': 'String',
          'include_vat': 'Bool',
          'notes': 'String',
          'service_id': 'Integer',
          'session_id': 'String',
          'time_zone': 'String'},
'session_id': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	by defaults from user session
email	N/A	False	String(allow_blank: False)	client billing email
date_to	N/A	False	String(allow_blank: False)	end date of invoice. Should be less then current
due_date	N/A	False	String(allow_blank: False)	time interval in days during which customer should pay invoice
notes	N/A	False	String(allow_blank: True)	
balance_id	N/A	False	Integer(gt: 0)	customer's balance id
date_from	N/A	False	String(allow_blank: False)	
include_vat	N/A	False	Bool	if True - VAT will be included in the invoice amount
service_id	1	True	Integer(gt: 0)	service type(voice - 1, sms - 2)
time_zone	0:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2})\$")	specify if company timezone different from GMT +00:00

2.175 incoming_invoice_get

Docstring:

Calls function `_incoming_outgoing_invoice_get`(with `"is_tp"=False`) that returns list of existing incoming invoices for specified time period, balance, company.

Example:

- Client Request:

```

{"session_id":"1f5adcacaa938949b74af1e14b8a0aab",
"data":{"limit":101,
"date_from":"2012-11-04T00:00:00",
"sort_by":[{"balance_name":""}],
"date_to":"2013-11-04T00:00:00",
"offset":0,
"fields":["number",
"balance_name",
"client_name",
"gen_date",
"from_date",
"to_date",
"amount_claimed",
"amount_protested",
"amount_total",
"status",
"mailed",
"email",
"id",
"amount_additional",
"notes",
"problems",
"owner_email",
"owner_name"]}
}

```

- Server Response:

```

{"status": 200,
"fields": ["number",
"balance_name",
"client_name",
"gen_date",
"from_date",
"to_date",
"amount_claimed",
"amount_protested",
"amount_total",
"status", "mailed",
"email",
"id",
"amount_additional",
"notes",
"problems",
"owner_email",
"owner_name"
],
"message": "OK",
"data": [[ "00044920130831c",
"Test_1_Balance (USD)",
"Test_1",
"2013-11-01T14:52:17.603884+00:00",
"2013-08-01T00:00:00",
"2013-08-31T23:59:59",
"29.20",
"0.00",
"29.20",
"In process",
false,
"test@test.com",
957,
"0.00",

```

```

        "",
        "",
        "test1@test.com",
        "Owner"]]
    }

```

_incoming_outgoing_invoice_get ([data_validation](#)) ([default_fields](#))

Returns list of existing incoming or outgoing invoices for specified time period, balance, company.

Request structure and fields description:

_incoming_outgoing_invoice_get

```

{'data': {'amount_additional': 'Float',
          'amount_greater': 'Float',
          'amount_less': 'Float',
          'balance_id': 'Integer',
          'company_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'fields': ["Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_
price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', '
balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'a
mount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additio
al', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email'
, 'vat', 'owner_id')"],
          'gen_date': 'String',
          'id': 'Integer',
          'is_tp': 'Bool',
          'limit': 'Integer',
          'notes': 'String',
          'number': 'String',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'period': 'Integer',
          'service_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_b
y_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date',
'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'a
mount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additio
nal', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email'
, 'vat', 'owner_id')": "Enum('asc', 'desc', '')"}]},
          'status_id': 'Integer',
          'type': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
amount_less	N/A	True	Float()	returns list of invoices with amount less than specified
type	N/A	True	Integer(gte: 0)	
gen_date	N/A	True	String(allow_blank: False)	creation date
session_id	N/A	True	String(allow_blank: False)	by defaults from user session
period	N/A	True	Integer(gte: 1, lte: 4)	filter by invoice period id
number	N/A	True	String(allow_blank: False)	search by invoice number
date_to	N/A	True	String(allow_blank: False)	upper boundary of request time slice. Combine inputted date with 23:59:59 time
amount_greater	N/A	True	Float()	returns list of invoices with amount more than specified
id	N/A	True	Integer(gt: 0)	invoice id in the database
date_from	N/A	True	String(allow_blank: False)	lower boundary of request time slice
is_tp	N/A	True	Bool	if "true" returns outgoing invoices, else - incoming
owner_id	N/A	True	Integer(gt: 0)	filter by owner

notes	""	True	String(allow_blank: True)	notes. amount protested is here
company_id	N/A	True	Integer(gt: 0)	filter by company id
amount_additional	N/A	True	Float()	additional amount for price correction
service_id	N/A	True	Integer(gt: 0, lt: 3)	filter by service(1 - voice, 2 - SMS). To get catalog info about service types use function services_get.
fields	['status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id']	True	List([Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
balance_id	N/A	True	Integer(gt: 0)	filter by balance id
sort_by	[]	True	List([Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id',	field on which result will be

			'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id'): Enum('asc', 'desc', ''), min_length: 0)	sorted
limit	0	True	Integer(gte: 0)	says how many rows will be in response
status_id	N/A	True	Integer(gte: 0)	invoice status id, 0 - "In process", 1- "Accepted", 2 - "Protested", 3 - "Closed", 4 - "Deleted", 5 - "Inv. changed"

Default fields:

_incoming_outgoing_invoice_get

Name	Type & Comments
status	String
due_date	String - date when invoice should be paid
attempts	Int
owner_name	String
correct_by_price	Boolean
status_id	Int

service_name	String
owner_email	String
problems	String
number	String
gen_date	String
balance_name	String
mailed	Boolean
to_date	String
balance_id	Int
duration	Int
id	Int
amount_total	numeric(16,2)
amount_protested	numeric(16,2)
amount_vat	numeric(16,2)
notes	String
time_zone	String
amount_claimed	numeric(16,2)
amount_additional	numeric(16,2)
from_date	String
client_email_real	String - actual client's billing email
vat_rate	String - current value of vat rate (without "%" sign)

client_name	String
service_id	Int
email	String - client's billing email that was at the time of invoice generation
vat	Boolean
owner_id	Int

2.176 incoming_invoice_set

Docstring:

Allows to change "amount_additional" value, recipient's email address and status of given client's invoice. The function adds new entry about changes to the action log and returns invoice id in database.

Example:

- Client Request

```
{"session_id":"86971e161ba0a6932fba4758ed1b99cf",
  "data":{"id":378,
          "amount_additional":0.0,
          "email":"managers@speedflow.com",
          "status_id":4,
          "notes":""}}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 378, "time_elapsed": 0.040393}
```

Request structure and fields description:

```
{'data': {'amount_additional': 'Float',
          'correct_by_price': 'Bool',
          'email': 'String',
          'id': 'Integer',
          'notes': 'String',
          'session_id': 'String',
          'status_id': 'Integer'},
 'session_id': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	invoice id
status_id	N/A	False	Integer(gte: 0, lte: 5)	new invoice status id, 0 - "In process", 1- "Accepted", 2 - "Protested", 3 - "Closed", 4 - "Deleted", 5 - "Inv. changed"
email	N/A	False	String(allow_blank: False)	invoice recipient's email address
amount_additional	N/A	True	Float()	This value will be added to the company balance if it allowed in database. If not - only affects on total invoice amount.
correct_by_price	False	True	Bool	is invoice was changed by price
notes	""	False	String(allow_blank: True)	notes. amount protested is here
session_id	N/A	True	String(allow_blank: False)	

2.177 incoming_sms_get

Docstring:

`_incoming_sms_get_individual` ([data_validation](#)) ([default_fields](#))

Returns authorized sms records for each sms individually ("type":"individual") for given time period ordered by 'setup_time' descending.

Example:

- Client REQUEST:

```
{"session_id":"b8dc6dcf5de3c7e7751c19233d1a157a",  
  "data":{"limit":1,  
          "time_zone":"0:00",  
          "end_date":"2014-03-03T23:59:59",
```

```

    "currency_id":1,
    "not_use_like":false,
    "offset":0,
    "type":"individual",
    "start_date":"2014-03-03T00:00:00",
    "fields":["setup_time","op_name",
              "tp_name","clear_number",
              "status","op_price",
              "tp_price","op_cost",
              "tp_cost","op_ip",
              "tp_host","nas",
              "incoming_ani","outgoing_ani",
              "bill_id","dialpeer",
              "country"]}
  }

```

- Server RESPONSE:

```

{'status': 200,
 'fields': ['setup_time', 'op_name', 'tp_name', 'clear_number',
            'status', 'op_price', 'tp_price', 'op_cost',
            'tp_cost', 'op_ip', 'tp_host', 'nas', 'incoming_ani',
            'outgoing_ani', 'dialpeer', 'country'],
 'message': 'OK',
 'data': [['2014-03-03T14:26:47.127183+00:00', 'SMS_OP', 'SMS_TP',
            '380632588588', 'ESME_RMSGQFUL', '0.00010', '0.00001',
            '0.00010', '0.00001', '188.230.126.178', '188.230.126.180',
            '193.161.87.53', '380632588333', '380632588333', 'SMS_Dialpeer',
            'Country']],
 'time_elapsed': 0.020364}

```

_incoming_sms_get_by_agg ([data_validation](#)) ([default_fields](#))

_incoming_sms_get_by_cdr ([data_validation](#)) ([default_fields](#))

_incoming_sms_get_by_cdr_by_state ([data_validation](#)) ([default_fields](#))

_incoming_sms_get_mcc_mnc_cdr ([data_validation](#)) ([default_fields](#))

_incoming_sms_get_mcc_mnc_agg ([data_validation](#)) ([default_fields](#))

Request structure and fields description:

_incoming_sms_get_individual

```
{'data': {'ani': 'String',
          'billed_sms_only': 'Bool',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('status', 'nas', 'tp_mnc', 'called_number_out', 'op_destinat
ion', 'tp_destination', 'tp_name', 'tp_client_id', 'tp_host', 'setup_time', 'message_id
', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'clear_number', 'dialpe
er', 'incoming_ani', 'tp_mcc', 'dlr_receive_time', 'tp_cost', 'op_ip', 'dlr', 'tp_proto
col', 'country', 'tp_price', 'op_client_id', 'op_name', 'called_number_in', 'tp_id', 'o
p_cost', 'state', 'bill_id', 'op_client_name', 'hlr_type', 'op_price', 'local_code_na
me', 'outgoing_ani', 'tp_caller_number', 'op_protocol')"],
          'hosts': "[ 'String' ] or String",
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_class_id': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('status', 'nas', 'tp_mnc', 'called_number_out', 'op_destin
ation', 'tp_destination', 'tp_name', 'tp_client_id', 'tp_host', 'setup_time', 'message
_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'clear_number', 'dial
peer', 'incoming_ani', 'tp_mcc', 'dlr_receive_time', 'tp_cost', 'op_ip', 'dlr', 'tp_pro
tocol', 'country', 'tp_price', 'op_client_id', 'op_name', 'called_number_in', 'tp_id',
'op_cost', 'state', 'bill_id', 'op_client_name', 'hlr_type', 'op_price', 'local_code_na
me', 'outgoing_ani', 'tp_caller_number', 'op_protocol')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELI
V', 'UNKNOWN', 'ENROUTE')",
          'status': 'Integer',
          'time_zone': 'String',
          'tp_host': 'String or String',
          'type': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
hosts	N/A	True	Or(List([String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]))\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])/([1-9] 12\\d 3[0-2]))?(0\\.){3}0(0)?\$")), min_length: 0), String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]))\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])/([1-9] 12\\d 3[0-2]))?(0\\.){3}0(0)?\$"))	filter by RTP server IP
sort_by	[]	True	List([Enum('status', 'nas', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'clear_number', 'dialpeer', 'incoming_ani', 'tp_mcc', 'dlr_receive_time', 'tp_cost', 'op_ip', 'dlr', 'tp_protocol', 'country', 'tp_price', 'op_client_id', 'op_name', 'called_number_in', 'tp_id', 'op_cost', 'state', 'bill_id', 'op_client_name', 'hlr_type', 'op_price', 'local_code_name', 'outgoing_ani', 'tp_caller_number', 'op_protocol'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
start_date	N/A	False	String(allow_blank: False)	start date and time for

				data gathering. Report returns calls which setup_time not less than given.
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
op_id	N/A	True	Integer(gt: 0)	filter by OP id.
op_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma.
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9] 12\\d 3[0-2]))?(0\\.){3}0(/0)?)\$")	Originator's IP.
owner_id	0	False	Integer(gte: 0)	filter by owner
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^(([a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9][A-Za-	Terminator's host or IP.

			z0-9][A-Za-z0-9\-*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d 1-9)\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d 1-9)\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?)\$"))	
client_id	N/A	True	Integer(gt: 0)	filter by OP company id.
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
limit	N/A	False	Integer(gt: 0)	says how many rows will be in response.
session_id	N/A	True	String(allow_blank: False)	not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
type	N/A	False	String(allow_blank: False, regex: "^individual\$")	statistical resolution type.
clients_ids	[]	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids.
ani	N/A	True	String(allow_blank: False, regex: "^[a-zA-Z0-9#*]*\$")	filter by caller number or first digits of caller number
op_class_id	N/A	True	Integer(gte: 0)	shows in report only calls with given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed

				wrong".
fields	['status', 'nas', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'clear_number', 'dialpeer', 'incoming_ani', 'tp_mcc', 'dlr_receive_time', 'tp_cost', 'op_ip', 'dlr', 'tp_protocol', 'country', 'tp_price', 'op_client_id', 'op_name', 'called_number_in', 'tp_id', 'op_cost', 'state', 'bill_id', 'op_client_name', 'hlr_type', 'op_price', 'local_code_name', 'outgoing_ani', 'tp_caller_number', 'op_protocol']	True	List([Enum('status', 'nas', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'clear_number', 'dialpeer', 'incoming_ani', 'tp_mcc', 'dlr_receive_time', 'tp_cost', 'op_ip', 'dlr', 'tp_protocol', 'country', 'tp_price', 'op_client_id', 'op_name', 'called_number_in', 'tp_id', 'op_cost', 'state', 'bill_id', 'op_client_name', 'hlr_type', 'op_price', 'local_code_name', 'outgoing_ani', 'tp_caller_number', 'op_protocol')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
offset	0	True	Integer(gte: 0)	offset says to skip that many rows before beginning to return rows.
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
not_use_like	False	True	Bool	if 'true' searches only calls by exact destination codes without subcodes
status	N/A	True	Integer()	sms status
billed_sms_only	False	False	Bool	show only billed sms filter.

_incoming_sms_get_by_agg

```
{'data': {'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
          'network_code': 'String',
          'offset': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'status': 'Integer',
          'time_zone': 'String',
          'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip')"},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	True	Integer(gt: 0)	# filter by OP id
type	N/A	False	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip')	method of grouping
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsence of filtration.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before

				beginning to return rows. Currently not in use.
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
op_destination	N/A	True	String(allow_blank: False, regex: " [^] (\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
status	N/A	True	Integer(gte: 0)	filter by SMS status
time_zone	+00:00	True	String(allow_blank: False, regex: " [^] ([+-]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
op_destination_exclude	N/A	True	String(allow_blank: False, regex: " [^] (\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given. End date will be rounded up to an hour.
op_ip	N/A	True	String(allow_blank: False, regex: " [^] (([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\/(([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(\\/0)?)")	IP address of call originator

			\$")	
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger ids.
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
hosts	[]	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])/(\\d [1-9] 12 \\d 3[0-2]))?(0\\.){3}0/(0)?)\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
client_id	N/A	True	Integer(gt: 0)	not in use
sort_by	[]	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
fields	['tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name']	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

_incoming_sms_get_by_cdr

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
```

```

'limit': 'Integer',
'longer_than': 'Integer',
'managers': ['Integer'],
'message_id': 'String or Null',
'network_code': 'String',
'not_use_like': 'Bool',
'offset': 'Integer',
'op_destination': 'String',
'op_destination_exclude': 'String',
'op_id': 'Integer',
'op_ip': 'String',
'owner_id': 'Integer',
'session_id': 'String',
'shorter_than': 'Integer',
'sort_by': [{"Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
'start_date': 'String',
'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
'status': 'Integer',
'time_zone': 'String',
'tp_host': 'String or String',
'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip')",
'session_id': 'String'

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
hosts	[]	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/(([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?)\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in

				specific currency. By defaults - from user session.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
op_id	N/A	True	Integer(gt: 0)	# filter by OP id
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(0\.){3}0(/0)?)\$")	Originator's IP.
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^([a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9][A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(0\.){3}0(/0)?)\$"))	Terminator's host or IP.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes

time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2}\$")	"start_date" and "end_date" timezone
client_id	N/A	True	Integer(gt: 0)	not in use
ani	N/A	True	String(allow_blank: False, regex: "^[a-zA-Z0-9#]*\$")	filter by caller number or first digits of caller number
fields	['tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name']	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
type	N/A	False	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip')	method of grouping
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
op_destinati on	N/A	True	String(allow_blank: False, regex: "^\\d+(, ,s)?*\$")	destination code(s), which need show in report, separated by comma
op_destinati on_exclude	N/A	True	String(allow_blank: False, regex: "^\\d+(, ,s)?*\$")	destination code(s), which should not show in report, separated

				by comma
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	
sort_by	[]	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adscence of filtration.
status	N/A	True	Integer(gte: 0)	filter by SMS status
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code

_incoming_sms_get_by_cdr_by_state

```
{'data': {'ani': 'String',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('state', 'share', 'attempts')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
```

```

    'sort_by': [{"Enum('state', 'share', 'attempts')": "Enum('asc', 'desc', '')"}
],
    'start_date': 'String',
    'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELI
V', 'UNKNOWN', 'ENROUTE')",
    'status': 'Integer',
    'time_zone': 'String',
    'type': "Enum('combine_by_state',)",
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
op_id	N/A	True	Integer(gt: 0)	# filter by OP id
op_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show

			s?)*\$")	in report, separated by comma
op_ip	N/A	True	String(allow_blank: False, regex: "^([1-9][1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9][12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$")	IP address of call originator
hosts	[]	True	List([String(allow_blank: False, regex: "^([1-9][1-9]\d{1}\d{2} 2[0-4]\d{25[0-5]})\.((\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})\.){2}(\d{1-9}\d{1}\d{2} 2[0-4]\d{25[0-5]})(/([1-9][12]\d{3[0-2]}))?(0\.){3}0(/0)?)\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes
type	N/A	False	Enum('combine_by_state',)	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+]??\d{1,2}):(\d{2})\$")	"start_date" and "end_date" timezone
status	N/A	True	Integer(gte: 0)	filter by SMS status
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
ani	N/A	True	String(allow_blank: False, regex: "^([a-zA-Z0-9#]*)*\$")	filter by caller number or first digits of caller number
fields	['state', 'share', 'attempts']	True	List([Enum('state', 'share', 'attempts')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response.

				Currently not in use.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
sort_by	[]	True	List([Enum('state', 'share', 'attempts'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means absence of filtration.

_incoming_sms_get_mcc_mnc_cdr

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String or String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'sort_by': [{"Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
          'status': 'Integer',
          'time_zone': 'String',
          'tp_host': 'String or String',
```

```
'type': "Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')"},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
hosts	[]	True	List([String(allow_blank: False, regex: "^[([1-9][1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.((\d [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.){2}(\d [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])/([1-9][12]\d{3}[0-2])?(0\.){3}0(0)?\$")], min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
op_id	N/A	True	Integer(gt: 0)	# filter by OP id
offset	N/A	True	Integer(gte: 0)	Offset says to skip

				that many rows before beginning to return rows. Currently not in use.
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
sort_by	[]	True	List([Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsence of filtration.
op_destinati on	N/A	True	String(allow_blank: False, regex: "^(\\d+ (, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
op_destinati on_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+ (, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
status	N/A	True	Integer(gte: 0)	filter by SMS status
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])) ([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(0)?)\$")	Originator's IP.
tp_host	N/A	True	Or(String(allow_blank: False, regex:	Terminator's host or

			"^(([a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9][A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.)(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])/(([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(0)?)\$")	IP.
type	N/A	False	Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
ani	N/A	True	String(allow_blank: False, regex: "^[a-zA-Z0-9#]**\$")	filter by caller number or first digits of caller number
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
client_id	N/A	True	Integer(gt: 0)	not in use
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
fields	['tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name']	True	List([Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

network_code	""	True	Or(String(allow_blank: False, regex: "^[0-9]*\$", min_length: 3, max_length: 6), String(allow_blank: True, max_length: 0))	filter by message MCC MCN. Works only with type 'combine_mcc_mnc' or 'combine_mcc_mnc_and_point'.
--------------	----	------	--	---

_incoming_sms_get_mcc_mnc_agg

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String or String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_destination': 'String',
          'op_destination_exclude': 'String',
          'op_id': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'sort_by': [{"Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
          'status': 'Integer',
          'time_zone': 'String',
          'tp_host': 'String or String',
          'type': "Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')"},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_ip	N/A	True	String(allow_blank: False, regex: "^[([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?{2}(\d [1-	Originator's IP.

			9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))? (0\\.){3}0(0)?)\$")	
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^(([a-zA-Z0-9] [a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9] [A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))? (0\\.){3}0(0)?)\$"))	Terminator's host or IP.
hosts	[]	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))? (0\\.){3}0(0)?)\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes
type	N/A	False	Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
ani	N/A	True	String(allow_blank: False, regex: "^[a-zA-Z0-9#]**\$")	filter by caller number or first digits of caller number
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
client_id	N/A	True	Integer(gt: 0)	not in use

state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
fields	['tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name']	True	List([Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	
sort_by	[]	True	List([Enum('tp_name', 'op_mcc', 'attempts', 'op_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering.

				Report returns calls which setup_time not more than given.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsence of filtration.
op_id	N/A	True	Integer(gt: 0)	# filter by OP id
status	N/A	True	Integer(gte: 0)	filter by SMS status
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
network_code	""	True	Or(String(allow_blank: False, regex: "^[0-9]*\$", min_length: 3, max_length: 6), String(allow_blank: True, max_length: 0))	filter by message MCC MCN. Works only with type 'combine_mcc_mnc' or 'combine_mcc_mnc_and_point'.
op_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
op_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma

Default fields:

_incoming_sms_get_individual

Name	Type & Comments
status	Int
nas	String(RTP IP address)
tp_mnc	String(3)
called_number_out	String
op_destination	String
tp_destination	String
tp_name	String
tp_client_id	Int
tp_host	String
setup_time	String
message_id	String
op_id	Int
op_mnc	String(3)
op_mcc	String(3)
local_code	String
tp_client_name	String
clear_number	String
dialpeer	String
incoming_ani	String

tp_mcc	String(3)
dlr_receive_time	String
tp_cost	numeric(16,5)
op_ip	String
dlr	Int
tp_protocol	String(100)
country	String
tp_price	numeric(16,5)
op_client_id	Int
op_name	String
called_number_in	String
tp_id	Int
op_cost	numeric(16,5)
state	String
bill_id	String
op_client_name	String
hlr_type	String ('no', 'external', 'local')
op_price	numeric(16,5)
local_code_name	String
outgoing_ani	String

tp_caller_number	String(200)
op_protocol	String(100)

_incoming_sms_get_by_agg

Name	Type & Comments
tp_name	String
attempts	Int
destination	String
tr_date	String
delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

_incoming_sms_get_by_cdr

Name	Type & Comments
tp_name	String
attempts	Int
destination	String
tr_date	String

delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

_incoming_sms_get_by_cdr_by_state

Name	Type & Comments
state	String
share	Numeric(6,2)
attempts	Int

_incoming_sms_get_mcc_mnc_cdr

Name	Type & Comments
tp_name	String
op_mcc	String(3)
attempts	Int
op_mnc	String(3)
delta_price	numeric(16,5)
op_cost	numeric

country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

_incoming_sms_get_mcc_mnc_agg

Name	Type & Comments
tp_name	String
op_mcc	String(3)
attempts	Int
op_mnc	String(3)
delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

2.178 information_messages_get

Docstring:

Returns messages about system failures such as:

- RTP/Balancer/Radius is unavailable;
- discrepancies in CDR's and aggregation data (check only if it enabled in

cron)

For message_code = 1 message_data contains two elements:

- host name
- host IP

For message_code 2 and 3 it contains single value - "CDR" or "XDR".

Example:

- Client Request

```
{
  "session_id": "24f5ed5f3d63d14ec4ff369de7f91dcd",
  "data": {"type": 1}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["short_message", "full_message", "type"],
  "message": "OK",
  "data": [
    ["RTP host is not available.", "RTP host 193.161.85.70 is not avail
able.", 1]
  ],
  "time_elapsed": 26.865136
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('message_data', 'full_message', 'type', 'message_code', 'short_message')"],
  'types': ['Integer']},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['message_data', 'full_message', 'type', 'message_code', 'short_message']	True	List([Enum('message_data', 'full_message', 'type', 'message_code', 'short_message']), min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

types	[1, 2, 3]	True	List([Integer(gte: 0)], min_length: 0)	filter by message type (level). 1 - critical, 2 - warning, 3 - info,
-------	-----------	------	---	---

Default fields:

Name	Type & Comments
message_data	List(String())
full_message	String
type	Int - message level: 1 - critical, 2 - warning, 3 - info
message_code	Int - 1: host(RTP/Radius/Balancer/SMS) access ability error, 2: CDR/XDR comparison hasn't been executed, 3: CDR/XDR agg comparison failure
short_message	String

2.179 internal_sms_disconnect_cause_get

Docstring:

Returns internal disconnect cause codes of mediacore and their description for SMS service.

Example:

- Client Request

```
{
  "session_id": "b7ea9fbc9d2c50d7fcdb26a000156c89",
  "data": {
    "sort_by": [{"disconnect_cause": ""}],
    "fields": ["disconnect_cause", "description"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["disconnect_cause", "description"],
  "message": "OK",
}
```

```

    "data": [
      [0, "OK"],
      [1000, "OP not found"],
      [1001, "OP Lines capacity is exceeded"],
      [1002, "OP not enabled"],
      [1003, "OP Tariff is not assigned"],
      [1004, "OP price is not found"],
      [1005, "OP price is not allowed"],
      [1006, "No TP found"],
      [1009, "OP called restricted"],
      [1010, "Called number exceed max digits limit after all manipulation. Due to internal limits"],
      ...
    ],
    "time_elapsed": 0.018605
  }

```

Request structure and fields description:

```

{
  'data': {
    'fields': ["Enum('disconnect_cause', 'description')"],
    'limit': 'Integer',
    'offset': 'Integer',
    'sort_by': [{"Enum('disconnect_cause', 'description')": "Enum('asc', 'desc', '')"}]},
  'session_id': 'String'
}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['disconnect_cause', 'description']	True	List([Enum('disconnect_cause', 'description')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([{"Enum('disconnect_cause', 'description')": "Enum('asc', 'desc', '')"}], min_length: 0)	field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
disconnect_cause	Int
description	String

2.180 invoice_amount_claimed_get

Docstring:

Function returns sum for selected balance and time period (considering time zone of the balance invoice settings) from aggregation in the currency of balance.

Function also returns keys "vat_pct", "amount_vat" and "amount" which have the same values as allowed within "data" key.

Function doesn't check if invoice for given time period and balance already exist.

Example:

- Client Request

```
{
  "session_id": "8efce2f78c7087eafbf88bb8c58390b9",
  "data": {
    "date_from": "2013-01-09T00:00:00",
    "include_vat": false,
    "direct_inc": true,
    "date_to": "2014-01-09T23:59:59",
    "balance_id": 844,
    "service_id": 1
  }
}
```

- Server Response

```
{
  "status": 200,
  "vat_pct": "0.0",
  "amount_vat": "0.0",
  "fields": ["amount", "amount_vat", "vat_pct", "vat_value"],
  "message": "OK",
  "data": ["0.0", "0.0", "0.0", "0.0"],
  "amount": "0.0",
  "time_elapsed": 0.014283
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'direct_inc': 'Bool',
```

```

        'fields': ["Enum('vat_pct', 'amount', 'amount_vat', 'vat_value')"],
        'include_vat': 'Bool',
        'service_id': 'Integer',
        'session_id': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
direct_in c	N/A	False	Bool	specify "true" for customer invoices or "false" for supplier.
session_i d	N/A	True	String(allow_blank: False)	not in use, by default from user's session
fields	['vat_pct', 'amount', 'amount_vat', 'vat_value']	True	List([Enum('vat_pct', 'amount', 'amount_vat', 'vat_value')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
include_v at	False	True	Bool	if "true" VAT will be added to the amount
balance_i d	N/A	False	Integer(gt: 0)	
date_to	N/A	False	String(allow_blank: False)	upper boundary for the time slice
service_i d	N/A	True	Integer(gt: 0, lt: 3)	filter by service id. 1 - SMS service, 2 - Voice. service
date_fro m	N/A	False	String(allow_blank: False)	lower boundary for the time slice

Default fields:

Name	Type & Comments
vat_pct	numeric(16,5) - VAT rate for whole system if "include_vat" enabled in the invoice settings, else - 0.0

amount	numeric(16,5) - amount without VAT
amount_vat	numeric(16,5) - amount including VAT
vat_value	numeric(16,5) - VAT amount = ("vat_pct" * "amount") / 100

2.181 invoice_data_cancel

Docstring:

Reverts correction of the given invoice, replacing corrected data with data from aggregation.

Example:

- Client Request

```
{"session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {"id": 7}}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'session_id': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer()	id of the corrected record which represents aggregated by destination code calls data for the invoice in the database. This id is returned by function invoice_data_get.
session_id	N/A	True	String(allow_blank: False)	

2.182 invoice_data_get

Docstring:

If 'id' of invoice is specified in request calls function `_invoice_data_get_by_id` that returns aggregated by destination code calls data for given invoice.

If in 'is_correct'="true" in query, then calls function `_invoice_correct_data_get` that returns invoice data from database table `invoices_correct`.

If there is no data on the invoice in given table, then inserts data from aggregation and returns them.

If 'is_correct' is false or is absent and 'id' is absent in query then calls function `_invoice_data_get`.

`_invoice_correct_data_get` ([data_validation](#)) ([default_fields](#))

Get data of invoice with given id from database combined by destination code.

If there is no data on the invoice in given table, then inserts data from aggregation and returns it.

Example:

- Client Request

```
{"session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {"is_correct": true,
           "invoice_id": 2,
           "fields": ["id", "destination", "country", "duration", "price",
                     "price_real", "amount", "is_correct"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "destination", "country", "duration", "price",
            "price_real", "amount", "is_correct"],
  "message": "OK",
  "data": [[1, "923", "Pakistan Mobile Other Carriers", "0",
            "0.04000", "0.04000", "0.00000", false],
           [2, "9230", "Pakistan Mobile Mobilink", "63560",
            "0.04000", "0.04000", "42.37331", false],
           [3, "9231", "Pakistan Mobile Paktel", "7996", "0.04000",
            "0.04000", "5.33063", false],
           [4, "9232", "Pakistan Mobile Warid", "5145", "0.04000",
            "0.04000", "3.43001", false],
           [5, "9233", "Pakistan Mobile Ufone", "1997", "0.04000",
            "0.04000", "1.33132", false],
           [6, "9234", "Pakistan Mobile Telnor", "23376", "0.04000",
            "0.04000", "15.58404", false]],
  "time_elapsed": 0.097993}
```

`_invoice_data_get_by_id` ([data_validation](#)) ([default_fields](#))

Returns aggregated by destination code calls data for given invoice.

Example:

- Client Request

```
{"session_id": "c8fccea343fde6df6355369d476af33d",
  "data": {"id": 1096,
    "fields": ["destination", "country", "duration", "price",
      "price_real", "amount", "is_correct",
      "from_date", "to_date"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["destination", "country", "duration", "price",
    "price_real", "amount", "is_correct", "from_date",
    "to_date"],
  "message": "OK",
  "data": [[ "923", "\u041f\u0430\u0430\u0438\u0441\u0442\u0430\u0434",
    "24655", "0.04000", "0.04000", "16.43665", false,
    "2013-11-01T00:00:00", "2013-11-03T23:59:59" ]],
  "time_elapsed": 0.083282}
```

`_invoice_data_get` ([data_validation](#)) ([default_fields](#))

Function returns calls data grouped by destination codes for invoice with given id.

Excluding fields mentioned in the "default_fields" function also returns following data:

- "problems", - message which reflects issues that happened during generation(for example - billing email is absent)
- "from_date", - start date of the invoicing period
- "to_date", - end date of the invoicing period
- "amount_claimed" - invoice amount without "amount_additional"
- "duration" - total duration of calls during the invoice period
- "number" - invoice number
- "attempts" - total number of call attempts during the invoice period
- "currency_sign" - balance's currency sign
- "currency_name" - name of balance currency
- "balance_name" - name of balance
- "client_name" - name of the client
- "due_date" - date when invoice should be paid
- "amount_vat" - VAT amount. It calculates as part of
- "amount_claimed" + "amount_additional"
- "time_zone" - balance's timezone
- "client_phone" - client's phone number
- "client_address" - client's address
- "email" - client's billing email address
- "invoicing_date" - date of the invoice creation

- "correct_by_price" - if True means that price in the invoice was corrected and additional amount was calculated according to the new one
- "amount_additional" - invoice additional amount. May be positive or negative.
- "owner_id" - id of the owner to which company's balance is assigned
- "vat_number" - client's vat number
- 'service_id' - 1 - Voice, 2 - SMS service
- 'sms_transmitted' - number of SMS have been successfully transmitted in case when service_id = 2 (SMS service) and the same as duration
- 'include_vat' - Boolean (says to include VAT value in invoices or not)

Request structure and fields description:

_invoice_correct_data_get

```
{'data': {'fields': ["Enum('origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id')"],
          'invoice_id': 'Integer',
          'session_id': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	by defaults from user session
invoice_id	N/A	False	Integer()	
fields	['origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id']	True	List([Enum('origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

_invoice_data_get_by_id

```
{'data': {'fields': ["Enum('origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id')"],
          'id': 'Integer',
          'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	by defaults from user session
id	N/A	True	Integer()	"id" column at billing.invoices_correct table
fields	['origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id']	True	List([Enum('origin_country', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'from_date', 'sms_transmitted', 'is_correct', 'amount', 'mcc', 'to_date', 'dst_ocn', 'duration', 'mnc', 'id')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

_invoice_data_get

```
{'data': {'fields': ["Enum('mnc', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'amount', 'sms_transmitted', 'time_zone', 'is_correct', 'attempts', 'origin_country', 'mcc', 'to_date', 'dst_ocn', 'duration', 'from_date', 'id')"],
          'invoice_id': 'Integer',
          'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_	N/A	True	String(allow_blank: False)	by defaults from

id				user session
invoice_id	N/A	False	Integer()	
fields	['mnc', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'amount', 'sms_transmitted', 'time_zone', 'is_correct', 'attempts', 'origin_country', 'mcc', 'to_date', 'dst_ocn', 'duration', 'from_date', 'id']	True	List([Enum('mnc', 'jurisdict', 'price_real', 'country', 'price', 'destination', 'dst_lata', 'amount', 'sms_transmitted', 'time_zone', 'is_correct', 'attempts', 'origin_country', 'mcc', 'to_date', 'dst_ocn', 'duration', 'from_date', 'id')], min_length: 0)	fields which will be present in response, if empty-fields mentioned in "default_fields"

Default fields:

_invoice_correct_data_get

Name	Type & Comments
origin_country	String
jurisdict	Boolean
price_real	numeric(16,5)
country	String
price	numeric(16,5)
destination	String
dst_lata	String
from_date	String
sms_transmitted	total number of sms have been received for region/destination and price
is_correct	boolean
amount	numeric(16,5)

mcc	String
to_date	String
dst_ocn	String
duration	Int
mnc	String
id	Int

_invoice_data_get_by_id

Name	Type & Comments
origin_country	String
jurisdict	Boolean
price_real	numeric(16,5)
country	String
price	numeric(16,5)
destination	String
dst_lata	String
from_date	String
sms_transmitted	total number of sms have been received for region/destination and price
is_correct	Boolean
amount	numeric(16,5)
mcc	String

to_date	String
dst_ocn	String
duration	Int
mnc	String
id	Int

_invoice_data_get

Name	Type & Comments
mnc	String
jurisdict	Boolean
price_real	Float
country	String
price	Float
destination	String
dst_lata	String
amount	numeric(16,2)
sms_transmitted	total number of sms have been received for region/destination and price
time_zone	String
is_correct	Boolean
attempts	Int
origin_country	String

mcc	String
to_date	end date of invoice period
dst_ocn	String
duration	Int
from_date	start date of invoice period
id	Int

2.183 invoice_data_set

Docstring:

Changes chosen price of given invoice and given time interval. Then updates "additional amount" (if new price different from old one) and recalculates "total amount" of the invoice considering "additional amount". This function returns error status 400 in the following cases:

- 1) If new price the same as old one,
- 2) if "from_date" less than start date of the invoice or "to_date" more than end date of chosen invoice
- 3) if no calls with duration for given time period.

Example:

- Client Request

```
{
  "session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {
    "id": 1,
    "price": 0.043,
    "to_date": "2014-01-15T23:00:00",
    "from_date": "2014-01-15T00:00:00"
  }
}
```

Request structure and fields description:

```
{'data': {'amount': 'Float',
          'from_date': 'String',
          'id': 'Integer',
          'price': 'Float',
          'session_id': 'String',
          'to_date': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer()	id of the record which represents aggregated by destination code calls data for the invoice in the database. This id is returned by function invoice_data_get.
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults from user session
price	N/A	False	Float()	new price value
amount	0	True	Float()	not in use
from_date	N/A	False	String(allow_blank: False)	start date
to_date	N/A	False	String(allow_blank: False)	end date

2.184 invoice_default_email_get

Docstring:

Returns billing email(s) of the company by chosen balance id or returns email of user who made request if the billing email is not specified.

Function also returns key "email" that contains company's billing email.

Example:

- Client Request

```
{"session_id": "8efce2f78c7087eafbf88bb8c58390b9",  
  "data": {"balance_id": 844}  
}
```

- Server Response

```
{"status": 200,  
  "time_elapsed": 0.0043,
```

```
"fields": ["email"],
"message": "OK",
"data": [{"test@mail.com"}],
"email": "test@mail.com"}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer', 'fields': ["Enum('email',)"]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	False	Integer()	
fields	['email']	True	List([Enum('email',)], min_length: 0)	required fields

Default fields:

Name	Type & Comments
email	String

2.185 invoice_del

Docstring:

Function sets status "deleted" to chosen invoice and fill template "email_financial_department" with appropriate data.

The notification will be send to emails specified in the appropriate template if any or to the owner's email.

Example:

- Client Request

```
{"session_id": "1f5adcacaa938949b74af1e14b8a0aab",
```

```
"data":{"id":959}
}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'session_id': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	
id	N/A	False	Integer(gt: 0)	invoice id

2.186 invoice_email_send

Docstring:

Generates invoice files(using API function invoice_report), attaches them to the mail, fill email template "invoice_report" with data of the invoice and add it to the sending queue. At the end function marks invoice as mailed.

This function is called by function "email_send" if "type"="invoice".

Request structure and fields description:

```
{'data': {'attach': ['String'],
            'body': 'String',
            'copy': ['String'],
            'from': 'String',
            'invoice_id': 'Integer',
            'mime_type': 'String',
            'subject': 'String',
            'to': ['String'],
            'type': 'String',
            'web_url': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
from	N/A	False	String(allow_blank: False)	email address of invoice sender
to	[]	True	List([String(allow_blank: True)], min_length: 0)	list email address(es) of the invoice recipient(s)
copy	[]	True	List([String(allow_blank: True)], min_length: 0)	list email address(es) to which copy of message will be sent
invoice_id	N/A	True	Integer(gt: 0)	invoice number
web_url	N/A	True	String(allow_blank: False)	download link for invoice file
subject	N/A	False	String(allow_blank: True)	
body	N/A	False	String(allow_blank: True)	
type	N/A	False	String(allow_blank: False, regex: "invoice")	"invoice"
attach	N/A	False	List([String(allow_blank: False)], min_length: 0)	list that contains invoices ids which will be attached to the mail
mime_type	html	True	String(allow_blank: False)	content type. Possible types: "plain" or "html". If chosen "html" then HTML tags will be interpreted, if "plain" - escape sequences.

2.187 invoice_periods_get

Docstring:

Get possible periods for invoice auto generation.

Example:

- Client Request

```
{"session_id": "4b679a9e6ce054ce74b60370754d5b05",  
  "data": {"fields": ["id", "name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[1, "each week"],  
            [2, "half-month"],  
            [3, "each month"],  
            [4, "daily"]],  
  "time_elapsed": 0.002887}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"]}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

Default fields:

Name	Type & Comments
id	Int
name	String

--	--

2.188 invoice_report

Docstring:

Generates invoice file in pdf or xls format depending on owner's settings or recreates already existing one.

Function returns download link for generated invoice. Download link TTL - is 2 hours.

Example:

- Client Request

```
{"session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {"regenerate": true,
           "invoice_id": 2,
           "fields": ["link"]}
}
```

- Server Response

```
{"status": 200,
  "link": "http://test.test.com:8070/cdn/download/5d8dd475bf98adfdecd908f7294c638e",
  "message": "OK",
  "data": [{"http://test.test.com:8070/cdn/download/5d8dd475bf98adfdecd908f7294c638e"}],
  "time_elapsed": 0.936588}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('link',)"],
           'invoice_id': 'Integer',
           'owner_id': 'Integer',
           'pdf_type': 'Integer',
           'regenerate': 'Bool',
           'use_xls_invoice': 'Bool'},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

invoice_id	N/A	False	Integer(gte: -1)	id of invoice is being generated. For preview (preset)generation of pdf or xls invoice id 0 (voice) or -1 (sms) should be used
owner_id	N/A	True	Integer()	owner's id for invoice preset generation
use_xls_invoice	N/A	True	Bool	option to generate xls or pdf preset
regenerate	False	True	Bool	if True regenerates already generated invoice
pdf_type	N/A	True	Integer()	option to generate pdf preset with given type id
fields	['link']	True	List([Enum('link',)], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

Default fields:

Name	Type & Comments
link	String

2.189 invoice_settings_get

Docstring:

Returns invoicing settings of the balance with given "balance_id".

Example:

- Client Request

```
{
  "session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {
    "balance_id": 1,
    "fields": [
      "period_in",
      "period_out",
      "time_zone",
      "due_days",
      "include_vat"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "period_in",
    "period_out",
    "time_zone",
    "due_days",
    "include_vat"
  ],
  "message": "OK",
  "data": [[1, 2, "0:00", 14, false]],
  "time_elapsed": 0.004888
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',  
          'fields': ["Enum('period_out', 'due_days', 'outgoing_time_zone', 'time_zone',  
                           'balance_id', 'period_in', 'incoming_time_zone')"]},  
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	False	Integer()	
fields	['period_out', 'due_days', 'outgoing_time_zone', 'time_zone', 'balance_id', 'period_in', 'incoming_time_zone']	True	List([Enum('period_out', 'due_days', 'outgoing_time_zone', 'time_zone', 'balance_id', 'period_in', 'incoming_time_zone')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"

Default fields:

Name	Type & Comments
period_out	Int
due_days	Int
outgoing_time_zone	String
time_zone	String
balance_id	Int
period_in	Int

incoming_time_zone	String
--------------------	--------

2.190 invoice_settings_set

Docstring:

Changes settings of invoice generation for given balance.

Example:

- Client Request

```
{
  "session_id": "4b679a9e6ce054ce74b60370754d5b05",
  "data": {
    "time_zone": "0:00",
    "period_out": 1,
    "period_in": 1,
    "balance_id": 1,
    "due_days": 14
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.034941
}
```

Request structure and fields description:

```
{
  'data': {
    'balance_id': 'Integer',
    'due_days': 'Integer',
    'incoming_time_zone': 'String',
    'outgoing_time_zone': 'String',
    'period_in': 'Integer',
    'period_out': 'Integer',
    'session_id': 'String',
    'time_zone': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
period_in	N/A	False	Integer()	generation period for customers invoices
outgoing_time	N/A	True	String(allow	This flag defines client's time zone. Supplier invoices of

e_zone			_blank: False)	the company generates accounting this value. Invoice amount depends from chosen timezone.
period_out	N/A	False	Integer()	generation period for suppliers invoices
time_zone	N/A	True	String(allow _blank: False)	Deprecated flag. This flag defines client's time zone. Invoices of the company generates accounting this value. Invoice amount depends from chosen timezone.
session_id	N/A	True	String(allow _blank: False)	not in use, by default from user session
due_days	N/A	False	Integer()	interval in days during which client should make payment for invoice
incoming_time_zone	N/A	True	String(allow _blank: False)	This flag defines client's time zone. Customer invoices of the company generates accounting this value. Invoice amount depends from chosen timezone.
balance_id	N/A	False	Integer()	

2.191 invoice_statuses_get

Docstring:

Returns list of possible invoice statuses.

Example:

- Client Request

```
{"session_id": "049a4d1709ba449a5588ceb602b3d001",
  "data": {"fields": ["id", "name"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[0, "In process"], [1, "Accepted"], [2, "Protested"],
            [3, "Closed"], [4, "Deleted"], [5, "Inv. changed"]],
  "time_elapsed": 2.058489}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],
            'limit': 'Integer',
            'offset': 'Integer',
            'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([{Enum('id', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted

Default fields:

Name	Type & Comments
id	int
name	String

2.192 invoices_gen

Docstring:

Generates auto invoices at given date and returns report about invoices which were created if 'is_report' = True.

If there is a time interval which overlaps with already created

invoice or if no traffic for given period on the balance sheet function adds new entry in the appropriate database table.

Info about invoices with problems will be added to the email notification "email_auto_invoice_create".

Example:

- Client Request

```
{
  "session_id": "12811adade330c03a976e691f97bc680",
  "data": {
    "gen_date": "2014-01-13T16:14:02",
    "fields": [
      "id", "balance_name", "problems", "from_date",
      "to_date", "amount_claimed", "duration", "number",
      "suppl_cust", "period", "currency_sign",
      "time_zone", "client_name"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "id", "balance_name", "problems", "from_date",
    "to_date", "amount_claimed", "duration", "number",
    "suppl_cust", "period", "currency_sign", "time_zone",
    "client_name"
  ],
  "message": "OK",
  "data": [
    [
      0, "Test_balance", "No traffic on the balance sheet.",
      "2014-01-06T00:00:00", "2014-01-12T00:00:00", "0.00",
      0, "0", "Customer", "each week", "$", "0:00",
      "Alexander_test_client"
    ],
    [
      0, "Test_supplier_balance", "No traffic on the balance sheet.",
      "2014-01-06T00:00:00", "2014-01-12T00:00:00", "0.00",
      0, "0", "Customer", "each week", "$", "0:00", "Test_supplier"
    ]
  ],
  "time_elapsed": 0.065751
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': ["Enum('service_name', 'problems', 'number', 'period', 'attempts',
    'to_date', 'duration', 'id', 'currency_sign', 'client_name', 'time_zone', 'amount_claimed',
    'from_date', 'suppl_cust', 'balance_name')"],
    'gen_date': 'String',
    'is_report': 'Bool',
    'session_id': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['service_name', 'problems',	True	List([Enum('service_name',	fields which will be

	'number', 'period', 'attempts', 'to_date', 'duration', 'id', 'currency_sign', 'client_name', 'time_zone', 'amount_claimed', 'from_date', 'suppl_cust', 'balance_name']		'problems', 'number', 'period', 'attempts', 'to_date', 'duration', 'id', 'currency_sign', 'client_name', 'time_zone', 'amount_claimed', 'from_date', 'suppl_cust', 'balance_name']], min_length: 0)	present in response, if empty - fields mentioned in "default_fields"
is_report	True	True	Bool	if True returns list of generated invoices within 'data' else 'data' will be an empty list
session_id	N/A	True	String(allow_blank: False)	not in use, by default from user session
gen_date	N/A	False	String(allow_blank: False)	date for which auto invoices will be generated

Default fields:

Name	Type & Comments
service_name	String
problems	String
number	String
period	String
attempts	Int
to_date	String
duration	Int
id	Int
currency_sign	String - balance's currency

	sign
client_name	String
time_zone	String
amount_claimed	String
from_date	String
suppl_cust	String
balance_name	String

2.193 ips_get

Docstring:

Returns list of IP addresses which assigned to Origination and(or) Termination points in chosen context(s).
The report displays the following information: point name to which IP assigned, prefix, port and IP record status - "enabled" or "disabled".

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "limit": 3,
    "sort_by": [{"ip": ""}],
    "show_tp": true,
    "contexts": [1],
    "offset": 0,
    "fields": ["id", "ip", "port", "prefix", "enabled", "pt_id", "pt_name", "is_op"]
  }
}
```

- Server response

```
{
  "status": 200,
  "fields": ["id", "ip", "port", "prefix", "enabled", "pt_id", "pt_name", "is_op"],
  "message": "OK",
  "data": [
    [2, "17.17.17.17/32", 5060, "741852963", true, 6, "TP1", false],
    [7, "181.233.162.19/32", null, "", true, 4, "Test_1_TP", false],
    [8, "181.233.162.18/32", 5060, "", true, 14, "pablo_tp", false]
  ]
}
```

```
"time_elapsed": 0.00966}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'enabled': 'Bool',
          'fields': ['String'],
          'ip': 'String',
          'limit': 'Integer',
          'offset': 'Integer',
          'pt_id': 'Integer',
          'pt_name': 'String',
          'show_op': 'Bool',
          'show_tp': 'Bool',
          'sort_by': [{"Enum('pt_name', 'ip', 'enabled', 'pt_id', 'prefix', 'is_op', 'port')": "Enum('asc', 'desc', '')"}]},
       'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
pt_id	N/A	True	Integer(gt: 0)	filter result by point id
show_op	N/A	True	Bool	if "true" returns only OPs where chosen IP is present
fields	['pt_name', 'ip', 'enabled', 'pt_id', 'prefix', 'is_op', 'port']	False	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
contexts	N/A	False	List([Integer()], min_length: 0)	filter by context id
limit	N/A	True	Integer(gte: 0)	says how many rows will be in response
show_tp	N/A	True	Bool	if "true" returns only TPs where chosen IP is present
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

ip	N/A	True	String(allow_blank: True)	filter by IP address
sort_by	[]	True	List([Enum('pt_name', 'ip', 'enabled', 'pt_id', 'prefix', 'is_op', 'port'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
enabled	N/A	True	Bool	show only enabled entries
pt_name	N/A	True	String(allow_blank: True)	filter by point name

Default fields:

Name	Type & Comments
pt_name	String
ip	String
enabled	Boolean
pt_id	Int
prefix	String(16)
is_op	Boolean
port	Int or null

2.194 jurisdictional_incoming_call_pathes_get

Docstring:

Returns list of Termination Points to which given OP have route for specific destination at the given time ordered according to the routing algorithm:

- 1) by priority of OP->Group relation
- 2) by dialpeer priority
- 3) by priority of dialpeer->TP relation
- 4) by weight of relation between dialpeer and TP if priority of dialpeer->TP relation is the same
- 5) by TP price priority
- 6) LCR (more profitable routes will be higher in list)

Function returns also boolean parameter - "dest_changed". It determines if tested number will be changed by some conversion defined in any route (only B number conversion in OP and dialpeer is taken into account).

Function returns parameters of the targeted OP within key 'op_data'.

'op_data' fields:

```
[
    'op_name',
    'op_capacity',
    'op_country',
    'op_price_id',
    'op_id',
    'op_enabled',
    'op_destination',
    'op_price_interstate',
    'op_price_intrastate',
    'op_price_indeterminate',
    'op_mc',
    'op_ci',
    'op_ft',
    'op_currency_id',
    'op_cid',
    'op_tarif_name',
    'op_price_tarif_interstate',
    'op_price_tarif_intrastate',
    'op_price_tarif_indeterminate',
    'op_currency_sign',
    'op_currency_name',
    'routing_algorithm',
    'op_tariff_invisible'
]
```

Example:

- Client Request

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "time_zone": "0:00",
    "time_period": "2014-01-13T15:32:50",
    "op_id": 7,
    "contexts": [1],
    "number": "923000000000",
    "fields": [
      "tp_enabled", "delta_price", "dif_price_interstate", "tp_id",
      "tp_price_id", "group_id", "dialpeer_id",
      "allow_internal", "tp_without_ip", "allowed_price",
      "tp_name", "tp_destination", "tp_country",
      "tp_price_interstate", "tp_price_tarif_interstate",
      "tp_currency_name", "tp_mc", "tp_ci", "tp_ft",
      "tp_capacity", "hunt_stop", "dp_name",

```

```

        "dp_tp_priority","dp_priority","weight",
        "group_name","op_group_priority","price_priority",
        "profit","client_id"]}]
    }

```

- Server Response

```

{"status": 200,
 "time_elapsed": 0.028728,
 "dest_changed": false,
 "fields": ["tp_enabled","delta_price","dif_price_interstate","tp_id",
            "tp_price_id","group_id","dialpeer_id",
            "allow_internal","tp_without_ip","allowed_price",
            "tp_name","tp_destination","tp_country",
            "tp_price_interstate","tp_price_tarif_interstate",
            "tp_currency_name","tp_mc","tp_ci","tp_ft",
            "tp_capacity","hunt_stop","dp_name",
            "dp_tp_priority","dp_priority","weight",
            "group_name","op_group_priority","price_priority",
            "profit","client_id"],
 "message": "OK",
 "op_data": ["H_in", 5, "Pakistan Mobile Mobilink", 13, 7, true,
             "9230", "10.4000", 1, 1, 0, 1, 5, "T_H_in1",
             "10.4000", "$", "USD", "QUALITY"],
 "data": [[true, "0.00000", "0.00500", 8, 20, 3, 4, false, false,
            true, "H_to", "9230", "Pakistan Mobile Mobilink",
            "10.3500", "10.3500", "USD", 1, 1, 0, 500, false,
            "H_p(CZ)", 0, 0, 0, "h_p (CZ)", 0, 0, "14.29", 6]]}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    'session_id': String(),
    'number': String(regex=B_NUMBER_REGEX),
    'time_period': String() ,
    'op_id': Int(gt=0),
    'time_zone': String(regex=TZ_REGEX),
    'show_disabled_pt': Bool(),
    'one_route': Bool(),
    'show_stat': Bool(),
    'fields': List(String()),
    'currency_id': Int(gt=0),
    'contexts': List(Int()),
    'limit': Int(gt=0),
    'offset': Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

'session_id'	session_info['id']	True	String()	user session id, need for checking user permissions. Currently not in use.
'number'	N/A	False	String(regex=B_NUMBER_REGEX)	called number
'time_period'	N/A	False	String()	request time in format yyyy-MM-ddT"HH:mm:ss
'op_id'	N/A	False	Int(gt=0)	id of chosen Origination Point
'time_zone'	'+00:00'	True	String(regex=TZ_REGEX)	time zone for 'time_period'
'show_disabled_pt'	False	True	Bool()	if "True" then disabled TPs also will be shown
'one_route'	False	True	Bool()	if "True" then report shows only one route per one Termination point.
'show_stat'	False	True	Bool()	if "True" then ACD,ASR for last 4 hours will returns
'fields'	default_fields	True	List(String())	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
'currency_id'	session_info['currency_id']	True	Int(gt=0)	currency in which price will display
'contexts'	N/A	False	List(Int())	ids context, in which request will be processed
'limit'	N/A	True	Int(gt=0)	says how many rows will be in response
'offset'	N/A	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'op_name'	String
'op_capacity'	Int
'op_country'	String
'op_price_id'	Int
'op_id'	Int
'op_enabled'	Boolean
'op_destination'	String
'op_ocn'	String(4)
'op_lata'	String(5)
'op_price_interstate'	numeric(16,5)
'op_price_intrastate'	numeric(16,5)
'op_price_indeterminate'	numeric(16,5)
'op_mc'	Int
'op_ci'	Int
'op_ft'	Int
'op_currency_id'	Int
'op_cid'	Int - OP currency id

'op_price_tarif_interstate'	numeric(16,5) - op price in the tariff's currency
'op_price_tarif_intrastate'	numeric(16,5) - op price in the tariff's currency
'op_price_tarif_indeterminate'	numeric(16,5) - op price in the tariff's currency
'op_currency_sign'	String
'op_currency_name'	String
'delta_price'	numeric(16,5)
'tp_name'	String
'tp_capacity'	Int
'tp_country'	String
'tp_price_id'	Int
'tp_id'	Int
'tp_enabled'	Boolean
'tp_destination'	String
'tp_ocn'	String(4)
'tp_lata'	String(5)
'tp_price_interstate'	numeric(16,5)
'tp_price_intrastate'	numeric(16,5)
'tp_price_indeterminate',	numeric(16,5)
'tp_mc'	Int

'tp_ci'	Int
'tp_ft'	Int
'tp_currency_id'	Int
'dif_price_interstate'	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default
'dif_price_intrastate'	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default
'dif_price_indeterminate'	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default
'profit'	numeric(15,2) = (dif_price/tp_price_default) * 100% if tp_price_default not equal 0, else 0
'dp_name'	String
'dp_capacity'	numeric(15,2)
'dp_tp_priority'	Int
'dp_priority'	Int
'group_name'	String
'op_group_priority'	Int
'price_priority'	Int
'hunt_stop'	Boolean
'group_id'	Int
'dialpeer_id'	Int
'weight'	Int
'client_id'	Int
'asr'	numeric or null

'acd'	Int or null
'delta_profit'	numeric(15,2)
'allow_internal'	Boolean
'tp_without_ip'	Boolean
'tp_price_tarif_interstate'	numeric(16,5) - tp price in the tariff's currency
'tp_price_tarif_intrastate'	numeric(16,5) - tp price in the tariff's currency
'tp_price_tarif_indeterminate'	numeric(16,5) - tp price in the tariff's currency
'tp_currency_sign'	String
'tp_currency_name'	String
'allowed_price'	Bool
'routing_algorithm'	String
'op_tariff_invisible'	Bool
'tp_tariff_invisible'	Bool
'op_charge_by'	String
'tp_charge_by'	String
'op_route_by'	String
'tp_route_by'	String
'delta_price_unit'	Int
'attempts'	Int

2.195 jurisdictional_outgoing_call_pathes_get

Docstring:

Function also used for SMS Routing Table.

If any global changes will be needed - create new function for SMS RT

Returns list of origination points which have route to given TP to specific destination at the given time.

If 'show_stat' not equal 'true' in request then list of routes will be ordered according to the routing algorithm:

- 1) by priority of OP->Group relation
 - 2) by dialpeer priority
 - 3) by priority of dialpeer->TP relation
 - 4) by weight of relation between dialpeer and TP if priority of dialpeer->TP relation is the same
 - 5) by TP price priority
 - 6) LCR (more profitable routes will be higher in list). Prices shows in the chosen in request currency, but compare in default currency.
- If in request 'show_stat' is 'true' list of routes will be ordered by difference between OP and TP price descending.

Function returns parameters of the targeted TP within key 'tp_data'.

Example:

- Client Request

```
{ "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": { "time_zone": "0:00",
            "time_period": "2014-01-13T16:01:18",
            "contexts": [1],
            "number": "9230000000000",
            "tp_id": 8,
            "fields": [ "op_enabled", "delta_price", "dif_price", "op_id",
                        "op_price_id", "group_id", "dialpeer_id",
                        "allowed_price", "op_name", "op_destination",
                        "op_country", "op_price_default",
                        "op_price_tarif", "op_currency_name", "op_mc",
                        "op_ci", "op_ft", "op_capacity", "dp_name",
                        "dp_tp_priority", "dp_priority", "group_name",
                        "op_group_priority", "profit", "client_id" ] }
}
```

- Server Response

```
{ "status": 200,
  "time_elapsed": 0.047496,
  "fields": [ "op_enabled", "delta_price", "dif_price", "op_id",
              "op_price_id", "group_id", "dialpeer_id",
              "allowed_price", "op_name", "op_destination",
              "op_country", "op_price_default", "op_price_tarif",
              "op_currency_name", "op_mc", "op_ci", "op_ft",
```

```

        "op_capacity", "dp_name", "dp_tp_priority",
        "dp_priority", "group_name", "op_group_priority",
        "profit", "client_id"],
    "message": "OK",
    "tp_data": ["H_to", 500, "Pakistan Mobile Mobilink", 20, 8,
        true, "9230", "10.3500", 1, 1, 0, 1, 6, 0],
    "data": [[true, "0.00000", "10.0500", 7, 13, 3, 4, true,
        "H_in", "9230", "Pakistan Mobile Mobilink",
        "10.4000", "10.4000", "USD", 1, 1, 0, 5, "H_p", 0, 0,
        "h_p", 0, "14.29", 5]]}

```

Request structure and fields description:

Request:

```

{
    "session_id": String(),
    "data": {
        'session_id': String(),
        'number': String(regex=B_NUMBER_REGEXP),
        'time_period': String() ,
        'tp_id': Int(gt=0),
        'time_zone': String(regex=TZ_REGEXP),
        'show_disabled_pt': Bool(),
        'show_stat': Bool(),
        'fields': List(String()),
        'currency_id': Int(gt=0),
        'contexts': List(Int()),
        'limit': Int(gt=0),
        'offset': Int(gte=0),
    }
}

```

Name	Default value	Optional	Type	Description
'session_id'	session_info['id']	True	String()	user session id, need for checking user permissions
'number'	N/A	False	String(regex=B_NUMBER_REGEXP)	Called number
'time_period'	N/A	False	String()	request time in format yyyy-MM-dd"THH:mm:ss
'tp_id'	N/A	False	Int(gt=0)	id of chosen TP
'time_zone'	'+00:00'	True	String(regex=TZ_REGEXP)	time zone for 'time_period'
'show_disabl	False	True	Bool()	boolean parameter, if 'True' then routes

ed_pt'				from disabled OPs will be shown also
'show_stat'	False	True	Bool()	if 'True' then ACD,ASR for last 4 hours will returns
'fields'	default_fields	True	List(String())	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
'currency_id'	session_info['currency_id']	True	Int(gt=0)	currency in which prices will display
'contexts'	N/A	False	List(Int())	ids context, in which request will be processed
'limit'	N/A	True	Int(gt=0)	says how many rows will be in response
'offset'	N/A	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'tp_name'	String
'tp_capacity'	Int
'tp_country'	String
'tp_price_id'	Int
'tp_id'	Int

'tp_enabled'	boolean
'tp_destination'	String
'tp_ocn'	String
'tp_lata'	String
'tp_price_interstate'	numeric(16,5)
'tp_price_intrastate'	numeric(16,5)
'tp_price_indeterminate'	numeric(16,5)
'tp_mc'	Int
'tp_ci'	Int
'tp_ft'	Int
'tp_currency_id'	Int
'tp_cid'	Int
'tp_price_priority'	
'delta_price'	numeric(16,5)
'op_name'	String
'op_capacity'	Int
'op_country'	String
'op_price_id'	Int
'op_id'	Int
'op_enabled'	Boolean
'op_destination'	String
'op_ocn'	String

'op_lata'	String
'op_price_interstate'	numeric(16,5)
'op_price_intrastate'	numeric(16,5)
'op_price_indeterminate'	numeric(16,5)
'op_mc'	Int
'op_ci'	Int
'op_ft'	Int
'op_currency_id'	Int
'dif_price_interstate'	numeric(16,5)
'dif_price_intrastate'	numeric(16,5)
'dif_price_indeterminate'	numeric(16,5)
'profit'	numeric(15,2)
'dp_name'	String
'dp_capacity'	numeric(15,2)
'dp_tp_priority'	Int
'dp_priority'	Int
'group_name'	String
'op_group_priority'	Int
'price_priority'	Int
'group_id'	Int
'dialpeer_id'	Int

'client_id'	Int
'asr'	numeric or null
'acd'	Int or null
'delta_profit'	numeric(15,2)
'op_price_tarif_interstate'	numeric(16,5)
'op_price_tarif_intrastate'	numeric(16,5)
'op_price_tarif_indeterminat e'	numeric(16,5)
'tp_price_tarif_interstate'	numeric(16,5)
'tp_price_tarif_intrastate'	numeric(16,5)
'tp_price_tarif_indeterminate'	numeric(16,5)
'op_currency_sign'	String
'tp_currency_sign'	String
'op_currency_name'	String
'tp_currency_name'	String
'allowed_price'	Bool
'op_tariff_invisible'	Bool
'tp_tariff_invisible'	Bool
'op_charge_by'	String
'tp_charge_by'	String
'op_route_by'	String

'tp_route_by'	String
'delta_price_unit'	Int
'attempts'	Int

2.196 jurisdictional_price_status_get

Docstring:

Shows status of Jurisdictional price uploading.

Statuses:

- "Applied on RTP" means that price list was successfully uploaded and applied.
- "Uploaded" - price list just uploaded but have not been applied yet.

Example:

- Client Request

```
{
  "session_id": "04f1c7e2e5f0bd62dd25e7aacb03d284",
  "data": {
    "limit": 101,
    "end_date": "2013-11-19T23:59:59",
    "sort_by": [{"tariff_name": ""}],
    "offset": 0,
    "start_date": "2013-11-17T00:00:00",
    "fields": [
      "id", "name", "insert_by", "insert_time",
      "price_type", "status", "notification_email",
      "status_time", "status_name", "tariff_name",
      "tariff_id"
    ]
  }
}
```

- Server RESPONSE:

```
{
  "status": 200,
  "fields": [
    "id", "name", "insert_by", "insert_time",
    "price_type", "status", "notification_email",
    "status_time", "status_name", "tariff_name",
    "tariff_id"
  ],
  "message": "OK",
  "data": [
    [
      7783, "Loaded by qw", "qw",
      "2013-11-19T15:23:13.678156", "incremental", 7,
      "test@test.com", "2013-11-19T15:23:13.888922",
      "Applied on RTP", "peer_out tariff", 4864
    ],
    [
      7757, "Loaded by SS", "SS",
      "2013-11-18T15:35:35.693572", "incremental", 7,
      "ikex@speedflow.com", "2013-11-18T15:38:17.399470",
      "Applied on RTP", "new1.peer_out tariff", 4864
    ]
  ],
  "time_elapsed": 0.08435
}
```

Request structure and fields description:

```
{'data': {'end_date': 'String',
          'fields': ["Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'price_id': 'Integer',
          'sort_by': [{"Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'tariff_id': 'Integer',
          'user_id': 'Integer',
          'user_name': 'String'},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tariff_id	N/A	True	Integer()	filter by tariff id
fields	['status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time']	True	List([Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')]), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
start_date	N/A	True	String(allow_blank: True)	Filter by price insertion time, shows only prices which were uploaded after given time
end_date	N/A	True	String(allow_blank: True)	Filter by price insertion time, shows only prices which were

				uploaded earlier given time
sort_by	[{'id': 'desc'}]	True	List([Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
price_id	N/A	True	Integer()	filter by price id
user_name	N/A	True	String(allow_blank: True)	filter by name or first chars of user name who uploaded price
user_id	N/A	True	Integer()	filter by id of user who uploaded price

Default fields:

Name	Type & Comments
status	Int
tariff_name	String
currency_name	String(3) - name_short
notification_email	String
id	Int
tariff_id	Int

insert_by	String
name	String
status_name	String
price_type	String
users_id	Int
insert_time	String
status_time	String

2.197 jurisdictional_price_upload

Docstring:

This function adds given Jurisdictional prices to the uploading queue. It allows to upload new prices only to the one tariff.

There are several possible prices formats which function accepts. You can get them with help of function 'prices_format_get'.

Function process request in the following order:

- 1) Sets direction ("1" for "in", 2 - for "out") and checks prices validity.
- 2) Uploads new prices.

Function will raise error with status 305 for single price uploading if 'start_date' in price less than current. Despite the error price will be uploaded but 'start_date' will be changed to the current date.

Possible fields names are mentioned in the default_fields.

Example with destination:

-Client Request

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
"data":{"price_type":"incremental",
"direction":"out",
"data":[[4700,
true,
true,
""],
```

```

        "3912",
        "",
        "",
        0.01,
        0.015,
        0.02345,
        0.0,
        1,
        1,
        0,
        0,
        "2013-10-29T12:34:37",
        "2030-01-01T23:59:59",
        "00:00:00",
        "23:59:59",
        "01111111"]],
    "fields":["tarif_id",
        "enabled",
        "allowed",
        "description",
        "destination",
        "ocn",
        "lata",
        "price_interstate",
        "price_intrastate",
        "price_indeterminate",
        "connection_price",
        "mc",
        "ci",
        "ft",
        "priority",
        "date_start",
        "date_end",
        "time_start",
        "time_end",
        "week_day"]}]
    }

```

Request structure and fields description:

Request:

```

{
    "session_id": String(),
    "data": {
        "tariff_id": Int(),
        "session_id": String(),
        "data": List(Any(), min_length=1),
        "fields": List(String()) ,
        "direction": String(regex="(in)|(out)"),
        "price_type": String(regex="(incremental)|(full)"),
        "notification_email": String(allow_blank=True),
        "operator": String(),
        "label": String(),

```

```

    "save_previous_price_priority": Bool(),
    "add_and_block_codes": Bool(),
    "replace_codes_on_future": Bool(),
    "flag_mc_ci_ft": Bool(),
}
}

```

Name	Default value	Optional	Type	Description
"tariff_id"	N/A	True	Int()	id of the target tariff
"session_id"	session_info["id"]	True	String()	current session id
"data"	N/A	False	List(Any(), min_length=1)	prices data values in the form of list
"fields"	N/A	False	List(String())	prices fields names in the same order as given prices values
"direction"	N/A	False	String(regex="(in) (out)")	"in" - for OP tariffs, "out" - for TP tariffs
"price_type"	"incremental"	True	String(regex="(incremental) (full)")	if type "full" all prices existed in the tariff will be closed and then uploaded new one. If "incremental" - new prices uploads over existed.
"notification_email"	session_info.get("user_email", "")	True	String(allow_blank=True)	email address to which will be send notification when price will be successfully uploaded. If doesn't specify in request - email will be sent to recipients addresses specified in price log template.
"operator"	session_info.get("login", "API")	True	String()	name of user who uploading price. By default - user login from current session
"label"	u"Loaded by {0}".format(unicode(session_info.get("login", "API")))	True	String()	label for price that will be recorded to database. Example: "Loaded by <Name of user who uploaded the price>". If label is absent in request - label will be set by defaults as "Loaded by <login of user from current

				session.>
"save_previous_price_priority"	False	True	Bool()	flag allows to preserve priority of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"add_and_block_codes"	False	True	Bool()	use in conjunction with 'type'='full'. If True than all currently existed prices which didn't present in uploading price list will be added to the price list with allowed=False. All other params will be preserved.
"replace_codes_on_future"	False	True	Bool()	if there are exists price for given code with older start date then in new one, then new will be uploaded over old one (it means that old price date_end becomes one second less then date_start in new one)
"flag_mc_ci_ft"	False	True	Bool()	flag allows to preserve tarification value of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.

Field "**data**" - is a list that contains prices data for different destination codes in the form of list.

```
"data": [
  [
    "tarif_id": Int(gt=0),
    "enabled": Bool(),
    "allowed": Bool(),
    "description": String(allow_blank=True),
    "destination": Destination(allow_blank=True),
    "ocn": String(allow_blank=True),
```

```

"lata": String(allow_blank=True),
"price_interstate": Float(gte=0),
"price_intrastate": Float(gte=0),
"price_indeterminate": Float(gte=0),
"connection_price": Float(gte=0),
"mc": Int(gte=0),
"ci": Int(gt=0),
"ft": Int(gte=0),
"priority": Int(gte=-100, lte=100),
"date_start": String(),
"date_end": String(),
"time_start": String(regex=TIME_CHECK_REGEX_PATTERN) ,
"time_end": String(regex=TIME_CHECK_REGEX_PATTERN) ,
"week_day": String(regex="(0|1){8}"),

```

], [...],[...], ...

]

Name	Default value	Optional	Type	Description
"tarif_id"	N/A	False	Int(gt=0)	id of the tariff to which price will be uploaded. Should be the same for all prices in price list.
"enabled"	True	True	Bool()	Defines price status. Disabled prices are not actual.
"allowed"	True	True	Bool()	"False" value makes destination disallowed.
"description"	N/A	True	String(allow_blank=True)	
"destination"	""	True	Destination(allow_blank=True)	Defines destination code for price
"ocn"	""	True	String(allow_blank=True)	operating company number
"lata"	""	True	String(allow_blank=True)	local access and transport area
"price_interstate"	0	True	Float(gte=0)	interstate price per minute
"price_intrastate"	0	True	Float(gte=0)	intrastate price per minute

te"				
"price_indeterminate"	0	True	Float(gte=0)	indeterminate price per minute
"connection_price"	0	True	Float(gte=0)	a connection fee
"mc"	1	True	Int(gte=0)	Minimal count interval. Call with duration less then this value - will have cost the same as call with duration specified in this field.
"ci"	1	True	Int(gt=0)	Tarification interval. For per second tarification put here "1" and "60" for per minute tarification
"ft"	0	True	Int(gte=0)	Time in seconds free of charge from beginning of call.
"priority"	0	True	Int(gte=-100, lte=100)	Price priority. You can specify this option only for TP tariff, i.e. if direction="out".
"date_start"	datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")	True	String()	Defines the start date and time for the price validity. It should not be less than current. Format: 'YYYY-MM-DDTHH:MM:SS', where T is delimiter between date and time. If price list is being uploaded with 'price_type' = 'full' all prices must have the same start date.
"date_end"	"2037-12-31 23:59:59+00"	True	String()	Defines the end date and time for the price validity. Format: 'YYYY-MM-DDTHH:MM:SS', where T is delimiter between date and time.
"time_start"	"00:00:00+00"	True	String(regex=TIM	Defines the start time for

			E_CHECK_REG EXP_PATTERN)	the price availability during the day time.
"time_end"	"23:59:59+00"	True	String(regex=TIME_CHECK_REG EXP_PATTERN)	Defines the end time for the price availability during the day time.
"week_day"	"01111111"	True	String(regex="(0 1){8}")	Bit mask, which defines days of week of price validity. First bit - not in use, second - Saturday, third - Friday, ...7th bit - Monday, 8th bit - Sunday. 1- means that price works in that day of week, 0 - not. Example - '01111111' - price works all days of week, '00111110' - price works every day excluding sunday and saturday.

Default fields:

Name	Type & Comments
"tarif_id"	
"enabled"	
"allowed"	
"description"	
"destination"	
"ocn"	
"lata"	

"price_interstate"	
"price_intrastate"	
"price_indeterminate"	
"connection_price"	
"mc"	
"ci"	
"ft"	
"priority"	
"date_start"	
"date_end"	
"time_start"	
"time_end"	
"week_day"	

2.198 jurisdictional_prices_list_get

Docstring:

Returns voice prices belonging to chosen OP tariff or TP tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Use only single of types - "current", "actual_prices" or "future", not all together.

Example:

- Client Request

```
{"session_id":"f28a3d96a0ecf3b1bbf44feed8b0c952",
  "data":{"limit":17,
```

```

"sort_by":[{"destination":""}, {"added":"desc"}],
"tariff_id":4201,
"offset":0,
"service_id":1,
"fields":["id","tariff_id","enabled","allowed",
"destination","price",
"connection_price","mc","ci","ft","priority",
"date_start","date_end","time_start","time_end",
"added","date_for_history","min_len","max_len",
"week_day","country"],"destination":""}
}

```

- Server Response:

```

{"status": 200,
"time_elapsed": 0.010708,
"fields": ["id", "tariff_id", "enabled", "allowed", "description",
"destination", "price", "connection_price", "mc", "ci",
"ft", "priority", "date_start", "date_end",
"time_start", "time_end", "added", "date_for_history",
"min_len", "max_len", "week_day", "country"],
"message": "OK",
"data": [[10129239, 4201, true, true, "", "1111111111", "0.00236",
"0.00000", 1, 1, 0, 0,
"2013-10-25T13:52:11.203884+00:00",
"2030-01-01T00:00:00+00:00", "00:00:00", "23:59:59",
"2013-10-25T13:52:11.161933+00:00",
"2030-01-01T00:00:00+00:00", 1, 24, "01111111", null],
[10129237, 4201, true, true,
"Record changed by system due to price overlap 10129239",
"1111111111", "0.00235", "0.00000", 1, 1, 0, 0,
"2013-10-25T13:43:26.997105+00:00",
"2013-10-25T13:52:10.203884+00:00", "00:00:00",
"23:59:59", "2013-10-25T13:43:26.942568+00:00",
"2030-01-01T00:00:00+00:00", 1, 24, "01111111", null],
[10129236, 4201, true, true,
"Record changed by system due to priceoverlap 10129237",
"1111111111", "0.00001", "0.00000", 1, 1, 0, 0,
"2013-10-25T13:40:15.196570+00:00",
"2013-10-25T13:43:25.997105+00:00", "00:00:00",
"23:59:59", "2013-10-25T13:40:15.158241+00:00",
"2030-01-01T00:00:00+00:00", 1, 24, "01111111", null]]
}

```

Request structure and fields description:

```

{'data': {'actual_prices': 'Bool',
'allowed': 'Bool',
'blocked_by_guardian': 'Bool',
'current': 'Bool',
'destination': 'String',
'enabled': 'Bool',
'end_date': 'String',
'fields': ["Enum('ci', 'date_start', 'date_end', 'time_end', 'week_day', 'all
owed', 'id', 'price_intrastate', 'actual_price_id', 'time_start', 'ocn', 'price_indeter
minate', 'destination', 'priority', 'lata', 'added', 'ft', 'description', 'price_inters
tate', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price')"],
'future': 'Bool',
'get_actual': 'Bool',
'id': 'Integer',

```

```

        'is_tp': 'Bool',
        'lata': 'String',
        'limit': 'Integer',
        'ocn': 'String',
        'offset': 'Integer',
        'pt_id': 'Integer',
        'sort_by': [{"Enum('ci', 'date_start', 'date_end', 'time_end', 'week_day', 'allowed', 'id', 'price_intrastate', 'actual_price_id', 'time_start', 'ocn', 'price_indeterminate', 'destination', 'priority', 'lata', 'added', 'ft', 'description', 'price_intrastate', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price')": "Enum('asc', 'desc', '')"}],
        'start_date': 'String',
        'strictly': 'Bool',
        'tariff_id': 'Integer'},
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
strictly	False	True	Bool	If "true" function searches strict match of given destination code in the database, else searches destinations code with subcodes.
lata	N/A	True	String(allow_blank: True, regex: "^%?[0-9]{0,5}")	
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
sort_by	[{'destination': 'asc'}, {'lata':	True	List([{'Enum('ci', 'date_start',	field on which result

	'asc'}, {'ocn': 'asc'}, {'added': 'desc'}]		'date_end', 'time_end', 'week_day', 'allowed', 'id', 'price_intrastate', 'actual_price_id', 'time_start', 'ocn', 'price_indeterminate', 'destination', 'priority', 'lata', 'added', 'ft', 'description', 'price_interstate', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price'): Enum('asc', 'desc', ")]], min_length: 0)	will be sorted. Result can be sorted ascending or descending.
actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current, date_start=
future	False	True	Bool	Used by Accucore. If "True" function searches only future prices for pt
ocn	N/A	True	String(allow_blank: True, regex: "^%?[0-9A-Fa-f]{0,4}")	
blocked_by_guardian	False	True	Bool	Used by Accucore. If "True" function searches only blocked destination codes for actual price.
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current, date_start=
start_date	N/A	True	String(allow_blank: False)	start date and time to show prices with start_date not less then given.
fields	['ci', 'date_start', 'date_end', 'time_end', 'week_day', 'allowed', 'id',	True	List([Enum('ci', 'date_start', 'date_end', 'time_end', 'week_day', 'allowed', 'id',	required fields. If absent - takes fields specified in "default

	'price_intrastate', 'actual_price_id', 'time_start', 'ocn', 'price_indeterminate', 'destination', 'priority', 'lata', 'added', 'ft', 'description', 'price_interstate', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price']		'price_intrastate', 'actual_price_id', 'time_start', 'ocn', 'price_indeterminate', 'destination', 'priority', 'lata', 'added', 'ft', 'description', 'price_interstate', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price')], min_length: 0)	fields". If empty - returns error.
tariff_id	N/A	True	Integer(gt: 0)	if "is_tp"=="true" then TP id, otherwise - OP id
pt_id	N/A	True	Integer(gt: 0)	if "is_tp"=="true" then TP id, otherwise - OP id
get_actual	N/A	True	Bool	Option should be used in conjunction with price id. If "get_actual" set to True, then function returns all actual prices for destination of the price with given id within price's tariff.
end_date	N/A	True	String(allow_blank: False)	end date and time to show prices with end_date not more then given.
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
allowed	N/A	True	Bool	if "true" returns allowed prices
enabled	N/A	True	Bool	if "true" returns enabled prices
id	N/A	True	Integer(gt: 0)	Price id in database. If "is_tp"=="true" then TP price id, else - OP price id

Default fields:

Name	Type & Comments
ci	Int
date_start	String
date_end	String
time_end	String
week_day	String
allowed	Boolean
id	Int
price_intrastate	numeric(16,5)
actual_price_id	Int
time_start	String
ocn	String
price_indeterminate	numeric(16,5)
destination	String
priority	Int
lata	String
added	String
ft	Int
description	String

price_interstate	numeric(16,5)
tariff_id	Int
mc	Int
country	String
enabled	Boolean
connection_price	numeric(16,5)

2.199 jurisdictional_tariff_add

Docstring:

Creates new OP or TP tariff.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "currency_id": 76,
    "name": "123456tar",
    "is_tp": true,
  }
}
```

Request structure and fields description:

```
{'data': {'currency_id': 'Integer',
          'invisible': 'Bool',
          'is_tp': 'Bool',
          'name': 'String',
          'session_id': 'String',
          'tariff_type_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
name	N/A	False	String(allow_blank : False)	name of the tariff. The name should unique.
session_id	N/A	True	String(allow_blank : False)	by default from user session
tariff_type_id	N/A	False	Integer(gt: 0, lt: 5)	defines tariff's type (1-destination, 2-OCN, 3-LATA, 4-LATA-OCN)
currency_id	N/A	False	Integer(gt: 0)	currency of the creating tariff.
invisible	False	True	Bool	invisible or not
is_tp	N/A	False	Bool	if "true" then function creates TP tariff, if "false" - creates OP tariff.

2.200 jurisdictional_tariff_set

Docstring:

Renames chosen tariff.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "id": 5006,
    "name": "1 tarif",
    "is_tp": true
  }
}
```

Request structure and fields description:

```
{'data': {'id': 'Integer',
          'invisible': 'Bool',
          'is_tp': 'Bool',
          'name': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	new name of the tariff
invisible	N/A	True	Bool	invisible or not
is_tp	N/A	False	Bool	if "true" then function searches tariff by id among TP tariffs
id	N/A	False	Integer(gt: 0)	OP tariff or TP tariff id. Generally OP tariff id is odd number, while TP tariff id is even number.

2.201 jurisdictional_tariffs_op_get

Docstring:

Calls API function `_tariffs_get` with option `'is_tp'=False`.

Example:

- Client Request

```
{"session_id": "0ab11223a707b59600993e99da72d167",  
  "data": {"currency_id": 1,  
           "sort_by": [{"name": ""}],  
           "fields": ["tariff_id", "name_currency"]}}
```

- Server Response:

```
{"status": 200,  
  "fields": ["tariff_id", "name_currency"],  
  "message": "OK",  
  "data": [[4233, "0001111 (USD)"],  
           [4149, "000_Zv_in (USD)"],  
           [4221, "011 (USD)"],  
           [4485, "1 (USD)"]],  
  "time_elapsed": 0.010708  
}
```

`_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of OP or TP tariffs existent in the database.

Request structure and fields description:

_tariffs_get

```
{'data': {'client_or_pt_name': 'String',
          'currency_id': 'Integer',
          'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')"],
          'is_tp': 'Bool',
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'point_name': 'String',
          'show_invisible': 'Bool or Null',
          'show_pts': 'Bool',
          'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer',
          'tariff_type_id': 'Integer'},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null
client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or point.
sort_by	[]	True	List([{'Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter',	field on which result will be sorted. Result can be sorted ascending or

			'prices_count', 'client_info', 'tariff_type_id'): Enum('asc', 'desc', ")}], min_length: 0)	descending.
tariff_type_id	N/A	True	Integer(gt: 0, lt: 5)	filter by tariff's type (1-destination, 2-OCN, 3-LATA, 4-LATA-OCN)
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
show_pts	True	True	Bool	not in use
is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the request.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

`_tariffs_get`

Name	Type & Comments
<code>dstcode_count</code>	Int - how many actual prices for different destination codes in the tariff
<code>name_currency</code>	String - tariff name with currency (example - "test_tariff(USD)")
<code>currency_name</code>	String
<code>invisible</code>	Boolean
<code>point_info</code>	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
<code>currency_id</code>	Int
<code>tariff_id</code>	Int
<code>name</code>	String
<code>counter</code>	Int - how many points use this tariff
<code>prices_count</code>	Int - how many actual prices in the tariff
<code>client_info</code>	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id
<code>tariff_type_id</code>	Int

2.202 jurisdictional_tariffs_tp_get

Docstring:

Calls API function `_tariffs_get` with option `'is_tp'=True`.

Example:

- Client Request

```
{
  "session_id": "90b28b952c569c96a967f0501138965e",
  "data": {
    "limit": 31,
    "sort_by": [{"name": ""}],
    "offset": 0,
    "fields": [
      "tariff_id", "name", "currency_id",
      "dstcode_count", "prices_count",
      "name_currency", "client_point_info"
    ]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": [
    "tariff_id", "name", "currency_id", "dstcode_count", "prices_count",
    "name_currency", "point_info", "client_info"
  ],
  "message": "OK",
  "data": [
    [4932, "000999", 7, 65501, 65501, "000999 (UAH)", [[4918, "000999TP1"], [5076, "1234567"]], [[564, "12Ls"], [609, "jur1001_"]]]
  ],
  "time_elapsed": 0.012435
}
```

`_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of OP or TP tariffs existent in the database.

Request structure and fields description:

`_tariffs_get`

```
{
  'data': {
    'client_or_pt_name': 'String',
    'currency_id': 'Integer',
    'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')"],
    'is_tp': 'Bool',
    'limit': 'Integer',
    'name': 'String',
    'offset': 'Integer',
    'point_name': 'String',
    'show_invisible': 'Bool or Null',
    'show_pts': 'Bool',
    'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')": "Enum('asc', 'desc', '')"}],
    'tariff_id': 'Integer',
    'tariff_type_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null
client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or point.
sort_by	[]	True	List([{Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id'): Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
tariff_type_id	N/A	True	Integer(gt: 0, lt: 5)	filter by tariff's type (1-destination, 2-OCN, 3-LATA, 4-LATA-OCN)
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
show_pts	True	True	Bool	not in use

is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the request.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

_tariffs_get

Name	Type & Comments
dstcode_count	Int - how many actual prices for different destination codes in the tariff
name_currency	String - tariff name with currency (example - "test_tariff(USD)")
currency_name	String
invisible	Boolean
point_info	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
currency_id	Int

tariff_id	Int
name	String
counter	Int - how many points use this tariff
prices_count	Int - how many actual prices in the tariff
client_info	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id
tariff_type_id	Int

2.203 kill_smpp_session

Docstring:

Kill SMPP session.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "is_inbound": true,
    "smpp_session_id": "82be5e45-6b9e-47be-8d5b-6174886c7552"
  }
}
```

- Server Response

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.204 lib_disconnect_cause_get

Docstring:

Returns description of SIP and(or) H323 disconnect codes existent in the database.

Example:

- Client Request

```
{"session_id": "db59903f45324059295712e6d6306ae4",
  "data": {"sort_by": [{"disconnect_cause": ""}],
          "fields": ["disconnect_cause", "description", "protocol"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["disconnect_cause", "description", "protocol"],
  "message": "OK",
  "data": [[0, "Normal call cleared (CANCEL or BYE received)", "sip"],
           [1, "Unallocated / unassigned number", "h323"],
           [2, "No route to specified transit network", "h323"],
           [3, "No route to destination", "h323"],
           [4, "Send special information tone", "h323"], ...],
  "time_elapsed": 0.41132}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('disconnect_cause', 'protocol', 'description')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'protocol': 'String',
          'sort_by': [{"Enum('disconnect_cause', 'protocol', 'description')": "Enum('asc', 'desc', '')"}]},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optiona	Type	Description
------	---------------	---------	------	-------------

		1		
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['disconnect_cause', 'protocol', 'description']	True	List([Enum('disconnect_cause', 'protocol', 'description')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
protocol	N/A	True	String(allow_blank: False, regex: "(sip) (h323)")	protocol name:"sip" or "h323". If absent in request returns cause codes for both protocols.
sort_by	[]	True	List([Enum('disconnect_cause', 'protocol', 'description'): Enum('asc', 'desc', ")]), min_length: 0)	defines field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause":"desc"}].

Default fields:

Name	Type & Comments
disconnect_cause	Int
protocol	String
description	String

2.205 lib_local_disconnect_cause_get

Docstring:

Returns local disconnect cause codes of mediacore and their description.

Example:

- Client Request

```
{"session_id":"b7ea9fbc9d2c50d7fcdb26a000156c89",
```

```

    "data":{"sort_by":[{"disconnect_cause":""}],
            "fields":["disconnect_cause","description"]}
}

```

- Server Response

```

{"status": 200,
 "fields": ["disconnect_cause", "description"],
 "message": "OK",
 "data": [[1000, "Point is not authorized by IP or Prefix"],
          [1001, "OP Lines capacity is exceeded"],
          [1002, "OP is Disabled"],
          [1003, "OP Tariff is not assigned"],
          [1004, "OP price is not found"], ... ],
 "time_elapsed": 0.43132}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('disconnect_cause', 'description')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('disconnect_cause', 'description')": "Enum('asc', 'desc',
''")}]},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([Enum('disconnect_cause', 'description'): Enum('asc', 'desc', "")]), min_length: 0)	field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['disconnect_cause', 'description']	True	List([Enum('disconnect_cause', 'description')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
disconnect_cause	Int
description	String

2.206 lib_local_disconnect_class_get

Docstring:

Returns classes of local disconnect cause codes with description.

"class_codes" contains default softswitch's accordance table for Local, SIP and H323 disconnect cause codes.

Example:

- Client Request

```
{
  "session_id": "c3275cff218c51429eae22a96a49e2d6",
  "data": {
    "sort_by": [{"disconnect_class": ""}],
    "fields": ["disconnect_class", "description", "class_codes"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["disconnect_class", "description", "class_codes"],
  "message": "OK",
  "data": [
    [0, 'System Disconnect Causes', [
      [1000, 'Point is not authorized by IP or Prefix.', 401, 34],
      [1001, 'OP Lines capacity is exceeded.', 503, 34],
      [1002, 'OP is Disabled.', 503, 34],
      [1003, 'OP Tariff is not assigned.', 404, 3],
      [1004, 'OP price is not found.', 503, 34],
      [1005, 'OP price is not allowed.', 503, 34],
      [1006, 'No TP found.', 503, 34],
      [1007, 'Insufficient volume of data received. Example \u2013 there is no IP address specified at TP.', 503, 34],
      [1008, 'RTP proxy is disabled, but it must be processed. Example \u2013 NAT call.', 404, 34],
      [1009, '', 503, 34],
      [1010, 'Called number exceed max digits limit after all manipulation. Due to internal limits.', 503, 34],
      [1011, 'A-number is blocked at origination point.', 503, 34],
      [1012, 'DST number contains an invalid number of digits.', 484, 34],
      [1013, 'Capability(codecs) missed or not supported in call initial message.', 415, 79],
      [1014, 'The Max-Forward parameter in the SIP message is less than 0 or equals zero.', 503, 34],
      [1015, 'No route (internal error).', 503, 34],
      [1016, 'Could not forward media stream, call rejected(only RTP proxy possible).', 415, 79],
      [1017, 'Could not start routing thread (internal error).', 503, 34],
    ]
  ]
}
```

```

[1018, 'Hunting is stopped (internal error).', 503, 34],
[1019, 'Error occurred during hunting process(internal error).', 503, 3
4],
[1020, 'Outgoing call did not pass authorization.', 503, 34],
[1021, 'Switch license is exceeded.', 503, 34],
[2001, 'Call is disconnected because point was blocked.', 480, 31],
[2002, 'Call is disconnected by administrator request.', 480, 31],
[2003, 'RTP timeout is detected.', 480, 31],
[2004, 'Call has been completed due to exceeding 'Max Call Time' limit
specified on OP/TP.', 480, 31]
]]
],
"time_elapsed": 0.023566}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('disconnect_class', 'class_codes', 'description')"],
            'limit': 'Integer',
            'offset': 'Integer',
            'sort_by': [{"Enum('disconnect_class', 'class_codes', 'description')": "Enum(
'asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['disconnect_class', 'class_codes', 'description']	True	List([Enum('disconnect_class', 'class_codes', 'description')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('disconnect_class', 'class_codes', 'description')": Enum('asc', 'desc', "")}], min_length: 0)	field name on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	False	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
disconnect_classes	String
class_codes	List(List(["local_disconnect_code"(Int), "description"(String), "sip_code"(Int), "h323_code"(Int)]))
description	String

2.207 login**Docstring:**

Creates new user session. If user have no access any point or banned error status will be returned.

If sessions limit enabled, other sessions of the given user will be terminated.

Example:

- Client Request

```
{"login": "test",  
  "password": "test",  
  "ip": "193.127.12.4"}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "time_elapsed": 0.05264,  
  "session_id": "8359eccf1ce015db729343d6f78186c0"}
```

Request structure and fields description:

```
{'description': 'String',  
  'ip': 'String',  
  'login': 'String',  
  'password': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
ip	N/A	False	String(allow_blank: False)	user IP address
login	N/A	False	String(allow_blank: False)	
password	N/A	False	String(allow_blank: False)	
description	""	True	String(allow_blank: True)	

2.208 logout

Docstring:

Releases given session.

Example:

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {}}  
}
```

Request structure and fields description:

```
{'data': {'forced_logout': 'Bool'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
forced_logout	N/A	True	Bool	flag which indicates forced logout to reflect it in the action log

2.209 maintenance_command

Docstring:

Executes given network diagnostic command (ping, mtr or traceroute) from given server (RTP sever or localhost) to the specified IP and returns command output.
Command have predefined options, for instance, "ping" sends 10 echo requests with interval 0,2 seconds and waits response up to 2 seconds after last echo request has been sent.

Example:

- Client Request

```
{"session_id": "7a1fe5eb3650e7f8908569c676fdec59",  
  "data": {"host": "194.165.82.80",  
           "command": "ping",  
           "fields": ["result"],  
           "ip": "8.8.8.8"}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["result"],  
  "message": "OK",  
  "result":  
    "PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
  
    64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=6 ttl=53 time=11.1 ms  
    64 bytes from 8.8.8.8: icmp_seq=7 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=8 ttl=53 time=11.0 ms  
    64 bytes from 8.8.8.8: icmp_seq=9 ttl=53 time=11.2 ms  
    64 bytes from 8.8.8.8: icmp_seq=10 ttl=53 time=11.1 ms  
  
    --- 8.8.8.8 ping statistics ---  
    10 packets transmitted, 10 received, 0% packet loss, time 1803ms  
    rtt min/avg/max/mdev = 11.099/11.207/11.246/0.039 ms  
    ",  
    "time_elapsed": 2.043441}
```

Request structure and fields description:

```
{'data': {'command': 'String',  
          'fields': ['String'],  
          'host': 'String',  
          'ip': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
ip	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12\d 3[0-2]))?(0\.){3}0(0)?)\$")	IP address to which need make command
host	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12\d 3[0-2]))?(0\.){3}0(0)?\$ (localhost)")	Certain mediacore server IP address from which necessary make command or "localhost".
fields	['result']	True	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
command	N/A	False	String(allow_blank: False, regex: "(ping) (mtr) (traceroute)")	command name: "ping", "mtr" or "traceroute".

Default fields:

Name	Type & Comments
result	String

2.210 manager_get

Docstring:

Returns list of users which assigned to the clients/companies as managers.

Example:

- Client Request

```
{"session_id": "fbb6a26407cad8f37fcb54ed7d94f049",  
  "data": {"sort_by": [{"name": ""}],  
          "fields": ["id", "name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[7, "test_user"], [6, "second_user"]],  
  "time_elapsed": 0.004521}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'id': 'Integer',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'owner_id': 'Integer',  
          'service_id': 'Integer',  
          'show_banned': 'Bool',  
          'show_invisible': 'Bool',  
          'show_invisible_client': 'Bool',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
show_banned	True	True	Bool	allows to get banned users in result, banned users will be returned by defaults
service_id	N/A	True	Integer(gt: 0, lt: 3)	id of service to show only managers which have SMS service (id=2) or Voice service (id=1) points assigned to them. If absent -

				will show both.
name	N/A	True	String(allow_blank: True)	applies 'like' search by user's name (percent symbol has special meaning - it is equal to any symbol(s))
show_invisible_client	False	True	Bool	allows to get managers which locked to invisible client(s). By default such users are hidden.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
id	N/A	True	Integer(gt: 0)	user id
limit	0	True	Integer(gte: 0)	rows quantity per one request
sort_by	[]	True	List([{Enum('id', 'name'): Enum('asc', 'desc', "")}], min_length: 0)	sorting by field(s) and ordering
owner_id	N/A	True	Integer(gt: 0)	filter by owner
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
show_invisible	False	True	Bool	allows to get invisible users, invisible users don't get by defaults

Default fields:

Name	Type & Comments
id	Int
name	String

2.211 mcc_mnc_add

Docstring:

Add a single MCC-MNC record to the database
(or updates existing one).

Example:

-Client Request

```
{"session_id": "ac2d9bc02a820409db8f0de6e9c8c812",  
  "data": {  
    "destination": "9377",  
    "name": "Afghanistan mobile Areeba",  
    "mcc": "412",  
    "mnc": "40"  
  }  
}
```

Request structure and fields description:

2.212 mcc_mnc_bulk_add

Docstring:

Adds a batch of new MCC-MNC records to the database (or updates existing ones)

Use this method when you need to add a large amount of records at once.

If the same destination code present at more than once - function will return error.

If "replace_all_mcc_mnc" and "renew_all_mcc_mnc" are False - then function only adds new codes.

Only "replace_all_mcc_mnc" or "renew_all_mcc_mnc" should be True.

Example:

-Client Request - add only new codes

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {  
    "mcc_mnc_data":  
      [[ "9377", "Afghanistan mobile Areeba", "412", "40"],  
        [ "9370", "Afghanistan mobile Awcc", "412", "1"] ],  
    "replace_all_mcc_mnc": false,  
    "renew_all_mcc_mnc": false  
  }  
}
```

Request structure and fields description:

```
{'data': {'mcc_mnc_data': [['String', 'String', 'String', 'String or Null']],
          'renew_all_mcc_mnc': 'Bool',
          'replace_all_mcc_mnc': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
mcc_mnc_data	N/A	False	List([List([Destination(allow_blank: False, _dest_pattern: ^\d*\$, min_length: 1, max_length: 24), String(allow_blank: False, min_length: 1, max_length: 100), String(allow_blank: False, regex: "^\d{1,3})?\$"), Or(String(allow_blank: False, regex: "^\d{1,3})?\$"), Null)], 4)], min_length: 0)	List of country codes and corresponding country names
replace_all_mcc_mnc	N/A	False	Bool	if True - deletes all existed entries from the database and uploads does not existent ones from the "country_codes_data"
renew_all_mcc_mnc	N/A	False	Bool	if True updates existed codes and adds new from the "country_codes_data"

2.213 mcc_mnc_del

Docstring:

Deletes specified MCC-MNC destination code record from database.

Example:

-Client Request

```
{"session_id": "ac2d9bc02a820409db8f0de6e9c8c812",
 "data": {"destinations": ["9377"]}}
```

Request structure and fields description:

```
{'data': {'destinations': ['String']}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
destinations	N/A	False	List([String(allow_blank: False, regex: "^[0-9#\\]**\$"), min_length: 0)	destination codes list

2.214 mcc_mnc_get

Docstring:

Function returns codes MCC-MNC by name, destination or exact MCC[MNC]

Square brackets means that MNC is optional.

Example:

- Client Request

```
{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "limit": 101,
    "sort_by": [{"code": "asc"}],
    "offset": 0,
    "code_or_name": "Af",
    "fields": ["code", "name", "mcc", "mnc"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["code", "name", "mcc", "mnc"],
  "message": "OK",
  "data": [
    ["93", "Afghanistan", "412", ""], ["9370", "Afghanistan mobile Awcc", "412", "1"],
    ["9372", "Afghanistan mobile Roshan", "412", "20"], ["9377", "Afghanistan mobile Areeba", "412", "40"],
    ["9378", "Afghanistan mobile Etisalat", "412", "50"], ["9379", "Afghanistan mobile Roshan", "412", "20"]
  ],
}
```



```

    "time_elapsed": 0.000888
}

```

Request structure and fields description:

```

{'data': {'code_or_name': 'String',
          'fields': ["Enum('mnc', 'code', 'mcc', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'nc_strictly': 'Bool',
          'network_code': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('mnc', 'code', 'mcc', 'name')": "Enum('asc', 'desc', '')"}]
},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
network_code	N/A	True	Destination(allow_blank: False, _dest_pattern: ^\d*\$/, min_length: 3, max_length: 6)	filter by network_code(mcc + mnc). Default behaviour: if MCC only provided (only 3 digits provided) all codes with given MCC will be returned. In case "network_code" may be divided to the MCC and MNC (more then 3 digits provided) then searches by exact MCC and MNC compliance.
fields	['mnc', 'code', 'mcc', 'name']	True	List([Enum('mnc', 'code', 'mcc', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
code_or_name	N/A	True	String(allow_blank: False)	filter by destination code or country name
sort_by	[]	True	List([{"Enum('mnc', 'code', 'mcc', 'name')": Enum('asc', 'desc', "")}], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
name	N/A	True	String(allow_blank: False)	filter by country name

limit	0	True	Integer(gte: 0)	says how many rows will be in response
nc_strictly	False	True	Bool	this option may be used only in conjunction with "network_code" filter. If option enabled and "network_code" length = 3 (MCC only provided) then MNC will be treated as 'null'.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
mnc	String(3)
code	String(24)
mcc	String(3)
name	String(64)

2.215 mcc_mnc_get_grouped

Docstring:

Returns list of existed unique network codes (concatenated mcc and mnc) using the filter by network code's first digits.

Example:

```
-Client Request
{
  "session_id": "78f18d2cb3693bf4bdefa529322de2aa",
  "data": {
    "network_code": '41',
    "fields": ['network_code'],
    "limit": 2
  }
}

- Server Response
{
  "status": 200,
```

```

    "fields": ["network_code"],
    "message": "OK",
    "data": [
        ['412'], ['4121']
    ],
    "time_elapsed": 0.46834
}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('network_code', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'network_code': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('network_code', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
network_code	N/A	True	Destination(allow_blank: False, _dest_pattern: ^\d*\$/, max_length: 6)	network code(mcc + mnc) or first digits of network code.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
name	N/A	True	String(allow_blank: False)	filter by mcc mnc name
fields	['network_code', 'name']	True	List([Enum('network_code', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'network_code': ""}]	True	List([{'Enum('network_code', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"country": "asc"}] - sorts result ascending by field "country"
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
network_code	String (mcc + mnc)
name	String

2.216 monitoring_options_get

Docstring:

Returns report about monitoring options in list of lists.

Example:

Example:

- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {}
}
```

- Server Response

```
{
  "status": 200,
  "data": [
    ["Swap1", "Swap balance by day", 0.0, -0.1],
    ["Swap1", "Minutes IN by day", 2, 1]
  ]
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.217 month_sms_count_get

Docstring:

Returns total SMS count for month on softswitch.

Request structure and fields description:

```
{'date': 'String', 'fields': ["Enum('web_value',)"]}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['web_value']	True	List([Enum('web_value',)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
date	2021-01-28	True	String(allow_blank: False)	

Default fields:

Name	Type & Comments
web_value	String

2.218 notification_email_add

Docstring:

Allows to send email notification manually to the given technical and(or) billi

ng email
addresses of chosen clients.

Example:

```
{ "session_id": "2200ae08bdd857adb0a3fd3b0ef42bf1",  
  "data": {  
    "body": "1",  
    "clients_ids": ["681"],  
    "subject": "1"  
  }  
}
```

Request structure and fields description:

```
{ 'data': { 'body': 'String',  
            'clients_ids': ['Integer'],  
            'send_to_billing_email': 'Bool',  
            'send_to_tech_email': 'Bool',  
            'subject': 'String'},  
  'session_id': 'String' }
```

data validation description:

Name	Default value	Optional	Type	Description
subject	N/A	False	String(allow_blank: False)	email's subject
body	N/A	False	String(allow_blank: False)	email's body
send_to_tech_email	False	True	Bool	send message to the client's tech email
send_to_billing_email	True	True	Bool	send message to the client's billing email
clients_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	if clients ids specified - sends notification to billing email addresses of listed companies

2.219 numbers_by_point_get

Docstring:

Function searches destination codes among prices of chosen point tariff which begin with given digits.

Then it searches free subcode for each code from obtained list and supplements it with zeros to the "min_len" specified in price.

Function returns list that contains two another lists:

- first list contains original codes from tariff
- second list contains corresponding to them autocompleted codes(in the same order)
- third list contains original countries codes(in the same order)

Example:

- Client Request

```
{"session_id": "bdf0fec748a6f7a0473389baf4fcb51b",  
  "data": {"limit": 100,  
           "pt_id": 4331,  
           "destination": "38",  
           "fields": ["numbers", "origin_country"]}  
}
```

- Server Response:

```
{"status": 200,  
  "fields": ["numbers"],  
  "message": "OK",  
  "data": [[ "380", "3800", "3805", "38050", "38051", "38063"],  
           [ "380100000000", "3800000000000", "3805200000000",  
             "3805000000000", "3805100000000", "3806300000000"],  
           [ "21", "232", "4", "6", "211", "9"]],  
  "time_elapsed": 0.008991  
}
```

Request structure and fields description:

```
{'data': {'destination': 'String',  
          'fields': ["Enum('origin_country', 'numbers')"],  
          'limit': 'Integer',  
          'pt_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
limit	N/A	True	Integer(gt: 0)	limit of results amount
pt_id	N/A	False	Integer(gt: 0)	Termination or Origination point id

destination	N/A	True	String(allow_blank: False)	first digits of destination code
fields	['origin_country', 'numbers']	True	List([Enum('origin_country', 'numbers')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

2.220 op_add_or_update

Docstring:

Creates new disabled origination point if "id" absent in request, otherwise updates parameters of given point:

- 1) If "contexts" present in request updates point data in the given context(s). If point status was changed to "disabled" - creates new email notification.
 - 2) if "contexts" not present in request function can change point tariff, status, "invisible" and "allow_internal" flags.
- If flag invisible was set as "true" then function disables given OP and removes all point relations with groups.

`_op_add` ([data_validation](#)) ([default_fields](#))

Creates new disabled origination point, adds new point to schema routing and each context.

Allows to specify balance, tariff, description, flags "invisible" and "allow_internal" for creating OP.

Function returns new Origination Point id.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "tariff_id": 29,
    "invisible": false,
    "description": "",
    "name": "my_op",
    "balance_id": 1,
    "allow_internal": false
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 25,
  "time_elapsed": 0.258111
}
```


`_op_update` ([data_validation](#)) ([default_fields](#))

Changes common parameters of given origination point: name, description, tariff, invisible and allow_internal flags in the schema routing. If flag "invisible" was set to "true" - function disables point and removes all relations of the OP with groups.

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"tarif_id":29,
        "id":25,
        "invisible":false,
        "description":"",
        "name":"my_op1",
        "allow_internal":false}
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.188672}
```

`_op_update_within_contexts` ([data_validation](#)) ([default_fields](#))

Updates origination point parameters within given context(s). Function creates notification email if point status was changed manually.

Function raises error if API user is trying to change invisible point.

Example:

- Client Request

```
{"session_id":"74c1cf47382eb1a7ffad856372e7242c",
  "data":{
    {
      "id":397,
      "contexts":[{"ani_block":"",
                  "max_call_time":3660,
                  "enabled":true,
                  "context_id":1,
                  "rtp_timeout":10,
                  "proxy_mode":1,
                  "ani_regexp":"",
                  "routes_id":1,
                  "capacity":500,
                  "dnis_regexp":""}
    ]
  }
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.061522}
```

Request structure and fields description:

_op_add

```
{'data': {'allow_internal': 'Bool',
          'balance_id': 'Integer',
          'charge_by': 'Integer',
          'client_id': 'Integer',
          'description': 'String',
          'ext_id': 'Integer',
          'invisible': 'Bool',
          'jurisdict_tariff_id': 'Integer',
          'lnp_lookup': 'Integer',
          'name': 'Name(at least one printable character). max_length=None',
          'north_american_routing': 'Bool',
          'point_manager_id': 'Integer or Null',
          'route_by': 'Integer',
          'rtp_timeout': 'Integer or Integer',
          'tarif_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	True	Integer(gt: 0)	client with given id should be visible
charge_by	N/A	True	Integer(gte: 1)	
name	N/A	False	Name(at least one printable character). max_length=None	point name should unique
description	""	True	String(allow_blank: True)	
route_by	N/A	True	Integer(gte: 1)	
tarif_id	0	True	Integer(gte: 0)	tariff currency has to be equal currency of chosen balance
lnp_lookup	N/A	True	Integer(gte: 1)	

balance_id	N/A	True	Integer(gt: 0)	client which owns given balance should be visible.
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	OP manager user id for update notifications
ext_id	0	True	Integer(gte: 0)	OP id for external billing
invisible	False	True	Bool	boolean option. If True - function makes point invisible and disabled, removes point relations with dialpeers and groups.
allow_internal	False	True	Bool	boolean option. If "true" - allows calls to TPs which belongs same company as given OP
rtp_timeout	10	True	Or(Integer(gte: 5, lte: 60), Integer(gte: 0, lte: 0))	rtp timeout in seconds
north_american_routing	N/A	True	Bool	
jurisdict_tariff_id	N/A	True	Integer(gte: 0)	

_op_update

```
{'data': {'allow_internal': 'Bool',
          'charge_by': 'Integer',
          'description': 'String',
          'id': 'Integer',
          'invisible': 'Bool',
          'jurisdict_tariff_id': 'Integer',
          'lnp_lookup': 'Integer',
          'name': 'Name(at least one printable character). max_length=None',
          'north_american_routing': 'Bool',
          'point_manager_id': 'Integer or Null',
          'route_by': 'Integer',
          'service_id': 'Integer',
          'tariff_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default	Optional	Type	Description
------	---------	----------	------	-------------

	value	l		
id	N/A	False	Integer(gt: 0)	point id
name	N/A	True	Name(at least one printable character). max_length=None	new point name, should be unique in the system.
description	N/A	True	String(allow_blank: True)	
tarif_id	N/A	True	Integer(gt: 0)	new point tariff. The currency of tariff have to be the same as the currency of balance.
invisible	N/A	True	Bool	if this flag set to "true" - function disables OP and removes all relations of point with groups.
allow_internal	N/A	True	Bool	boolean option. If "true" - allows calls to TPs which belongs same company as given OP
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)
north_american_routing	N/A	True	Bool	
jurisdiction_tariff_id	N/A	True	Integer(gte: 0)	
charge_by	N/A	True	Integer(gte: 1)	
route_by	N/A	True	Integer(gte: 1)	
lnp_lookup	N/A	True	Integer(gte: 1)	
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	OP manager user id for update notifications. If key value is "NULL" manager user will be cleared. If key not present in function request data user id will not be updated.

_op_update_within_contexts

```
{'data': {'contexts': [{'ani_block': 'String',
                        'ani_regexp': 'String',
                        'capacity': 'Integer',
                        'charge_by': 'Integer',
                        'context_id': 'Integer',
```

```

'dnis_regex': 'String',
'enabled': 'Bool',
'jurisdict_tariff_id': 'Integer',
'lnp_lookup': 'Integer',
'max_call_time': 'Integer',
'north_american_routing': 'Bool',
'proxy_mode': 'Integer',
'route_by': 'Integer',
'routes_id': 'Integer',
'rtp_timeout': 'Integer or Integer'}],
'id': 'Integer'},
'session_id': 'String'}

```

data.contexts validation description:

Name	Default value	Optional	Type	Description
ani_regex	N/A	True	String(allow_blank: True, max_length: 64)	A number regex
max_call_time	N/A	True	Integer(gte: 0, lte: 10800)	maximum call time allowed for the point in seconds
proxy_mode	N/A	True	Integer()	"1" - Full Proxy, "2" - No Proxy if Possible, "3" - No Proxy
north_american_routing	N/A	True	Bool	
lnp_lookup	N/A	True	Integer(gte: 1)	
enabled	N/A	True	Bool	point status. If True - point can participate in routing. If point status was changed creates new email notification.
routes_id	N/A	True	Integer(gt: 0)	Currently this option not in use. The option determines method of routing: "1" - LCR (routing without considering ASR and ACD. It is a standard routing algorithm of mediacore.), "2" - Quality (According to this kind of routing suppliers orders by destination priority, which counts by formula: $\text{priority} = \text{ASR} * \text{ACD} * 100\%$), "3" - Profit

				(According to this kind of routing suppliers orders by destination priority, which counts by formula: ASR*ACD*100%*delta_price).
dnis_regexp	N/A	True	String(allow_blank: True, max_length: 64)	B number regexp
capacity	N/A	True	Integer(gte: 0, lte: 1000)	defines the limit on the number of simultaneous calls that can pass through this point.
rtp_timeout	N/A	True	Or(Integer(gte: 5, lte: 60), Integer(gte: 0, lte: 0))	rtp timeout in seconds
jurisdict_tariff_id	N/A	True	Integer(gte: 0)	
route_by	N/A	True	Integer(gte: 1)	
ani_block	N/A	True	String(allow_blank: True, max_length: 255)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
context_id	N/A	False	Integer(gt: 0)	
charge_by	N/A	True	Integer(gte: 1)	

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([data.contexts], min_length: 0)	

id	N/A	False	Integer(gt: 0)	
----	-----	-------	----------------	--

2.221 op_add_simple

Docstring:

Creates OP in "Simple routing" mode with flag allow_internal = True and OP tarif with name [OP_name tarif].
Function raises error if user is trying assign point to invisible company.

Function returns id of created Origination Point.

Request structure and fields description:

```
{'data': {'allow_internal': 'Bool',
          'balance_id': 'Integer',
          'client_id': 'Integer',
          'description': 'String',
          'invisible': 'Bool',
          'name': 'String',
          'service_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	True	Integer(gt: 0)	balance id of company. Company should be visible.
name	N/A	False	String(allow_blank: False)	OP name, should be unique
description	""	True	String(allow_blank: True)	point description
invisible	False	True	Bool	boolean value, if True then point is invisible and disabled
allow_internal	False	True	Bool	boolean option. If "true" - allows calls to TPs which

				belongs same company as given OP
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)
client_id	N/A	True	Integer(gt: 0)	company owner identifier in table billing.client. Client should be visible.

2.222 op_call_path_analysis_get

Docstring:

Returns problem routes to for given Origination point at the given date and time.

'Problem route' means that for given destination(destination code from tariff of given OP) there is no route or TP price doesn't meet delta price criteria.

Example:

- Client Request

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "op_id": 3,
    "contexts": [1],
    "date": "2014-01-13T15:29:13",
    "fields": [
      "op_destination", "op_country", "op_price",
      "op_settings", "tp_name", "tp_destination",
      "tp_country", "tp_price", "tp_settings"
    ],
    "destination": "380"
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "op_destination", "op_country", "op_price", "op_settings",
    "tp_name", "tp_destination", "tp_country", "tp_price",
    "tp_settings"
  ],
  "message": "OK",
  "data": [
    [
      "380507", "Ukraine Mobile Mts", "0.50000", "1/1/0",
      "No route have been found", "", "", "0.00000", ""
    ],
    [
      "38040", "Ukraine", "0.06000", "1/1/0",
      "No route have been found", "", "", "0.00000", ""
    ],
    [
      "38050", "Ukraine Mobile Umc Mts", "0.06000", "1/1/0",
      "No route have been found", "", "", "0.00000", ""
    ]
  ],
  "time_elapsed": 2.072495
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'date': 'String',
    'destination': 'String',
  }
}
```



```

        'fields': ["Enum('tp_time_end', 'tp_week_day', 'tp_tarif_id', 'op_time_end',
'op_destination', 'tp_destination', 'tp_time_start', 'op_settings', 'tp_id', 'op_time_s
tart', 'tp_country', 'tp_name', 'tp_price', 'op_id', 'tp_settings', 'op_tarif_id', 'tp_
ci', 'op_week', 'op_country', 'op_ci', 'op_price')"],
        'limit': 'Integer',
        'offset': 'Integer',
        'op_id': 'Integer',
        'session_id': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optiona l	Type	Description
limit	N/A	True	Integer(gt: 0)	not in use
fields	['tp_time_end', 'tp_week_day', 'tp_tarif_id', 'op_time_end', 'op_destination', 'tp_destination', 'tp_time_start', 'op_settings', 'tp_id', 'op_time_start', 'tp_country', 'tp_name', 'tp_price', 'op_id', 'tp_settings', 'op_tarif_id', 'tp_ci', 'op_week', 'op_country', 'op_ci', 'op_price']	True	List([Enum('tp_time_end', 'tp_week_day', 'tp_tarif_id', 'op_time_end', 'op_destination', 'tp_destination', 'tp_time_start', 'op_settings', 'tp_id', 'op_time_start', 'tp_country', 'tp_name', 'tp_price', 'op_id', 'tp_settings', 'op_tarif_id', 'tp_ci', 'op_week', 'op_country', 'op_ci', 'op_price')], min_length: 0)	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
date	N/A	True	String(allow_blank: False)	By defaults current date and time
session_id	N/A	True	String(allow_blank: False)	User's session id, need for checking user permissions. By defaults - from current session.
destination	N/A	True	String(allow_blank: True, regex: "^(d+(, ,s)?)*\$")	destination for filtering
offset	N/A	True	Integer(gte: 0)	not in use
op_id	N/A	False	Integer(gt: 0)	id of chosen

				Origination Point
contexts	N/A	False	List([Integer()], min_length: 0)	ids context, in which request will be processed

Default fields:

Name	Type & Comments
tp_time_end	String
tp_week_day	String
tp_tarif_id	Int
op_time_end	String
op_destination	String
tp_destination	String
tp_time_start	String
op_settings	String
tp_id	Int
op_time_start	String
tp_country	String
tp_name	String
tp_price	numeric(16,5)
op_id	Int
tp_settings	String
op_tarif_id	Int

tp_ci	Int
op_week	String
op_country	String
op_ci	Int
op_price	numeric(16,5)

2.223 op_dest_capacity_add

Docstring:

Creates new traffic limit for given context, OP and destination.

Function raises error if context does not exist in DB or there is no OP with given id in context.

Returns new record id.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "context_id": 1,
    "op_id": 14,
    "dest": "312,213",
    "capacity": 123
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 22,
  "time_elapsed": 0.260494
}
```

Request structure and fields description:

2.224 op_dest_capacity_del

Docstring:

Delete op_dest_capacity record with given id in given context.

Example:

- Client Request

```
{
  "session_id":"63db616315b9e7d6515e9a24aa8f0e2c",
  "data":{
    "id":2,
    "context_id":1
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 2,
  "time_elapsed": 0.051203
}
```

Request structure and fields description:

2.225 op_dest_capacity_get

Docstring:

Returns data about op destination limits.

Example:

- Client Request

```
{
  "session_id":"63db616315b9e7d6515e9a24aa8f0e2c",
  "data":{
    "context_id":1,
    "id":32,
    "fields":
      ["id","op_id","capacity","dest"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id","op_id","capacity","dest"],
  "message": "OK",
  "data": [[32,14,999,"312,213"]],
  "time_elapsed": 0.03936
}
```

Request structure and fields description:

2.226 op_dest_capacity_set

Docstring:

Change op_dest_capacity record with given id.

Example:

- Client Request

```
{
  "session_id": "63db616315b9e7d6515e9a24aa8f0e2c",
  "data": {
    "id": 14,
    "capacity": 999
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.073294
}
```

Request structure and fields description:

2.227 op_group_set

Docstring:

Changes priority of relation between given OP and group.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fcla089d8c8f",
  "data": {
    "op_id": 3,
    "priority": 1,
    "group_id": 6,
    "contexts": [1]
  }
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.032511}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'group_id': 'Integer',  
          'op_id': 'Integer',  
          'priority': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	False	Integer(gt: 0)	
group_id	N/A	False	Integer(gt: 0)	
priority	0	True	Integer(gte: -100, lte: 100)	new relation priority
contexts	N/A	False	List([Integer()], min_length: 0)	contexts id where need change relations

2.228 op_ip_add

Docstring:

Add combination of IP address and prefix to chosen origination point and context(s)

with specified weight and priority.

The function disallows to add new entry in the following cases:

- 1) if in some OP in chosen context(s) the same IP with prefix which starts with the same digit as given one already exists,
- 2) if in some other OP in chosen context(s) the same IP without any prefix already exists,
- 3) if such IP with any prefix is already exist in the given point and context(s).

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "op_id": 3,
    "contexts": [
      {
        "enabled": true,
        "context_id": 1,
        "prefix": "123",
        "ip": "1.2.3.4"
      }
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.072589
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': [
      {
        'context_id': 'Integer',
        'enabled': 'Bool',
        'ext_id': 'Integer',
        'ip': 'String or ['String']',
        'prefix': 'String'
      }
    ],
    'op_id': 'Integer',
    'session_id': 'String'
  }
}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	
ip	N/A	False	Or(String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?\$"), List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?\$")), min_length: 0))	single IP address or subnet IP address or list of them.
ext_id	N/A	True	Integer(gte: 0)	not in use
enabled	False	True	Bool	entry status, if "true" then IP is enabled

prefix	""	True	String(allow_blank: True, regex: "^[\\d#+]*\$")	prefix may contain digits and "#" sign
--------	----	------	---	--

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	False	Integer(gt: 0)	
contexts	N/A	False	List([data.contexts], min_length: 0)	list of data for each context separately

2.229 op_ip_del

Docstring:

Removes combination of IP address and prefix from chosen OP and context.
Removes all entries from given OP, if IP address id specified in request.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "id": 23,
    "contexts": [1]
  }
}
```

- Server response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.06506
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'id': 'Integer',
    'op_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of contexts id where need to remove the entry
op_id	N/A	True	Integer(gt: 0)	if present deletes all entries from chosen OP
id	N/A	True	Integer(gt: 0)	entry id in database

2.230 op_ip_set

Docstring:

Changes parameters of the given IP address in given context.
The function disallows to change chosen entry in the following cases:
1) if in some OP in chosen context(s) the same IP with prefix which starts with the same digit as given one already exists,
2) if in some other OP in chosen context(s) the same IP without any prefix already exists,
3) if such IP with any prefix is already exist in the given point and context(s).

Function returns id of the ip address in database.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "id": 23,
    "contexts": [
      {
        "enabled": false,
        "context_id": 1,
        "prefix": "123",
        "ip": "1.2.3.4/32"
      }
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 23,
  "time_elapsed": 0.070987
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': [
      {
        'context_id': 'Integer',
        'enabled': 'Bool',
        'ext_id': 'Integer',
        'ip': 'String',
        'prefix': 'String'
      }
    ],
    'id': 'Integer'
  },
}
```

```
'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
enabled	N/A	True	Bool	new entry status, if "true" then IP is enabled
ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ((([1-9] [12]\d 3[0-2]))? (0\.){3}0(/0)?)\$")	single IP address or IP address + mask (example : 2.2.2.2/32).
context_id	N/A	False	Integer(gt: 0)	
prefix	N/A	True	String(allow_blank: True, regex: "^[d#+]*\$")	prefix may contain digits and "#" sign
ext_id	N/A	True	Integer(gte: 0)	not in use

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	IP address id in database
contexts	N/A	False	List([data.contexts], min_length: 0)	list of data for each context separately

2.231 op_ips_get

Docstring:

Returns IP addresses records of chosen OP and context(s) or records of

all OPs filtered by given IP address(es).
This function also allows to get list of OPs to which given IP address(es) assigned.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "limit": 101,
    "op_id": 3,
    "contexts": [1],
    "offset": 0,
    "fields": ["id", "op_id", "ip", "prefix", "enabled"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "op_id", "ip", "prefix", "enabled"],
  "message": "OK",
  "data": [[13, 3, "181.232.133.18/32", "1#", true]],
  "time_elapsed": 0.006448
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ['String'],
          'ip_list': ['String'],
          'limit': 'Integer',
          'offset': 'Integer',
          'op_id': 'Integer',
          'sort_by': [{"Enum('ip', 'enabled', 'prefix', 'op_id', 'context_id', 'id')":
"Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	True	Integer(gt: 0)	filter by origination point id
sort_by	[]	True	List([{"Enum('ip', 'enabled', 'prefix', 'op_id', 'context_id', 'id')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
ip_list	N/A	True	List([String(allow_blank: False, regex:	filter by ip addresses

			"^(([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.((\d [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.){2}(\d [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])/([1-9] [12]\d{3}[0-2]))?(0\.){3}0(/0)?)\$"), min_length: 0)	
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
contexts	[]	False	List([Integer()], min_length: 0)	list of contexts, where to seek given IP addresses or point. If absent - function searches in each context.
fields	['ip', 'enabled', 'prefix', 'op_id', 'context_id', 'id']	False	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
ip	String
enabled	Boolean
prefix	String(16)
op_id	Int
context_id	Int
id	Int

2.232 ops_free_group_get

Docstring:

Returns list of OPs which aren't connected with chosen group.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {"sort_by": [{"op_name": ""}],
           "group_id": 6,
           "contexts": [1],
           "fields": ["op_id", "op_name"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["op_id", "op_name"],
  "message": "OK",
  "data": [[13, "s_test_op"], [7, "H_P_in"], [9, "H_in_simple"],
           [5, "st.peer1"], [15, "OP-14"], [19, "pablo_op"]],
  "time_elapsed": 0.00085}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ["Enum('op_id', 'op_name')"],
          'group_id': 'Integer',
          'limit': 'Integer',
          'offset': 'Integer',
          'op_id': 'Integer',
          'op_name': 'String',
          'service_id': 'Integer',
          'sort_by': [{"Enum('op_id', 'op_name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
op_id	N/A	True	Integer()	Filter by OP
op_name	N/A	True	String(allow_blank:)	filter by OP name. Function will return only

			True)	points which names starts with given symbols
contexts	[]	False	List([Integer()], min_length: 0)	context(s) where search relations
group_id	N/A	True	Integer(gt: 0)	filter by group
sort_by	[]	True	List([Enum('op_id', 'op_name'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"op_name": "asc"}] - sorts result ascending by field "op_name"
fields	['op_id', 'op_name']	True	List([Enum('op_id', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
op_id	Int
op_name	String

2.233 ops_free_tp_get

Docstring:

Returns list of OPs which aren't connected with chosen OP in simple and advanced mode in the chosen context(s).

Example:

- Client Request

```
{"session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {"sort_by": [{"name": ""}],
           "name": ""},
```

```

    "contexts": [1],
    "tp_id": 8,
    "fields": ["id", "name"]}
}

```

- Server Response

```

{"status": 200,
 "fields": ["id", "name"],
 "message": "OK",
 "data": [[3, "Test_1_OP"], [21, "ss_OP"], [13, "a_test_op"],
          [23, "H_in"], [5, "peer1"], [27, "NewOP"], [25, "my_op"],
          [15, "OP-14"]],
 "time_elapsed": 0.00972}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'fields': ["Enum('id', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	False	Integer(gt: 0)	
name	N/A	True	String(allow_blank: True)	filter by OP name
contexts	[]	False	List([Integer()], min_length: 0)	contexts ids
fields	['id', 'name']	False	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'name': ""}]	True	List([{"Enum('id', 'name')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
--------	---	------	-----------------	---

Default fields:

Name	Type & Comments
id	Int
name	String - OP name

2.234 ops_from_groups_del

Docstring:

Removes relations between given OPs and groups within given context(s).

Function diallows to delete relations which was created in the Simple Routing mode.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fcla089d8c8f",
  "data": {"op_ids": [3],
           "contexts": [1],
           "groups_ids": [8]}
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.060283}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'groups_ids': ['Integer'],
          'op_ids': ['Integer']},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of OPs
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of group ids
contexts	N/A	False	List([Integer()], min_length: 0)	contexts id where need remove relation

2.235 ops_get

Docstring:

Returns list of origination points existing in database in given context.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "limit": 2,
    "sort_by": [{ "name": "" }],
    "invisible": false,
    "contexts": [1],
    "offset": 0,
    "fields": [
      "id", "name", "client_name", "client_id", "enabled",
      "capacity", "tariff_id", "tariff_name",
      "tariff_name_currency"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "id", "name", "client_name", "client_id", "enabled",
    "capacity", "tariff_id", "tariff_name",
    "tariff_name_currency"
  ],
  "message": "OK",
  "data": [
    [3, "Test_1_OP", "Test_client", 1, true, 500, 9, "test", "test (USD)"],
    [21, "ss_OP", "ss_Co", 15, true, 500, 39, "ss_1.1", "ss_1.1 (EUR)"]
  ],
  "time_elapsed": 0.010607
}
```

Request structure and fields description:

```
{'data': {'clients_ids': ['Integer'],
```

```

'contexts': ['Integer'],
'enabled': 'Bool',
'ext_id': 'Integer',
'fields': ["Enum('client_invisible', 'tariff_name', 'description', 'lnp_looku
p', 'allow_internal', 'currency_name_short', 'rtp_timeout', 'route_by', 'currency_id',
'balance_name', 'client_id', 'invisible', 'ani_regexp', 'jurisdict_tariff_name', 'id',
'jurisdict_tariff_name_currency', 'ext_id', 'tariff_id', 'capacity', 'name', 'point_man
ager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_nam
e', 'simple_pt', 'charge_by', 'tt_timeout', 'proxy_mode', 'dnis_regexp', 'tariff_name_c
urrency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing',
'max_call_time', 'routes_id')"],
'id': 'Integer',
'invisible': 'Bool or Null',
'limit': 'Integer',
'manager_id': 'Integer',
'name': 'String',
'name_or_client_name': 'String',
'offset': 'Integer',
'owner_id': 'Integer',
'points_type': "Enum('jurisdictional', 'non_jurisdictional', 'all')",
'service_id': 'Integer',
'sort_by': [{"Enum('client_invisible', 'tariff_name', 'description', 'lnp_loo
kup', 'allow_internal', 'currency_name_short', 'rtp_timeout', 'route_by', 'currency_id'
, 'balance_name', 'client_id', 'invisible', 'ani_regexp', 'jurisdict_tariff_name', 'id'
, 'jurisdict_tariff_name_currency', 'ext_id', 'tariff_id', 'capacity', 'name', 'point_m
anager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_n
ame', 'simple_pt', 'charge_by', 'tt_timeout', 'proxy_mode', 'dnis_regexp', 'tariff_name
_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing'
, 'max_call_time', 'routes_id')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	True	String(allow_blank: True)	filter by point name or first chars of name
name_or_client_name	N/A	True	String(allow_blank: True)	filter by point/client name or first chars of point/client name
owner_id	N/A	True	Integer(gt: 0)	filter_by owner
service_id	1	True	Integer(gt: 0)	filter by service

				type(e.g. voice, sms)
enabled	N/A	True	Bool	boolean option, if equal True returns only enabled OPs
id	N/A	True	Integer(gt: 0)	filter by OP id
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
invisible	False	True	Or(Bool, Null)	boolean option, if True returns only invisible OPs (flag is ignored if id is set)
clients_ids	N/A	True	List([Integer()], min_length: 0)	filter by client_ids
fields	['client_invisible', 'tariff_name', 'description', 'lnp_lookup', 'allow_internal', 'currency_name_short', 'rtp_timeout', 'route_by', 'currency_id', 'balance_name', 'client_id', 'invisible', 'ani_regex', 'jurisdict_tariff_name', 'id', 'jurisdict_tariff_name_currency', 'ext_id', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'simple_pt', 'charge_by', 'tt_timeout', 'proxy_mode', 'dnis_regex', 'tariff_name_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'max_call_time', 'routes_id']	True	List([Enum('client_invisible', 'tariff_name', 'description', 'lnp_lookup', 'allow_internal', 'currency_name_short', 'rtp_timeout', 'route_by', 'currency_id', 'balance_name', 'client_id', 'invisible', 'ani_regex', 'jurisdict_tariff_name', 'id', 'jurisdict_tariff_name_currency', 'ext_id', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'simple_pt', 'charge_by', 'tt_timeout', 'proxy_mode', 'dnis_regex', 'tariff_name_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'max_call_time', 'routes_id']), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

contexts	[]	True	List([Integer()], min_length: 0)	filter by contexts. If is absent function searches origination points in all contexts.
points_type	all	True	Enum('jurisdictional', 'non_jurisdictional', 'all')	filter points by routing type
ext_id	N/A	True	Integer(gt: 0)	OP id for external billing
sort_by	[]	True	List([{'client_invisible', 'tariff_name', 'description', 'lnp_lookup', 'allow_internal', 'currency_name_short', 'rtp_timeout', 'route_by', 'currency_id', 'balance_name', 'client_id', 'invisible', 'ani_regex', 'jurisdiction_tariff_name', 'id', 'jurisdiction_tariff_name_currency', 'ext_id', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'simple_pt', 'charge_by', 'tt_timeout', 'proxy_mode', 'dnis_regex', 'tariff_name_currency', 'jurisdiction_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'max_call_time', 'routes_id'}], Enum('asc', 'desc', ''), min_length: 0)	field on which result sorts
limit	0	True	Integer(gte: 0)	says how many rows will be in response
manager_id	N/A	True	Integer(gt: 0)	filter by client's manager

Default fields:

Name	Type & Comments
client_invisible	Boolean
tariff_name	String
description	String
lnp_lookup	Int
allow_internal	Boolean
currency_name_short	String
rtp_timeout	Int
route_by	Int
currency_id	Int
balance_name	String
client_id	Int
invisible	Boolean
ani_regexp	String
jurisdict_tariff_name	String
id	Int
jurisdict_tariff_name_currency	String - DEPRECATED
ext_id	Int
tariff_id	Int
capacity	Int
name	String

point_manager_id	Integer - Point manager user id for update notifications
active_tariff_id	Int
ani_block	String
client_name	String
enabled	Boolean
active_tariff_name	String
simple_pt	Boolean
charge_by	Int
tt_timeout	Int
proxy_mode	Int
dnis_regexp	String
tariff_name_currency	String - DEPRECATED
jurisdict_tariff_id	Int
service_id	Int
balance_id	Int
north_american_routing	Boolean
max_call_time	Int
routes_id	Int

2.236 ops_groups_get

Docstring:

Returns data of relations between chosen group and selected OP. Function doesn't show service group in the resulted groups list.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {"sort_by": [{"op_name": ""}],
           "group_id": 6,
           "contexts": [1],
           "fields": ["op_id", "op_name"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["op_id", "op_name"],
  "message": "OK",
  "data": [[3, "Alexander_test_1_OP"]],
  "time_elapsed": 0.009095}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
           'fields': ["Enum('priority', 'op_id', 'group_id', 'op_name', 'group_name')"],
           'group_id': 'Integer',
           'group_name': 'String',
           'limit': 'Integer',
           'offset': 'Integer',
           'op_id': 'Integer',
           'op_name': 'String',
           'service_id': 'Integer',
           'sort_by': [{"Enum('priority', 'op_id', 'group_id', 'op_name', 'group_name')"
: "Enum('asc', 'desc', '')"}]}},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
service_id	N/A	True	Integer(gt: 0, lt: 3)	filter by service id(1 for voice service, 2 for SMS)
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

limit	0	True	Integer(gte: 0)	says how many rows will be in response
op_name	N/A	True	String(allow_blank: True)	filter by OP name or first symbols of OP name
sort_by	[]	True	List([Enum('priority', 'op_id', 'group_id', 'op_name', 'group_name'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"op_name": "asc"}] - sorts result ascending by field "op_name"
group_id	N/A	True	Integer()	filter by group
op_id	N/A	True	Integer()	filter by OP
group_name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name
fields	['priority', 'op_id', 'group_id', 'op_name', 'group_name']	True	List([Enum('priority', 'op_id', 'group_id', 'op_name', 'group_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
contexts	[]	False	List([Integer()], min_length: 0)	context(s) where search relations

Default fields:

Name	Type & Comments
priority	Int
op_id	Int
group_id	Int
op_name	String
group_name	String

2.237 ops_to_groups_add

Docstring:

Adds relations between given OP(s) and group(s) in the chosen context(s) with specific relation priority. Given OP(s) and group(s) should have the same services.

Function disallows to add relation if chosen OP already configured in Simple mode.

Function raises error if user tries to create relation with invisible point.

Example:

- Client Request

```
{"session_id": "91af15315af776928e1abfa0e88b6641",  
  "data": {"priority": 0,  
           "op_ids": ["19"],  
           "contexts": [1],  
           "groups_ids": [9]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.047786}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'groups_ids': ['Integer'],  
          'op_ids': ['Integer'],  
          'priority': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of OPs id, points should be visible
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of groups id
priority	0	True	Integer(gte: -100, lte: 100)	relation priority in range from -100 to

				100
contexts	N/A	False	List([Integer()], min_length: 0)	contexts id where need add relations

2.238 ops_tp_get

Docstring:

Returns a list of OPs which are connected with given TP in simple and advanced routing modes, i.e. OPs that satisfy condition:

- OPs and dialpeers binded with given TP must belong to the same group as TP within chosen context.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "limit": 101,
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "offset": 0,
    "tp_id": 8,
    "fields": ["id", "name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[7, "H_in"], [9, "H_in_simple"]],
  "time_elapsed": 0.009807}
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
  'fields': ["Enum('id', 'name')"],
  'limit': 'Integer',
  'name': 'String',
  'offset': 'Integer',
  'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}],
  'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	False	Integer(gt: 0)	filter by certain TP
name	N/A	True	String(allow_blank: True)	filter by OP name
contexts	[]	False	List([Integer()], min_length: 0)	contexts ids where search OP-TP relations
fields	['id', 'name']	False	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'name': ''}]	True	List([{'name': 'id', 'name': 'name': Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
id	Int
name	String - OP name

2.239 ops_tp_simple_get

Docstring:

Returns a list of OPs which are connected with given TP only in simple mode.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "sort_by": [{"name": ""}]
  }
}
```

```

        "contexts": [1],
        "tp_id": 20,
        "fields": ["id", "name"]}
    }

```

- Server Response

```

{"status": 200,
 "fields": ["id", "name"],
 "message": "OK",
 "data": [[23, "H_in"]],
 "time_elapsed": 0.011045}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'fields': ["Enum('id', 'name')"],
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
contexts	[]	False	List([Integer()], min_length: 0)	contexts ids where search OP-TP relations
fields	['id', 'name']	False	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
tp_id	N/A	False	Integer(gt: 0)	filter by certain TP
sort_by	[{'name': ''}]	True	List([{'Enum('id', 'name')': Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
name	N/A	True	String(allow_blank: True)	filter by OP name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before

				beginning to return rows.
--	--	--	--	---------------------------

Default fields:

Name	Type & Comments
id	Int
name	String - OP name

2.240 outgoing_call_pathes_get

Docstring:

Returns list of origination points which have route to given TP to specific destination at the given time.
 If 'show_stat' not equal 'true' in request then list of routes will be ordered according to the routing algorithm:

- 1) by priority of OP->Group relation
- 2) by dialpeer priority
- 3) by priority of dialpeer->TP relation
- 4) by weight of relation between dialpeer and TP if priority of dialpeer->TP relation is the same
- 5) by TP price priority
- 6) LCR (more profitable routes will be higher in list). Prices shows in the chosen in request currency, but compare in default currency.

If in request 'show_stat' is 'true' list of routes will be ordered by difference between OP and TP price descending.

Function returns parameters of the targeted TP within key 'tp_data'.

'tp_data' fields:

```
[
    'tp_name',
    'tp_capacity',
    'tp_country',
    'tp_price_id',
    'tp_id',
    'tp_enabled',
    'tp_destination',
    'tp_price_default',
    'tp_mc',
    'tp_ci',
    'tp_ft',
    'tp_currency_id',
    'tp_cid',
    'tp_price_priority',
    'tp_tariff_invisible',
    'tp_price_tarif',
    'tp_currency_name',
    'tp_origin_country'
```

```
]
```

Example:

- Client Request

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "time_zone": "0:00",
    "time_period": "2014-01-13T16:01:18",
    "contexts": [1],
    "number": "923000000000",
    "tp_id": 8,
    "fields": [
      "op_enabled", "delta_price", "dif_price", "op_id",
      "op_price_id", "group_id", "dialpeer_id",
      "allowed_price", "op_name", "op_destination",
      "op_country", "op_price_default",
      "op_price_tariff", "op_currency_name", "op_mc",
      "op_ci", "op_ft", "op_capacity", "hunt_stop", "dp_name",
      "dp_tp_priority", "dp_priority", "group_name",
      "op_group_priority", "profit", "client_id"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.047496,
  "fields": [
    "op_enabled", "delta_price", "dif_price", "op_id",
    "op_price_id", "group_id", "dialpeer_id",
    "allowed_price", "op_name", "op_destination",
    "op_country", "op_price_default", "op_price_tariff",
    "op_currency_name", "op_mc", "op_ci", "op_ft",
    "op_capacity", "hunt_stop", "dp_name", "dp_tp_priority",
    "dp_priority", "group_name", "op_group_priority",
    "profit", "client_id"
  ],
  "message": "OK",
  "tp_data": [
    ["H_to", 500, "Pakistan Mobile Mobilink", 20, 8,
      true, "9230", "10.3500", 1, 1, 0, 1, 6, 0, false, "20.7000", "
USD", "92"],
    ["data": [
      [true, "0.00000", "10.0500", 7, 13, 3, 4, true,
        "H_in", "9230", "Pakistan Mobile Mobilink",
        "10.4000", "10.4000", "USD", 1, 1, 0, 5, "H_p", 0, 0,
        "h_p", 0, "14.29", 5]]
    ]
  ]
}
```

Request structure and fields description:

```
{
  'data': {
    'caller_number': 'String',
    'contexts': ['Integer'],
    'currency_id': 'Integer',
    'fields': [
      "Enum('dp_tp_priority', 'price_priority', 'op_destination', 'tp_de
stination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'o
p_tariff_invisible', 'tp_currency_sign', 'tp_price_priority', 'tp_capacity', 'dif_price
', 'op_id', 'client_id', 'op_mc', 'tp_cid', 'tp_price_default', 'op_name', 'dp_capacity
', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_group_pri
ority', 'op_price_tariff', 'op_capacity', 'op_origin_country', 'tp_name', 'op_currency_n
ame', 'delta_price', 'tp_price_tariff', 'hunt_stop', 'op_without_ip', 'profit', 'op_enab
led', 'dp_priority', 'tp_tariff_invisible', 'op_price_default', 'dp_name', 'op_ci', 'as
r', 'tp_enabled', 'op_currency_sign', 'tp_id', 'tp_mc', 'tp_origin_country', 'dialpeer
_id', 'op_price_id', 'allowed_price', 'acd', 'op_ft', 'tp_country', 'tp_price_id', 'op_c
urrency_id')"],
    'limit': 'Integer',
  }
}
```

```

'number': 'String',
'offset': 'Integer',
'session_id': 'String',
'show_disabled_pt': 'Bool',
'show_stat': 'Bool',
'time_period': 'String',
'time_zone': 'String',
'tp_id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	User session id, need for checking user permissions. By defaults from user session.
contexts	N/A	False	List([Integer()], min_length: 0)	ids context, in which request will be processed
number	N/A	False	String(allow_blank: False, regex: "^[0-9#*]+\$")	Called number
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
caller_number	""	True	String(allow_blank: True)	caller number
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
time_period	N/A	False	String(allow_blank: False)	request time in format yyyy-MM-ddT'HH:mm:ss
currency_id	N/A	True	Integer(gt: 0)	Currency in which prices will display. By defaults from user

				session.
tp_id	N/A	False	Integer(gt: 0)	termination point id
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2})\$")	time zone for 'time_period'
show_disabled_pt	False	True	Bool	boolean parameter, if 'True' then routes from disabled OPs will be shown also
show_stat	False	True	Bool	if 'True' then ACD,ASR for last 4 hours will returns
fields	['dp_tp_priority', 'price_priority', 'op_destination', 'tp_destination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'op_tariff_invisible', 'tp_currency_sign', 'tp_price_priority', 'tp_capacity', 'dif_price', 'op_id', 'client_id', 'op_mc', 'tp_cid', 'tp_price_default', 'op_name', 'dp_capacity', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_group_priority', 'op_price_tarif', 'op_capacity', 'op_origin_country', 'tp_name', 'op_currency_name', 'delta_price', 'tp_price_tarif', 'hunt_stop', 'op_without_ip', 'profit', 'op_enabled', 'dp_priority', 'tp_tariff_invisible', 'op_price_default', 'dp_name', 'op_ci', 'asr', 'tp_enabled', 'op_currency_sign', 'tp_id', 'tp_mc', 'tp_origin_country', 'dialpeer_id', 'op_price_id', 'allowed_price', 'acd', 'op_ft', 'allowed_price', 'acd', 'op_ft',	True	List([Enum('dp_tp_priority', 'price_priority', 'op_destination', 'tp_destination', 'tp_currency_id', 'group_name', 'attempts', 'tp_ft', 'tp_currency_name', 'op_tariff_invisible', 'tp_currency_sign', 'tp_price_priority', 'tp_capacity', 'dif_price', 'op_id', 'client_id', 'op_mc', 'tp_cid', 'tp_price_default', 'op_name', 'dp_capacity', 'delta_profit', 'delta_price_unit', 'tp_ci', 'group_id', 'op_country', 'op_group_priority', 'op_price_tarif', 'op_capacity', 'op_origin_country', 'tp_name', 'op_currency_name', 'delta_price', 'tp_price_tarif', 'hunt_stop', 'op_without_ip', 'profit', 'op_enabled', 'dp_priority', 'tp_tariff_invisible', 'op_price_default', 'dp_name', 'op_ci', 'asr', 'tp_enabled', 'op_currency_sign', 'tp_id', 'tp_mc', 'tp_origin_country', 'dialpeer_id', 'op_price_id', 'allowed_price', 'acd', 'op_ft', 'tp_country', 'tp_price_id',	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.

	'tp_country', 'tp_price_id', 'op_currency_id']		'op_currency_id')], min_length: 0)	
--	---	--	---------------------------------------	--

Default fields:

Name	Type & Comments
dp_tp_priority	Int
price_priority	Int
op_destination	String
tp_destination	String
tp_currency_id	Int
group_name	String
attempts	Int
tp_ft	Int
tp_currency_name	String(3) - name_short
op_tariff_invisible	Bool
tp_currency_sign	String
tp_price_priority	Int
tp_capacity	Int
dif_price	numeric(16,5)
op_id	Int
client_id	Int
op_mc	Int

tp_cid	Int
tp_price_default	numeric(16,5)
op_name	String
dp_capacity	numeric(15,2)
delta_profit	numeric(15,2)
delta_price_unit	Int
tp_ci	Int
group_id	Int
op_country	String
op_group_priority	Int
op_price_tarif	numeric(16,5)
op_capacity	Int
op_origin_country	String
tp_name	String
op_currency_name	String(3) - name_short
delta_price	numeric(16,5)
tp_price_tarif	numeric(16,5)
hunt_stop	Boolean
op_without_ip	Boolean
profit	numeric(15,2)
op_enabled	Boolean

dp_priority	Int
tp_tariff_invisible	Bool
op_price_default	numeric(16,5)
dp_name	String
op_ci	Int
asr	numeric or null
tp_enabled	boolean
op_currency_sign	String
tp_id	Int
tp_mc	Int
tp_origin_country	String
dialpeer_id	Int
op_price_id	Int
allowed_price	Bool
acd	Int or null
op_ft	Int
tp_country	String
tp_price_id	Int
op_currency_id	Int

2.241 outgoing_calls_get

Docstring:

Returns outgoing calls attempts which has been sent to TP during given time period.

To define from where to get data (from cdrs or aggregation) options "type" and "by_cdr" has been designed.

Function stores responses in cache during 5 minutes. That's why user should wait 5 minutes to get new calls data or change request params.

To get each call separately chose type "individual". With type "individual" it invokes function '_outgoing_calls_get_individual' which returns data from cdrs ordered by 'setup_time' descending.

To get calls data combined in some way use necessary "type" ("combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" or "combine_disconnect_code") in conjunction with boolean option "by_cdr".

If type not individual key 'by_cdr' should be defined.

If "by_cdr" is "true" - it calls function '_outgoing_calls_get_by_cdr' which takes data from cdrs, otherwise - calls '_outgoing_calls_get_by_agg' that takes data from aggregation. In the second case(type = "combine_dest", "month" etc.) function may return less precise result for last several hours.

Note, that this function has limitation of maximal time period which depends on chosen "type":

- 1) 7 days for "type" "individual";
- 2) 31 day for "type": "combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point" and "combine_disconnect_code";
- 3) 12 hours for "type" listed in 2) if flag "by_cdr" is "true".

If 'bill_id' of call (incoming and outgoing parts of the same call has the same 'bill_id') is specified in request - then calls '_outgoing_calls_get_by_bill_id' function which returns all outgoing attempts with given bill_id. Use this option in conjunction with 'nas'.

`_outgoing_calls_get_by_bill_id` ([data validation](#)) ([default fields](#))

Returns outgoing attempts for given incoming call bill_id.

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",  
  "data": {"time_zone": "0:00",  
           "end_date": "2014-01-13T23:59:59",  
           "currency_id": 1,  
           "nas": "193.167.82.77",  
           "start_date": "2014-01-13T00:00:00",  
           "bill_id": "1335380414-2489136"}  
}
```

- Server Response

```
{"status": 200,
```

```

"fields": ["destination", "setup_time", "connect_time",
           "disconnect_time", "name", "tp_id", "called_number_in",
           "clear_num", "called_number_out", "tp_client_id",
           "tp_client_name", "duration", "disconnect_cause",
           "local_disconnect_cause", "tp_price", "tp_cost",
           "tp_ip", "nas", "dst_codec", "outgoing_ani",
           "country"],
"message": "OK",
"data": [[ "9232", "2014-01-13T12:05:38.022000+00:00",
           "2014-01-13T12:05:42.945000+00:00",
           "2014-01-13T12:05:42.945000+00:00", "H_to", 8,
           "400923249159924", "923249459951", "400923245159921", 6,
           "H_To", 0, 16, 12000, "0.03500", null, "195.24.165.1",
           "193.167.82.77", null, "1959", "Pakistan Mobile Warid"]],
"time_elapsed": 0.013695}

```

`_outgoing_calls_get_by_cdr` ([data_validation](#)) ([default_fields](#))

It returns outgoing calls statistics from cdrs combined by given criteria. This function more precise result then from aggregation because it accounts for current traffic.

Example:

- Client Request

```

{"session_id": "f345e24573e77cadffbc28a11ae64af4",
 "data": {"limit": 1,
          "time_zone": "0:00",
          "currency_id": 1,
          "end_date": "2014-01-12T23:59:59",
          "longer_than": "0",
          "not_use_like": false,
          "offset": 0,
          "type": "day",
          "start_date": "2014-01-12T00:00:00",
          "by_cdr": true,
          "fields": ["some_xy", "tr_date", "asr", "acd", "pdd", "sc",
                     "duration", "attempts", "sa", "op_sum", "tp_sum",
                     "delta_price"]}
}

```

- Server Response

```

{"status": 200,
 "fields": ["some_xy", "tr_date", "asr", "acd", "pdd", "sc",
            "duration", "attempts", "sa", "op_sum", "tp_sum",
            "delta_price"],
 "message": "OK",
 "data": [[false, "2014-01-12T00:00:00+00:00", "100.00", 209, "1.8",
              "23.27", 260675, 1246, "1246",
              "173.783320000000000000000000000000", "152.062170000000000000000000000000",
              "21.72115"]],
 "time_elapsed": 0.519934}

```

`_outgoing_calls_get_individual` ([data_validation](#)) ([default_fields](#))

Returns authorized calls records that were routed to supplier during given time period.
Each call shows individually ("type":"individual").

If current user doesn't belong to the company of mediacore owner ('is_client' = True), then function hides fields that relates to the OP.

Example:

- Client Request

```
{"session_id":"f345e24573e77cadffbc28a11ae64af4",
  "data":{"limit":1,
    "time_zone":"0:00",
    "currency_id":1,
    "end_date":"2014-01-13T23:59:59",
    "longer_than":"0",
    "not_use_like":false,
    "offset":0,
    "type":"individual",
    "start_date":"2014-01-13T00:00:00",
    "fields":["nas","op_name","setup_time","country",
      "called_number_in","clear_number",
      "op_caller_number","tp_caller_number",
      "called_number_out","duration","connect_time",
      "disconnect_time","disconnect_cause_name",
      "disconnect_cause","local_disconnect_cause_name",
      "local_disconnect_cause","op_price","tp_price",
      "op_cost","tp_cost","op_ip","op_protocol","tp_ip",
      "tp_protocol","nas","op_codecs","tp_codecs",
      "dialpeer","pdd","op_class_name","tp_class_name",
      "ds","scd"],
    "fas_possible":false}}
```

- Server Response

```
{"status": 200,
  "fields": ["nas", "op_name", "setup_time", "country",
    "called_number_in", "clear_number", "op_caller_number",
    "tp_caller_number", "called_number_out", "duration",
    "connect_time", "disconnect_time",
    "disconnect_cause_name", "disconnect_cause",
    "local_disconnect_cause_name", "local_disconnect_cause",
    "op_price", "tp_price", "op_cost", "tp_cost", "op_ip",
    "op_protocol", "tp_ip", "tp_protocol", "nas",
    "op_codecs", "tp_codecs", "dialpeer", "pdd",
    "op_class_name", "tp_class_name", "ds", "scd"],
  "message": "OK",
  "data": [[ "194.151.88.89", "H_in",
    "2014-01-13T12:09:30.066000+00:00", "Pakistan Mobile Telnor",
    "200923474115816", "923474115816", "unknown", "unknown",
    "400923474115816", 32, "2014-01-13T12:09:45.819000+00:00",
    "2014-01-13T12:10:17.061000+00:00",
    "Normal call cleared (CANCEL or BYE received)", 0,
    "BYE received from origination side", 13000, "0.04000",
    "0.03500", "0.02133", "0.01867", "193.27.163.12", "sip",
    "193.27.163.12", "h323", "194.151.88.89",
    "G729,ALAW,G723,MULAW", "G729", "H_p", 0,
```

```
        "Completed, successful", "Completed, successful", 1, 15]],
    "time_elapsed": 0.016223}
```

`_outgoing_calls_get_by_agg` ([data_validation](#)) ([default_fields](#))

Function returns outgoing calls statistics combined by chosen criteria from aggregation and thats why it doesn't account current traffic.

Example:

- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {
    "limit": 1,
    "time_zone": "0:00",
    "currency_id": 1,
    "end_date": "2014-01-12T23:59:59",
    "longer_than": "0",
    "not_use_like": false,
    "offset": 0,
    "type": "day",
    "start_date": "2014-01-12T00:00:00",
    "by_cdr": false,
    "fields": [
      "some_xy", "tr_date", "asr", "acd", "pdd", "sc",
      "duration", "attempts", "sa", "op_sum", "tp_sum", "delta_pric
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "some_xy", "tr_date", "asr", "acd", "pdd", "sc",
    "duration", "attempts", "sa", "op_sum", "tp_sum",
    "delta_price"
  ],
  "message": "OK",
  "data": [
    [
      true, "2014-01-12T00:00:00+00:00", "0.78", 206, "2.4",
      "24.13", 260675, 161745, "1260",
      "173.783320000000000000000000000000",
      "152.070290000000000000000000000000", "21.71303"
    ]
  ],
  "time_elapsed": 0.167038
}
```

Request structure and fields description:

`_outgoing_calls_get_by_bill_id`

```
{
  'data': {
    'bill_id': 'String',
    'currency_id': 'Integer',
    'end_date': 'String',
    'fields': [
      "Enum('tp_client_id', 'clear_num', 'duration', 'tp_client_name', 'dst_codec', 'ocn', 'tp_protocol', 'destination', 'nas', 'lata', 'disconnect_time', 'connect_time', 'called_number_out', 'tp_id', 'setup_time', 'tp_media_ip', 'disconnect_cause', 'calc_duration', 'tp_ip', 'name', 'local_disconnect_cause', 'country', 'tp_price', 'outgoing_ani', 'tp_cost', 'called_number_in')"]
    ]
  }
}
```

```

'nas': 'String',
'start_date': 'String',
'time_zone': 'String',
'user_id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['tp_client_id', 'clear_num', 'duration', 'tp_client_name', 'dst_codec', 'ocn', 'tp_protocol', 'destination', 'nas', 'lata', 'disconnect_time', 'connect_time', 'called_number_out', 'tp_id', 'setup_time', 'tp_media_ip', 'disconnect_cause', 'calc_duration', 'tp_ip', 'name', 'local_disconnect_cause', 'country', 'tp_price', 'outgoing_ani', 'tp_cost', 'called_number_in']	True	List([Enum('tp_client_id', 'clear_num', 'duration', 'tp_client_name', 'dst_codec', 'ocn', 'tp_protocol', 'destination', 'nas', 'lata', 'disconnect_time', 'connect_time', 'called_number_out', 'tp_id', 'setup_time', 'tp_media_ip', 'disconnect_cause', 'calc_duration', 'tp_ip', 'name', 'local_disconnect_cause', 'country', 'tp_price', 'outgoing_ani', 'tp_cost', 'called_number_in'), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error
nas	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))? (0\.){3}0(/0)?\$")	RTP server IP address
currency_id	1	True	Integer()	currency for tp price
user_id	N/A	True	Integer()	user id to filter by locked companies and show IP permissions, by defaults is taken from current session

time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2}\$")	"start_date" and "end_date" timezone
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
bill_id	N/A	False	String(allow_blank: False)	incoming call bill id

_outgoing_calls_get_by_cdr

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd')"],
          'hosts': ['String'],
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'min_attempts': 'Integer',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'outgoing_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'sort_by': [{"Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'termcause': 'String',
          'time_zone': 'String',
          'tp_class_id': 'Integer',
          'tp_destination': 'String',
```

```

'tp_destination_exclude': 'String',
'tp_id': 'Integer',
'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_latest_stat')",
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d{3}[0-2])?(0\.){3}0(0)?)\$\$")	IP address of call originator
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults session_id from current session
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\d+(, \s)?)*\$")	destination code(s), which should not show in report, separated by comma
tp_id	N/A	True	Integer(gt: 0)	
hosts	[]	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d{3}[0-2])?(0\.){3}0(0)?)\$\$")])	filter by RTP server IP. A few addresses need separate by comma.

			d[[1-9]\d 1\d{2} 2[0-4]\d 25[0-5]]/([1-9] [12]\d 3[0-2]))? (0\.){3}0(/0)?\$")], min_length: 0)	
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
not_use_like	N/A	True	Bool	if 'true' searches only calls by exact destination codes without subcodes
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
type	N/A	True	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_latest_stat')	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):(?\\d{2})\$")	
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from usersession.
termcause	N/A	True	String(allow_blank: False)	filter by disconnect cause code
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
tp_class_id	N/A	True	Integer(gte: 0)	Shows in report only calls with

				given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed wrong".
outgoing_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?\$")	IP address of the termination side
min_attempts	0	True	Integer(gte: 0)	filter by attempts quantity
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
client_id	N/A	True	Integer(gt: 0)	not in use
sort_by	[]	True	List([{'enum': 'tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd'}])	field on which result will be sorted. Result can be sorted ascending or descending.

			Enum('asc', 'desc', ")}], min_length: 0)	
fields	['tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd']	True	List([Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'callback_rate', 'sc', 'some_xy', 'sa', 'pdd')], min_length: 0)	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsence of filtration.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.

_outgoing_calls_get_individual

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fas_possible': 'Bool',
          'fields': ["Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name',
'op_caller_number', 'clear_number', 'tp_protocol', 'op_price', 'tp_client_id', 'op_prot
ocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_id', 'op_class_id', 'ds', 'dial
peer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op_na
me', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', '
tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'tp_media_ip',
'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause_nam
e', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'pdd', 'calle
d_number_in')"],
          'hosts': "['String'] or String",
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'outgoing_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'show_stat': 'Bool',
          'sort_by': [{"Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name
', 'op_caller_number', 'clear_number', 'tp_protocol', 'op_price', 'tp_client_id', 'op_p
rotocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_id', 'op_class_id', 'ds', 'd
```

```

ialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op
_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time'
, 'tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'tp_media_i
p', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause
_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'pdd', 'c
alled_number_in'): "Enum('asc', 'desc', '')"}]],
    'start_date': 'String',
    'termcause': 'String',
    'time_zone': 'String',
    'tp_class_id': 'Integer',
    'tp_destination': 'String',
    'tp_destination_exclude': 'String',
    'tp_id': 'Integer',
    'type': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])?/([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?)\$")	IP address of call originator
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(d+(, s)?)*\$")	destination code(s), which should not show in report, separated by comma
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+ -] ? \d { 1, 2 }) : (\d { 2 }) \$")	"start_date" and "end_date" timezone
termcause	N/A	True	String(allow_blank: False)	filter by disconnect cause code
outgoing_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])?/([1-9]	IP address of the termination side

			[12]\d{3}[0-2]))?(0\.){3}0(/0)?\$")	
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsense of filtration.
hosts	[]	True	Or(List([String(allow_blank: False, regex: "^([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.((\d{1}[1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.){2}(\d{1}[1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])/([1-9] [12]\d{3}[0-2]))?(0\.){3}0(/0)?\$")], min_length: 0), String(allow_blank: False, regex: "^([1-9] [1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.((\d{1}[1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])\.){2}(\d{1}[1-9]\d{1}\d{2} 2[0-4]\d{25}[0-5])/([1-9] [12]\d{3}[0-2]))?(0\.){3}0(/0)?\$"))	filter by RTP server IP. A few addresses need separate by comma.
sort_by	[]	True	List([{'op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codecs', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'tp_media_ip', 'calc_duration', 'tp_ip',	field on which result will be sorted. Result can be sorted ascending or descending.

			'incoming_ani', 'op_codecs', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'pdd', 'called_number_in'): Enum('asc', 'desc', ''), min_length: 0)	
not_use_like	False	True	Bool	if "true" searches only calls by exact destination codes without subcodes
session_id	N/A	True	String(allow_blank: False)	not in use
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
tp_class_id	N/A	True	Integer(gte: 0)	Shows in report only calls with given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed wrong".
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
client_id	N/A	True	Integer(gt: 0)	filter by company id.

limit	N/A	False	Integer(gt: 0)	says how many rows will be in response
fields	['op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codec', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'tp_media_ip', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codec', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'pdd', 'called_number_in']	True	List([Enum('op_destination', 'tp_destination', 'scd', 'tp_client_name', 'op_caller_number', 'clear_number', 'tp_protocol', 'op_price', 'tp_client_id', 'op_protocol', 'tp_class_id', 'called_number_out', 'op_ip', 'op_id', 'op_class_id', 'ds', 'dialpeer', 'local_disconnect_cause_name', 'tp_class_name', 'local_disconnect_cause', 'op_name', 'op_client_name', 'outgoing_ani', 'tp_caller_number', 'tp_name', 'connect_time', 'tp_codec', 'duration', 'nas', 'disconnect_time', 'tp_id', 'setup_time', 'tp_media_ip', 'calc_duration', 'tp_ip', 'incoming_ani', 'op_codec', 'tp_cost', 'disconnect_cause_name', 'op_cost', 'country', 'tp_price', 'op_client_id', 'disconnect_cause', 'pdd', 'called_number_in')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
fas_possible	False	True	Bool	filter calls for signs of FAS (SCD < 1 sec, SCD is Setup Connect Delay: connect_time - setup_time)
tp_destination	N/A	True	String(allow_blank: False, regex: "^(%?d+(, \s?))*\$")	destination code(s), which need show in report, separated by comma
shorter_than	N/A	True	Integer(gte: 0)	filter that allows to return calls with duration (in seconds) less than given.

end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
longer_than	N/A	True	Integer(gte: 0)	filter that allows to return calls with duration (in seconds) more than given.
clients_ids	[]	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids. Currently it takes into account only first id from list.
type	N/A	False	String(allow_blank: False, regex: "^individual\$")	statistical resolution type
show_stat	False	True	Bool	not in use
tp_id	N/A	True	Integer(gt: 0)	

_outgoing_calls_get_by_agg

```
{'data': {'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd')"],
          'hosts': ['String'],
          'incoming_ip': 'String',
          'limit': 'Integer',
          'longer_than': 'Integer',
          'managers': ['Integer'],
          'min_attempts': 'Integer',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'session_id': 'String',
          'shorter_than': 'Integer',
          'sort_by': [{"Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pd
```

```
d'): "Enum('asc', 'desc', '')"}],
    'start_date': 'String',
    'time_zone': 'String',
    'tp_destination': 'String',
    'tp_destination_exclude': 'String',
    'tp_id': 'Integer',
    'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_a
nd_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destina
tion', 'combine_point_nas_ip')"}},
    'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given. End date will be rounded up to an hour.
longer_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) more than given.
shorter_than	N/A	True	Integer(gte: 0)	filter that accounts calls with duration (in seconds) less than given.
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3})\$")	IP address of call originator

			1\d{2} 2[0-4]\d 25[0-5])\.)}{2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?)\$")	
hosts	[]	True	List([String(allow_blank: False, regex: "^([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?)\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2})\$")	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
client_id	N/A	True	Integer(gt: 0)	not in use
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
min_attempts	0	True	Integer(gte: 0)	filter by attempts

				quantity
type	N/A	False	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip')	method of grouping
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
sort_by	[]	True	List([Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd'): Enum('asc', 'desc', '')), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
tp_id	N/A	True	Integer(gt: 0)	
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means absence of filtration.
fields	['tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd']	True	List([Enum('tp_name', 'op_sum', 'tp_sum', 'delta_price', 'duration', 'destination', 'nas', 'percentage', 'ner', 'termcode', 'asr', 'tr_date', 'calc_duration', 'name_termcode', 'acd', 'country', 'attempts', 'op_name', 'sc', 'some_xy', 'sa', 'pdd')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

_outgoing_calls_get_by_bill_id

Name	Type & Comments
tp_client_id	Int
clear_num	String(200)
duration	Int
tp_client_name	String(64)
dst_codec	String(255)
ocn	String
tp_protocol	String(100)
destination	String(24)
nas	String (RTP IP address)
lata	String
disconnect_time	String
connect_time	String
called_number_out	String(200)
tp_id	Int
setup_time	String
tp_media_ip	String
disconnect_cause	Int
calc_duration	Int
tp_ip	String

name	String(64) (TP name)
local_disconnect_cause	Int
country	String(64)
tp_price	numeric(16,5)
outgoing_ani	String(200)
tp_cost	numeric(16,5)
called_number_in	String(200)

_outgoing_calls_get_by_cdr

Name	Type & Comments
tp_name	String
op_sum	numeric
tp_sum	numeric
delta_price	numeric(16,5) - difference between op_sum and tp_sum
duration	Int
destination	String
nas	String (RTP IP address)
percentage	String - rate of calls per disconnect code (available only for types 'combine_disconnect_code' and 'combine_disconnect_code_and_destination')
ner	String
termcode	Int

asr	String
tr_date	String - start date of group period (begin of the month for "type" - "month" and so on)
calc_duration	Int
name_termcode	String
acd	Int
country	String
attempts	Int
op_name	String
callback_rate	numeric(10,2)
sc	numeric (short calls percentage (N_short/N_success)*100%)
some_xy	Boolean ('True' if ASR <= 5% or ACD <= 60 seconds)
sa	numeric (number of successful attempts)
pdd	numeric - post dial delay in seconds

_outgoing_calls_get_individual

Name	Type & Comments
op_destination	String
tp_destination	String
scd	Int (setup connect delay, connect_time - setup_time in seconds)
tp_client_name	String
op_caller_number	String(200)

clear_number	String
tp_protocol	String(100)
op_price	numeric(16,5)
tp_client_id	Int
op_protocol	String(100)
tp_class_id	Int
called_number_out	String
op_ip	String
op_id	Int
op_class_id	Int
ds	Int (disconnect side, may take values 0, 1 or 2)
dialpeer	String
local_disconnect_cause_name	String
tp_class_name	String
local_disconnect_cause	Int
op_name	String
op_client_name	String
outgoing_ani	String
tp_caller_number	String(200)
tp_name	String
connect_time	String

tp_codecs	String
duration	Int
nas	String
disconnect_time	String
tp_id	Int
setup_time	String
tp_media_ip	String
calc_duration	Int
tp_ip	String
incoming_ani	String
op_codecs	String
tp_cost	numeric(16,5)
disconnect_cause_name	String
op_cost	numeric(16,5)
country	String
tp_price	numeric(16,5)
op_client_id	Int
disconnect_cause	Int
pdd	Int
called_number_in	String

_outgoing_calls_get_by_agg

Name	Type & Comments
tp_name	String
op_sum	numeric
tp_sum	numeric
delta_price	numeric(16,5) - difference between op_sum and tp_sum
duration	Int
destination	String
nas	String (RTP IP address)
percentage	String - rate of calls per disconnect code (available only for types 'combine_disconnect_code' and 'combine_disconnect_code_and_destination')
ner	String
termcode	Int
asr	String
tr_date	String - start date of group period (begin of the month for 'type' - 'month' and so on)
calc_duration	Int
name_termcode	String
acd	Int
country	String
attempts	Int
op_name	String
sc	numeric (short calls percentage ($N_{short}/N_{success}$)*100%)
some_xy	Boolean ('True' if ASR <= 5% or ACD <= 60 seconds)

sa	numeric (number of successful attempts)
pdd	Int - Post Dial Delay in seconds

2.242 outgoing_invoice_add

Docstring:

Creates new outgoing invoice for chosen supplier's balance. Total invoice amount calculates by formula:

Total Amount="Initial Claimed"+"Additional Amount"-"Amount Protested".

Function raises error in the following cases:

- 1) "date_to">=current or "date_from">=current
- 2) if invoice for given period already exist
- 3) invoice "total amount" = 0
- 4) If session id incorrect.

If time period in query covers several month function creates invoice for each one separately. Function returns list of created invoices ids.

Function returns message within key "error".

In case function returned status 305 - then it is just informational message. In other cases key "error" contains message which describes error has been occurred.

Example:

- Client Request

```
{
  "session_id": "049a4d1709ba449a5588ceb602b3d001",
  "data": {
    "time_zone": "0:00",
    "amount_additional": 0.0,
    "date_from": "2014-01-10T00:00:00",
    "include_vat": false,
    "email": "test@test.com",
    "date_to": "2014-01-15T23:59:59",
    "balance_id": 6,
    "notes": "test"
  }
}
```

- Server Response

```
{
  "status": 305,
  "message": "Informal",
  "details": "Invoice for the period from 2014-01-10 00:00:00 to 2014-01-15 23:59:59 successfully created!"
}
```

Request structure and fields description:

```
{'data': {'amount_additional': 'Float',  
          'balance_id': 'Integer',  
          'date_from': 'String',  
          'date_to': 'String',  
          'email': 'String',  
          'include_vat': 'Bool',  
          'notes': 'String',  
          'service_id': 'Integer',  
          'session_id': 'String',  
          'time_zone': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
include_vat	N/A	False	Bool	if True - VAT will be included in the invoice amount
date_from	N/A	False	String(allow_blank: False)	
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}:\\d{2}\$")	specify invoice timezone if it different from GMT +0
email	""	True	String(allow_blank: True)	not in use
notes	N/A	False	String(allow_blank: True)	some note
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults from user session
balance_id	N/A	False	Integer(gt: 0)	
amount_additional	0.0	True	Float()	amount which may correct balance if it allowed in database. This value will be added to balance amount.
service_id	1	True	Integer(gt: 0)	service type(voice - 1, sms - 2)
date_to	N/A	False	String(allow_blank:	

			False)	
--	--	--	--------	--

2.243 outgoing_invoice_get

Docstring:

Calls function `_incoming_outgoing_invoice_get`(with `"is_tp"=True`) that returns list of existing outgoing invoices for specified time period, balance, company.

Example:

- Client Request

```
{
  "session_id": "049a4d1709ba449a5588ceb602b3d001",
  "data": {
    "limit": 101,
    "date_from": "2014-01-08T00:00:00",
    "sort_by": [{"balance_name": ""}],
    "date_to": "2014-01-15T00:00:00",
    "balance_id": 6,
    "offset": 0,
    "fields": [
      "number", "balance_name", "client_name",
      "gen_date", "from_date", "to_date",
      "amount_claimed", "amount_total", "status", "id",
      "amount_additional", "notes", "owner_name",
      "time_zone"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "number", "balance_name", "client_name", "gen_date",
    "from_date", "to_date", "amount_claimed",
    "amount_total", "status", "id", "amount_additional",
    "notes", "owner_name", "time_zone"
  ],
  "message": "OK",
  "data": [
    [
      "000120140115s", "H_To_balance (USD)", "H_To",
      "2014-01-16T14:03:23.340568+00:00", "2014-01-10T00:00:00",
      "2014-01-15T23:59:59", "691.90", "691.90", "In process",
      1, "0.00", "test", "MC v3.1 owner", "+0:00"
    ]
  ],
  "time_elapsed": 0.011348
}
```

`_incoming_outgoing_invoice_get` ([data_validation](#)) ([default_fields](#))

Returns list of existing incoming or outgoing invoices for specified time period, balance, company.

Request structure and fields description:

_incoming_outgoing_invoice_get

```
{'data': {'amount_additional': 'Float',
          'amount_greater': 'Float',
          'amount_less': 'Float',
          'balance_id': 'Integer',
          'company_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'fields': ["Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id')"],
          'gen_date': 'String',
          'id': 'Integer',
          'is_tp': 'Bool',
          'limit': 'Integer',
          'notes': 'String',
          'number': 'String',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'period': 'Integer',
          'service_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id')": "Enum('asc', 'desc', '')"}],
          'status_id': 'Integer',
          'type': 'Integer'},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
amount_less	N/A	True	Float()	returns list of invoices with amount less than specified
type	N/A	True	Integer(gte: 0)	
gen_date	N/A	True	String(allow_blank: False)	creation date

session_id	N/A	True	String(allow_blank: False)	by defaults from user session
period	N/A	True	Integer(gte: 1, lte: 4)	filter by invoice period id
number	N/A	True	String(allow_blank: False)	search by invoice number
date_to	N/A	True	String(allow_blank: False)	upper boundary of request time slice. Combine inputted date with 23:59:59 time
amount_greater	N/A	True	Float()	returns list of invoices with amount more than specified
id	N/A	True	Integer(gt: 0)	invoice id in the database
date_from	N/A	True	String(allow_blank: False)	lower boundary of request time slice
is_tp	N/A	True	Bool	if "true" returns outgoing invoices, else - incoming
owner_id	N/A	True	Integer(gt: 0)	filter by owner
notes	""	True	String(allow_blank: True)	notes. amount protested is here
company_id	N/A	True	Integer(gt: 0)	filter by company id
amount_additional	N/A	True	Float()	additional amount for price correction
service_id	N/A	True	Integer(gt: 0, lt: 3)	filter by service(1 - voice, 2 - SMS). To get catalog

				info about service types use function services_get.
fields	['status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id']	True	List([Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
balance_id	N/A	True	Integer(gt: 0)	filter by balance id
sort_by	[]	True	List([Enum('status', 'due_date', 'attempts', 'owner_name', 'correct_by_price', 'status_id', 'service_name', 'owner_email', 'problems', 'number', 'gen_date', 'balance_name', 'mailed', 'to_date', 'balance_id', 'duration', 'id', 'amount_total', 'amount_protested', 'amount_vat', 'notes', 'time_zone', 'amount_claimed', 'amount_additional', 'from_date', 'client_email_real', 'vat_rate', 'client_name', 'service_id', 'email', 'vat', 'owner_id'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted
limit	0	True	Integer(gte: 0)	says how many

				rows will be in response
status_id	N/A	True	Integer(gte: 0)	invoice status id, 0 - "In process", 1- "Accepted", 2 - "Protested", 3 - "Closed", 4 - "Deleted", 5 - "Inv. changed"

Default fields:

_incoming_outgoing_invoice_get

Name	Type & Comments
status	String
due_date	String - date when invoice should be paid
attempts	Int
owner_name	String
correct_by_price	Boolean
status_id	Int
service_name	String
owner_email	String
problems	String
number	String
gen_date	String
balance_name	String
mailed	Boolean

to_date	String
balance_id	Int
duration	Int
id	Int
amount_total	numeric(16,2)
amount_protested	numeric(16,2)
amount_vat	numeric(16,2)
notes	String
time_zone	String
amount_claimed	numeric(16,2)
amount_additional	numeric(16,2)
from_date	String
client_email_real	String - actual client's billing email
vat_rate	String - current value of vat rate (without "%" sign)
client_name	String
service_id	Int
email	String - client's billing email that was at the time of invoice generation
vat	Boolean
owner_id	Int

2.244 outgoing_invoice_set

Docstring:

Function changes amount protested and additional amount of chosen supplier invoice. Total invoice amount calculates by formula:

Total Amount="Initial Claimed"+"Additional Amount"- "Amount Protested".

Function returns invoice id in the DB.

Examples:

- Client Request

```
{"session_id": "6aca832a15e5cb49484434ec98903607",  
  "data": {"id": 1038,  
            "amount_additional": 66.0,  
            "session_id": "6aca832a15e5cb49484434ec98903607",  
            "status_id": 5,  
            "notes": ""}  
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "id": 1038,  
  "time_elapsed": 0.103676}
```

Request structure and fields description:

```
{'data': {'amount_additional': 'Float',  
          'id': 'Integer',  
          'notes': 'String',  
          'session_id': 'String',  
          'status_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
amount_additional	N/A	False	Float()	This value will be added to the company balance if it allowed in database. If not - only affects on total invoice amount.
session_id	N/A	True	String(allow_b	by defaults from user session

			lank: False)	
notes	N/A	True	String(allow_b lank: True)	
id	N/A	False	Integer(gt: 0)	invoice id
status_id	N/A	False	Integer(gte: 0)	new invoice status id, 0 - "In process", 1- "Accepted", 2 - "Protested", 3 - "Closed", 4 - "Deleted", 5 - "Inv. changed"

2.245 outgoing_sms_get

Docstring:

`_outgoing_sms_get_individual` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_get_by_bill_id` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_by_agg_get` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_by_cdr_get` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_get_by_cdr_by_state` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_get_mcc_mnc_cdr` ([data_validation](#)) ([default_fields](#))

`_outgoing_sms_get_mcc_mnc_agg` ([data_validation](#)) ([default_fields](#))

Request structure and fields description:

`_outgoing_sms_get_individual`

```
{'data': {'ani': 'String',
          'billed_sms_only': 'Bool',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('status', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'op_client_name', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'op_caller_number', 'clear_number', 'dialpeer', 'op_price', 'tp_mcc', 'dlr_receive_time', 'op_ip', 'dlr', 'op_cost', 'country', 'tp_price', 'op_client_id', 'op_name', 'outgoing_ani', 'hlr_type', 'nas', 'state', 'incoming_ani', 'tp_caller_number', 'tp_id', 'tp_cost', 'local_code_name', 'called_number_in', 'tp_protocol', 'op_protocol')"],
          'hosts': "['String'] or String",
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'show_stat': 'Bool',
          'sort_by': [{"Enum('status', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'op_client_name', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'op_caller_number', 'clear_number', 'dialpeer', 'op_price', 'tp_mcc', 'dlr_receive_time', 'op_ip', 'dlr', 'op_cost', 'country', 'tp_price', 'op_client_id', 'op_name', 'outgoing_ani', 'hlr_type', 'nas', 'state', 'incoming_ani', 'tp_caller_number', 'tp_id', 'tp_cost', 'local_code_name', 'called_number_in', 'tp_protocol', 'op_protocol')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
          'status': 'String',
          'time_zone': 'String',
```

```

'tp_class_id': 'Integer',
'tp_destination': 'String',
'tp_destination_exclude': 'String',
'tp_host': 'String or String',
'tp_id': 'Integer',
'type': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	True	Integer(gt: 0)	TP id
billed_sms_only	False	False	Bool	show only billed SMSs
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12\d 3[0-2]))?(0\.){3}0(0)?)\$")	Originator's IP.
sort_by	[]	True	List([Enum('status', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'op_client_name', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'op_caller_number', 'clear_number', 'dialpeer', 'op_price', 'tp_mcc', 'dlr_receive_time', 'op_ip', 'dlr', 'op_cost', 'country', 'tp_price', 'op_client_id', 'op_name', 'outgoing_ani',	field on which result will be sorted. Result can be sorted ascending or descending.

			'hlr_type', 'nas', 'state', 'incoming_ani', 'tp_caller_number', 'tp_id', 'tp_cost', 'local_code_name', 'called_number_in', 'tp_protocol', 'op_protocol': Enum('asc', 'desc', ''), min_length: 0)	
hosts	[]	True	Or(List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2})2[0-4]\d25[0-5])\.((\d [1-9]\d 1\d{2})2[0-4]\d25[0- 5])\.){2}(\d[1-9]\d 1\d{2} 2[0-4]\d25[0-5])/([1-9][12]\ d3[0-2]))?(0\.){3}0(/0)?)\$")), min_length: 0), String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\ d{2})2[0-4]\d25[0-5])\.((\d [1-9]\d 1\d{2})2[0-4]\d25[0- 5])\.){2}(\d[1-9]\d 1\d{2} 2[0-4]\d25[0-5])/([1-9][12]\ d3[0-2]))?(0\.){3}0(/0)?)\$"))	filter by RTP server IP. A few addresses need separate by comma.
fields	['status', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'op_client_name', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'op_caller_number', 'clear_number', 'dialpeer', 'op_price', 'tp_mcc', 'dlr_receive_time', 'op_ip', 'dlr', 'op_cost', 'country', 'tp_price', 'op_client_id', 'op_name', 'outgoing_ani', 'hlr_type', 'nas', 'state', 'incoming_ani', 'tp_caller_number', 'tp_id', 'tp_cost', 'local_code_name', 'called_number_in', 'tp_protocol',	True	List([Enum('status', 'tp_mnc', 'called_number_out', 'op_destination', 'tp_destination', 'tp_name', 'tp_client_id', 'op_client_name', 'tp_host', 'setup_time', 'message_id', 'op_id', 'op_mnc', 'op_mcc', 'local_code', 'tp_client_name', 'op_caller_number', 'clear_number', 'dialpeer', 'op_price', 'tp_mcc', 'dlr_receive_time', 'op_ip', 'dlr', 'op_cost', 'country', 'tp_price', 'op_client_id', 'op_name', 'outgoing_ani', 'hlr_type', 'nas', 'state', 'incoming_ani', 'tp_caller_number', 'tp_id', 'tp_cost', 'local_code_name', 'called_number_in', 'tp_protocol', 'op_protocol')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

	'op_protocol']			
tp_class_id	N/A	True	Integer(gte: 0)	Shows in report only SMS with given OP class id. Possible values: 0 - "Other successful", 1 - "Completed, unsuccessful", 2 - "Completed, ok", 3 - "Non-routable", 4 - "Other unsuccessful", 5 - "Routed, misconfiguration possible", 6 - "Routed, uncompleted", 7 - "Completed, zero", 8 - "Unrouted", 9 - "Routed wrong".
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2}\$")	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
session_id	N/A	True	String(allow_blank: False)	not in use
status	N/A	True	String(allow_blank: False)	filter by status
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns sms which setup_time not more than given.
tp_destination_exclude	N/A	True	String(allow_blank: False,	destination code(s), which should not show

			regex: "^(\d+(, \s)?)*\$")	in report, separated by comma
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
show_stat	False	True	Bool	not in use
client_id	N/A	True	Integer(gt: 0)	filter by company id.
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^([a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9][A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d{3}[0-2]))? (0\.){3}0(/0)?)\$"))	Terminator's host or IP.
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.
owner_id	0	False	Integer(gte: 0)	filter by owner
limit	N/A	False	Integer(gt: 0)	says how many rows will be in response
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number

clients_ids	[]	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids. Currently it takes into account only first id from list.
not_use_like	False	True	Bool	if "true" searches only SMSs by exact destination codes without subcodes
type	N/A	False	String(allow_blank: False, regex: "^individual\$")	statistical resolution type

_outgoing_sms_get_by_bill_id

```
{'data': {'bill_id': 'String',
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('status', 'tp_mcc', 'tp_mnc', 'clear_number', 'name', 'nas',
'tp_id', 'tp_client_id', 'destination', 'message_id', 'tp_host', 'setup_time', 'tp_price', 'outgoing_ani', 'country', 'local_code', 'tp_client_name', 'tp_cost', 'local_code_name', 'called_number_in', 'called_number_out')"],
          'nas': 'String',
          'start_date': 'String',
          'time_zone': 'String',
          'user_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns sms which setup_time not more than given.
user_id	N/A	True	Integer()	user id to filter by locked companies and show IP permissions, by defaults is taken from current

				session
bill_id	N/A	False	String(allow_blank: False)	incoming call bill id
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):(?\\d{2})\$")	"start_date" and "end_date" timezone
fields	['status', 'tp_mcc', 'tp_mnc', 'clear_number', 'name', 'nas', 'tp_id', 'tp_client_id', 'destination', 'message_id', 'tp_host', 'setup_time', 'tp_price', 'outgoing_ani', 'country', 'local_code', 'tp_client_name', 'tp_cost', 'local_code_name', 'called_number_in', 'called_number_out']	True	List([Enum('status', 'tp_mcc', 'tp_mnc', 'clear_number', 'name', 'nas', 'tp_id', 'tp_client_id', 'destination', 'message_id', 'tp_host', 'setup_time', 'tp_price', 'outgoing_ani', 'country', 'local_code', 'tp_client_name', 'tp_cost', 'local_code_name', 'called_number_in', 'called_number_out')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error
currency_id	1	True	Integer()	currency for tp price
nas	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ((\\d [1-9] 12)\\d 3[0-2]))?(0\\.){3}0(0)?)\$")	RTP server IP address
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.

_outgoing_sms_by_agg_get

```
{'data': {'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
```

```

'network_code': 'String',
'offset': 'Integer',
'op_ip': 'String',
'owner_id': 'Integer',
'session_id': 'String',
'sort_by': [{"Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
'start_date': 'String',
'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
'status': 'Integer',
'time_zone': 'String',
'tp_destination': 'String',
'tp_destination_exclude': 'String',
'tp_id': 'Integer',
'type': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.
tp_id	N/A	True	Integer(gt: 0)	TP id
hosts	[]	True	List([String(allow_blank: False, regex: "^((([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?\$")], min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
type	N/A	False	String(allow_blank: False)	method of grouping: "combine_dest", "month", "week", "day", "hour", "combine_dest_and_point", "combine_point", "combine_disconnect_code"
op_ip	N/A	True	String(allow_blank: False, regex:	IP address of sms

			"^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(\0\.){3}0(0)?)\$")	originator
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids.
client_id	N/A	True	Integer(gt: 0)	client_id. not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given. End date will be rounded up to an hour.
status	N/A	True	Integer(gte: 0)	filter by status.
owner_id	0	False	Integer(gte: 0)	filter by owner
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
fields	['tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name']	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response.

				Currently not in use.
offset	N/A	True	Integer(gte: 0)	offset says to skip that many rows before beginning to return rows. currently not in use.
sort_by	[]	False	List([{'Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')': Enum('asc', 'desc', ")"}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
time_zone	+00:00	False	String(allow_blank: False, regex: "^[+-]?\\d{1,2}:\\d{2}\$")	"start_date" and "end_date" timezone
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma.
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma.

_outgoing_sms_by_cdr_get

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
```

```

'op_ip': 'String',
'owner_id': 'Integer',
'session_id': 'String',
'sort_by': [{"Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')": "Enum('asc', 'desc', '')"}],
'start_date': 'String',
'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
'status': 'Integer',
'time_zone': 'String',
'tp_destination': 'String',
'tp_destination_exclude': 'String',
'tp_host': 'String or String',
'tp_id': 'Integer',
'type': "Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip', 'combine_latest_stat')",
'session_id': 'String'

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults session_id from current session
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns sms which setup_time not more than given.
fields	['tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country',	True	List([Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in 'default fields'. If empty - returns

	'sa', 'tp_cost', 'op_name']			error.
status	N/A	True	Integer(gte: 0)	shows in report only sms with given status.
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, ,s)?)*\$")	destination code(s), which need show in report, separated by comma.
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, ,s)?)*\$")	destination code(s), which should not show in report, separated by comma
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
client_id	N/A	True	Integer(gt: 0)	not in use
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9] 12\\d 3[0-2]))?(0\\.){3}0(/0)?)\$")	Originator's IP.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
type	N/A	True	Enum('combine_dest', 'month', 'week', 'day', 'hour', 'combine_dest_and_point', 'combine_point', 'combine_disconnect_code', 'combine_disconnect_code_destination', 'combine_point_nas_ip',	method of grouping

			'combine_latest_stat')	
owner_id	0	False	Integer(gte: 0)	filter by owner
time_zone	+00:00	True	String(allow_blank: False, regex: " <code>^[+-]?\\d{1,2}:\\d{2}\$</code> ")	<code>`start_date`</code> and <code>`end_date`</code> timezone.
hosts	[]	True	List([String(allow_blank: False, regex: " <code>^((\\d{1-9} \\d{1-9}\\d{1}\\d{2}) 2[0-4]\\d 25[0-5])\\.((\\d{1-9} \\d{1-9}\\d{1}\\d{2}) 2[0-4]\\d 25[0-5])\\.){2}(\\d{1-9} \\d{1-9}\\d{1}\\d{2}) 2[0-4]\\d 25[0-5])/(\\d{1-9} 12\\d 3[0-2]))?(0\\.){3}0(0)?\$</code> "), min_length: 0)	filter by RTP server IP. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if 'true' searches only calls by exact destination codes without subcodes
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from usersession.
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
tp_host	N/A	True	Or(String(allow_blank: False, regex: " <code>^([a-zA-Z0-9] [a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9] [A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$</code> ", max_length: 255), String(allow_blank: False, regex: " <code>^((\\d{1-9} \\d{1-9}\\d{1}\\d{2}) 2[0-</code> "))	Terminator's host or IP.

			4]\d 25[0-5])\.(\\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)}{2}(\\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/(\\d [1-9][12]\d 3[0-2]))?(0\\.){3}0(0)?)\$")	
network_code	N/A	True	String(allow_blank: False, regex: "^(\\d{1,6})?\$")	filter by network code
tp_id	N/A	True	Integer(gt: 0)	
sort_by	[]	True	List([{'Enum('tp_name', 'attempts', 'destination', 'tr_date', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'op_name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

_outgoing_sms_get_by_cdr_by_state

```
{'data': {'ani': 'String',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('state', 'share', 'attempts')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String or String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('state', 'share', 'attempts')": "Enum('asc', 'desc', '')"}]
},
          'start_date': 'String',
          'status': 'Integer',
          'time_zone': 'String',
          'tp_destination': 'String',
          'tp_destination_exclude': 'String',
          'tp_host': 'String or String',
          'tp_id': 'Integer',
          'type': "Enum('combine_by_state',)",
          'session_id': 'String'}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
tp_id	N/A	True	Integer(gt: 0)	# filter by TP id
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which need show in report, separated by comma
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9][1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9][12]\\d 3[0-2]))?(0\\.){3}0(0)?)\$")	Originator's IP.
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^(([a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9][A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])\$", max_length: 255), String(allow_blank: False, regex: "^(([1-9][1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9][12]\\d 3[0-2]))?(0\\.){3}0(0)?)\$"))	Terminator's host or IP.

hosts	[]	True	List([String(allow_blank: False, regex: "^([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?)?(0\.){3}0(/)?\$"), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if "true" searches only calls by exact destination codes without subcodes
type	N/A	False	Enum('combine_by_state',)	method of grouping
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):(\d{2})\$")	"start_date" and "end_date" timezone
ani	N/A	True	String(allow_blank: False, regex: "^[a-zA-Z0-9#*]*\$")	filter by caller number or first digits of caller number
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id.
owner_id	0	True	Integer(gte: 0)	filter by owner id. 0 means adsence of filtration.
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
status	N/A	True	Integer(gte: 0)	filter by SMS status
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
fields	['state', 'share', 'attempts']	True	List([Enum('state', 'share', 'attempts')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response.

				Currently not in use.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
sort_by	[]	True	List([{Enum('state', 'share', 'attempts'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
network_code	""	True	Or(String(allow_blank: False, regex: "[0-9]*\$", min_length: 3, max_length: 6), String(allow_blank: True, max_length: 0))	filter by message MCC MCN.

_outgoing_sms_get_mcc_mnc_cdr

```
{'data': {'ani': 'String',
          'client_id': 'Integer',
          'clients_ids': ['Integer'],
          'currency_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name')"],
          'hosts': ['String'],
          'limit': 'Integer',
          'managers': ['Integer'],
          'message_id': 'String or Null',
          'network_code': 'String or String',
          'not_use_like': 'Bool',
          'offset': 'Integer',
          'op_ip': 'String',
          'owner_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
          'status': 'Integer',
          'time_zone': 'String',
          'tp_destination': 'String',
          'tp_destination_exclude': 'String',
          'tp_host': 'String or String',
          'tp_id': 'Integer',
          'type': "Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')"},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults session_id from current session
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns sms which setup_time not more than given.
tp_id	N/A	True	Integer(gt: 0)	
tp_destination	N/A	True	String(allow_blank: False, regex: "^(\\d+ (, \\s)?)*\$")	destination code(s), which need show in report, separated by comma.
type	N/A	True	Enum('combine_mcc_mnc', 'combine_mcc_mnc_and_point')	method of grouping
op_ip	N/A	True	String(allow_blank: False, regex: "^(\\[[1-9]\\[[1-9]\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5]\\]\\.(\\d{1-9}\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5])\\.)\\{2\\}(\\d{1-9}\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5])/(\\[[1-9]\\[[12]\\d{3}[0-2]\\])?(0\\.){3}0(0)?\$")	Originator's IP.
tp_host	N/A	True	Or(String(allow_blank: False, regex: "^(\\[[a-zA-Z0-9]\\[[a-zA-Z0-9]\\[[a-zA-Z0-9\\-]*[a-zA-Z0-9]\\]\\.)\\{2\\}([A-Za-z0-9]\\[[A-Za-z0-9\\-]*[A-Za-z0-9]\\])\$"), String(allow_blank: False, regex: "^(\\[[1-9]\\[[1-9]\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5]\\]\\.(\\d{1-9}\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5])\\.)\\{2\\}(\\d{1-9}\\d{1}\\d{2}\\d{2}[0-4]\\d{25}[0-5])/(\\[[1-9]\\[[12]\\d{3}[0-2]\\])?(0\\.){3}0(0)?\$")	Terminator's host or IP.

			0)?)\$"))	
hosts	[]	True	List(String(allow_blank: False, regex: "^[([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d 3[0-2]))?(0\.){3}0(0)?)\$"), min_length: 0)	filter by RTP server IP. A few addresses need separate by comma.
not_use_like	N/A	True	Bool	if 'true' searches only calls by exact destination codes without subcodes
managers	N/A	True	List(Integer(gt: 0), min_length: 0)	filter by manger id
status	N/A	True	Integer(gte: 0)	shows in report only sms with given status.
owner_id	0	False	Integer(gte: 0)	filter by owner
time_zone	+00:00	True	String(allow_blank: False, regex: "^[([+-]?[d{1,2}):(\d{2})\$")	`start_date` and `end_date` timezone.
ani	N/A	True	String(allow_blank: True)	filter by caller number or first digits of caller number
message_id	None	True	Or(String(allow_blank: False), Null)	filter by message id
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from usersession.
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
clients_ids	N/A	True	List(Integer(gt: 0), min_length: 0)	filter by company ids
fields	['tp_mcc',	True	List(Enum('tp_mcc', 'attempts', 'tp_mnc',	required fields. If

	'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name']		'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name']), min_length: 0)	absent - takes fields specified in 'default fields'. If empty - returns error.
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+ (, \\s)?)*\$")	destination code(s), which should not show in report, separated by comma
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows. Currently not in use.
sort_by	[]	True	List([{Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name'): Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
client_id	N/A	True	Integer(gt: 0)	not in use
network_code	""	True	Or(String(allow_blank: False, regex: "^[0-9]*\$", min_length: 3, max_length: 6), String(allow_blank: True, max_length: 0))	filter by message MCC MCN. Works only with type 'combine_mcc_mnc' ' or 'combine_mcc_mnc _and_point'.

_outgoing_sms_get_mcc_mnc_agg

```
{'data': {'client_id': 'Integer',  
          'clients_ids': ['Integer'],  
          'currency_id': 'Integer',  
          'end_date': 'String',
```

```

'fields': ["Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name')"],
'hosts': ['String'],
'limit': 'Integer',
'managers': ['Integer'],
'network_code': 'String or String',
'offset': 'Integer',
'op_ip': 'String',
'owner_id': 'Integer',
'session_id': 'String',
'sort_by': [{"Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name')": "Enum('asc', 'desc', '')"}],
'start_date': 'String',
'state': "Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')",
'status': 'Integer',
'time_zone': 'String',
'tp_destination': 'String',
'tp_destination_exclude': 'String',
'tp_id': 'Integer',
'type': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns sms which setup_time not more than given.
session_id	N/A	True	String(allow_blank: False)	not in use, by defaults id from session
tp_id	N/A	True	Integer(gt: 0)	TP id
tp_destination	N/A	True	String(allow_blank: False, regex: "^(d+(, ,\\s)?)*\$")	destination code(s), which need show in report, separated by comma.
tp_destination_exclude	N/A	True	String(allow_blank: False, regex: "^(d+(, ,\\s)?)*\$")	destination code(s), which should not show in report, separated by comma.

op_ip	N/A	True	String(allow_blank: False, regex: <code>"^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]))/([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?\$"</code>)	IP address of sms originator
hosts	[]	True	List([String(allow_blank: False, regex: <code>"^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]))/([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?\$"</code>), min_length: 0)	filter by RTP servers IP address. A few addresses need separate by comma.
type	N/A	False	String(allow_blank: False)	method of grouping: "combine_mcc_mnc", "combine_mcc_mnc_and_point"
time_zone	+00:00	False	String(allow_blank: False, regex: <code>"^([+-]?d{1,2}):(\d{2})\$"</code>)	"start_date" and "end_date" timezone
currency_id	N/A	True	Integer(gt: 0)	shows prices and costs in report in specific currency. By defaults - from user session.
clients_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by company ids.
client_id	N/A	True	Integer(gt: 0)	client_id. not in use.
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id.
status	N/A	True	Integer(gte: 0)	filter by status.
owner_id	0	False	Integer(gte: 0)	filter by owner
state	N/A	True	Enum('ACCEPTD', 'DELIVRD', 'DELETED', 'EXPIRED', 'REJECTD', 'UNDELIV', 'UNKNOWN', 'ENROUTE')	Filter by DLR state.
fields	['tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country',	True	List([Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty -

	'sa', 'tp_cost', 'tp_name', 'op_name']			returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response. Currently not in use.
offset	N/A	True	Integer(gte: 0)	offset says to skip that many rows before beginning to return rows. currently not in use.
sort_by	[]	False	List([{Enum('tp_mcc', 'attempts', 'tp_mnc', 'delta_price', 'op_cost', 'country', 'sa', 'tp_cost', 'tp_name', 'op_name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
network_code	""	True	Or(String(allow_blank: False, regex: "^([0-9]*)\$", min_length: 3, max_length: 6), String(allow_blank: True, max_length: 0))	filter by message MCC MCN. Works only with type 'combine_mcc_mnc' or 'combine_mcc_mnc_and_point'.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns sms which setup_time not less than given.

Default fields:

_outgoing_sms_get_individual

Name	Type & Comments
status	Int
tp_mnc	String(3)
called_number_ou	String

t	
op_destination	String
tp_destination	String
tp_name	String
tp_client_id	Int
op_client_name	String
tp_host	String
setup_time	String
message_id	String
op_id	Int
op_mnc	String(3)
op_mcc	String(3)
local_code	String
tp_client_name	String
op_caller_number	String(200)
clear_number	String
dialpeer	String
op_price	numeric(16,5)
tp_mcc	String(3)
dlr_receive_time	String
op_ip	String
dlr	Int

op_cost	numeric(16,5)
country	String
tp_price	numeric(16,5)
op_client_id	Int
op_name	String
outgoing_ani	String
hlr_type	String ('no', 'external', 'local')
nas	String (RTP IP address)
state	String
incoming_ani	String
tp_caller_number	String(200)
tp_id	Int
tp_cost	numeric(16,5)
local_code_name	String
called_number_in	String
tp_protocol	String(100)
op_protocol	String(100)

_outgoing_sms_get_by_bill_id

Name	Type & Comments
------	-----------------

status	String
tp_mcc	String(3)
tp_mnc	String(3)
clear_number	String(200)
name	String(64) (TP name)
nas	String (RTP IP address)
tp_id	Int
tp_client_id	Int
destination	String(24)
message_id	String(65)
tp_host	String
setup_time	String
tp_price	numeric(16,5)
outgoing_ani	String(200)
country	String(64)
local_code	String
tp_client_name	String(64)
tp_cost	numeric(16,5)
local_code_name	String
called_number_in	String(200)
called_number_out	String(200)

_outgoing_sms_by_agg_get

Name	Type & Comments
tp_name	String
attempts	Int
destination	String
tr_date	String
delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

_outgoing_sms_by_cdr_get

Name	Type & Comments
tp_name	String
attempts	Int
destination	String
tr_date	String
delta_price	numeric(16,5)
op_cost	numeric

country	String
sa	numeric - number of successful attempts
tp_cost	numeric
op_name	String

_outgoing_sms_get_by_cdr_by_state

Name	Type & Comments
state	String
share	Numeric(6,2)
attempts	Int

_outgoing_sms_get_mcc_mnc_cdr

Name	Type & Comments
tp_mcc	String(3)
attempts	Int
tp_mnc	String(3)
delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts

tp_cost	numeric
tp_name	String
op_name	String

`_outgoing_sms_get_mcc_mnc_agg`

Name	Type & Comments
tp_mcc	String(3)
attempts	Int
tp_mnc	String(3)
delta_price	numeric(16,5)
op_cost	numeric
country	String
sa	numeric - number of successful attempts
tp_cost	numeric
tp_name	String
op_name	String

2.246 owner_add

Docstring:

Creates new owner with given parameters. Currently NOT IN USE!

Function returns id of new owner(key "id").

Request structure and fields description:

```
{'data': {'additional_info': 'String',
          'address': 'String',
          'bank_property': 'String',
          'email': 'String',
          'invoice_number_rule_cust': 'String',
          'invoice_number_rule_supl': 'String',
          'name_company': 'String',
          'pdf_type': 'Integer',
          'phone_number': 'String',
          'show_invoice_below_attributes': 'Bool',
          'show_invoice_number': 'Bool',
          'show_invoice_vat': 'Bool',
          'show_invoices_by_region': 'Bool',
          'show_personal_invoice_sms': 'Bool',
          'use_xls_invoice': 'Bool',
          'xls_invoice_pass': 'String',
          'xls_invoice_user': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
xls_invoice_user	N/A	True	String(allow_blank: True)	User name for xls invoices. Necessary if you want to protect invoices editing by customers
additional_info	N/A	True	String(allow_blank: True)	
use_xls_invoice	N/A	True	Bool	if true - invoices generates in .xls format
phone_number	N/A	True	String(allow_blank: True)	
invoice_number_rule_supl	N/A	True	String(allow_blank: True)	pattern for invoice number. Possible rules: "000000.id.i", "000000.id.yyyyMMdd.i", "id", where id - invoice id in database
xls_invoice_pass	N/A	True	String(allow_blank: True)	password for xls invoice user.

show_invoices_by_region	N/A	True	Bool	
name_company	N/A	False	String(allow_blank: True)	owner's company name
show_invoice_number	N/A	True	Bool	If 'True' shows invoice number in invoices.
show_invoice_below_attributes	N/A	True	Bool	if 'True', shows owner bank details in invoice
use_personal_invoice_sms	N/A	True	Bool	if true - SMS invoices generates in custom type in .pdf
address	N/A	True	String(allow_blank: True)	owner's address
show_invoice_vat	N/A	True	Bool	if true, total cost amount with VAT and VAT rate shows in invoices
bank_property	N/A	True	String(allow_blank: True)	owner's bank credentials
email	N/A	True	String(allow_blank: True)	
pdf_type	N/A	True	Integer()	pdf invoice type: '1','2' or '3'
invoice_number_rule_cust	N/A	True	String(allow_blank: True)	pattern for invoice number. Possible rules: "000000.id.i", "000000.id.yyyyMMdd.i", "id", where id - invoice id in database

2.247 owner_del

Docstring:

NOT IN USE at the moment!

Removes owner with given id. You should first remove all bindings with

companies before proceed to removing.

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer()	

2.248 owner_info_by_invoice_get

Docstring:

Invoice id or owner id is required.
Invoice id is 0 and owner id not 0 - data for presets.
Invoice id is not 0(owner id is absent) - data for invoice generating.

Example:

- Client Request

```
{"session_id": "8467a6682be3bdbca3785eb14039ab1b",  
  "data": {"invoice_id": 2,  
           "fields": ["email", "name_company", "vat_rate"]  
          }  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["email", "name_company", "vat_rate"],  
  "message": "OK",  
  "data": [["test1@test.com", "MC v3.1 Test", 10.0]],  
  "time_elapsed": 0.030999}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('phone_number', 'bank_property', 'show_invoice_below_attributes', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_p
```

```

ass', 'show_invoice_number', 'email', 'pdf_type', 'xls_invoice_user', 'name_company', '
vat_rate', 'address', 'use_xls_invoice_sms', 'additional_info')"],
    'invoice_id': 'Integer',
    'owner_id': 'Integer'},
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['phone_number', 'bank_property', 'show_invoice_below_attributes', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_pass', 'show_invoice_number', 'email', 'pdf_type', 'xls_invoice_user', 'name_company', 'vat_rate', 'address', 'use_xls_invoice_sms', 'additional_info']	True	List([Enum('phone_number', 'bank_property', 'show_invoice_below_attributes', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_pass', 'show_invoice_number', 'email', 'pdf_type', 'xls_invoice_user', 'name_company', 'vat_rate', 'address', 'use_xls_invoice_sms', 'additional_info']), min_length: 0)	required fields.If absent - takes fields specified in"default fields". If empty - returns error.
owner_id	N/A	True	Integer(gt: 0)	
invoice_id	N/A	True	Integer(gte: 0)	invoice id. can be 0 for presets

Default fields:

Name	Type & Comments
phone_number	String
bank_property	String
show_invoice_below_attributes	Boolean
use_personal_invoice_sms	Boolean

show_invoice_vat	Boolean
use_xls_invoice	Boolean
xls_invoice_pass	String
show_invoice_number	Boolean
email	String(100)
pdf_type	Int
xls_invoice_user	String
name_company	String(1000)
vat_rate	numeric
address	String
use_xls_invoice_sms	Boolean
additional_info	String

2.249 owner_info_get

Docstring:

Returns information about owner of the system and currently specified system parameters.

Example:

- Client Request

```
{
  "session_id": "832a3d94029e3c1f64f5bc569e1b42c4",
  "data": {
    "fields": [
      "currency_id", "email", "name_company",
      "vat_rate", "web_url", "business_id",
      "alarm_not_allow_ip", "password_expire_term",
      "user_expire_term", "use_password_expire_term",
      "use_user_expire_term", "use_invoice_cut"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
}
```

```

"fields": ["currency_id", "email", "name_company", "vat_rate",
           "web_url", "business_id", "alarm_not_allow_ip",
           "password_expire_term", "user_expire_term",
           "use_password_expire_term", "use_user_expire_term",
           "use_invoice_cut"],
"message": "OK",
"data": [[1, "test@test.com", "MC v3.1 Test", 11.0,
           "http://test.telecom.com", "test1@test.com", true, 60,
           60, false, false, false]],
"time_elapsed": 0.003558}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('business_id', 'block_spam_ip', 'currency_id', 'web_url', 'u
se_invoice_cut', 'password_expire_term', 'alarm_not_allow_ip', 'use_password_expire_ter
m', 'user_expire_term', 'name_company', 'vat_rate', 'use_user_expire_term', 'email')"],
          'invoice_id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optiona l	Type	Description
invoice_id	N/A	True	Integer(gt: 0)	
fields	['business_id', 'block_spam_ip', 'currency_id', 'web_url', 'use_invoice_cut', 'password_expire_term', 'alarm_not_allow_ip', 'use_password_expire_term', 'user_expire_term', 'name_company', 'vat_rate', 'use_user_expire_term', 'email']	True	List([Enum('business_id', 'block_spam_ip', 'currency_id', 'web_url', 'use_invoice_cut', 'password_expire_term', 'alarm_not_allow_ip', 'use_password_expire_term', 'user_expire_term', 'name_company', 'vat_rate', 'use_user_expire_term', 'email')], min_length: 0)	required fields.If absent - takes fields specified in "default fields".If empty - returns error.

Default fields:

Name	Type & Comments
business_id	String(1000)

block_spam_ip	Boolean
currency_id	Int
web_url	String(DEFAULT 'http://localhost/')
use_invoice_cut	Boolean
password_expire_term	Int
alarm_not_allow_ip	Boolean
use_password_expire_term	Boolean
user_expire_term	Int
name_company	String(1000)
vat_rate	numeric
use_user_expire_term	Boolean
email	String(100)

2.250 owner_info_set

Docstring:

Changes system owner info and owner parameters.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "business_id": "",
    "currency_id": 1,
    "web_url": "http://test.telecom.com",
    "use_invoice_cut": False,
    "password_expire_term": 60,
    "alarm_not_allow_ip": True,
    "use_password_expire_term": False,
    "user_expire_term": 1,
    "name_company": "MC v3.1 Test"
  }
}
```

```

        "vat_rate":11.0,
        "use_user_expire_term":False,
        "email":"'test@test.com',}
    }

```

- Server Response

```

{"status": 200, "message": "OK", "time_elapsed": 0.075261}

```

Request structure and fields description:

```

{'data': {'alarm_not_allow_ip': 'Bool',
          'block_spam_ip': 'Bool',
          'business_id': 'String',
          'currency_id': 'Integer',
          'email': 'String',
          'invoice_number_rule': 'String',
          'name_company': 'String',
          'password_expire_term': 'Integer',
          'use_invoice_cut': 'Bool',
          'use_password_expire_term': 'Bool',
          'use_user_expire_term': 'Bool',
          'user_expire_term': 'Integer',
          'web_url': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
business_id	N/A	True	String(allow_blank: True)	paypal id
alarm_not_allow_ip	N/A	True	Bool	
block_spam_ip	N/A	True	Bool	if True - enable block spam IPs
password_expire_term	N/A	True	Integer()	password expire term in days.
name_company	N/A	True	String(allow_blank: True)	owner's company name

user_expire_term	N/A	True	Integer()	user expire term in days.
email	N/A	True	String(allow_blank: True)	main email
web_url	N/A	True	String(allow_blank: False, regex: "^(?:http ftp)s?://(?:((?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\\. [A-Z0-9-]{2,}\\. localhost \\d{1,3}\\. \\d{1,3}\\. \\d{1,3}\\. \\d{1,3})((?:\\d+)?(?:/? [/?]\\S+)\$)")	billing URL which shows in emails from billing
currency_id	N/A	True	Integer()	default system currency
use_password_expire_term	N/A	True	Bool	if True - password_expire_term will be used in login
use_user_expire_term	N/A	True	Bool	if True - user_expire_term will be used in login
invoice_number_rule	N/A	True	String(allow_blank: True)	pattern for invoice number. Possible rules: "000000.id.i", "000000.id.yyyyMMdd.i", "id", where id - invoice id in database
use_invoice_cut	N/A	True	Bool	if True - invoice will be divided by months

2.251 owner_set

Docstring:

Allows to change parameters of the given owner. Function returns owner id(key "id").

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "show_invoice_vat": true,
    "phone_number": "",
    "additional_info": "",
    "invoice_number_rule_cust": "000000.id.yyyyMMdd.c",
  }
}
```

```

        "show_invoice_below_attributes":true,
        "use_xls_invoice":false,
        "id":3,
        "address":"","
        "xls_invoice_pass":"","
        "bank_property":"Vabank",
        "name_company":"Lemgrove International, S.A",
        "invoice_number_rule_supl":"000000.id.yyyyMMdd.s",
        "xls_invoice_user":"","
        "show_invoice_number":true,
        "pdf_type":1}
    }

```

- Server Response

```

{"status": 200, "message": "OK", "id": 3, "time_elapsed": 0.069904}

```

Request structure and fields description:

```

{'data': {'additional_info': 'String',
          'address': 'String',
          'bank_property': 'String',
          'email': 'String',
          'id': 'Integer',
          'invoice_number_rule_cust': 'String',
          'invoice_number_rule_supl': 'String',
          'name_company': 'String',
          'pdf_type': 'Integer',
          'phone_number': 'String',
          'show_invoice_below_attributes': 'Bool',
          'show_invoice_number': 'Bool',
          'show_invoice_vat': 'Bool',
          'show_invoices_by_region': 'Bool',
          'use_personal_invoice_sms': 'Bool',
          'use_xls_invoice': 'Bool',
          'use_xls_invoice_sms': 'Bool',
          'xls_invoice_pass': 'String',
          'xls_invoice_user': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
use_personal_invoice_sms	N/A	True	Bool	if true - SMS invoices generates in custom type in .pdf

xls_invoice_user	N/A	True	String(allow_blank: True)	User name for xls invoices. Necessary if you want to protect invoices editing by customers
use_xls_invoice	N/A	True	Bool	if true - Voice invoices generates in .xls format
name_company	N/A	True	String(allow_blank: True)	owner's company name
phone_number	N/A	True	String(allow_blank: True)	
additional_info	N/A	True	String(allow_blank: True)	
address	N/A	True	String(allow_blank: True)	owner's address
use_xls_invoice_sms	N/A	True	Bool	if true - SMS invoices generates in .xls format else in .pdf
xls_invoice_pass	N/A	True	String(allow_blank: True)	password for xls invoice user.
id	N/A	False	Integer()	owner id
show_invoice_number	N/A	True	Bool	if true, in invoice shows invoice number
email	N/A	True	String(allow_blank: True)	
bank_property	N/A	True	String(allow_blank: True)	bank information of the owner company
show_invoice_vat	N/A	True	Bool	if true, total cost amount with VAT and VAT rate shows in invoices
pdf_type	N/A	True	Integer()	pdf invoice type: 1,2 or 3

show_invoice_below_attributes	N/A	True	Bool	if "true" owner bank details shows in invoice
invoice_number_rule_cust	N/A	True	String(allow_blank: True)	pattern for invoice number. Possible rules: "000000.id.i", "000000.id.yyyyMMdd.i", "id", where id - invoice id in database
show_invoices_by_region	N/A	True	Bool	
invoice_number_rule_supl	N/A	True	String(allow_blank: True)	pattern for invoice number. Possible rules: "000000.id.i", "000000.id.yyyyMMdd.i", "id", where id - invoice id in database

2.252 owners_get

Docstring:

Returns information about owner(s) of the system and owner parameters

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "id": 3,
    "fields": [
      "additional_info",
      "address",
      "bank_property",
      "invoice_number_rule_supl",
      "invoice_number_rule_cust",
      "name_company",
      "phone_number",
      "xls_invoice_pass",
      "xls_invoice_user",
      "show_invoice_number",
      "show_invoice_below_attributes",
      "pdf_type",
      "show_invoice_vat",
      "use_xls_invoice"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "additional_info",
    "address",
    "bank_property",
    "invoice_number_rule_supl",
    "invoice_number_rule_cust",
    "name_company",
    "phone_number",
    "xls_invoice_pass",
    "xls_invoice_user",
    "show_invoice_number",
    "show_invoice_below_attributes",
    "pdf_type",
    "show_invoice_vat",
    "use_xls_invoice"
  ],
  "message": "OK",
  "data": [
    [
      "",
      "",
      "",
      "",
      "",
      "Lemgrove International, S.A",
      "",
      "",
      "",
      true,
      true,
      1,
      true,
      false
    ]
  ],
  "time_elapsed": 0.005672
}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'client_id': 'Integer',
          'fields': ['String'],
          'id': 'Integer',
          'limit': 'Integer',
          'manager_id': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('phone_number', 'show_invoice_number', 'bank_property', 'vat_rate', 'show_invoices_by_region', 'pdf_type', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_pass', 'invoice_number_rule_cust', 'additional_info', 'email', 'xls_invoice_user', 'name_company', 'show_invoice_below_attributes', 'address', 'use_xls_invoice_sms', 'invoice_number_rule_supl', 'id')": "Enum('asc', 'desc', '')"}]},
          'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	True	Integer(gt: 0)	Balance id. Used if "id" of owner is not specified.
fields	['phone_number', 'show_invoice_number', 'bank_property', 'vat_rate', 'show_invoices_by_region', 'pdf_type', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_pass', 'invoice_number_rule_cust', 'additional_info', 'email', 'xls_invoice_user', 'name_company', 'show_invoice_below_attributes', 'address', 'use_xls_invoice_sms', 'invoice_number_rule_supl', 'id']	True	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
client_id	N/A	True	Integer(gt: 0)	filter by client

sort_by	[]	True	List([{'phone_number', 'show_invoice_number', 'bank_property', 'vat_rate', 'show_invoices_by_region', 'pdf_type', 'use_personal_invoice_sms', 'show_invoice_vat', 'use_xls_invoice', 'xls_invoice_pass', 'invoice_number_rule_cust', 'additional_info', 'email', 'xls_invoice_user', 'name_company', 'show_invoice_below_attributes', 'address', 'use_xls_invoice_sms', 'invoice_number_rule_supl', 'id'}): Enum('asc', 'desc', ''), min_length: 0)	field on which result will be sorted
id	N/A	True	Integer()	owner id
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
manager_id	N/A	True	Integer(gt: 0)	filter by manager
limit	N/A	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
phone_number	String(100)
show_invoice_number	Boolean
bank_property	String
vat_rate	numeric

show_invoices_by_region	Boolean
pdf_type	Int (Default 1)
use_personal_invoice_sms	Boolean - tells to use custom type for PDF invoice
show_invoice_vat	Boolean
use_xls_invoice	Boolean - tells in which format(xls or pdf) generate Voice invoices
xls_invoice_pass	String(100)
invoice_number_rule_cust	String(100)
additional_info	String
email	String(100)
xls_invoice_user	String(100)
name_company	String(1000)
show_invoice_below_attributes	Boolean
address	String(1000)
use_xls_invoice_sms	Boolean - the same as previous but for SMS service invoices
invoice_number_rule_supl	String(100)
id	Int

2.253 payment_add

Docstring:

Makes new payment to chosen balance. Payment amount may be positive or negative.

Negative payments will decrease balance.

Example:

```
- Client Request
{"session_id":"f345e24573e77cadffbc28a11ae64af4",
 "data":{"amount_paid":1.0,
        "direction":"outgoing",
        "balance_id":1,
        "notes":"test",
        "date":"2014-01-13T00:00:00","type_id":1}
}
```

Request structure and fields description:

```
{'data': {'amount_paid_vat': 'Float',
          'balance_id': 'Integer',
          'date': 'String',
          'direction': 'String',
          'notes': 'String',
          'session_id': 'String',
          'type_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
type_id	N/A	False	Integer(gt: 0)	payment type id which returns function payment_type_get
balance_id	N/A	False	Integer(gt: 0)	
notes	N/A	False	String(allow_blank: False)	any note, obligatory should be present in the request
direction	N/A	False	String(allow_blank: False, regex: "^(incoming outgoing)\$")	field describes 'direction' of payment: 'inc' means incoming, 'out' - outgoing and 'undefined'(can't be set from web)
session_id	N/A	True	String(allow_blank: False)	By default from user's session
date	N/A	False	String(allow_blank: False)	payment date, obligatory should be present in the request
amount_pai	N/A	False	Float()	positive or negative value with VAT

d_vat				
-------	--	--	--	--

2.254 payment_calendar_get

Docstring:

Returns payment calendar data.
Fields which are necessary: "balance_id", "amount_due", "due_date", "count_invoice", "balance", "period", "soa_from".

Request structure and fields description:

2.255 payment_del

Docstring:

Deletes chosen payment. Deleted payment log will be saved in payments history in database.

If call to this function initiate deleting of payment for amount(absolute value) more than 1 (\$ or some other currency) then it sends message to the mediacore owner (email template - 'email_payment_del').

Example:

- Client Request

```
{"session_id":"f345e24573e77cadffbc28a11ae64af4",
  "data":{"id":12,"notes":"del"}}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.060899}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'notes': 'String', 'session_id': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By defaults from user's session
id	N/A	False	Integer(gt: 0)	payment id
notes	N/A	False	String(allow_blank: False)	any note, obligatory should be present in the request

2.256 payment_get

Docstring:

Returns payments history with information about them. If payment was changed displays changes log like function "payment_history_get".

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {"limit": 11,
    "date_from": "2014-01-01T00:00:00",
    "sort_by": [{"client_name": ""}],
    "date_to": "2014-01-13T00:00:00",
    "offset": 0,
    "fields": ["id", "amount_paid", "client_id", "client_name",
      "curr_balance", "last_payment_date", "notes",
      "payment_date", "balance_id", "balance_name",
      "currency", "type_id", "type_name", "user_name",
      "status", "direction"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "amount_paid", "client_id", "client_name",
    "curr_balance", "last_payment_date", "notes",
    "payment_date", "balance_id", "balance_name",
    "currency", "type_id", "type_name", "user_name",
    "status", "direction"],
  "message": "OK",
  "data": [[1, "-169.12000", 1, "Alexander_test_client", "0.00000",
    "2014-01-11T01:00:02.660744+00:00", "fdfgdg",
    "2014-01-11T00:00:00+00:00", 1, "Test_balance",
    "USD", 6, "external", "dev(dev)", "added", "incoming"],
    [5, "200.00000", 1, "Test_client", "-169.12000",
    "2014-01-11T12:11:01.623916+00:00", "test",
    "2014-01-11T00:00:00+00:00", 1, "Alexander_test_balance",
```

```

        "USD", 1, "payment for traffic", "dev(dev)", "added", "incoming"
    ],
    [6, "-200.00000", 1, "Test_client", "30.88000",
      "2014-01-11T12:12:35.175940+00:00", "test2",
      "2014-01-11T00:00:00+00:00", 1, "Test_balance",
      "USD", 1, "payment for traffic", "dev(dev)", "added", "incoming"
    ],
    [12, "1.00000", 1, "Test_client", "-169.12000",
      "2014-01-13T10:52:19.392418+00:00", "test",
      "2014-01-13T00:00:00+00:00", 1, "Test_balance",
      "USD", 1, "payment for traffic", "dev(dev)", "added", "outgoing"
    ],
    [7, "100.00000", 8, "Test1", "0.00000",
      "2014-01-11T19:11:26.138797+00:00", "test",
      "2014-01-11T00:00:00+00:00", 8, "Test1_balance", "USD",
      6, "external", "dev(dev)", "added", "outgoing"]],
    "time_elapsed": 0.010093}

```

Request structure and fields description:

```

{'data': {'balance_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'direction': 'String',
          'fields': ["Enum('status', 'payment_date', 'direction', 'last_payment_date',
'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance',
'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')"],
          'id': 'Integer',
          'limit': 'Integer',
          'offset': 'Integer',
          'owner_id': 'Integer or Null',
          'session_id': 'String',
          'show_deleted': 'Bool',
          'sort_by': [{"Enum('status', 'payment_date', 'direction', 'last_payment_date',
'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance',
'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')"
: "Enum('asc', 'desc', '')"}],
          'type_id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By defaults from user's session
balance_id	N/A	True	Integer(gt: 0)	

owner_id	None	True	Or(Integer(gt: 0), Null)	filter by owner id
date_from	N/A	True	String(allow_blank: False)	lower boundary of request time slice
id	N/A	True	Integer(gt: 0)	payment id in the database
date_to	N/A	True	String(allow_blank: False)	upper boundary of request time slice. Combine inputted date with 23:59:59 time
show_deleted	N/A	True	Bool	if "true" displays payments which had been deleted. All payments logs are store in the database table payments_history.
sort_by	['client_name': '']	True	List([Enum('status', 'payment_date', 'direction', 'last_payment_date', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id'): Enum('asc', 'desc', '')), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
fields	['status', 'payment_date', 'direction', 'last_payment_date', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id']	True	List([Enum('status', 'payment_date', 'direction', 'last_payment_date', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
type_id	N/A	True	Integer(gt: 0)	payment type id. Function payment_type_get returns accordance of type_id and type

				name.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
direction	all	True	String(allow_blank: False, regex: "^(incoming outgoing all)\$")	incoming payments/outgoing payments/all payments. by default - all

Default fields:

Name	Type & Comments
status	String
payment_date	String
direction	String
last_payment_date	String
client_id	Int
type_id	Int
amount_paid	numeric(16,5)
type_name	String
amount_paid_vat	Numeric(16,5)
curr_balance	numeric(16,5)
currency	String

balance_name	String
notes	String
balance_id	Int
client_name	String
user_name	String
id	Int

2.257 payment_history_get

Docstring:

Returns payments changes log for payments by given balance and time interval or changes log of chosen payment specified by id.

Example:

- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {
    "id": 13,
    "fields": [
      "id", "amount_paid", "client_id", "client_name",
      "curr_balance", "last_payment_date", "notes",
      "payment_date", "balance_id", "balance_name",
      "type_id", "type_name", "user_name", "status"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "id", "amount_paid", "client_id", "client_name",
    "curr_balance", "last_payment_date", "notes",
    "payment_date", "balance_id", "balance_name",
    "type_id", "type_name", "user_name", "status"
  ],
  "message": "OK",
  "data": [
    [
      8, "1.00000", 1, "Test_client", "-169.12000",
      "2014-01-13T11:36:17.953745+00:00", "0",
      "2014-01-13T00:00:00+00:00", 1, "Test_balance",
      1, "payment for traffic", "dev(dev)", "added"
    ],
    [
      9, "2.00000", 1, "Test_client", "-169.12000",
      "2014-01-13T11:36:30.864907+00:00", "1",
      "2014-01-13T00:00:00+00:00", 1, "Test_balance",
      1, "payment for traffic", "dev(dev)", "modified"
    ]
  ],
  "time_elapsed": 0.008908
}
```


Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'client_id': 'Integer',
          'end_date': 'String',
          'fields': ["Enum('status', 'payment_date', 'last_payment_date', 'user_id', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')"],
          'id': 'Integer',
          'limit': 'Integer',
          'offset': 'Integer',
          'payment_date': 'String',
          'show_deleted': 'Bool',
          'sort_by': [{"Enum('status', 'payment_date', 'last_payment_date', 'user_id', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'type_id': 'Integer'},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	True	Integer(gt: 0)	if specified, function returns changes log of chosen payment
balance_id	N/A	True	Integer(gt: 0)	
sort_by	[]	True	List([{Enum('status', 'payment_date', 'last_payment_date', 'user_id', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
type_id	N/A	True	Integer(gt: 0)	payment type id. Function payment_type_get

				returns accordance of type_id and type name.
fields	['status', 'payment_date', 'last_payment_date', 'user_id', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id']	True	List([Enum('status', 'payment_date', 'last_payment_date', 'user_id', 'client_id', 'type_id', 'amount_paid', 'type_name', 'amount_paid_vat', 'curr_balance', 'currency', 'balance_name', 'notes', 'balance_id', 'client_name', 'user_name', 'id')], min_length: 0)	required fields. if absent - takes fields specified in "default fields". if empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
end_date	N/A	True	String(allow_blank: False)	
payment_date	N/A	True	String(allow_blank: False)	
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
start_date	N/A	True	String(allow_blank: False)	
show_deleted	N/A	True	Bool	if "true" displays deleted payments
client_id	N/A	True	Integer(gt: 0)	company id

Default fields:

Name	Type & Comments
status	String(10)
payment_date	String

last_payment_date	String
user_id	Int
client_id	Int
type_id	Int
amount_paid	numeric(16,5)
type_name	String
amount_paid_vat	numeric(16,5)
curr_balance	numeric(16,5)
currency	String
balance_name	String
notes	String
balance_id	Int
client_name	String
user_name	String
id	Int

2.258 payment_set

Docstring:

Changes existed payment. Negative payment amount will decrease balance.

Also function creates new email notification (template - 'email_payment_change') which will be sent to the owner's email address.

Example:
- Client Request

```
{
  "session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {
    "id": 12,
    "amount_paid": 2.0,
    "direction": "outgoing",
    "balance_id": 1,
    "notes": "test1",
    "date": "2014-01-13T00:00:00",
    "type_id": 1
  }
}
```

- Server Response:

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.063514
}
```

Request structure and fields description:

```
{
  'data': {
    'amount_paid_vat': 'Float',
    'balance_id': 'Integer',
    'date': 'String',
    'direction': 'String',
    'id': 'Integer',
    'notes': 'String',
    'session_id': 'String',
    'type_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
balance_id	N/A	False	Integer(gt: 0)	
id	N/A	False	Integer(gt: 0)	
amount_paid_vat	N/A	False	Float()	positive or negative value with VAT
type_id	N/A	False	Integer(gt: 0)	payment type id. Function payment_type_get returns accordance of type_id and type name.
session_id	N/A	True	String(allow_blank: False)	By defaults from user's session
direction	N/A	False	String(allow_blank: False, regex:	field describes 'direction' of payment: 'inc' means incoming, 'out' - outgoing and

			"^(incoming outgoing)\$")	'undefined'(can't be set from web)
notes	N/A	False	String(allow_blank: False)	any note, obligatory should be present in the request
date	N/A	False	String(allow_blank: False)	payment date, obligatory should be present in the request

2.259 payment_type_get

Docstring:

Returns names and ids of payment types which exists in database.

Example:

- Client Request

```
{"session_id": "f345e24573e77cadffbc28a11ae64af4",
  "data": {"fields": ["id", "name"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[0, "Traffic"],
            [1, "payment for traffic"],
            [2, "customer's dispute"],
            [3, "supplier's dispute"],
            [4, "bank charges"],
            [5, "netting, other"],
            [6, "external"]],
  "time_elapsed": 0.002735}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default	Optiona	Type	Description
------	---------	---------	------	-------------

	value	l		
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'id': ""}]	True	List([Enum('id', 'name'): Enum('asc', 'desc', "")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

Default fields:

Name	Type & Comments
id	Int
name	String

2.260 payments_get_ac

Docstring:

Function for accucore.

Request structure and fields description:

```
{'data': {'client_id': 'Integer',
          'date_from': 'String',
          'date_to': 'String',
          'direction': 'String',
          'fields': ["Enum('date', 'direction', 'amount_vat', 'amount_paid', 'client_id', 'balance_id', 'type_id', 'notes', 'client_name', 'id')"],
          'id': 'Integer',
          'session_id': 'String',
          'traffic_payment': 'Bool'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
direction	all	True	String(allow_blank: False, regex: "^(incoming outgoing all)\$")	
date_from	N/A	True	String(allow_blank: False)	
id	N/A	True	Integer(gt: 0)	
fields	['date', 'direction', 'amount_vat', 'amount_paid', 'client_id', 'balance_id', 'type_id', 'notes', 'client_name', 'id']	True	List([Enum('date', 'direction', 'amount_vat', 'amount_paid', 'client_id', 'balance_id', 'type_id', 'notes', 'client_name', 'id')], min_length: 0)	
date_to	N/A	True	String(allow_blank: False)	
client_id	N/A	True	Integer(gt: 0)	
session_id	N/A	True	String(allow_blank: False)	By defaults from user's session
traffic_payment	N/A	True	Bool	

Default fields:

Name	Type & Comments
date	String
direction	String
amount_vat	(25,15)
amount_paid	numeric(16,5)

client_id	Int
balance_id	Int
type_id	Int
notes	String
client_name	String
id	Int

2.261 payments_total_get_ac

Docstring:

Function for accucore.

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.262 pc_notification_send

Docstring:

Sends payment notification based on payment notification preset via email.

Request structure and fields description:

Request:


```

{
  "session_id": String(),
  "data": {
    "balances_ids_list": List(Int(gt=0)),
    "weekly_notification": Bool(),
  }
}

```

Name	Default value	Optional	Type	Description
"balances_ids_list"	N/A	True	List(Int(gt=0))	list of balances to be sent a notification
"weekly_notification"	N/A	False	Bool()	if true - data from payment_calendar_get(), not from redis

2.263 pc_weekly_get

Docstring:

Returns payment calendar weekly notification flag.

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "fields": List(String()) ,
  }
}

```

Name	Default value	Optional	Type	Description
"fields"	default_fields	False	List(String())	

Default fields:

Name	Type & Comments
"payment_calendar_weekly"	Bool

2.264 pc_weekly_set**Docstring:**

Sets payment calendar weekly notification flag.

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "payment_calendar_weekly": Bool(),
  }
}
```

Name	Default value	Optional	Type	Description
"payment_calendar_weekly"	N/A	False	Bool()	

2.265 permission_set

Docstring:

Changes state of the given property in specified role. If property enabled then disables it and vice versa.
Only 'Owner' can edit roles. Pre-installed roles are not editable i.e. roles with id less than 11.

Example:

- Client Request:

```
{"session_id": "71e35ce9184432c3cddea2b041629405",  
  "data": {"role_id": 116,  
           "properties_id": 108}  
}
```

- Server Response:

```
{"status": 200,  
  "message": "OK",  
  "time_elapsed": 0.023663,  
  "role_id": 116}
```

Request structure and fields description:

```
{'data': {'properties_id': 'Integer', 'role_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
role_id	N/A	False	Integer(gt: 0)	role id should be more than 10. Table of accordance of roles and their ids you can get using function 'roles_get'.
properties_id	N/A	False	Integer(gt: 0)	all properties with their ids you can get from function 'permission_set'

2.266 permissions_enabled_get

Docstring:

DEPRECATED

Returns only enabled properties of the given role.

If "role_id" is absent in query then function returns permissions of current user role.

Example:

- Client Request:

```
{"session_id": "8a4710eb8e328397352edf524b262c57",
  "data": {"limit": 100,
           "offset": 0,
           "fields": ["id", "nick", "label", "new_tab", "parent_id"]}}
```

- Server Response:

```
{"status": 200,
  "fields": ["id", "nick", "label", "new_tab", "parent_id"],
  "message": "OK",
  "data": [[1, "traffic_tools", "Traffic Tools", false, 0],
           [101, "active_calls", "Active Calls", true, 1],
           [2, "reports", "Reports", false, 0],
           [3, "client", "Clients", false, 0],
           [4, "prices", "Prices", false, 0],
           [5, "financial_tools", "Financial Tools", false, 0],
           [6, "guardian", "Guardian", false, 0],
           [7, "maintenance", "Maintenance", false, 0],
           [8, "help", "Help", false, 0],
           ....
          ],
  "time_elapsed": 0.008991
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('nick', 'new_tab', 'id', 'parent_id', 'label')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'role_id': 'Integer',
          'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
		1		

session_id	N/A	True	String(allow_blank: False)	Not in use. By default from current user's session
role_id	N/A	True	Integer(gt: 0)	id of role "Owner" is 1. Table of accordance of roles and their ids you can get using function 'roles_get'.
fields	['nick', 'new_tab', 'id', 'parent_id', 'label']	True	List([Enum('nick', 'new_tab', 'id', 'parent_id', 'label')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
nick	String
new_tab	Boolean
id	Int
parent_id	Int
label	String

2.267 permissions_get

Docstring:

Returns permissions of role with chosen "role_id". If "role_id" is absent in query, then returns permissions of current user role.

Function always returns one additional field which equals to page id if parent_id = 0, otherwise it equals parent_id.

Function orders permissions in the same way as web menus and submenus are displayed.

Example:

- Client Request:

```
{"session_id": "9926e3709067fde99af86b40b35fae35",
  "data": {"fields": ["nick", "new_tab", "show_in_menu", "property",
                    "properties_id", "enabled", "parent_id"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["nick", "new_tab", "show_in_menu", "property",
            "properties_id", "enabled", "parent_id"],
  "message": "OK",
  "data": [
    ["traffic_tools", false, true, "view", 1, true, 0, 1],
    ["active_calls", true, true, "view", 39, true, 1, 1],
    ["active_calls", true, true, "show_ip", 153, true, 1, 1],
    ["incoming_calls", true, true, "view", 20, true, 1, 1],
    ["incoming_calls", true, true, "show_ip", 154, true, 1, 1],
    ["outgoing_calls", true, true, "view", 21, true, 1, 1],
    ["outgoing_calls", true, true, "show_ip", 155, true, 1, 1],
    ["incoming_call_path", true, true, "view", 25, true, 1, 1],
    ["outgoing_call_path", true, true, "view", 26, true, 1, 1],
    ["calls_charts", false, true, "view", 45, true, 1, 1],
    ["call_trace", true, true, "view", 56, true, 1, 1],
    ["reports", false, true, "view", 2, true, 0, 2],
    ["financial_summary", true, true, "view", 43, true, 2, 2],
    .... ],
  "time_elapsed": 0.003879}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('nick', 'parent_id', 'new_tab', 'property', 'properties_id',
'id', 'enabled', 'show_in_menu')"],
  'limit': 'Integer',
  'offset': 'Integer',
  'role_id': 'Integer',
  'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
limit	0	True	Integer(gte: 0)	says how many rows will be in response

offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
session_id	N/A	True	String(allow_blank: False)	not in use
fields	['nick', 'parent_id', 'new_tab', 'property', 'properties_id', 'id', 'enabled', 'show_in_menu']	True	List([Enum('nick', 'parent_id', 'new_tab', 'property', 'properties_id', 'id', 'enabled', 'show_in_menu')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
role_id	N/A	True	Integer(gt: 0)	Table of accordance of role names and their ids you can get using function 'roles_get'.

Default fields:

Name	Type & Comments
nick	String
parent_id	Int
new_tab	Boolean
property	String
properties_id	Int
id	Int
enabled	Boolean
show_in_menu	Boolean

2.268 points_enabled_get

Docstring:

Calls functions ops_get() and tps_get() with option "enabled" = True and combine result in one list.

Request structure and fields description:

```
{'data': {}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description

2.269 price_analyze_request_add

Docstring:

Request file structure:

```
{user id, request date, description and other given info in json}
{destination},{price}
{destination},{price}
{destination},{price}
[...]
```

Request structure and fields description:

```
{'data': {'data': [['String or Float']],
          'fields': ['String'],
          'file_name': 'String',
          'send_email': 'Bool',
          'service_id': 'Integer',
          'tp_id': 'Integer',
          'tp_tariff_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default	Optional	Type	Description

	value	l		
tp_id	N/A	True	Integer(gte: 0)	used just in description in email subject
data	N/A	True	List([List([Or(Destination(allow_blank: False, _dest_pattern: ^\d*\$), Float(gte: 0))], min_length: 2, max_length: 2)], min_length: 1)	prices data values in the form of list
service_id	N/A	False	Integer(gt: 0)	service to check prices for
tp_tariff_id	N/A	True	Integer(gt: 0)	TP tariff id to get current prices. Cannot be used with "data" key.
fields	N/A	True	List([String(allow_blank: False)], min_length: 2)	Prices fields names in the same order as given prices values. Allowed fields are 'destination': String and 'price': Float
file_name	N/A	True	String(allow_blank: True)	used only in description
send_email	True	False	Bool	send analyze result by email

2.270 price_analyze_request_del

Docstring:

Request structure and fields description:

```
{'data': {'request_id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description

request_id	N/A	False	Integer(gt: 0)	Id of request to delete
------------	-----	-------	----------------	-------------------------

2.271 price_analyze_requests_get

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('file', 'tariff_name', 'last_status_date', 'description', 'request_date', 'request_id', 'service_id', 'point_name', 'user_name')"],
          'request_id': 'Integer',
          'service_id': 'Integer',
          'show_all_results': 'Bool',
          'tp_name': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_name	N/A	True	String(allow_blank: True)	filter by TP or TP Tariff name by given prefix
show_all_results	False	False	Bool	don't filter by current user. Works only if user is owner
request_id	N/A	True	Integer()	
service_id	N/A	True	Integer(gt: 0)	filter requests by service
fields	['file', 'tariff_name', 'last_status_date', 'description', 'request_date', 'request_id', 'service_id', 'point_name', 'user_name']	True	List([Enum('file', 'tariff_name', 'last_status_date', 'description', 'request_date', 'request_id', 'service_id', 'point_name', 'user_name')], min_length: 0)	fields, which will be present in response. If empty - returns fields specified in "default_fields" section

Default fields:

Name	Type & Comments
file	List(String(), String(), Int()) - List that contains file name, download link and file size in bytes.
tariff_name	String
last_status_date	String
description	String
request_date	String
request_id	Int
service_id	Int
point_name	String
user_name	String

2.272 price_analyze_top_routes_get**Docstring:****Request structure and fields description:**

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.273 price_status_get

Docstring:

Shows status of voice or SMS price uploading.

Statuses:

- "Applied on RTP" means that price list was successfully uploaded and applied.
- "Uploaded" - price list just uploaded but have not been applied yet.

Example:

- Client Request

```
{"session_id": "04f1c7e2e5f0bd62dd25e7aacb03d284",
  "data": {"limit": 101,
    "end_date": "2013-11-19T23:59:59",
    "sort_by": [{"tariff_name": ""}],
    "offset": 0,
    "service_id": 1,
    "start_date": "2013-11-17T00:00:00",
    "fields": ["id", "name", "insert_by", "insert_time",
      "price_type", "status", "notification_email",
      "status_time", "status_name", "tariff_name",
      "tariff_id"]}
```

- Server RESPONSE:

```
{"status": 200,
  "fields": ["id", "name", "insert_by", "insert_time",
    "price_type", "status", "notification_email",
    "status_time", "status_name", "tariff_name",
    "tariff_id"],
  "message": "OK",
  "data": [[7783, "Loaded by qw", "qw",
    "2013-11-19T15:23:13.678156", "incremental", 7,
    "test@test.com", "2013-11-19T15:23:13.888922",
    "Applied on RTP", "peer_out tariff", 4864],
    [7757, "Loaded by SS", "SS",
    "2013-11-18T15:35:35.693572", "incremental", 7,
    "ikex@speedflow.com", "2013-11-18T15:38:17.399470",
    "Applied on RTP", "new.peer_out tariff", 4864],
    "time_elapsed": 0.08435
}
```

Request structure and fields description:

```
{'data': {'end_date': 'String',
  'fields': ["Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')"]},
```

```

        'limit': 'Integer',
        'offset': 'Integer',
        'price_id': 'Integer',
        'service_id': 'Integer',
        'sort_by': [{"Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')": "Enum('asc', 'desc', '')"}],
        'start_date': 'String',
        'tariff_id': 'Integer',
        'user_id': 'Integer',
        'user_name': 'String'},
        'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
end_date	N/A	True	String(allow_blank: True)	Filter by price insertion time, shows only prices which were uploaded earlier given time
user_name	N/A	True	String(allow_blank: True)	filter by name or first chars of user name who uploaded price
user_id	N/A	True	Integer()	filter by id of user who uploaded price
service_id	1	True	Integer(gte: 1, lte: 2)	filter by service id. 1 - voice, 2 - sms
price_id	N/A	True	Integer()	filter by price id
tariff_id	N/A	True	Integer()	filter by tariff id

fields	['status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time']	True	List([Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'id': 'desc'}]	True	List([Enum('status', 'tariff_name', 'currency_name', 'notification_email', 'id', 'tariff_id', 'insert_by', 'name', 'status_name', 'price_type', 'users_id', 'insert_time', 'status_time'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
start_date	N/A	True	String(allow_blank: True)	Filter by price insertion time, shows only prices which were uploaded after given time

Default fields:

Name	Type & Comments
status	Int
tariff_name	String
currency_name	String(3) - name_short

notification_email	String
id	Int
tariff_id	Int
insert_by	String
name	String
status_name	String
price_type	String
users_id	Int
insert_time	String
status_time	String

2.274 price_upload

Docstring:

This function adds given prices to the uploading queue. It allows to upload new prices only to the one tariff.

There are several possible prices formats which function accepts. You can get them with help of function 'prices_format_get'.

Function process request in the following order:

- 1) Sets direction ("1" for "in", 2 - for "out") and checks prices validity.
- 2) Uploads new prices.
- 3) If uploaded price is present in the list of destinations blocked by Guardian - function changes their state (in the same way as actions "leave closed" and "open now" in function guardian_blocked_dest_set) depending on 'allowed' flag in price that has been uploaded.

Function will raise error with status 305 for single price uploading if 'start_date' in price less than current. Despite the error price will be uploaded but 'start_date' will be changed to the current date.

Possible fields names are mentioned in the default_fields.

Example:

-Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "price_type": "incremental",
    "direction": "out",
    "data": [
      4700,
      true,
      true,
      "",
      "3912",
      0.02345,
      0.0,
      1,
      1,
      0,
      0,
      "2013-10-29T12:34:37",
      "2030-01-01T23:59:59",
      "00:00:00",
      "23:59:59",
      1,
      24,
      "01111111"
    ],
    "fields": [
      "tarif_id",
      "enabled",
      "allowed",
      "description",
      "destination",
      "price",
      "connection_price",
      "mc",
      "ci",
      "ft",
      "priority",
      "date_start",
      "date_end",
      "time_start",
      "time_end",
      "min_len",
      "max_len",
      "week_day"
    ]
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tariff_id": Int(),
```



```

"session_id": String(),
"data": List(Any(), min_length=1),
"fields": List(String()) ,
"direction": String(regex="(in)|(out)"),
"price_type": String(regex="(incremental)|(full)"),
"notification_email": String(allow_blank=True),
"guardian": Bool(),
"operator": String(),
"label": String(),
"save_previous_price_priority": Bool(),
"save_previous_min_max_digits": Bool(),
"add_and_block_codes": Bool(),
"replace_codes_on_future": Bool(),
"flag_mc_ci_ft": Bool(),
"save_privious_price_priority": Bool(),
"save_privious_min_max_digits": Bool(),
"analyze_price": Bool(),
}
}

```

Name	Default value	Optional	Type	Description
"tariff_id"	N/A	True	Int()	id of the target tariff
"session_id"	session_info["id"]	True	String()	current session id
"data"	N/A	False	List(Any(), min_length=1)	prices data values in the form of list
"fields"	N/A	False	List(String())	prices fields names in the same order as given prices values
"direction"	N/A	False	String(regex="(in) (out)")	"in" - for OP tariffs, "out" - for TP tariffs
"price_type"	"incremental"	True	String(regex="(incremental) (full)")	if type "full" all prices existed in the tarif will be closed and then uploaded new one. If "incremental" - new prices uploads over existed.
"notification_email"	session_info.get("user_email", "")	True	String(allow_blank=True)	email address to which will be send notification when price will be successfully uploaded. If doesn't specify in request - email will be sent to recipients addresses specified in price log template.

"guardian"	False	True	Bool()	for internal usage
"operator"	session_info.get("login", "API")	True	String()	name of user who uploading price. By default - user login from current session
"label"	u"Loaded by {0}".format(unicode(session_info.get("login", "API")))	True	String()	label for price that will be recorded to database. Example: "Loaded by <Name of user who uploaded the price>". If label is absent in request - label will be set by defaults as "Loaded by <login of user from current session.>
"save_previous_price_priority"	False	True	Bool()	flag allows to preserve priority of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"save_previous_min_max_digits"	False	True	Bool()	flag allows to preserve min and max digits of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"add_and_block_codes"	False	True	Bool()	use in conjunction with 'type'='full'. If True than all currently existed prices which didn't present in uploading price list will be added to the price list with allowed=False. All other params will be preserved.
"replace_codes_on_future"	False	True	Bool()	if there are exists price for given code with older start date then in new one, then new will be uploaded over old one (it means that old price date_end becomes one second less then date_start in new one)
"flag_mc_ci_ft"	False	True	Bool()	flag allows to preserve

				tarification value of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"save_privious_price_priority"	False	True	Bool()	DEPRECATED, backport
"save_privious_min_max_digits"	False	True	Bool()	DEPRECATED, backport
"analyze_price"	False	True	Bool()	flag for Price Analyzer. If 'True' price will be send for analyzing after processing.

Field **"data"** - is a list that contains prices data for different destination codes in the form of list.

```
"data": [
  [
    "tarif_id": Int(gt=0),
    "enabled": Bool(),
    "allowed": Bool(),
    "description": String(allow_blank=True),
    "destination": String(regex="^[0-9]*$"),
    "price": Float(gte=0),
    "connection_price": Float(gte=0),
    "mc": Int(gte=0),
    "ci": Int(gt=0),
    "ft": Int(gte=0),
    "priority": Int(gte=-100, lte=100),
    "date_start": String(),
    "date_end": String(),
    "time_start": String(regex=TIME_CHECK_REGEXP_PATTERN),
    "time_end": String(regex=TIME_CHECK_REGEXP_PATTERN),
    "week_day": String(regex="(0|1){8}"),
    "min_len": Int(gte=1, lte=100),
    "max_len": Int(gte=1, lte=100),
    "origin_country": String(regex="^[0-9\\*\\#]{0,13}$", allow_blank=True, max_length=13),

  ], [...],[...], ...
]
```

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

"tarif_id"	N/A	False	Int(gt=0)	id of the tariff to which price will be uploaded. Should be the same for all prices in price list.
"enabled"	True	True	Bool()	Defines price status. Disabled prices are not actual.
"allowed"	True	True	Bool()	"False" value makes destination disallowed.
"description"	N/A	True	String(allow_blank=True)	
"destination"	N/A	False	String(regex="^[0-9]*\$")	Defines destination code for price
"price"	0	True	Float(gte=0)	price per minute
"connection_price"	0	True	Float(gte=0)	a connection fee
"mc"	1	True	Int(gte=0)	Minimal count interval. Call with duration less then this value - will have cost the same as call with duration specified in this field.
"ci"	1	True	Int(gt=0)	Tarification interval. For per second tarification put here "1" and "60" for per minute tarification
"ft"	0	True	Int(gte=0)	Time in seconds free of charge from beginning of call.
"priority"	0	True	Int(gte=-100, lte=100)	Price priority. You can specify this option only for TP tariff, i.e. if direction="out".
"date_start"	datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")	True	String()	Defines the start date and time for the price validity. It should not be less than current. Format: "YYYY-

				MM-DDTHH:MM:SS", where T is delimiter between date and time. If price list is being uploaded with "price_type" = "full" all prices must have the same start date.
"date_end"	"2037-12-31 23:59:59+00"	True	String()	Defines the end date and time for the price validity. Format: "YYYY-MM-DDTHH:MM:SS", where T is delimiter between date and time.
"time_start"	"00:00:00+00"	True	String(regex=TIME_CHECK_REGEX_PATTERN)	Defines the start time for the price availability during the day time.
"time_end"	"23:59:59+00"	True	String(regex=TIME_CHECK_REGEX_PATTERN)	Defines the end time for the price availability during the day time.
"week_day"	"01111111"	True	String(regex="(0 1){8}")	Bit mask, which defines days of week of price validity. First bit - not in use, second - Saturday, third - Friday, ...7th bit - Monday, 8th bit - Sunday. 1- means that price works in that day of week, 0 - not. Example - "01111111" - price works all days of week, "00111110" - price works every day excluding sunday and saturday.
"min_len"	1	True	Int(gte=1, lte=100)	Defines the minimal length of destination code for current price in range from 1 till 100 inclusive.
"max_len"	24	True	Int(gte=1, lte=100)	Defines the maximum length of destination code for current price in range from 2 till 100 inclusive.
"origin_country"	""	True	String(regex="^[0-9*\\#]{0,13}\$", allow_blank=True,	Defines Origin Country for price

			max_length=13)	
--	--	--	----------------	--

Default fields:

Name	Type & Comments
"tarif_id"	
"enabled"	
"allowed"	
"description"	
"destination"	
"price"	
"connection_price"	
"mc"	
"ci"	
"ft"	
"priority"	
"date_start"	
"date_end"	
"time_start"	
"time_end"	

"min_len"	
"max_len"	
"week_day"	
"origin_country"	

2.275 prices_creation_get

Docstring:

Returns new price list created from currently actual prices (not considering active time and week day) of given TP tariffs using chosen price increase and charging options.

All prices in the resulted price list are in system's default currency and ordered by destination code.

Field 'is_covered' = True tells that:

- destination of covered price is subdestination of covering (e.g. 380 and 38050 are subdests for 3);
- both rates of 'covered' and 'covering' prices are equal.

Coverage logic is based on Mediacore routing algorithm and supposes that covered price is redundant because it duplicates price for shorter destination code which covers given one.

Excluding keys 'data' and 'fields' function also returns:

- 'rows_not_enough_tps' - number of prices which are present not in all selected tariffs
- 'rows_recently_closed' - quantity of prices which will be closed during 'recently_closed' period
- 'rows' - total quantity of prices in resulted price list
- 'rows_covered' - quantity of prices which are covered by shorter destination with the same price

If "backups_quantity" not specified and "price_type" = 4 then 404 error raised:
"Option 'backups_quantity' is required for 'price_type' 4"

Example:

- Client Request

```
{
  "session_id": "30aa6ac62d855c37268dcd2ed2d02226",
  "data": {
    "charging_options": 1,
    "price_type": 1,
    "mc": 1,
    "increase_type": 1,
    "charging_formula": [],
    "tariff_ids": [12, 4],
    "ci": 1,
    "increase_formula": []
  }
}
```

```

        "increase":0.02,
        "ft":0,
        "is_recently_closed":1,
        "fields":["tariff_id","tariff_name","destination","price",
            "date_start","description","delta_price",
            "is_recently_closed","is_covered",
            "is_not_enough_tariffs","country","max_nlen",
            "min_nlen","mc","ci","ft"]}
    }

```

- Server Response

```

{"status": 200,
 "rows_recently_closed": 0,
 "rows": 2,
 "fields": ["tariff_id", "tariff_name", "destination", "price",
    "date_start", "description", "delta_price",
    "is_recently_closed", "is_covered",
    "is_not_enough_tariffs", "country", "max_nlen",
    "min_nlen", "mc", "ci", "ft"],
 "rows_covered": 0,
 "rows_not_enough_tps": 2,
 "time_elapsed": 0.011831,
 "message": "OK",
 "data": [[4, "Test_tariff_TP", "380507", "0.42000",
    "2014-01-10T10:28:24.293351+00:00",
    "Test_tariff_TP (USD) (0.40000) till 2030-01-01 00:00:00",
    0.02, false, false, true, "Ukraine Mobile Mts", 24, 1,
    1, 1, 0],
    [4, "Test_tariff_TP", "5", "1.52000",
    "2014-01-10T15:44:34.797310+00:00",
    "Test_tariff_TP (USD) (1.50000) till 2015-10-25 23:59:59",
    0.02, false, false, true, "Unknown Code", 100, 1, 1, 1, 0]]
}

```

Request structure and fields description:

```

{'data': {'backups_quantity': 'Integer',
  'charging_formula': [{'ci': 'Integer',
    'destination': 'String',
    'ft': 'Integer',
    'mc': 'Integer'}],
  'charging_options': 'Integer',
  'ci': 'Integer',
  'destination': 'String',
  'destination_exclude': 'String',
  'fields': ["Enum('extended_description', 'ci', 'tariff_id', 'ft', 'descriptio
n', 'mc', 'country', 'price', 'destination', 'date_start', 'max_nlen', 'is_covered', 'i
s_recently_closed', 'tariff_name', 'delta_price', 'is_not_enough_tariffs', 'extended_de
scription_switch_currency', 'min_nlen', 'date_end')"],
  'ft': 'Integer',
  'ignore_dest_without_backups': 'Bool',
  'increase': 'Float',
  'increase_formula': [{'from': 'Float',
    'increase': 'Float',
    'increase_type': 'Integer',
    'to': 'Float'}],
  'increase_type': 'Integer',
  'mc': 'Integer',
  'price_type': 'Integer',

```



```

    'recently_closed': 'Integer',
    'session_id': 'String',
    'tariff_ids': ['Integer']},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
destination_exclude	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination codes which have to be excluded from price list
charging_options	1	True	Integer(gte: 1, lte: 3)	1 - sets values specified in params 'mc', 'ci', 'ft'; 2 - takes values for charging parameters from chosen by option 'price_type' parent price, 3 - takes values defined in 'charging_formula'(default values specified in params 'mc', 'ci', 'ft') if any formula matches else default values are used.
charging_formula	N/A	True	List([{'ci': 'Integer', 'destination': 'String', 'ft': 'Integer', 'mc': 'Integer'}], min_length: 0)	should be used in conjunction with "charging_options":3. Formula checks that destination starts from specified. Example: "charging_formula": [{"mc":1,"ci":1,"ft":0,"destination":"12"}]. First matched 'formula' will be used. If no one formula matches - values of params 'mc', 'ci', 'ft' are used.
recently_closed	3	True	Integer(gte: 1, lte: 3)	Checks that price will be ended in N weeks. 1 - one week, 2 - two weeks, 3 - three weeks.

price_type	N/A	False	Integer(gte: 1, lte: 4)	prices selection option for further processing: 1- price with minimal rate will be used as parent, 2 - price with maximal rate will be used as parent, 3 - counts average price rate and takes data from other data from price with maximal rate, 4 - select price with "backup" ("backups_quantity" is required)
backups_quantity	N/A	True	Integer(gte: 1, lte: 10)	has sense when 'price_type' = 4. Skips first "backups_quantity" prices for the same destination ordered by price ascending and takes next one. If "backups_quantity" >= number of prices for a destination then option "ignore_dest_without_backups" will determine a price to be taken.
fields	['extended_description', 'ci', 'tariff_id', 'ft', 'description', 'mc', 'country', 'price', 'destination', 'date_start', 'max_nlen', 'is_covered', 'is_recently_closed', 'tariff_name', 'delta_price', 'is_not_enough_tariffs', 'extended_description_switch_currency', 'min_nlen', 'date_end']	True	List([Enum('extended_description', 'ci', 'tariff_id', 'ft', 'description', 'mc', 'country', 'price', 'destination', 'date_start', 'max_nlen', 'is_covered', 'is_recently_closed', 'tariff_name', 'delta_price', 'is_not_enough_tariffs', 'extended_description_switch_currency', 'min_nlen', 'date_end')], min_length: 0)	required fields.If absent - takes fields specified in "default fields". If empty - returns error.
mc	1	False	Integer(gte: 1)	Default minimal count interval for 'charging_options'=3 or value for all prices if 'charging_options'=1.
destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	destination which required
increase	0	True	Float()	defines value or percent on

				which price have to be increasing. To increase price in 1.5 'increase' should be 50.0.
ft	0	False	Integer(gte: 0)	Default call free time for 'charging_options'=3 or value for all prices if 'charging_options'=1.
increase_formula	[]	False	List([{'increase': 'Float', 'to': 'Float', 'increase_type': 'Integer', 'from': 'Float'}], min_length: 0)	this option defines additional price increase for given rate ranges. Keys "increase" and "increase_type" have the same meaning as mentioned above. First matched 'formula' will be used to count new price and 'delta_price' values. First formula that matches price's rate value will be used to count new one.
increase_type	2	True	Integer(gte: 1, lte: 2)	defines default method of price increasing: 1 - increase in the value specified in 'increase' ('increase' is added to the price), 2 - % increasing that means percentage on which price have to be increased
session_id	N/A	True	String(allow_blank: False)	not in use, by default from user session
ignore_destination_without_backups	False	True	Bool	Option should be used in conjunction with 'price_type' = 4. If True when "backups_quantity" more or equal to number of prices for a destination this destination will be skipped in resulted list. Otherwise last destination price (actually price with largest price) will be chosen.
ci	1	False	Integer(gte: 1)	Default tarification interval for 'charging_options'=3 or

				value for all prices if 'charging_options'=1.
tariff_ids	N/A	False	List([Integer(gt: 0)], min_length: 1)	List of TP tariffs from which price list will be created

Default fields:

Name	Type & Comments
extended_description	String
ci	Int
tariff_id	Int
ft	Int
description	String
mc	Int
country	String - country name
price	Float - new price in default currency
destination	String - destination code
date_start	String
max_nlen	Int
is_covered	Boolean
is_recently_closed	Boolean
tariff_name	String
delta_price	Float - difference between increased price value and original one in default currency

is_not_enough_tariffs	Boolean - True if price for dest is not present in all chosen tariffs
extended_description_switch_currency	String
min_nlen	Int
date_end	String

2.276 prices_format_get

Docstring:

Returns list of possible price list formats. Format describes possible fields in price list.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {}
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('format',)"],
           'limit': 'Integer',
           'offset': 'Integer',
           'sort_by': [{"Enum('format',)": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['format']	True	List([Enum('format',)],	required fields. If absent - takes fields specified in "default fields". If empty -

			min_length: 0)	returns error.
sort_by	[]	True	List([Enum('format'), Enum('asc', 'desc', ")]], min_length: 0)	field name on which result will be sorted
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
format	String

2.277 prices_list_get

Docstring:

Returns voice prices belonging to chosen OP tariff or TP tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Use only single of types - "current", "actual_prices" or "future", not all together.

Example:

- Client Request

```
{
  "session_id": "f28a3d96a0ecf3b1bbf44feed8b0c952",
  "data": {
    "limit": 17,
    "sort_by": [{"destination": ""}, {"added": "desc"}],
    "tariff_id": 4201,
    "offset": 0,
    "service_id": 1,
    "fields": [
      "id", "tariff_id", "enabled", "allowed",
      "description", "destination", "price",
      "connection_price", "mc", "ci", "ft", "priority",
      "date_start", "date_end", "time_start", "time_end",
      "added", "date_for_history", "min_len", "max_len",
      "week_day", "country"], "destination": ""
    ]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "time_elapsed": 0.010708,
  "fields": [
    "id", "tariff_id", "enabled", "allowed", "description",

```

```

        "destination", "price", "connection_price", "mc", "ci",
        "ft", "priority", "date_start", "date_end",
        "time_start", "time_end", "added", "date_for_history",
        "min_len", "max_len", "week_day", "country"],
    "message": "OK",
    "data": [[10129239, 4201, true, true, "", "111111111", "0.00236",
        "0.00000", 1, 1, 0, 0,
        "2013-10-25T13:52:11.203884+00:00",
        "2030-01-01T00:00:00+00:00", "00:00:00", "23:59:59",
        "2013-10-25T13:52:11.161933+00:00",
        "2030-01-01T00:00:00+00:00", 1, 24, "01111111", null],
        [10129237, 4201, true, true,
        "Record changed by system due to price overlap 10129239",
        "111111111", "0.00235", "0.00000", 1, 1, 0, 0,
        "2013-10-25T13:43:26.997105+00:00",
        "2013-10-25T13:52:10.203884+00:00", "00:00:00",
        "23:59:59", "2013-10-25T13:43:26.942568+00:00",
        "2030-01-01T00:00:00+00:00", 1, 24, "01111111", null],
        [10129236, 4201, true, true,
        "Record changed by system due to priceoverlap 10129237",
        "111111111", "0.00001", "0.00000", 1, 1, 0, 0,
        "2013-10-25T13:40:15.196570+00:00",
        "2013-10-25T13:43:25.997105+00:00", "00:00:00",
        "23:59:59", "2013-10-25T13:40:15.158241+00:00",
        "2030-01-01T00:00:00+00:00", 1, 24, "01111111", null]]
    }

```

Request structure and fields description:

```

{
  'data': {
    'actual_prices': 'Bool',
    'allowed': 'Bool',
    'blocked_by_guardian': 'Bool',
    'current': 'Bool',
    'destination': 'String',
    'enabled': 'Bool',
    'end_date': 'String',
    'fields': ["Enum('ci', 'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'id', 'origin_country', 'actual_price_id', 'time_start', 'date_start', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price', 'date_for_history')"],
    'future': 'Bool',
    'get_actual': 'Bool',
    'id': 'Integer',
    'is_tp': 'Bool',
    'latest': 'Bool',
    'limit': 'Integer',
    'offset': 'Integer',
    'origin_country': 'String',
    'pt_id': 'Integer',
    'service_id': 'Integer',
    'sort_by': [{"Enum('ci', 'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'id', 'origin_country', 'actual_price_id', 'time_start', 'date_start', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price', 'date_for_history')": "Enum('asc', 'desc', '')"}],
    'start_date': 'String',
    'strictly': 'Bool',
    'tariff_id': 'Integer'},
  'session_id': 'String'
}

```

data validation description:

Name	Default value	Optional	Type	Description
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current, date_start=
fields	['ci', 'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'id', 'origin_country', 'actual_price_id', 'time_start', 'date_start', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price', 'date_for_history']	True	List([Enum('ci', 'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'id', 'origin_country', 'actual_price_id', 'time_start', 'date_start', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price', 'date_for_history')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
sort_by	[]	True	List([Enum('ci', 'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'id', 'origin_country', 'actual_price_id', 'time_start', 'date_start', 'priority', 'price_status', 'added', 'ft', 'description', 'min_len', 'price', 'tariff_id', 'mc', 'country', 'enabled', 'connection_price', 'date_for_history'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
id	N/A	True	Integer(gt: 0)	Price id in database. If "is_tp"=="true" then TP price id, else - OP price id
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
tariff_id	N/A	True	Integer(gt: 0)	tariff id
latest	False	True	Bool	If "true" returns last uploaded prices by destination. May cause timeout if tariff_id is not passed!
pt_id	N/A	True	Integer(gt: 0)	if "is_tp"=="true" then TP id, otherwise - OP id
origin_country	N/A	True	String(allow_blank: True)	Origin country
enabled	N/A	True	Bool	if "true" returns enabled prices
future	False	True	Bool	Used by Accucore. If "True" function searches only future prices for pt
service_id	N/A	False	Integer(gt: 0)	filter by service: 1 - voice, 2 - SMS
strictly	False	True	Bool	If "true" function searches strict match of

				given destination code in the database, else searches destinations code with subcodes.
start_date	N/A	True	String(allow_blank: False)	start date and time to show prices with start_date not less then given.
get_actual	N/A	True	Bool	Option should be used in conjunction with price id. If "get_actual" set to True, then function returns all actual prices for destination of the price with given id within price's tariff.
actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current, date_start=
end_date	N/A	True	String(allow_blank: False)	end date and time to show prices with end_date not more then given.
blocked_by_guardian	False	True	Bool	Used by Accucore. If "True" function searches only blocked destination codes for actual price.
allowed	N/A	True	Bool	if "true" returns allowed prices

Default fields:

Name	Type & Comments
ci	Int

destination	String
date_end	String
time_end	String
max_len	Int
week_day	String
allowed	Boolean
id	Int
origin_country	String
actual_price_id	Int
time_start	String
date_start	String
priority	Int
price_status	String
added	String
ft	Int
description	String
min_len	Int
price	numeric(16,5)
tariff_id	Int
mc	Int
country	String
enabled	Boolean

connection_price	numeric(16,5)
date_for_history	String

2.278 problem_calls_get_out

Docstring:

Function for accucore.

Request structure and fields description:

```
{'data': {'clear_number': 'String',
          'client_id': 'Integer',
          'end_date': 'String',
          'fields': ['String'],
          'limit': 'Integer',
          'longer_then': 'Integer',
          'offset': 'Integer',
          'shorter_then': 'Integer',
          'start_date': 'String',
          'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	True	Integer(gt: 0)	filter by company id.
fields	['called_number', 'nas', 'end_time', 'called_number_out', 'duration', 'start_time', 'termination_cause', 'dst_codec', 'tp_ip', 'setup_time']	True	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
start_date	N/A	False	String(allow_blank: False)	start date and time for data gathering. Report returns calls which setup_time not

				less than given.
shorter_than	N/A	True	Integer(gte: 0)	filter that allows to return calls with duration (in seconds) less than given.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
tp_id	N/A	True	Integer(gt: 0)	filter by TP id
end_date	N/A	False	String(allow_blank: False)	end date and time for data gathering. Report returns calls which setup_time not more than given.
clear_number	N/A	True	String(allow_blank: False, regex: "^[0-9,]*\$")	called number without prefix
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
longer_than	N/A	True	Integer(gte: 0)	filter that allows to return calls with duration (in seconds) more than given.

Default fields:

Name	Type & Comments
called_number	String(200)
nas	String (RTP IP address)
end_time	String
called_number_out	String(200)
duration	Int

start_time	String
termination_cause	Int
dst_codec	String(255)
tp_ip	String
setup_time	String

2.279 problem_types_get

Docstring:

Returns list of problem types which exist in the database.

Example:

- Client Request

```
{"session_id": "32e62a88ed92dc437f3379f2535a31d7",
  "data": {"session_id": "32e62a88ed92dc437f3379f2535a31d7",
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "fields": ["id", "name"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[7, "Incorrect DC"], [1, "Junk traffic"],
    [4, "Low quality"], [6, "Route drop"],
    [3, "Supplier overloaded"], [2, "Technical problems"],
    [5, "Terrible quality"], [8, "Unrouted calls"]],
  "time_elapsed": 0.041712}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('description', 'id', 'name')"],
  'limit': 'Integer',
  'offset': 'Integer',
  'sort_by': [{"Enum('description', 'id', 'name')": "Enum('asc', 'desc', '')"}]
},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['description', 'id', 'name']	True	List([Enum('description', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('description', 'id', 'name'), Enum('asc', 'desc', '')]), min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
description	String
id	Int
name	String

2.280 process_new_price

Docstring:

Process uploaded prices (changes its status to '7' - 'Applied') and creates notification email after price uploading.
Function for internal usage.

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {
```

```
}  
}
```

2.281 radiuses_by_context_get

Docstring:

Function returns radiuses data(ip, name and so on) for different collocations.

Example:

- Client Request

```
{  
  "session_id": "4595aceb14c686aa9752bb9860462bf2",  
  "data": {  
    "sort_by": [{"name": ""}],  
    "contexts": [1],  
    "fields": ["id", "name"]} } }
```

- Server Response

```
{  
  "status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[1, "CZ"]],  
  "time_elapsed": 0.013664 } }
```

Request structure and fields description:

```
{  
  'data': {  
    'contexts': ['Integer'],  
    'fields': ["Enum('colocation_id', 'ip', 'id', 'name')"],  
    'limit': 'Integer',  
    'offset': 'Integer',  
    'sort_by': [{"Enum('colocation_id', 'ip', 'id', 'name')": "Enum('asc', 'desc', '')"}]},  
  'session_id': 'String' }
```


data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer()], min_length: 1)	filter by selected contexts
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['colocation_id', 'ip', 'id', 'name']	True	List([Enum('colocation_id', 'ip', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('colocation_id', 'ip', 'id', 'name'): Enum('asc', 'desc', "")}], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
colocation_id	Int
ip	String
id	Int
name	String

2.282 regexp_test

Docstring:

Checks regexp for Dnis/Ani Regexp, Dstregexp, Ani Block, Denyregexp and returns result within the appropriate key.

For Ani Block and other not replacing expressions returns string 'Match' or 'No Match'. For Dnis/Ani Regexp returns changed string.

Example:

```
- Client's Request:
  {"session_id": "98c16a9f07e24b5e4c1844d31530b593",
   "data": {"change_value": false,
            "number": "1234",
            "regexp": "^12.*"}}

- Server Response:
  {"status": 200, "message": "OK", "result": "Match",
   "time_elapsed": 0.016811}
```

Request structure and fields description:

```
{'data': {'change_value': 'Bool', 'number': 'String', 'regexp': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
change_value	N/A	False	Bool	for Dnis/Ani Regexp - True, for other cases - False
number	N/A	False	String(allow_blank: True)	test number
regexp	N/A	False	String(allow_blank: False)	regular expression that is being checked

2.283 regions_by_tp_get

Docstring:

Function returns list of currently actual regions for given termination point within chosen context.

Function was created for creating AutoTests from Incoming Calls (MCD-4295). Actual means that price for the destination works at the moment.

Example:

- Client Request

```
{
  "data": {
    "tp_id": 4692,
    "context_id": 1,
    "fields": ["destinations", "country"]
  }
  "session_id": "ca84cd3eaa596f2e3f7836db065777da"
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.010708,
  "message": "OK",
  "data": [
    [{"38050"}, None],
    [{"8801"}, 'BANGLADESH MOBILE'],
  ],
  "fields": ["destinations", "country"]
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',
          'country_or_code': 'String',
          'fields': ["Enum('country', 'destinations')"],
          'hide_without_country': 'Bool',
          'limit': 'Integer',
          'offset': 'Integer',
          'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['country', 'destinations']	True	List([Enum('country', 'destinations')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
tp_id	N/A	False	Integer(gt: 0)	Termination Point ID

offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
hide_without_country	N/A	True	Bool	show only records with non-empty country
country_or_code	N/A	True	String(allow_blank: False)	filter by country name or destination code
context_id	N/A	False	Integer(gt: 0)	ID of chosen context

Default fields:

Name	Type & Comments
country	String or NULL
destinations	Array of Strings

2.284 rel_op_tp_add

Docstring:

Creates relations between chosen OP and TP(s), i.e.:

- 1) creates service dialpeer '.*' if such dialpeer doesn't exist in the given context,
- 2) creates service group for given OP if such doesn't exist in the given context,
- 3) creates relation between service dialpeer and given TP, if such doesn't exist in the chosen context,
- 4) adds dialpeer, OP and TP to the service group of given OP in specified context.

Function raises error if relation between OP and some non service group is already exists.

Example:

- Client Request

```
{
  "session_id": "fbea69d4b49be804dfe87f2e1d852f2f",
  "data": {
    "op_ids": [5],
    "contexts": [1],
  }
}
```

```
        "tp_ids": [14]}
    }
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.311718}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'op_ids': ['Integer'],
          'tp_ids': ['Integer']},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of TP ids
op_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of OP ids
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	context ids to which relations will be added

2.285 rel_op_tp_del

Docstring:

Removes relations between chosen OP and TPs which were created in the simple mode, i.e. removes relations between TPs and service group of given OP and service dialpeer in the chosen context(s). If given OP has lost all relation with TPs then service group of the OP will be removed too.

Example:

- Client Request

```
{"session_id": "6726cebe6ca27ef6a8da30867d6050a8",
 "data": {"op_ids": [17],
```

```

        "contexts":[1],
        "tp_ids":[24]}
    }

```

- Server Response

```

{"status": 200, "message": "OK", "time_elapsed": 2.182171}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'op_ids': ['Integer'],
          'tp_ids': ['Integer']},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	TP ids which need remove from relations with given OP
contexts	N/A	False	List([Integer(gt: 0)], min_length: 1, max_length: 1)	context ids in which relation need to delete
op_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	

2.286 role_add

Docstring:

Creates new role (user profile). Only user with profile "Owner" has permissions to add new role.

Function returns id of new role in the database.

Example:

- Client Request

```

{"session_id":"0945535c018205fee591b31fd11a1a8d",
 "data":{"name":"NewRole"}}

```

```
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 25, "time_elapsed": 0.065114}
```

Request structure and fields description:

```
{'data': {'name': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	role name

2.287 role_del

Docstring:

```
Removes role with specified id. The function disallows to remove profile with id 0 ("Banned" profile). Only user with profile "Owner" has permissions to do it. You can't remove pre-installed roles, i.e. roles with id less than 11.
```

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(id of role which necessary to remove. It should be more than

)	10.
--	--	--	---	-----

2.288 role_set

Docstring:

Changes name of the role with given id. Only user with profile "Owner" has permissions to do it. You can't rename pre-installed roles, i.e. roles with id less than 11.

Function returns role id in the database.

Example:

- Client Request

```
{"session_id":"0945535c018205fee591b31fd11a1a8d",
  "data":{"id":25,
          "name":"New_Role"}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 25, "time_elapsed": 0.032494}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'name': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	new name of the role
id	N/A	False	Integer(gte: 0)	role id should be more than 10.

2.289 roles_get

Docstring:

Returns list of existing roles in the database.

Example:

- Client Request

```
{"session_id": "0945535c018205fee591b31fd11a1a8d",  
  "data": {"limit": 101,  
           "offset": 0,  
           "fields": ["id", "name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[1, "Owner"], [2, "Billing Department"],  
           [3, "Client."], [4, "Client. Show Ballance."],  
           [5, "Engineer"], [7, "Support"], [6, "Manager"],  
           [22, "super_user"], [23, "rrrr1"], [25, "New_Role"]],  
  "time_elapsed": 0.003879}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'id': 'Integer',  
          'is_client': 'Bool',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
is_client	False	True	Bool	
sort_by	[]	True	List([{"Enum('id', 'name')": Enum('asc', 'desc', "")}],	field name on which result will be sorted

			min_length: 0)	
limit	0	True	Integer(gte: 0)	rows quantity per one request
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
id	N/A	True	Integer(gte: 0)	role id
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
name	N/A	True	String(allow_blank: True)	role name

Default fields:

Name	Type & Comments
id	Int
name	String

2.290 routing_types_get

Docstring:

Returns list of possible routing algorithms for origination points.

Example:

- Client Request

```
{
  "session_id": "a9ce4915db464e2d30b22d1a1a5ef103",
  "data": {
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[1, "LCR"], [3, "PROFIT"], [2, "QUALITY"]]
}
```

```
"time_elapsed": 0.004059}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('id', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
id	Int
name	String

2.291 services_get

Docstring:

Function returns service name and appropriate service id.

Example:

- Client Request:

```
{"session_id": "4595aceb14c686aa9752bb9860462bf2",  
  "data": {"sort_by": [{"name": ""}],  
           "limit": 10, "offset": 0,  
           "fields": ["id", "name"]}  
}
```

- Server Response:

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[2, "sms_service"], [1, "voice_service"]],  
  "time_elapsed": 0.013664}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('id', 'name'): Enum('asc', 'desc', "")], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
id	Int
name	String

2.292 sip_authorization_add

Docstring:

Creates new SIP authoriation account (account for authorization by login and password) within chosen context and OP.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "op_id": 4331,
    "context_id": 1,
    "login": "test_sip",
    "password": "121314"
  }
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',
          'login': 'String',
          'op_id': 'Integer',
          'password': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	context to which authorization will be added
password	N/A	False	String(allow_blank:	SIP auth password

			False)	
login	N/A	False	String(allow_blank: False)	SIP auth login. Should be unique.
op_id	N/A	False	Integer(gt: 0)	Origination point id to which given authorization will be added

2.293 sip_authorization_del

Docstring:

Removes chosen SIP authorization from the given context.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "context_id": 1,
    "login": "test_sip"
  }
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer', 'login': 'String', 'op_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
login	N/A	True	String(allow_blank: False)	SIP account login
context_id	N/A	False	Integer(gt: 0)	context in which given SIP authorization have to be removed
op_id	N/A	True	Integer(gt: 0)	filter by OP id

2.294 sip_authorization_get

Docstring:

Function retrieves SIP accounts existent in the database.

Example:

- Client Request:

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {"op_id": 231,  
           "context_id": 1,  
           "fields": ["login"]}  
}
```

- Server Response:

```
{"status": 200,  
  "fields": ["login"],  
  "message": "OK",  
  "data": [["test_sip"]],  
  "time_elapsed": 0.008435}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',  
          'fields': ["Enum('context_id', 'op_id', 'password', 'login')"],  
          'login': 'String',  
          'op_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	True	Integer(gt: 0)	filter by Origination Point
login	N/A	True	String(allow_blank: False)	filter by SIP account login
fields	['context_id', 'op_id', 'password', 'login']	True	List([Enum('context_id', 'op_id', 'password', 'login')], min_length: 0)	required fields If absent - takes fields specified in "default fields". If empty - returns error.
context_id	N/A	True	Integer(gt: 0)	filter by context. If absent function retrieves authorizations from all

				contexts.
--	--	--	--	-----------

Default fields:

Name	Type & Comments
context_id	Int
op_id	Int
password	String
login	String

2.295 sip_authorization_set

Docstring:

Function allows to change password of chosen SIP authorization in the given context.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "login": "test_sip",
    "context_id": 1,
    "password": "132456567"
  }
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer', 'login': 'String', 'password': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
context_i	N/A	False	Integer(gt: 0)	context where given SIP authorization have to

d				be changed
password	N/A	False	String(allow_blank: False)	new password
login	N/A	False	String(allow_blank: False)	SIP account login which password have to be changed

2.296 sms_activity_points_get

Docstring:

Returns data to draw attempts charts for the given period.
 "start_time" field contains start time of the period as POSIX timestamp (int).

"day" period contains data 2 last days.

The "data" field contains a list of 1440 samples, data[i], i=0..1439.
 Here i - sequence number.

The data[i] item denotes the instantaneous SMS rate (expressed in messages per second) at the moment of time t[i]. This moment of time is computed as follows:

$t[i] = \text{start_time} + \text{delta} * (i+1)$,
 where delta = 120 seconds.

Example:

- Client Request

```
{
  "session_id": "1d81ff9b02a2ea9f55df0cf4327f3124",
  "data": {
    "period": "month"
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "data": [
    ["194.145.39.19", [426.819576966, 219.767931342, 9.5516722038, 1.85
424038046, 1.03966881114, ...]]
  ],
  "time_elapsed": 0.226724,
  "start_time": 1459345800
}
```

Request structure and fields description:

```
{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')",
          'rate': "Enum('per_sec', 'per_min')"},
        'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.
rate	N/A	False	Enum('per_sec', 'per_min')	Sms interest rate.

2.297 sms_change_dests_min_max

Docstring:

Function adds new request 'change min/max digits' to the queue to change 'min_len/max_len' in SMS prices with given destination code (or given code with subcodes) or given MCC and MNC for all SMS OP/TP tariffs.

Example:

- Client Request

```
{ "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "sub_dest": false,
    "is_tp": true,
    "mcc": "255",
    "mnc": "05",
    "min_len": 11,
    "max_len": 12
  }
}
```

Request structure and fields description:

```
{'data': {'destination': 'String',
          'is_tp': 'Bool',
          'max_len': 'Integer',
          'mcc': 'String',
          'min_len': 'Integer',
          'mnc': 'String',
          'sub_dest': 'Bool'}}
```

```
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sub_dest	False	True	Bool	if True changes "min_len"/"max_len" digits prices for given destination with sub codes. It works with destinations tariffs only!
mnc	""	True	String(allow_blank: True, regex: "^(\d{1,3})?\$")	MNC
min_len	1	True	Integer(gte: 1, lte: 100)	Defines the minimal length of destination code for current price.
is_tp	True	True	Bool	if True (is default value) changes "min_len"/"max_len" only for prices of TP tariffs, else - only for prices of OP tariffs
max_len	24	True	Integer(gte: 1, lte: 100)	Defines the maximum length of destination code for current price.
destination	N/A	True	String(allow_blank: True, regex: "^([0-9]*)?\$")	destination code
mcc	""	True	String(allow_blank: True, regex: "^(\d{1,3})?\$")	MCC

2.298 sms_content_get

Docstring:

Request structure and fields description:

```
{'data': {'bill_id': 'String',  
          'fields': ["Enum('sms_text',)"]},
```

```

        'limit': 'Integer',
        'offset': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	Rows quantity per one request
fields	['sms_text']	True	List([Enum('sms_text'),], min_length: 0)	Required fields. If absent - takes fields specified in "default fields". If empty - returns error.
bill_id	N/A	False	String(allow_blank: False)	Incoming SMS bill id

Default fields:

Name	Type & Comments
sms_text	String

2.299 sms_content_replacement_rule_add

Docstring:

Request structure and fields description:

```

{'data': {'data_coding': "Enum('UCS2', 'Default(GSM7)', 'Latin1')",
          'destinations': 'String',
          'point_id': 'Integer',
          'replacement_pairs': [['String']]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
data_coding	N/A	False	Enum('UCS2', 'Default(GSM7)', 'Latin1')	SMS data coding. Possible values: "Default(GSM7)", "Latin1", "UCS2"
replacement_pairs	N/A	False	List([List([String(allow_blank: False, max_length: 50)], min_length: 0)], min_length: 0, max_length: 10)	List of replacement from->to pairs lists. Max length is 10 pairs. Example: [{"From_1", "ToTo_1"}, {"From_2", "ToTo_2"}, ...]
point_id	N/A	False	Integer(gte: 0)	SMS OP/TP id. "0" value apply rules for all OP/TP points
destinations	N/A	False	String(allow_blank: False, regex: "^(\\d+(, \\s - \\s)?)*\$", max_length: 120)	Destination code(s) for replacement rule. Comma-separated string. Max length is 120 symbols

2.300 sms_content_replacement_rule_del

Docstring:

Request structure and fields description:

```
{'data': {'id': 'Integer'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer()	Replacement Rule id to be deleted

2.301 sms_content_replacement_rule_get

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('destinations', 'text_from', 'data_coding', 'point_id', 'point_type', 'text_to', 'id', 'point_name')"],
            'id': 'Integer',
            'limit': 'Integer',
            'offset': 'Integer',
            'point_id': 'Integer',
            'point_name': 'Integer',
            'sort_by': [{"Enum('destinations', 'text_from', 'data_coding', 'point_id', 'point_type', 'text_to', 'id', 'point_name')": "Enum('asc', 'desc', '')"}]},
            'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	True	Integer(gt: 0)	Replacement Rule id in the database
limit	0	True	Integer(gte: 0)	Rows quantity returned per one request
fields	['destinations', 'text_from', 'data_coding', 'point_id', 'point_type', 'text_to', 'id', 'point_name']	True	List([Enum('destinations', 'text_from', 'data_coding', 'point_id', 'point_type', 'text_to', 'id', 'point_name')], min_length: 0)	Required field. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('destinations', 'text_from', 'data_coding', 'point_id', 'point_type', 'text_to', 'id', 'point_name'): Enum('asc', 'desc', '')}], min_length: 0)	Field for result list sorting
point_name	N/A	True	Integer()	SMS OP/TP name in the database
point_id	N/A	True	Integer(gte: 0)	SMS OP/TP id in the database. "point_id = 0"

				means rules that works for all SMS points
offset	0	True	Integer(gte: 0)	Offset says how many rows will be skipped before beginning of result

Default fields:

Name	Type & Comments
destinations	String(120)
text_from	String(50)
data_coding	String
point_id	Int
point_type	String
text_to	String(50)
id	Int
point_name	String

2.302 sms_content_replacement_rules_get_all

Docstring:

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.303 sms_dialpeer_add

Docstring:

Creates a new dialpeer. This function also used for copying already existent dialpeer to given context and its parameters:

- dialpeer -> TP relations
 - dialpeer -> Group relations
- and so on.

Function returns id of created dialpeer. If dialpeer was added to several contexts their id will be the same.

Examples:

- add simple dialpeer:

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
 "data": {
   "name": "test_dp",
   "dstregex": ".*",
   "enabled": True,
   "contexts": [1],
 }
}
```

- add dialpeer with predefined Groups and TPs relations:

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
 "data": {
   "name": "test_dp",
   "dstregex": ".*",
   "enabled": True,
   "contexts": [1],
   "groups_ids": [[935], [855]],
   "tp_ids": [[4536, 0, 0], [4730, 0, 0]],
   "current_context": 2,
 }
}
```

where:

"tp_ids" - is a lists which contain following data:

- TP id linked with the dialpeer,
- dialpeer -> TP relation priority,
- dialpeer -> TP relation weight;

"group_ids" - list of group ids which have to be connected with chosen dialpeer;

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "name": String(),
    "dstregexp": String(),
    "priority": Int(gte=-100, lte=100),
    "enabled": Bool(),
    "stop": Bool(),
    "denyregexp": String(allow_blank=True, max_length=dst_deny_limit),
    "ani_regexp": Ani_Regexp(max_length=64, allow_blank=True),
    "dnis_regexp": Dnis_Regexp(max_length=64, allow_blank=True),
    "ani_block": String(allow_blank=True),
    "delta_price": Float(),
    "delta_price_unit": Int(gt=0),
    "capacity": Int(gte=0, lte=DP_CAPACITY_LIM),
    "contexts": List(Int(gt=0)),
    "current_context": Int(gt=0),
    "tp_ids": List(Tuple(Int(gt=0), Int(gte=-100, lte=100), Int(gt=0, lte=100))),
    "groups_ids": List(List(Int(gt=0))),
    "invisible": Bool(),
  }
}
```

Name	Default value	Optional	Type	Description
"name"	N/A	False	String()	name of dialpeer or new name for dialpeer's copy. Name should be unique. Dialpeer's name can't be "Service_dialpeer_" because this name reserved for configuration in Simple Routing mode.
"dstregexp"	N/A	False	String()	
"priority"	0	True	Int(gte=-100, lte=100)	integer number, dialpeer's priority in range [-100:+100]
"enabled"	False	True	Bool()	state of dialpeer, if "True" then dialpeer is enabled
"stop"	True	True	Bool()	boolean option, if "True" then all dialpeers from global dialpeers tree that shorter than current won't

				participate in the routing.
"denyregexp"	"	True	String(allow_blank=True, max_length=dst_deny_limit)	regular expression which defines denied destinations for given dialpeer. For example "^3.*\$" block calls to destinations which begins with 3, a few expressions should be separated by comma.
"ani_regexp"	"	True	Ani_Regexp(max_length=64, allow_blank=True)	string variable which defines conversion of the caller number with help of regular expression
"dnis_regexp"	"	True	Dnis_Regexp(max_length=64, allow_blank=True)	string variable which defines conversion of the called number with help of regular expressions
"ani_block"	"	True	String(allow_blank=True)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
"delta_price"	0	True	Float()	this floating value allows to permit only routes with necessary profitability. If distinction between OP price and TP price in the default currency less than this value then given TP won't participate in routing. This value specifies in default currency.
"delta_price_unit"	1	True	Int(gt=0)	delta price unit type
"capacity"	1000	True	Int(gte=0, lte=DP_CAPACITY_LIM)	integer value from 0 to 1000, which means amount of concurrent calls and call attempts through the dialpeer
"contexts"	N/A	True	List(Int(gt=0))	context ids to which dialpeer will be added
"current_context"	N/A	True	Int(gt=0)	context id to which groups relations will be added
"tp_ids"	N/A	True	List(Tuple(Int(gt=0), Int(gte=-100, lte=100), Int(gt=0, lte=100)))	nested lists contains TP -> Dialpeer relation data which can be copied to another context(s) in the following order: TP id to link it with the dialpeer, dialpeer -> TP relation

				priority and dialpeer -> TP relation weight
"groups_ids"	N/A	True	List(List(Int(gt=0)))	list of groups existed in the chosen context, to which given dialpeer will be added. Use this parameter in conjunction with "current_context" option.
"invisible"	False	True	Bool()	option that makes dialpeer invisible. relations between dp and tp, dp and group will be removed.

2.304 sms_dialpeer_groups_get

Docstring:

Returns list of groups connected with chosen dialpeer.

Example:

- Client Request

```
{
  "session_id": "593a9260cb1406179301fe2c2680e0fd",
  "data": {
    "sort_by": [
      {
        "name": ""
      }
    ],
    "name": "",
    "contexts": [1],
    "dialpeer_id": 4,
    "fields": ["group_id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["group_id", "name"],
  "message": "OK",
  "data": [[3, "h_p (CZ)"]],
  "time_elapsed": 0.007634
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "dialpeer_id": Int(gt=0),
    "name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"dialpeer_id"	N/A	False	Int(gt=0)	dialpeer id
"name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"contexts"	[]	False	List(Int())	contexts id where need search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field name on which result will be sorted.
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'dialpeer_id'	Int
'group_id'	Int
'name'	String

2.305 sms_dialpeer_set**Docstring:**

Updates general settings of dialpeer with given id within given context(s) and in schema routing.

This function disallows to edit service dialpeers.
Function returns id of dialpeer that was edited.

Example:

- Client Request

```
{ "session_id": "ba6c2e2f97ebddf6910369b2e48b19a1",  
  "data": { "id": 6,  
            "stop": true,  
            "enabled": true,  
            "denyregexp": "^123.*$",  
            "priority": 0,  
            "delta_price": -100.0,  
            "name": "Test_DP",  
            "ani_regexp": "",  
            "contexts": [1],  
            "dstregexp": "^.*",  
            "dnis_regexp": "" }  
}
```

- Server Response

```
{ "status": 200, "message": "OK", "id": 6, "time_elapsed": 0.039199 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "dstregex": String(),
    "priority": Int(gte=-100, lte=100),
    "enabled": Bool(),
    "stop": Bool(),
    "ani_regex": Ani_Regex(max_length=64, allow_blank=True),
    "dnis_regex": Dnis_Regex(max_length=64, allow_blank=True),
    "denyregex": String(allow_blank=True, max_length=dst_deny_limit),
    "ani_block": String(allow_blank=True),
    "delta_price": Float(),
    "delta_price_unit": Int(gt=0),
    "capacity": Int(gte=0, lte=DP_CAPACITY_LIM),
    "contexts": List(Int(gt=0)),
    "invisible": Bool(),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	id of the dialpeer in database
"name"	N/A	True	String()	dialpeer name
"dstregex"	N/A	False	String()	
"priority"	N/A	True	Int(gte=-100, lte=100)	dialpeer's priority in range from -100 to +100
"enabled"	N/A	True	Bool()	dialpeer's state, if True - dialpeer is enabled
"stop"	N/A	True	Bool()	boolean option, if "True" then all dialpeers from global dialpeers tree that shorter than current won't participate in the routing
"ani_regex"	N/A	True	Ani_Regex(max_length=64 , allow_blank=True)	string variable which defines conversion of the caller number with help of regular expression; max length of regex is 64

"dnis_regexp"	N/A	True	Dnis_Regexp(max_length=64, allow_blank=True)	string variable which defines conversion of the called number with help of regular expressions; max length of regex is 64
"denyregexp"	N/A	True	String(allow_blank=True, max_length=dst_deny_limit)	regular expression which defines denied destinations for given dialpeer. For example "^3.*\$" block calls to destinations which begins with 3, a few expressions should be separated by comma.
"ani_block"	N/A	True	String(allow_blank=True)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
"delta_price"	N/A	True	Float()	this floating value allows to permit only routes with necessary profitability. If distinction between OP price and TP price in the default currency less than this value then given TP won't participate in routing. This value specifies in default currency.
"delta_price_unit"	N/A	True	Int(gt=0)	delta price unit type
"capacity"	N/A	True	Int(gte=0, lte=DP_CAPACITY_LIM)	integer value from 0 to 1000 which means amount of concurrent calls and call attempts through the dialpeer
"contexts"	N/A	False	List(Int(gt=0))	contexts ids where chosen dialpeer will be changed
"invisible"	N/A	True	Bool()	option that makes dialpeer invisible. relations between dp and tp, dp and group will be removed.

2.306 sms_dialpeer_tp_set

Docstring:

Changes priority and weight of given TP <--> Dialpeer relations within chosen context(s). Function doesn't allow to change relations of service dialpeer.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"weight": 0,  
           "priority": 3,  
           "contexts": [1],  
           "dialpeer_id": 6, "tp_id": 8}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.030787}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "tp_id": Int(gt=0),  
    "dialpeer_id": Int(gt=0),  
    "priority": Int(gte=-100, lte=100),  
    "weight": Int(gte=0, lte=100),  
    "contexts": List(Int(gt=0)),  
  }  
}
```

Name	Default value	Optional	Type	Description
"tp_id"	N/A	False	Int(gt=0)	termination point id in database
"dialpeer_id"	N/A	False	Int(gt=0)	id of dialpeer in database
"priority"	N/A	True	Int(gte=-100, lte=100)	relation priority in range from -100 to 100 inclusive

"weight"	N/A	True	Int(gte=0, lte=100)	weight of the relation in range from 0 to 100 inclusive
"contexts"	N/A	False	List(Int(gt=0))	contexts ids where need change given relation

2.307 sms_dialpeer_tps_op_get

Docstring:

Returns termination points connected with chosen OP via chosen group and dialpeer in the given context(s).

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "limit": 101,
    "op_id": 3,
    "sort_by": [{"tp_name": ""}],
    "group_id": 6,
    "contexts": [1],
    "offset": 0,
    "dialpeer_id": 6,
    "fields": ["dialpeer_id", "tp_id", "tp_name", "priority", "weight"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["dialpeer_id", "tp_id", "tp_name", "priority", "weight"],
  "message": "OK",
  "data": [[6, 4, "Test_1_TP", 0, 0]],
  "time_elapsed": 0.010412
}
```

Request structure and fields description:

Request:

```
{
```

```

"session_id": String(),
"data": {
  "op_id": Int(gt=0),
  "dialpeer_id": Int(gt=0),
  "group_id": Int(gt=0),
  "tp_name": String(allow_blank=True),
  "contexts": List(Int(gt=0)),
  "fields": List(String()),
  "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
  "limit": Int(gte=0),
  "offset": Int(gte=0),
}
}

```

Name	Default value	Optional	Type	Description
"op_id"	N/A	False	Int(gt=0)	chosen OP id
"dialpeer_id"	N/A	False	Int(gt=0)	chosen dialpeer id
"group_id"	N/A	False	Int(gt=0)	chosen group id
"tp_name"	N/A	True	String(allow_blank=True)	filter by first symbols of termination point name
"contexts"	N/A	False	List(Int(gt=0))	required contexts ids
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field on which list of groups and dialpeers will be sorted
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'dialpeer_id'	Int
'tp_id'	Int
'priority'	Int
'tp_name'	String
'weight'	Int

2.308 sms_dialpeers_free_group_get**Docstring:**

Returns list of dialpeers which don't related with chosen group.

Example:

- Client Request

```
{ "session_id": "677232544e5f894e13cc8a3f906504bd",  
  "data": { "sort_by": [ { "name": "" } ],  
            "name": "",  
            "group_id": 3,  
            "contexts": [ 1 ],  
            "fields": [ "id", "name" ] }  
}
```

- Server Response

```
{ "status": 200,  
  "fields": [ "id", "name" ],  
  "message": "OK",  
  "data": [ [ 3, "st-dialpeer2 (CZ)" ], [ 2, "st-dialpeer (CZ)" ] ],  
  "time_elapsed": 0.007284 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "group_id": Int(gt=0),
    "name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"group_id"	N/A	True	Int(gt=0)	group id
"name"	N/A	True	String(allow_blank=True)	filter by dialpeer name or fist symbols of name
"contexts"	[]	False	List(Int())	contexts id where need search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field name on which result will be sorted.
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows

Default fields:

Name	Type & Comments
"id"	Int
"name"	String

2.309 sms_dialpeers_from_groups_del**Docstring:**

Removes relations between chosen dialpeers and groups. Function doesn't allow to remove relations between groups and service dialpeers and raises error if user tries to do it.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"dialpeers_ids": [6],  
           "contexts": [1],  
           "groups_ids": [1]}}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.041017}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "dialpeers_ids": List(Int(gt=0)),  
    "groups_ids": List(Int(gt=0)),  
    "contexts": List(Int(gt=0)),  
  }  
}
```

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

"dialpeers_ids"	N/A	False	List(Int(gt=0))	list of dialpeer ids
"groups_ids"	N/A	False	List(Int(gt=0))	list of group ids
"contexts"	N/A	False	List(Int(gt=0))	contexts ids where need remove specified relations

2.310 sms_dialpeers_get

Docstring:

Returns dialpeers data by chosen criteria.

If dialpeer's id don't set explicitly in query function returns parameters of dialpeers within given context(s) which satisfy 'name' filter.

If dialpeer id specified - function returns settings of chosen dialpeer within given context(s).

Example:

- Client Request

```
{
  "session_id": "4dcfe8adb532e2b3856ada8caea50506",
  "data": {
    "id": 2,
    "contexts": [1],
    "fields": [
      "id", "name", "dstregexp", "priority", "enabled",
      "stop", "denyregexp", "ani_regexp", "dnis_regexp",
      "delta_price", "name_context"
    ]
  }
}
```

- Server response

```
{
  "status": 200,
}
```

```

"fields": ["id", "name", "dstregex", "priority", "enabled",
           "stop", "denyregex", "ani_regex", "dnis_regex",
           "delta_price", "name_context"],
"message": "OK",
"data": [[2, "ikex.dialpeer", "^.*", 0, true, false, "", "",
           "^(.*)$/0001\1", "0.00000", "ikex.dialpeer (CZ)"]],
"time_elapsed": 0.008622}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(allow_blank=True),
    "dstregex": String(allow_blank=True, max_length=dst_deny_limit),
    "enabled": Bool(),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
    "invisible": Bool() | Null(),
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	True	Int(gt=0)	dialpeer id
"name"	N/A	True	String(allow_blank=True)	first symbols of dialpeer name
"dstregex"	N/A	True	String(allow_blank=True, max_length=dst_deny_limit)	
"enabled"	N/A	True	Bool()	filter by enabled/disabled dialpeers
"contexts"	[]	False	List(Int(gt=0))	required context ids
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) (\$\$)))	field name on which result will be sorted
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.
"invisible"	False	True	Bool() Null()	filter by visible/invisible

Default fields:

Name	Type & Comments
'id'	Int
'name'	String
'name_context'	String (dialpeer name with context name)
'dstregexp'	String
'priority'	Int
'enabled'	Boolean
'stop'	Boolean
'denyregexp'	String
'ani_regexp'	String

'dnis_regexp'	String
'ani_block'	String
'capacity'	String
'delta_price'	numeric(16,5)
'invisible'	Boolean
'delta_price_unit',	Int

2.311 sms_dialpeers_tp_get

Docstring:

Returns information about dialpeers('dstregexp', 'denyregexp', 'ani_regexp', dialpeer status and so on) that connected with chosen termination point and chosen group.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "limit": 101,
    "sort_by": {
      "name": ""
    },
    "group_id": 6,
    "contexts": [1],
    "offset": 0,
    "tp_id": 4,
    "fields": ["id", "name", "dstregexp", "priority", "enabled", "stop"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name", "dstregexp", "priority", "enabled", "stop"],
  "message": "OK",
  "data": [[6, "Test_DP", "^.*", 0, true, true]],
  "time_elapsed": 0.008475
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tp_id": Int(gt=0),
    "group_id": Int(gt=0),
    "dialpeer_name": String(allow_blank=True),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"tp_id"	N/A	False	Int(gt=0)	termination point id
"group_id"	N/A	False	Int(gt=0)	group_id
"dialpeer_name"	N/A	True	String(allow_blank=True)	filter by first symbols of dialpeer name
"contexts"	[]	False	List(Int(gt=0))	required contexts ids
"fields"	default_fields	True	List(String())	
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	column name on which result sorts
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'id'	Int
'name'	String - dialpeer's name
'dstregexp'	String
'priority'	Int
'enabled'	Boolean
'stop'	Boolean
'denyregexp'	String
'ani_regexp'	String
'dnis_regexp'	String
'delta_price'	numeric(16,5)

2.312 sms_dialpeers_tps_get**Docstring:**

Returns list of dialpeers binded with chosen TP or TPS binded with chosen dialpeer and parameters (priority and weight) of their relation.

Function can filter output by TP's name or dialpeer's name. Filter by TP name should be used in conjunction with dialpeer's id, filter by dialpeer name - in conjunction with TP's id.

Example:

- Client Request

```
{"session_id":"773b6b683024ac6b6cfd95c8b034e659",
```

```

"data":{"sort_by":[{"dialpeer_name":""}],
"contexts":[1],
"tp_id":4,
"dialpeer_name":"","
"fields":["dialpeer_id","dialpeer_name"]}
}

```

- Server Response

```

{"status": 200,
"fields": ["dialpeer_id", "dialpeer_name"],
"message": "OK",
"data": [[6, "Test_DP (CZ)"]],
"time_elapsed": 0.008504}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "tp_id": Int(gt=0),
    "dialpeer_id": Int(gt=0),
    "tp_name": String(allow_blank=True),
    "dialpeer_name": String(allow_blank=True),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"tp_id"	N/A	True	Int(gt=0)	termination point id
"dialpeer_id"	N/A	True	Int(gt=0)	dialpeer id
"tp_name"	N/A	True	String(allow_blank=True)	filter by first symbols of TP name
"dialpeer_name"	N/A	True	String(allow_blank=True)	filter by first symbols of dialpeer name
"contexts"	N/A	False	List(Int(gt=0))	required contexts ids
"fields"	default_fields	True	List(String())	requested fields, by defaults names

	elds			mentioned in "default fields"
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field on which list will be sorted
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
"dialpeer_id"	Int
"tp_id"	Int
"priority"	Int
"tp_name"	String
"dialpeer_name"	String
"weight"	Int

2.313 sms_end_upload

Docstring:

Function for internal usage and Accucore.

Raise error in case if db function returns price status not equal 2.

If prices are correct it starts prices processing in background.

Function raises error in the following cases:

- 1) Price list is empty
 - 2) Price list contains several prices for the same destination
 - 3) 'full' price list contains prices with different 'start_date'
- etc.

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.314 sms_fin_summary_get

Docstring:

Returns sms financial report for chosen time period and company(ies) with statistical information.

There are time interval limitations for each "type" in the API configuration.

If "type"=6 function returns list of manager ids and names("managers" param) for which financial data is available in case of interval has been exceeded.
For other types function returns list of companies ids and names("clients" param) for which financial data is available in case of interval has been exceeded.
If there are no clients or managers available then returns empty data and required fields list.

By defaults limits are following:

- 1) for "type" 1, 2 and 3 - 2 days
- 2) for "type" 4, 5, 6 and 7 - 7 days

Function provides for each grouped data "inbound" and "outbound" parts if any (

```
"type" field in resulted data)
and "direction":0.
"inbound" - OP sms traffic
"outbound" - TP sms traffic
```

NOTE! "agg_last_time" field - defines aggregation last time.

Example:

- Client Request (report interval has been exceeded)

```
{
  "session_id":"24608c71d51739a386d5947d37a00aab",
  "data":{
    "time_zone":"0:00",
    "currency_id":1,
    "end_date":"2014-10-13T08:59:59",
    "direction":0,
    "type":4,
    "start_date":"2014-10-01T08:00:00",
    "fields":[
      "company","tot","type","point_name","destination","country","op
_price","tp_price",
      "attempts","sa","asr","op_sum","tp_sum","delta_price","delta_pr
ofit","is_closed","manager"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.148999,
  "agg_last_time": "2014-10-13 08:00:00",
  "fields": [
    "company", "tot", "type", "point_name", "destination", "country", "
op_price",
    "tp_price", "attempts", "sa", "asr", "op_sum", "tp_sum", "delta_pri
ce",
    "delta_profit", "is_closed", "manager"
  ],
  "message": "OK",
  "clients": [
    [112, "Infobip"], [136, "SMS Test"], [105, "SMS_NEW"]
  ],
  "data": [
    ["Infobip", "", "outbound", "Info_flat_out", "", "", null, null, 32
, 31, "96.88%",
    "0.00310", "0.41633", "-0.41323", "-99.26%", false, "sb"],
    ["Infobip", "Total", "outbound", "", "", "", null, null, 32, 31, "9
6.88%",
    "0.00310", "0.41633", "-0.41323", "-99.26%", false, "sb"]
  ]
}
```

Request structure and fields description:

```
{'data': {'asr_less_than': 'Float',
          'asr_more_than': 'Float',
```

```

        'codes_exc': ['String'],
        'codes_inc': ['String'],
        'companies': ['Integer'],
        'context_id': 'Integer',
        'currency_id': 'Integer',
        'direction': 'Integer',
        'end_date': 'String',
        'fields': ["Enum('sa', 'direction', 'mnc', 'asr', 'manager', 'country', 'company', 'destination', 'delta_price', 'tp_sum', 'tot', 'attempts', 'delta_profit', 'traffic_share', 'op_sum', 'mcc', 'point_name', 'total', 'tp_price', 'is_closed', 'op_price')"],
        'managers': ['Integer'],
        'min_attempts': 'Integer',
        'network_code': 'String',
        'owner_id': 'Integer',
        'session_id': 'String',
        'start_date': 'String',
        'time_zone': 'String',
        'type': 'Integer'},
    'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
min_attempts	0	True	Integer(gte: 0)	display rows where attempts more than "min_attempts"
codes_inc	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that should be present in reply
asr_less_than	N/A	True	Float(gt: 0.0)	filter by asr, returns rows with asr less than specified
codes_exc	[]	True	List([String(allow_blank: True)], min_length: 0)	filter by destination codes, specify here codes that should not be present in reply
fields	['sa', 'direction', 'mnc', 'asr', 'manager', 'country', 'company', 'destination', 'delta_price', 'tp_sum', 'tot', 'attempts', 'delta_profit', 'traffic_share', 'op_sum', 'mcc', 'point_name', 'total', 'tp_price',	True	List([Enum('sa', 'direction', 'mnc', 'asr', 'manager', 'country', 'company', 'destination', 'delta_price', 'tp_sum', 'tot', 'attempts', 'delta_profit', 'traffic_share', 'op_sum', 'mcc', 'point_name', 'total', 'tp_price', 'is_closed',	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

	'is_closed', 'op_price']		'op_price')], min_length: 0)	
asr_more_than	N/A	True	Float(gte: 0.0)	filter by asr, returns rows with asr more than specified
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):(\d{2})\$")	requires if company works in timezone different from GMT+0
owner_id	0	True	Integer(gte: 0)	filter by owner
direction	0	True	Integer(gte: 0)	display only incoming(by OPs, 1) or outgoing(TPs, 2) traffic, For 0 (default value) - both directions
context_id	N/A	True	Integer(gt: 0)	filter by context
type	N/A	False	Integer(gte: 1, lte: 9)	report type id. 1 - Individual report for each destination, price, point and company (inbound and outbound), 2 - by destination name, 3 - by destination name and prices, 4 - by points, 5 - total for all companies, 6 - by manager, 7 - by companies, 8 - by country, 9 - by MCC MNC.
start_date	N/A	False	String(allow_blank: False)	
managers	N/A	True	List([Integer(gt: 0)], min_length: 0)	filter by manger id
network_code	""	True	String(allow_blank: True)	filter by MCC MNC
session_id	N/A	True	String(allow_blank: False)	by defaults from user's session
end_date	N/A	False	String(allow_blank: False)	
companies	[0]	True	List([Integer(gte: 0)], min_length: 0)	filter by company ids. [0] means all companies. Currently only first company id from the list is being taken

				into account.
currency_id	N/A	True	Integer(gt: 0)	display prices and cost in chosen currency. By defaults from user's session

Default fields:

Name	Type & Comments
sa	Int - quantity of successful attempts
direction	Int (1 - inbound, 2 - outbound)
mnc	String
asr	String
manager	String
country	String
company	String
destination	String
delta_price	String or Null (Null if hide_owners_profit enabled and different manager)
tp_sum	String or Null (Null if hide_owners_profit enabled and different manager)
tot	String - like as 'total' field (DEPRECATED, backport)
attempts	Int
delta_profit	String or Null (Null if hide_owners_profit enabled and different manager)
traffic_share	String

op_sum	String or Null (Null if hide_owners_profit enabled and different manager)
mcc	String
point_name	String
total	String
tp_price	String or Null (Null if hide_owners_profit enabled and different manager)
is_closed	Boolean (if True - dest is closed)
op_price	String or Null (Null if hide_owners_profit enabled and different manager)

2.315 sms_group_add

Docstring:

Creates new group in the given context(s).
Function returns id of the created group.

Example:

- Client Request

```
{ "session_id": "4595aceb14c686aa9752bb9860462bf2",
  "data": { "name": "1212121",
            "contexts": [1],
            "internal": false }
}
```

- Server Response

```
{ "status": 200,
  "message": "OK",
  "id": 1398,
  "time_elapsed": 0.081872 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
```

```

    "name": String(max_length=80),
    "internal": Bool(),
    "contexts": List(Int(gt=0)),
}
}

```

Name	Default value	Optional	Type	Description
"name"	N/A	False	String(max_length=80)	group name, should be no longer than 24 symbols length. Group name shouldn't ends with '_group_simple' because this suffix is designed for service groups in Simple Routing.
"internal"	False	True	Bool()	option that makes group internal (if this flag is "true") or not (if "false"). Point, which is added to the internal group can not be added or belong to another group. Thus, adding point to internal group you at the same time remove existant group relations of given point.
"contexts"	N/A	False	List(Int(gt=0))	list of contexts id to which group will be added

2.316 sms_group_copy

Docstring:

Creates new group with same parameters as given one, including group relations with points and dialpeers.

New group will be created in the same context as existent one but with another name.

Function returns id of created group.

Example:

- Client Request

```
{ "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": { "id": 6,
            "name": "test_group",
            "contexts": [1],
            "internal": false }
}
```

- Server Response

```
{ "status": 200, "message": "OK", "id": 8, "time_elapsed": 0.062069 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "contexts": List(Int(gt=0)),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	id of existent group
"name"	N/A	False	String()	name of new group, should be unique
"contexts"	N/A	False	List(Int(gt=0))	context id of parental group, on which new group will be created.

2.317 sms_group_del

Docstring:

Deletes chosen group from the given context(s).
Function returns group id.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",  
  "data": {"id": 7,  
           "contexts": [1]}  
}
```

- Server response

```
{"status": 200, "message": "OK", "id": 7, "time_elapsed": 0.07674}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "id": Int(gt=0),  
    "contexts": List(Int(gt=0)),  
  }  
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	group id
"contexts"	N/A	False	List(Int(gt=0))	context(s) where need to delete group with given id.

2.318 sms_group_dialpeers_get

Docstring:

Returns list of dialpeers that assigned to chosen group and dialpeers's settings.

Example:

- Client Request

```
{"session_id":"677232544e5f894e13cc8a3f906504bd",
  "data":{"dp_name":"","
    "sort_by":[{"dp_name":""}],
    "group_id":1,
    "contexts":[1],
    "fields":["dialpeer_id","dp_name"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["dialpeer_id", "dp_name"],
  "message": "OK",
  "data": [[2, "Test_DP (CZ)"]],
  "time_elapsed": 0.006144}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "dialpeer_id": Int(gt=0),
    "group_id": Int(gt=0),
    "dp_name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"dialpeer_id"	N/A	True	Int(gt=0)	filter by dialpeer id
"group_id"	N/A	True	Int(gt=0)	filter by group id
"dp_name"	N/A	True	String(allow_blank=True)	filter by dialpeer name or first symbols of dialpeer name
"contexts"	[]	False	List(Int())	contexts id where should search

				relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) (0\$")))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"dp_name": "asc"}] - sorts result ascending by field "dp_name"
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows..

Default fields:

Name	Type & Comments
'dialpeer_id'	Int
'group_id'	Int
'dp_name'	String
'dstregex'	String
'priority'	Int
'enabled'	Boolean
'stop'	Boolean

'group_name'	String
'dp_name_short'	String

2.319 sms_group_dp_tp_dnis_get

Docstring:

Returns list of Dialpeers, TP's and Dnis_regex that assigned to chosen group.

Example:

- Client Request

```
{ "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": { "group_id": 11,
            "contexts": [1]
            "fields": [ "group_id",
                        "group_name",
                        "dialpeer_id",
                        "dialpeer_name",
                        "dnis_regex",
                        "tp_id",
                        "tp_name"] }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "group_id": Int(gt=0),
    "group_name": String(allow_blank=True),
    "dialpeer_id": Int(gt=0),
    "dialpeer_name": String(allow_blank=True),
    "dnis_regex": String(allow_blank=True),
    "tp_id": Int(gt=0),
    "tp_name": String(allow_blank=True),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"group_id"	N/A	True	Int(gt=0)	chosen group id
"group_name"	N/A	True	String(allow_blank=True)	chosen group name
"dialpeer_id"	N/A	True	Int(gt=0)	filter by dialpeer id
"dialpeer_name"	N/A	True	String(allow_blank=True)	filter by dialpeer name
"dnis_regex"	N/A	True	String(allow_blank=True)	filter by Dnis_regex
"tp_id"	N/A	True	Int(gt=0)	filter by Termination Point id
"tp_name"	N/A	True	String(allow_blank=True)	filter by Termination Point name
"contexts"	N/A	False	List(Int(gt=0))	required contexts ids
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field on which list of groups and dialpeers will be sorted
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	offset says to skip that many rows before beginning to return rows

Default fields:

Name	Type & Comments
'group_id'	Int
'group_name'	String
'dialpeer_id'	Int
'dialpeer_name'	String
'dnis_regexp'	String
'tp_id'	Int
'tp_name'	String

2.320 sms_group_set**Docstring:**

Allows to rename or(and) set chosen group internal.
Function returns id of group which was modified.

Example:

- Client Request

```
{"session_id":"4595aceb14c686aa9752bb9860462bf2",  
  "data":{"id":1399,  
          "name":"121211",  
          "contexts":[1]}  
}
```

- Server response

```
{"status": 200,  
  "message": "OK",  
  "id": 1399,  
  "time_elapsed": 0.088758}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "name": String(),
    "internal": Bool(),
    "contexts": List(Int(gt=0)),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	group id
"name"	N/A	True	String()	new name of group. Group name shouldn't ends with '_group_simple' because this suffix is designed for service groups in Simple Routing.
"internal"	N/A	True	Bool()	Deprecated
"contexts"	N/A	False	List(Int(gt=0))	context where group parameters will be changed

2.321 sms_groups_free_dialpeer_get

Docstring:

Returns list of groups which are not assigned with chosen dialpeer.

Example:

- Client Request

```
{"session_id":"593a9260cb1406179301fe2c2680e0fd",
 "data":{"group_name":"","
        "sort_by":[{"group_name":""}]},
```

```

    "contexts":[1],
    "dialpeer_id":4,
    "fields":["group_id","group_name"]}
}

```

- Server Response

```

{"status": 200,
 "fields": ["group_id", "group_name"],
 "message": "OK",
 "data": [[2, "H_P_in_group_simple (CZ)", [1, "st.group (CZ)"]],
 "time_elapsed": 0.008413}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "dialpeer_id": Int(gte=0),
    "group_name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"dialpeer_id"	0	True	Int(gte=0)	
"group_name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"contexts"	[]	False	List(Int())	contexts id where search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field name on which result will be sorted.

"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
"group_id"	Int
"group_name"	String

2.322 sms_groups_free_op_get

Docstring:

Returns list of groups aren't connected with chosen OP.

Example:

- Client Request

```
{
  "session_id": "91af15315af776928e1abfa0e88b6641",
  "data": {
    "op_id": 19,
    "sort_by": [{"name": ""}],
    "name": "",
    "contexts": [1],
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
}
```

```

"message": "OK",
"data": [[6, "Alexander_test_group (CZ)", [3, "h_pak (CZ)"],
        [2, "H_Pak_in_group_simple (CZ)"], [1, "ikex.group (CZ)"],
        [8, "test_group (CZ)"]],
"time_elapsed": 0.000902}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "op_id": Int(),
    "name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"op_id"	N/A	True	Int()	
"name"	N/A	True	String(allow_blank=True)	filter results by group name or first symbols of group name
"contexts"	N/A	False	List(Int())	context(s) where need to search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"name": "asc"}] - sorts result ascending by field "name"
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
"id"	Int
"name"	String

2.323 sms_groups_free_tp_get**Docstring:**

Returns list of groups are not related with chosen Termination point.

Example:

- Client Request

```
{"session_id":"593a9260cb1406179301fe2c2680e0fd",  
  "data":{"sort_by":[{"name":""}],  
         "name":"","  
         "contexts":[1],  
         "tp_id":8,  
         "fields":["id","name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[2, "H_P_in_group_simple (CZ)"], [1, "st.group (CZ)"]],  
  "time_elapsed": 0.009848}
```

Request structure and fields description:

Request:


```

{
  "session_id": String(),
  "data": {
    "tp_id": Int(gt=0),
    "name": String(allow_blank=True),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
"tp_id"	N/A	True	Int(gt=0)	filter by termination point id
"name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"contexts"	[]	False	List(Int())	contexts id where need search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field name on which result will be sorted.
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows

Default fields:

Name	Type & Comments
'id'	Int
'name'	String

2.324 sms_groups_get**Docstring:**

Returns list of groups existent in the chosen context(s) and information about them.

Example:

- Client Request

```
{"session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",  
  "data": {"limit": 101,  
           "sort_by": [{"name": ""}],  
           "contexts": [1],  
           "offset": 0,  
           "fields": ["id", "name", "internal"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name", "internal"],  
  "message": "OK",  
  "data": [[6, "Test_group (CZ)", false],  
           [3, "h_p (CZ)", false],  
           [2, "H_in_group_simple (CZ)", false],  
           [1, "st.group (CZ)", false],  
           [9, "pablo_gr (CZ)", false],  
           [8, "test_group1 (CZ)", false]],  
  "time_elapsed": 0.008566}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "id": Int(gt=0),  
    "name": String(allow_blank=True),
```

```

    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
    "manager_id": Int(gte=0),
    "companies_id": List(Int(gte=0)),
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	True	Int(gt=0)	group id
"name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"contexts"	N/A	True	List(Int(gt=0))	context(s) id where search groups
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"name": "asc"}] - sorts result ascending by field "name"
"limit"	N/A	True	Int(gte=0)	says how many rows will be in response.
"offset"	N/A	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.
"manager_id"	N/A	True	Int(gte=0)	filtration by manager.
"companies_id"	N/A	True	List(Int(gte=0))	filtration by companies. It fills up automatically when manager id is given.

Default fields:

Name	Type & Comments
"id"	Int
"name"	String
"name_context"	String
"internal"	Boolean
"context_id"	Int

2.325 sms_groups_to_dialpeers_add**Docstring:**

Creates relations between chosen dialpeer and groups or chosen group and dialpeers.

This function disallows to add relation between groups and service dialpeer.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"dialpeers_ids": [6],  
           "contexts": [1],  
           "groups_ids": [1]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.039321}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),
```

```

"data": {
  "groups_ids": List(Int(gt=0)),
  "dialpeers_ids": List(Int(gt=0)),
  "contexts": List(Int(gt=0)),
}
}

```

Name	Default value	Optional	Type	Description
"groups_ids"	N/A	False	List(Int(gt=0))	list of groups ids
"dialpeers_ids"	N/A	False	List(Int(gt=0))	list of dialpeers ids which shouldn't contain ids of service dialpeers.
"contexts"	N/A	False	List(Int(gt=0))	contexts ids where need add relations will be added

2.326 sms_hlr_report_get

Docstring:

Request structure and fields description:

```

{'data': {'date_from': 'String',
          'date_to': 'String',
          'fields': ["Enum('price', 'mccmnc', 'cost', 'country_name', 'network_name', 'messages_count')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('price', 'mccmnc', 'cost', 'country_name', 'network_name', 'messages_count')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
date_to	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})(T)(\d{2}):(\d{2}):(\d{2})(\.\d{6}){0,1}\$")	start date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
date_from	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})(T)(\d{2}):(\d{2}):(\d{2})(\.\d{6}){0,1}\$")	start date in format YYYY-MM-DDTHH:MM:SS, where 'T' is delimiter between date and time
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows
sort_by	[]	False	List([Enum('price', 'mccmnc', 'cost', 'country_name', 'network_name', 'messages_count'): Enum('asc', 'desc', ")]), min_length: 0)	
fields	{'price': 'Decimal', 'mccmnc': 'String', 'cost': 'Decimal', 'country_name': 'String', 'network_name': 'String', 'messages_count': 'Int'}	True	List([Enum('price', 'mccmnc', 'cost', 'country_name', 'network_name', 'messages_count')], min_length: 0)	required fields If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
price	Decimal

mccmnc	String
cost	Decimal
country_name	String
network_name	String
messages_count	Int

2.327 sms_hosts_get

Docstring:

Returns list of sms hosts which assigned to SMS Origination and(or) SMS Termination points in chosen context(s).
The report displays the following information: point name to which host assigned, port and host record status - "enabled" or "disabled".

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "limit": 3,
    "sort_by": [{"host": ""}],
    "contexts": [1],
    "offset": 0,
    "fields": ["id", "host", "port", "enabled", "pt_id",
               "pt_name", "is_op"]}
}
```

- Server response

```
{
  "status": 200,
  "fields": ["id", "host", "port", "enabled", "pt_id",
             "pt_name", "is_op"],
  "message": "OK",
  "data": [[129, '188.230.126.180', 2775, True, 6, 'sms_tp_6', False]],
  "time_elapsed": 0.00966}
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'enabled': 'Bool',
          'fields': ['String'],
          'host': 'String',
          'limit': 'Integer',
          'offset': 'Integer',
          'pt_id': 'Integer',
```

```

        'pt_name': 'String',
        'sort_by': [{"Enum('pt_name', 'enabled', 'pt_id', 'host', 'is_op', 'port')":
"Enum('asc', 'desc', '')"}]],
        'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
host	N/A	True	String(allow_blank: True)	filter by IP address
pt_name	N/A	True	String(allow_blank: True)	filter by point name
enabled	N/A	True	Bool	show only enabled entries
limit	N/A	True	Integer(gte: 0)	says how many rows will be in response
contexts	N/A	False	List([Integer()], min_length: 0)	filter by context id
sort_by	[]	True	List([{"Enum('pt_name', 'enabled', 'pt_id', 'host', 'is_op', 'port')": Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
fields	['pt_name', 'enabled', 'pt_id', 'host', 'is_op', 'port']	False	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
pt_id	N/A	True	Integer(gt: 0)	filter result by point id

Default fields:

Name	Type & Comments
pt_name	String

enabled	Boolean
pt_id	Int
host	String
is_op	Boolean
port	Int or null

2.328 sms_numbers_by_point_get

Docstring:

Function searches destination codes among prices of chosen point tariff which begin with given digits.
Then it searches free subcode for each code from obtained list and supplements it with zeros to the "min_len" specified in price.
Function returns list that contains two another lists:
- first list contains original codes from tariff
- second list contains corresponding to them autocompleted codes(in the same order).

Example:

```
- Client Request
{"session_id":"bdf0fec748a6f7a0473389baf4fcb51b",
 "data":{"limit":100,
        "pt_id":"4331",
        "sort_by":[{"numbers":""}],
        "contexts":[1],
        "destination":"38",
        "fields":["numbers"]}
}

- Server Response:

{"status": 200,
 "fields": ["numbers"],
 "message": "OK",
 "data": [["380", "3800", "3805", "38050", "38051", "38063"],
          ["380100000000", "3800000000000", "3805200000000",
           "3805000000000", "3805100000000", "3806300000000"]],
 "time_elapsed": 0.008991
}
```

Request structure and fields description:

```
{'data': {'destination': 'String', 'limit': 'Integer', 'pt_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
limit	N/A	True	Integer(gt: 0)	limit of results amount
pt_id	N/A	False	Integer()	Termination or Origination point id
destination	N/A	True	String(allow_blank: False)	first digits of destination code

2.329 sms_op_add_or_update

Docstring:

Creates new disabled origination point if "id" absent in request, otherwise updates parameters of given point:

- 1) If "contexts" present in request updates point data in the given context(s). If point status was changed to "disabled" - creates new email notification.
 - 2) if "contexts" not present in request function can change point tariff, status, "invisible" and "allow_internal" flags.
- If flag invisible was set as "true" then function disables given OP and removes all point relations with groups.

`_sms_op_add` ([data_validation](#)) ([default_fields](#))

Creates new disabled origination point, adds new point to schema sms_routing and each context.

Allows to specify balance, tariff, description, flags "invisible" and "allow_internal" for creating OP.

Function returns new Origination Point id.

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"tarif_id":29,
    "invisible":false,
    "description":"",
    "name":"my_op",
    "balance_id":1,
    "allow_internal":false}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 25, "time_elapsed": 0.258111}
```

`_sms_op_update` ([data validation](#)) ([default fields](#))

Changes common parameters of given origination point: name, description, tariff, invisible and allow_internal flags in the schema routing. If flag "invisible" was set to "true" - function disables point and removes all relations of the OP with groups.

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"tarif_id":29,
    "id":25,
    "invisible":false,
    "description":"",
    "name":"my_op1",
    "allow_internal":false}
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.188672}
```

`_sms_op_update_within_contexts` ([data validation](#)) ([default fields](#))

Updates origination point parameters within given context(s). Function creates notification email if point status was changed manually.

Function raises error if API user is trying to change invisible point.

Example:

- Client Request

```
{"session_id":"74c1cf47382eb1a7ffad856372e7242c",
  "data":
  {
    "id":397,
```

```

        "contexts": [{
            "ani_block": "",
            "enabled": true,
            "context_id": 1,
            "rtp_timeout": 10,
            "ani_regexp": "",
            "capacity": 500,
            "dnis_regexp": ""
        }
    ]
}

```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.061522
}
```

Request structure and fields description:

_sms_op_add

```

{
  'data': {
    'allow_internal': 'Bool',
    'balance_id': 'Integer',
    'client_id': 'Integer',
    'currency_id': 'Integer',
    'description': 'String',
    'dlr_mode': "Enum('default_dlr', 'force_dlr')",
    'ext_id': 'Integer',
    'hlr_lookup': 'Bool',
    'hlr_lookup_type': "Enum('no', 'local', 'external')",
    'invisible': 'Bool',
    'mps_limit': 'Integer',
    'name': 'String',
    'point_manager_id': 'Integer or Null',
    'store_forward': 'Bool',
    'tariff_id': 'Integer',
    'tariff_type_id': 'Integer'},
  'session_id': 'String'
}

```

data validation description:

Name	Default value	Optional	Type	Description
description	""	True	String(allow_blank: True)	
allow_internal	False	True	Bool	boolean option. If "True" - allows calls to TPs which belongs same company as given OP
name	N/A	False	String(allow_b	point name should be unique

			lank: False)	
store_forwar d	False	True	Bool	If "True" - Store and Forward mode will be used for SMS
hlr_lookup	N/A	True	Bool	DEPRECATED boolean option. (If "True" - HLR request will be used for routing)
tariff_type_i d	1	True	Integer(gte: 1, lte: 2)	(ONLY FOR DEVELOPER USE) Type of SMS tariff: 1 - Destination, 2 - MCCMNC
client_id	N/A	True	Integer(gt: 0)	client with given id should be visible
invisible	False	True	Bool	boolean option. If "True" - function makes point invisible and disabled, removes point relations with dialpeers and groups.
hlr_lookup_t ype	N/A	True	Enum('no', 'local', 'external')	HLR option. If "external" - HLR request will be used for routing, if "local" MCC MNC Table will be used.
dlr_mode	default_d lr	True	Enum('default_dlr', 'force_dlr')	"default_dlr" - simple DLR behaviour"force_dlr" - status "DELIVRD" will be returned in case if status has been requested by OP;"dlr_replacement" - not in use.
mps_limit	100	True	Integer(gt: 0, lte: 1000)	Messages per second limit
tarif_id	0	True	Integer(gte: 0)	tariff currency has to be equal currency of chosen balance
ext_id	0	True	Integer(gte: 0)	OP id for external billing
balance_id	N/A	True	Integer(gt: 0)	client which owns given balance should be visible
currency_id	N/A	True	Integer(gte: 0)	
point_manag er_id	N/A	True	Or(Integer(gt: 0), Null)	OP manager user id for update notifications

_sms_op_update

```
{'data': {'allow_internal': 'Bool',
'description': 'String',
'dlr_mode': "Enum('default_dlr', 'force_dlr')",
'hlr_lookup': 'Bool',
```

```

'hlr_lookup_type': "Enum('no', 'local', 'external')",
'id': 'Integer',
'invisible': 'Bool',
'name': 'String',
'point_manager_id': 'Integer or Null',
'store_forward': 'Bool',
'tarif_id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
allow_internal	N/A	True	Bool	boolean option. If "true" - allows calls to TPs which belongs same company as given OP
hlr_lookup_type	N/A	False	Enum('no', 'local', 'external')	HLR Lookup type option. If "external" - HLR request will be used for routing, if "local" MCC MNC Table will be used.
tarif_id	N/A	True	Integer(gt: 0)	new point tariff. The currency of tariff have to be the same as the currency of balance.
invisible	N/A	True	Bool	if this flag is set to "true" - function disables OP and removes all relations of point with groups.
dlr_mode	default_dlr	True	Enum('default_dlr', 'force_dlr')	"default_dlr" - simple DLR behaviour"force_dlr" - status "DELIVRD" will be returned in case if status has been requested by OP;"dlr_replacement" - not in use.
id	N/A	False	Integer(gt: 0)	point id
store_forward	N/A	True	Bool	If "True" - Store and Forward mode will be used for SMS
name	N/A	True	String(allow_blank: False)	new point name, should be unique in the system.
hlr_lookup	N/A	True	Bool	DEPRECATED boolean option. (If "True" - HLR request will be used for routing)
description	N/A	True	String(allow_blank: True)	

point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	OP manager user id for update notifications. If key value is "NULL" manager user will be cleared. If key not present in function request data user id will not be updated.
------------------	-----	------	--------------------------	--

_sms_op_update_within_contexts

```
{'data': {'contexts': [{'ani_block': 'String',
                        'ani_regexp': 'String',
                        'capacity': 'Integer',
                        'context_id': 'Integer',
                        'dnis_regexp': 'String',
                        'enabled': 'Bool',
                        'mps_limit': 'Integer'}],
          'id': 'Integer'},
  'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	
enabled	N/A	True	Bool	point status. If "True" - point can participate in routing. If point status was changed creates new email notification.
capacity	N/A	True	Integer(gte: 0, lte: 1000)	defines the limit on the number of simultaneous calls that can pass through this point.
ani_regexp	N/A	True	String(allow_blank: True, max_length: 64)	A number regexp
dnis_regexp	N/A	True	String(allow_blank: True, max_length: 64)	B number regexp
ani_block	N/A	True	String(allow_blank: True, max_length: 255)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
mps_limit	N/A	True	Integer(gt: 0, lte: 1000)	Messages per second limit

			1000)	
--	--	--	-------	--

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	
contexts	N/A	False	List([data.contexts], min_length: 0)	

2.330 sms_op_authorization_add

Docstring:

Creates new SMS authorization account (account for authorization by login and password) within chosen context and OP.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "op_id": 4331,
    "context_id": 1,
    "login": "test_sip",
    "password": "121314"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'op_id': Int(gt=0),
    'login': String(),
    'password': String(max_length=SMPP_PASSWORD_MAX_LENGTH),
    'trusted_ips': List(String(regex=IP_REGEX)),
    "context_id": Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
'op_id'	N/A	False	Int(gt=0)	Origination point id to which given

				authorization will be added
'login'	N/A	False	String()	SMS auth login. Should be unique.
'password'	N/A	False	String(max_length=SMPP_PASSWORD_MAX_LENGTH)	SMS auth password
'trusted_ips'	N/A	False	List(String(regex=IP_REGEX))	trusted ips
"context_id"	N/A	False	Int(gt=0)	context to which authorization will be added

2.331 sms_op_authorization_del

Docstring:

Removes chosen SMS authorization from the given context.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "context_id": 1,
    "login": "test_sip"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'login': String(),
    'context_id': Int(gt=0),
    'op_id': Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
'login'	N/A	True	String()	SMS account login
'context_id'	N/A	False	Int(gt=0)	context in which given SMS authorization have to be removed
'op_id'	N/A	True	Int(gt=0)	filter by OP id

2.332 sms_op_authorization_get

Docstring:

Function retrieves SMS accounts existent in the database.

Example:

- Client Request:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "op_id": 231,
    "contexts": [1],
    "fields": ["login"]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": ["login"],
  "message": "OK",
  "data": ["test_sip"],
  "time_elapsed": 0.008435
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
```

```

'op_id': Int(gt=0),
'limit': Int(gte=0),
'offset': Int(gte=0),
'login': String(),
'sort_by': List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
'fields': List(String()),
"contexts": List(Int()),
}
}

```

Name	Default value	Optional	Type	Description
'op_id'	N/A	True	Int(gt=0)	filter by Origination Point
'limit'	0	False	Int(gte=0)	filter by Origination Point
'offset'	0	False	Int(gte=0)	filter by Origination Point
'login'	N/A	True	String()	filter by SMS account login
'sort_by'	[]	False	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	
'fields'	default_fields	True	List(String())	required fields If absent - takes fields specified in "default fields". If empty - returns error.
"contexts"	[]	False	List(Int())	filter by context id. If absent in request - returns data from all contexts.

Default fields:

Name	Type & Comments
------	-----------------

"id"	Int
"op_id"	Int
"login"	String
"context_id"	Int
"password"	String
"trusted_ips"	List(String)

2.333 sms_op_authorization_set

Docstring:

Function allows to change password of chosen SMS authorization in the given context.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "login": "test_sip",
    "context_id": 1,
    "password": "132456567"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(),
    'login': String(),
    'password': String(max_length=SMPP_PASSWORD_MAX_LENGTH),
    'trusted_ips': List(String(regex=IP_REGEX)),
    'context_id': Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int()	id

'login'	N/A	False	String()	SMS account login which password have to be changed
'password'	N/A	True	String(max_length=SMPP_PASSWORD_MAX_LENGTH)	new password
'trusted_ips'	N/A	False	List(String(regex=IP_REGEX))	trusted ips
'context_id'	N/A	False	Int(gt=0)	context where given SMS authorization have to be changed

2.334 sms_op_group_set

Docstring:

Changes priority of relation between given OP and group.

Example:

- Client Request

```
{ "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": { "op_id": 3,
            "priority": 1,
            "group_id": 6,
            "contexts": [1] }
}
```

- Server Response

```
{ "status": 200, "message": "OK", "time_elapsed": 0.032511 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "op_id": Int(gt=0),
    "group_id": Int(gt=0),
    "priority": Int(gte=-100, lte=100),
    "contexts": List(Int()),
  }
}
```

Name	Default value	Optional	Type	Description
"op_id"	N/A	False	Int(gt=0)	
"group_id"	N/A	False	Int(gt=0)	
"priority"	0	True	Int(gte=-100, lte=100)	new relation priority
"contexts"	N/A	False	List(Int())	contexts id where need change relations

2.335 sms_op_settings_copy

Docstring:

Copies specified SMS OP settings from given context to contexts that in given contexts list.

Request structure and fields description:

```
{'data': {'context_id_from': 'Integer',
          'context_ids_to': ['Integer'],
```

```

'copy_ani_block': 'Bool',
'copy_ani_regexp': 'Bool',
'copy_auth_list': 'Bool',
'copy_capacity': 'Bool',
'copy_dnis_regexp': 'Bool',
'copy_enabled': 'Bool',
'copy_mps_limit': 'Bool',
'id': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
copy_mps_limit	False	True	Bool	messages per second limit will be copied or not
copy_capacity	False	True	Bool	capacity value will be copied or not
copy_ani_block	False	True	Bool	ani block will be copied or not
context_id_from	N/A	False	Integer(gt: 0)	context id from which data will be copied to another context/contexts
copy_ani_regexp	False	True	Bool	ani regexp will be copied or not
id	N/A	False	Integer(gt: 0)	point id
context_ids_to	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of context ids where data must be copied from another context
copy_enabled	False	True	Bool	enabled flag will be copied or not
copy_auth_list	False	True	Bool	authentication list will be copied or not
copy_dnis_regexp	False	True	Bool	dnis regexp will be copied or not

2.336 sms_ops_free_group_get

Docstring:

Returns list of OPs which aren't connected with chosen group.

Example:

- Client Request

```
{ "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",  
  "data": { "sort_by": [ { "op_name": "" } ],  
            "group_id": 6,  
            "contexts": [1],  
            "fields": [ "op_id", "op_name" ] }  
}
```

- Server Response

```
{ "status": 200,  
  "fields": [ "op_id", "op_name" ],  
  "message": "OK",  
  "data": [[13, "s_test_op"], [7, "H_P_in"], [9, "H_in_simple"],  
            [5, "st.peer1"], [15, "OP-14"], [19, "pablo_op"]],  
  "time_elapsed": 0.00085 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "op_id": Int(),  
    "op_name": String(allow_blank=True),  
    "group_id": Int(gt=0),  
    "contexts": List(Int()),  
    "fields": List(String()),  
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),  
    "limit": Int(gte=0),  
    "offset": Int(gte=0),  
  }  
}
```

Name	Default value	Optional	Type	Description
"op_id"	N/A	True	Int()	Filter by OP
"op_name"	N/A	True	String(allow_blank=True)	filter by OP name. Function will return only points which names starts with given symbols

"group_id"	N/A	True	Int(gt=0)	
"contexts"	[]	False	List(Int())	context(s) where search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"op_name":"asc"}] - sorts result ascending by field "op_name"
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
"op_id"	Int
"op_name"	String

2.337 sms_ops_free_tp_get

Docstring:

Returns list of OPs which aren't connected with chosen TP in simple and advanced mode in the chosen context(s).

Example:

- Client Request

```
{ "session_id": "6726cebe6ca27ef6a8da30867d6050a8",  
  "data": { "sort_by": [ { "name": "" } ],  
            "name": "",  
            "contexts": [1],  
            "tp_id": 8,  
            "fields": [ "id", "name" ] }  
}
```

- Server Response

```
{ "status": 200,  
  "fields": [ "id", "name" ],  
  "message": "OK",  
  "data": [ [3, "Test_1_OP"], [21, "ss_OP"], [13, "a_test_op"],  
            [23, "H_in"], [5, "peer1"], [27, "NewOP"], [25, "my_op"],  
            [15, "OP-14"] ],  
  "time_elapsed": 0.00972 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.338 sms_ops_from_groups_del

Docstring:

Removes relations between given OPs and groups within given context(s).

Function diallows to delete relations which was created in the Simple

Routing mode.

Example:

- Client Request

```
{ "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",  
  "data": { "op_ids": [3],  
            "contexts": [1],  
            "groups_ids": [8] }  
}
```

- Server Response

```
{ "status": 200, "message": "OK", "time_elapsed": 0.060283 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "op_ids": List(Int(gt=0)),  
    "groups_ids": List(Int(gt=0)),  
    "contexts": List(Int()),  
  }  
}
```

Name	Default value	Optional	Type	Description
"op_ids"	N/A	False	List(Int(gt=0))	list of OPs id
"groups_ids"	N/A	False	List(Int(gt=0))	list of group ids
"contexts"	N/A	False	List(Int())	contexts id where need remove relation

2.339 sms_ops_get

Docstring:

Returns list of origination points existing in database in given context.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "limit": 2,
    "sort_by": [{"name": ""}],
    "invisible": false,
    "contexts": [1],
    "offset": 0,
    "fields": [
      "id", "name", "client_name", "client_id", "enabled",
      "capacity", "tariff_id", "tariff_name",
      "tariff_name_currency"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "id", "name", "client_name", "client_id", "enabled",
    "capacity", "tariff_id", "tariff_name",
    "tariff_name_currency"
  ],
  "message": "OK",
  "data": [
    [3, "Test_1_OP", "Test_client", 1, true, 500, 9, "test",
      "test (USD)"],
    [21, "ss_OP", "ss_Co", 15, true, 500, 39, "ss_1.1",
      "ss_1.1 (EUR)"]
  ],
  "time_elapsed": 0.010607
}
```

Request structure and fields description:

```
{
  'data': {
    'clients_ids': ['Integer'],
    'contexts': ['Integer'],
    'enabled': 'Bool',
    'ext_id': 'Integer',
    'fields': [
      "Enum('client_invisible', 'ext_id', 'hlr_lookup_type', 'currency_id', 'balance_name', 'balance_id', 'id', 'dlr_mode', 'store_forward', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp')",
      'id': 'Integer',
      'invisible': 'Bool or Null',
      'limit': 'Integer',
      'manager_id': 'Integer',
      'name': 'String',
      'name_or_client_name': 'String',
      'offset': 'Integer',
      'owner_id': 'Integer',
      'sort_by': [
        "Enum('client_invisible', 'ext_id', 'hlr_lookup_type', 'currency_id', 'balance_name', 'balance_id', 'id', 'dlr_mode', 'store_forward', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp')"]
      ]
    }
  }
```

```

_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp')": "Enum('asc', 'desc', '')"}]],
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
id	N/A	True	Integer(gt: 0)	id of Origination Point
manager_id	N/A	True	Integer(gt: 0)	filter by client's manager
fields	['client_invisible', 'ext_id', 'hlr_lookup_type', 'currency_id', 'balance_name', 'balance_id', 'id', 'dlr_mode', 'store_forward', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp']	True	List([Enum('client_invisible', 'ext_id', 'hlr_lookup_type', 'currency_id', 'balance_name', 'balance_id', 'id', 'dlr_mode', 'store_forward', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
owner_id	N/A	True	Integer(gt: 0)	filter by owner
clients_ids	N/A	True	List([Integer()], min_length: 0)	filter by client_ids

invisible	False	True	Or(Bool, Null)	boolean option, if "True" returns only invisible OPs
name	N/A	True	String(allow_blank: True)	filter by point name or first chars of name
enabled	N/A	True	Bool	boolean option, if equal "True" returns only enabled OPs
contexts	[]	True	List([Integer()], min_length: 0)	filter by contexts. If is absent function searches origination points in all contexts.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
sort_by	[]	True	List([Enum('client_invisible', 'ext_id', 'hlr_lookup_type', 'currency_id', 'balance_name', 'balance_id', 'id', 'dlr_mode', 'store_forward', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'context_id', 'tariff_name', 'description', 'hlr_lookup', 'allow_internal', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'simple_pt', 'dnis_regexp'): Enum('asc', 'desc', ")]], min_length: 0)	field on which result sorts

name_or_client_name	N/A	True	String(allow_blank: True)	filter by point/client name or first chars of point/client name
ext_id	N/A	True	Integer(gt: 0)	OP id for external billing

Default fields:

Name	Type & Comments
client_invisible	Boolean
ext_id	Integer
hlr_lookup_type	String
currency_id	Integer
balance_name	String
balance_id	Integer
id	Integer
dlr_mode	String
store_forward	Boolean
capacity	Integer
ani_block	String
client_name	String
mps_limit	Integer
tariff_type_id	Integer

tariff_name_currency	String - DEPRECATED
context_id	Integer
tariff_name	String
description	String
hlr_lookup	Boolean - DEPRECATED
allow_internal	Boolean
currency_name_short	String
client_id	Integer
invisible	Boolean
ani_regexp	String
tariff_id	Integer
name	String
point_manager_id	Integer - Point manager user id for update notifications
enabled	Boolean
simple_pt	Boolean
dnis_regexp	String

2.340 sms_ops_groups_get

Docstring:

Returns data of relations between chosen group and selected OP. Function doesn't show service group in the resulted groups list.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "sort_by": [
      {
        "op_name": ""
      }
    ],
    "group_id": 6,
    "contexts": [1],
    "fields": ["op_id", "op_name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["op_id", "op_name"],
  "message": "OK",
  "data": [[3, "Alexander_test_1_OP"]],
  "time_elapsed": 0.009095
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "op_name": String(allow_blank=True),
    "group_name": String(allow_blank=True),
    "op_id": Int(),
    "group_id": Int(),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"op_name"	N/A	True	String(allow_blank=True)	filter by OP name or first symbols of OP name
"group_name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"op_id"	N/A	True	Int()	filter by OP id
"group_id"	N/A	True	Int()	filter by group id

"contexts"	[]	False	List(Int())	context(s) where search relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"op_name": "asc"}] - sorts result ascending by field "op_name"
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows..

Default fields:

Name	Type & Comments
'group_id'	Int
'op_id'	Int
'priority'	Int
'op_name'	String
'group_name'	String

2.341 sms_ops_to_groups_add

Docstring:

Adds relations between given OP(s) and group(s) in the chosen context(s) with specific relation priority. Given OP(s) and group(s) should have the same services.

Function disallows to add relation if chosen OP already configured in Simple mode.
Function raises error if user tries to create relation with invisible point.

Example:

- Client Request

```
{
  "session_id": "91af15315af776928e1abfa0e88b6641",
  "data": {
    "priority": 0,
    "op_ids": ["19"],
    "contexts": [1],
    "groups_ids": [9]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.047786
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "op_ids": List(Int(gt=0)),
    "groups_ids": List(Int(gt=0)),
    "priority": Int(gte=-100, lte=100),
    "contexts": List(Int()),
  }
}
```

Name	Default value	Optional	Type	Description
"op_ids"	N/A	False	List(Int(gt=0))	list of OPs id, points should be visible
"groups_ids"	N/A	False	List(Int(gt=0))	list of groups id
"priority"	0	True	Int(gte=-100, lte=100)	relation priority in range from -100 to 100
"contexts"	N/A	False	List(Int())	contexts id where need add relations

--	--	--	--	--

2.342 sms_ops_tp_get

Docstring:

Returns a list of OPs which are connected with given TP in simple and advanced routing modes, i.e. OPs that satisfy condition:

- OPs and dialpeers binded with given TP must belong to the same group as TP within chosen context.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "limit": 101,
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "offset": 0,
    "tp_id": 8,
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[7, "H_in"], [9, "H_in_simple"]],
  "time_elapsed": 0.009807
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
```

```
}
```

2.343 sms_ops_tp_simple_get

Docstring:

Returns a list of OPs which are connected with given TP only in simple mode.

Example:

- Client Request

```
{ "session_id": "6726cebe6ca27ef6a8da30867d6050a8",  
  "data": { "sort_by": [ { "name": "" } ],  
            "contexts": [1],  
            "tp_id": 20,  
            "fields": ["id", "name"] }  
}
```

- Server Response

```
{ "status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[23, "H_in"]],  
  "time_elapsed": 0.011045 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.344 sms_points_enabled_get

Docstring:

Calls functions sms_ops_get() and sms_tps_get() with option "enabled" = True and combine result in one list.

Request structure and fields description:

2.345 sms_price_upload

Docstring:

This function adds given prices to the uploading queue. It allows to upload new prices only to the one tariff.

There are several possible prices formats which function accepts. You can get them with help of function 'prices_format_get'.

Function process request in the following order:

- 1) Sets direction ("1" for "in", 2 - for "out") and checks prices validity.
- 2) Uploads new prices.

Function will raise error with status 305 for single price uploading if 'start_date' in price less than current. Despite the error price will be uploaded but 'start_date' will be changed to the current date.

Function will raise error with status 308 in case if destinations for given mcc and mnc not found in DB.

Possible fields names are mentioned in the default_fields.

Example with destination:

-Client Request

```
{"session_id":"ca84cd3eaa596f2e3f7836db065777da",
```

```

"data":{"price_type":"incremental",
"direction":"out",
"data":[[4700,
true,
true,
"",
"3912",
0.02345,
0.0,
1,
1,
0,
0,
"2013-10-29T12:34:37",
"2030-01-01T23:59:59",
"00:00:00",
"23:59:59",
1,
24,
"01111111"]],
"fields":["tarif_id",
"enabled",
"allowed",
"description",
"destination",
"price",
"connection_price",
"mc",
"ci",
"ft",
"priority",
"date_start",
"date_end",
"time_start",
"time_end",
"min_len",
"max_len",
"week_day"]}]
}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    "tariff_id": Int(),
    "session_id": String(),
    "data": List(Any(), min_length=1),
    "fields": List(String()) ,
    "direction": String(regex="(in)|(out)"),
    "price_type": String(regex="(incremental)|(full)"),
    "notification_email": String(allow_blank=True),

```

```

    "guardian": Bool(),
    "operator": String(),
    "label": String(),
    "save_previous_price_priority": Bool(),
    "save_previous_min_max_digits": Bool(),
    "save_privious_price_priority": Bool(),
    "save_privious_min_max_digits": Bool(),
}
}

```

Name	Default value	Optional	Type	Description
"tariff_id"	N/A	True	Int()	id of the target tariff
"session_id"	session_info["id"]	True	String()	current session id
"data"	N/A	False	List(Any(), min_length=1)	prices data values in the form of list
"fields"	N/A	False	List(String())	prices fields names in the same order as given prices values
"direction"	N/A	False	String(regex="(in) (out)")	"in" - for OP tariffs, "out" - for TP tariffs
"price_type"	"incremental"	True	String(regex="(incremental) (full)")	if type "full" all prices existed in the tariff will be closed and then uploaded new one. If "incremental" - new prices uploads over existed.
"notification_email"	session_info.get("user_email", "")	True	String(allow_blank=True)	email address to which will be send notification when price will be successfully uploaded. If doesn't specify in request - email will be sent to recipients addresses specified in price log template.
"guardian"	False	True	Bool()	for internal usage
"operator"	session_info.get("login", "API")	True	String()	name of user who uploading price. By default - user login from current session
"label"	u"Loaded by {0}".format(unicode(session_info.get("login"	True	String()	label for price that will be recorded to database. Example: "Loaded by <Name of user who

	, "API"))))			uploaded the price>". If label is absent in request - label will be set by defaults as "Loaded by <login of user from current session.>
"save_previous_price_priority"	False	True	Bool()	flag allows to preserve priority of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"save_previous_min_max_digits"	False	True	Bool()	flag allows to preserve min and max digits of existent actual price (previous actual price with the same destcode, start and end time, tariff id and week days). This option may be used in conjunction with 'price_type' = incremental.
"save_privious_p rice_priority"	False	True	Bool()	DEPRECATED, backport
"save_privious_min_max_digits"	False	True	Bool()	DEPRECATED, backport

Field "**data**" - is a list that contains prices data for different destination codes in the form of list.

```
"data": [
  [
    "tarif_id": Int(gt=0),
    "enabled": Bool(),
    "allowed": Bool(),
    "description": String(allow_blank=True),
    "destination": String(allow_blank=True, regex="^[0-9]*$"),
    "network_code": String(regex="^(\d{3,6})?$"),
    "mcc": String(allow_blank=True, regex="^(\d{3})?$"),
    "mnc": String(allow_blank=True, regex="^(\d{1,3})?$"),
    "price": Float(gte=0),
    "priority": Int(gte=-100, lte=100),
    "date_start": String(),
    "date_end": String(),
```

"time_start": String(regex=TIME_CHECK_REGEX_PATTERN),
 "time_end": String(regex=TIME_CHECK_REGEX_PATTERN),
 "week_day": String(regex="(0|1){8}"),
 "min_len": Int(gte=1, lte=100),
 "max_len": Int(gte=1, lte=100),

], [...],[...], ...

]

Name	Default value	Optional	Type	Description
"tarif_id"	N/A	False	Int(gt=0)	id of the tariff to which price will be uploaded. Should be the same for all prices in price list.
"enabled"	True	True	Bool()	Defines price status. Disabled prices are not actual.
"allowed"	True	True	Bool()	"False" value makes destination disallowed.
"description"	N/A	True	String(allow_blank=True)	
"destination"	""	True	String(allow_blank=True, regex="^[0-9]*\$")	Defines destination code for price
"network_code"	N/A	True	String(regex="^(\\d{3,6})?\$")	network code(mcc + mnc)
"mcc"	""	True	String(allow_blank=True, regex="^(\\d{3})?\$")	mcc
"mnc"	""	True	String(allow_blank=True, regex="^(\\d{1,3})?\$")	mnc
"price"	0	True	Float(gte=0)	price per minute
"priority"	0	True	Int(gte=-100, lte=100)	Price priority. You can specify this option only for TP tariff, i.e. if direction="out".

"date_start"	<code>datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")</code>	True	String()	Defines the start date and time for the price validity. It should not be less than current. Format: 'YYYY-MM-DDTHH:MM:SS', where T is delimiter between date and time. If price list is being uploaded with 'price_type' = 'full' all prices must have the same start date.
"date_end"	"2037-12-31 23:59:59+00"	True	String()	Defines the end date and time for the price validity. Format: 'YYYY-MM-DDTHH:MM:SS', where T is delimiter between date and time.
"time_start"	"00:00:00+00"	True	String(regex=TIME_CHECK_REG_EXP_PATTERN)	Defines the start time for the price availability during the day time.
"time_end"	"23:59:59+00"	True	String(regex=TIME_CHECK_REG_EXP_PATTERN)	Defines the end time for the price availability during the day time.
"week_day"	"01111111"	True	String(regex="(0 1){8}")	Bit mask, which defines days of week of price validity. First bit - not in use, second - Saturday, third - Friday, ...7th bit - Monday, 8th bit - Sunday. 1- means that price works in that day of week, 0 - not. Example - '01111111' - price works all days of week, '00111110' - price works every day excluding sunday and saturday.
"min_len"	1	True	Int(gte=1, lte=100)	Defines the minimal length of destination code for current price in range from 1 till 100 inclusive.
"max_len"	24	True	Int(gte=1, lte=100)	Defines the maximum length of destination code for current price in range from 2 till 100 inclusive.

Default fields:

Name	Type & Comments
"tarif_id"	
"enabled"	
"allowed"	
"description"	
"destination"	
"network_code"	
"mcc"	
"mnc"	
"price"	
"priority"	
"date_start"	
"date_end"	
"time_start"	
"time_end"	
"min_len"	
"max_len"	

"week_day"	
------------	--

2.346 sms_prices_list_get

Docstring:

Returns SMS prices belonging to chosen SMS OP tariff or SMS TP tariff with detailed information about them. User can specify in request price type to be shown: "actual" (only actual prices), "current" (only prices actual at the current time), "enabled" (only enabled prices).

Request structure and fields description:

```
{'data': {'actual_prices': 'Bool',
          'allowed': 'Bool',
          'blocked_by_guardian': 'Bool',
          'current': 'Bool',
          'destination': 'String',
          'enabled': 'Bool',
          'end_date': 'String',
          'fields': ["Enum('destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'mnc', 'id', 'actual_price_id', 'time_start', 'date_start', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'tariff_id', 'country', 'network_code', 'enabled', 'date_for_history')"],
          'future': 'Bool',
          'get_actual': 'Bool',
          'id': 'Integer',
          'is_tp': 'Bool',
          'limit': 'Integer',
          'mcc': 'String',
          'mnc': 'String',
          'network_code': 'String',
          'offset': 'Integer',
          'pt_id': 'Integer',
          'session_id': 'String',
          'sort_by': [{"Enum('destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'mnc', 'id', 'actual_price_id', 'time_start', 'date_start', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'tariff_id', 'country', 'network_code', 'enabled', 'date_for_history')": "Enum('asc', 'desc', '')"}],
          'start_date': 'String',
          'strictly': 'Bool',
          'tariff_id': 'Integer'},
        'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
------	---------------	----------	------	-------------

		I		
session_id	N/A	True	String(allow_blank: False)	by default from user session
id	N/A	True	Integer(gt: 0)	Price id in database. If "is_tp"=="true" then TP price id, else - OP price id
is_tp	N/A	False	Bool	if "true" returns prices only from TP tariffs, otherwise - OP. This flag is internal. Not need to send it in request.
tariff_id	N/A	True	Integer(gt: 0)	if "is_tp"=="true" then TP id, otherwise - OP id
pt_id	N/A	True	Integer(gt: 0)	if "is_tp"=="true" then TP id, otherwise - OP id
destination	N/A	True	Destination(allow_blank: True, _dest_pattern: ^\d*\$)	filter by destination code or first digits of code
mcc	N/A	True	String(allow_blank: True, regex: "^(\\d{1,3})?\$")	filter by mobile country code
current	N/A	True	Bool	If "true" returns last uploaded prices by tariff and destination, which satisfy following conditions: date_end>=current, date_start=
future	False	True	Bool	Used by Accucore. If "True" function searches only future prices for point
mnc	N/A	True	String(allow_blank: True, regex: "^(\\d{1,3})?\$")	filter by mobile network code

enabled	N/A	True	Bool	if "true" returns enabled prices
allowed	N/A	True	Bool	if "true" returns allowed prices
blocked_by_guardian	False	True	Bool	Used by Accucore. If "True" function searches only blocked destination codes for actual price.
strictly	False	True	Bool	If "true" function searches strict match of given destination code in the database, else searches destinations code with subcodes.
start_date	N/A	True	String(allow_blank: False)	start date and time to show prices with start_date not less then given.
end_date	N/A	True	String(allow_blank: False)	end date and time to show prices with end_date not more then given.
actual_prices	N/A	True	Bool	if "true" returns prices, which satisfy following conditions: date_end>=current, date_start=
fields	['destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'mnc', 'id', 'actual_price_id', 'time_start', 'date_start', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'tariff_id', 'country', 'network_code', 'enabled', 'date_for_history']	True	List([Enum('destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'mnc', 'id', 'actual_price_id', 'time_start', 'date_start', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'tariff_id', 'country', 'network_code', 'enabled', 'date_for_history')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows

				will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
get_actual	N/A	True	Bool	Option should be used in conjunction with price id. If "get_actual" set to True, then function returns all actual prices for destination of the price with given id within price's tariff.
network_code	N/A	True	String(allow_blank: False, min_length: 3, max_length: 6)	filter by network code(mcc+mnc)
sort_by	[]	True	List([{'destination', 'date_end', 'time_end', 'max_len', 'week_day', 'allowed', 'mnc', 'id', 'actual_price_id', 'time_start', 'date_start', 'priority', 'added', 'description', 'min_len', 'price', 'mcc', 'tariff_id', 'country', 'network_code', 'enabled', 'date_for_history'}: Enum('asc', 'desc', ")]], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

Default fields:

Name	Type & Comments
destination	String
date_end	String
time_end	String
max_len	Int
week_day	String

allowed	Boolean
mnc	String
id	Int
actual_price_id	Int
time_start	String
date_start	String
priority	Int
added	String
description	String
min_len	Int
price	numeric(16,5)
mcc	String
tariff_id	Int
country	String
network_code	String
enabled	Boolean
date_for_history	String

2.347 sms_process_new_price

Docstring:

Process uploaded SMS prices (changes its status to '7' - 'Applied') and creates notification email after price uploading.
Function for internal usage.

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.348 sms_radiuses_by_context_get

Docstring:

Function returns sms radiuses data(ip, name and so on) for different collocations.

Example:

- Client Request

```
{
  "session_id": "4595aceb14c686aa9752bb9860462bf2",
  "data": {
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "fields": ["id", "name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[1, "CZ"]],
  "time_elapsed": 0.013664
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'fields': ["Enum('colocation_id', 'ip', 'id', 'name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'sort_by': [{"Enum('colocation_id', 'ip', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
, '' )}]},
```

```
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer()], min_length: 1)	filter by selected contexts
fields	['colocation_id', 'ip', 'id', 'name']	True	List([Enum('colocation_id', 'ip', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('colocation_id', 'ip', 'id', 'name'): Enum('asc', 'desc', ")]], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
colocation_id	Int
ip	String
id	Int
name	String

2.349 sms_rel_op_tp_add

Docstring:

Creates relations between chosen OP and TP(s), i.e.:

- 1) creates service dialpeer '.*' if such dialpeer doesn't exist in the given context,
- 2) creates service group for given OP if such doesn't exist in the given context,
- 3) creates relation between service dialpeer and given TP, if such doesn't exist in the chosen context,
- 4) adds dialpeer, OP and TP to the service group of given OP in specified context.

Function raises error if relation between OP and some non service group is already exists.

Example:

- Client Request

```
{"session_id": "fbea69d4b49be804dfe87f2e1d852f2f",  
  "data": {"op_ids": [5],  
           "contexts": [1],  
           "tp_ids": [14]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.311718}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.350 sms_rel_op_tp_del

Docstring:

Removes relations between chosen OP and TPs which were created in the simple mode, i.e. removes relations between TPs and service group of given OP and service dialpeer in the chosen context(s). If given OP has lost all relation with TPs then service group of the OP will be removed too.

Example:

- Client Request

```
{ "session_id": "6726cebe6ca27ef6a8da30867d6050a8",  
  "data": { "op_ids": [17],  
            "contexts": [1],  
            "tp_ids": [24] }  
}
```

- Server Response

```
{ "status": 200, "message": "OK", "time_elapsed": 2.182171 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.351 sms_routing_table_get

Docstring:

Returns list of SMS Termination Points to which given SMS OP have route for specific destination at the given time ordered according to the routing algorithm:

1) by priority of SMS OP->SMS Group relation

- 2) by SMS Dialpeer priority
- 3) by priority of SMS Dialpeer->SMS TP relation
- 4) by weight of relation between SMS Dialpeer and SMS TP if priority of SMS Dialpeer->SMS TP relation is the same
- 5) by SMS TP price priority
- 6) LCR (more profitable routes will be higher in list)

Function returns also boolean parameter - "dest_changed". It determines if tested number will be changed by some conversion defined in any route (only B number conversion in OP and dialpeer is taken into account).

Function returns parameters of the targeted SMS OP within key 'op_data'.

Example:

- Client Request

```
{
  "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": {
    "time_zone": "0:00",
    "time_period": "2014-01-13T15:32:50",
    "op_id": 7,
    "contexts": [1],
    "number": "923000000000",
    "fields": [
      "tp_enabled", "delta_price", "dif_price", "tp_id",
      "tp_price_id", "group_id", "dialpeer_id",
      "allow_internal", "tp_without_ip", "allowed_price",
      "tp_name", "tp_destination", "tp_country",
      "tp_price_default", "tp_price_tarif",
      "tp_currency_name",
      "tp_capacity", "hunt_stop", "dp_name",
      "dp_tp_priority", "dp_priority", "weight",
      "group_name", "op_group_priority", "price_priority",
      "profit", "client_id"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.028728,
  "dest_changed": false,
  "fields": [
    "tp_enabled", "delta_price", "dif_price", "tp_id",
    "tp_price_id", "group_id", "dialpeer_id",
    "allow_internal", "tp_without_ip", "allowed_price",
    "tp_name", "tp_destination", "tp_country",
    "tp_price_default", "tp_price_tarif", "tp_currency_name",
    "tp_capacity", "hunt_stop",
    "dp_name", "dp_tp_priority", "dp_priority", "weight",
    "group_name", "op_group_priority", "price_priority",
    "profit", "client_id"
  ],
  "message": "OK",
  "op_data": [
    "H_in", 5, "Pakistan Mobile Mobilink", 13, 7, true,
    "9230", "10.4000", 1, 1, 0, 1, 5, "T_H_in1",
  ]
}
```

```
"10.4000", "$", "USD", "QUALITY"],
"data": [[true, "0.00000", "0.00500", 8, 20, 3, 4, false, false,
true, "H_to", "9230", "Pakistan Mobile Mobilink",
"10.3500", "10.3500", "USD", 1, 1, 0, 500, false,
"H_p(CZ)", 0, 0, 0, "h_p (CZ)", 0, 0, "14.29", 6]]}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'session_id': String(),
    'number': String(regex=B_NUMBER_REGEXP),
    'time_period': String(),
    'op_id': Int(gt=0),
    'time_zone': String(regex=TZ_REGEXP),
    'show_disabled_pt': Bool(),
    'one_route': Bool(),
    "op_fields": List(String()),
    'fields': List(String()),
    'currency_id': Int(gt=0),
    'contexts': List(Int()),
    "mcc": String(allow_blank=True, regex="^(\\d{3})?$"),
    "mnc": String(allow_blank=True, regex="^(\\d{1,3})?$"),
  }
}
```

Name	Default value	Optional	Type	Description
'session_id'	session_info['id']	True	String()	user session id, need for checking user permissions. Currently not in use.
'number'	N/A	False	String(regex=B_NUMBER_REGEXP)	called number
'time_period'	N/A	False	String()	request time in format yyyy-MM-dd'T'HH:mm:ss
'op_id'	N/A	False	Int(gt=0)	id of chosen Origination Point
'time_zone'	'+00:00'	True	String(regex=TZ_REGEXP)	time zone for 'time_period'
'show_disabled_pt'	False	True	Bool()	if "True" then disabled TPs also will be shown
'one_route'	False	True	Bool()	if "True" then report shows only one

				route per one Termination point.
"op_fields"	default_fields_op	True	List(String())	OP information which will be displayed in response. If absent - returns fields specified in "default_fields_op" section. If empty - returns error.
'fields'	default_fields	True	List(String())	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
'currency_id'	session_info['currency_id']	True	Int(gt=0)	currency in which price will display
'contexts'	N/A	False	List(Int())	ids context, in which request will be processed
"mcc"	N/A	True	String(allow_blank=True, regex="^(\\d{3})?\$")	
"mnc"	N/A	True	String(allow_blank=True, regex="^(\\d{1,3})?\$")	

Default fields:

Default fields OP:

Name	Type & Comments
'op_name'	String
'op_capacity'	Int

'op_country'	String
'op_price_id'	Int
'op_id'	Int
'op_enabled'	Boolean
'op_destination'	String
'op_price_default'	numeric(16,5)
'op_currency_id'	Int
'op_cid'	Int
'op_tarif_name'	String
'op_price_tarif'	numeric(16,5)
'op_currency_sign'	String
'op_currency_name'	String(3) - name_short
'op_tariff_invisible'	Boolean
'op_mcc'	String
'op_mnc'	String
'hlr_lookup'	Boolean DEPRECATED since release version 36!
'hlr_lookup_type'	String

Name	Type & Comments
'delta_price'	numeric(16,5)
'tp_name'	String
'tp_capacity'	Int

'tp_country'	String
'tp_price_id'	Int
'tp_id'	Int
'tp_enabled'	Boolean
'tp_destination'	String
'tp_price_default',	numeric(16,5)
'tp_currency_id'	Int
'dif_price'	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default
'profit'	numeric(15,2) = (dif_price/tp_price_default) * 100% if tp_price_default not equal 0, else 0
'dp_name'	String
'dp_capacity'	numeric(15,2)
'dp_tp_priority'	Int
'dp_priority'	Int
'group_name'	String
'op_group_priority'	Int
'price_priority'	Int
'hunt_stop'	Boolean
'group_id'	Int
'dialpeer_id'	Int
'weight'	Int

'client_id'	Int
'allow_internal'	Boolean
'tp_without_ip'	Boolean
'tp_price_tarif'	numeric(16,5) - tp price in the tariff's currency
'tp_currency_sign'	String
'tp_currency_name'	String(3) - name_short
'allowed_price'	Bool
'tp_tariff_invisible'	Bool
'delta_price_unit'	Int
'tp_mcc'	String
'tp_mnc'	String

2.352 sms_routing_table_get_old

Docstring:

Returns list of SMS Termination Points to which given SMS OP have route for specific destination at the given time ordered according to the routing algorithm:

- 1) by priority of SMS OP->SMS Group relation
- 2) by SMS Dialpeer priority
- 3) by priority of SMS Dialpeer->SMS TP relation
- 4) by weight of relation between SMS Dialpeer and SMS TP if priority of SMS Dialpeer->SMS TP relation is the same
- 5) by SMS TP price priority
- 6) LCR (more profitable routes will be higher in list)

Function returns also boolean parameter - "dest_changed". It determines if tested number will be changed by some conversion defined in any route (only B number conversion in OP and dialpeer is taken into account).

Function returns parameters of the targeted SMS OP within key 'op_data'.

'op_data' fields:

```
[
  'op_name',
  'op_capacity',
  'op_country',
  'op_price_id',
  'op_id',
  'op_enabled',
  'op_destination',
  'op_price_default',
  'op_currency_id',
  'op_cid',
  'op_tarif_name',
  'op_price_tarif',
  'op_currency_sign',
  'op_currency_name',
  'routing_algorithm',
]
```

Example:

- Client Request

```
{ "session_id": "7a1fe5eb3650e7f8908569c676fdec59",
  "data": { "time_zone": "0:00",
            "time_period": "2014-01-13T15:32:50",
            "op_id": 7,
            "contexts": [1],
            "number": "923000000000",
            "fields": ["tp_enabled", "delta_price", "dif_price", "tp_id",
                      "tp_price_id", "group_id", "dialpeer_id",
                      "allow_internal", "tp_without_ip", "allowed_price",
                      "tp_name", "tp_destination", "tp_country",
                      "tp_price_default", "tp_price_tarif",
                      "tp_currency_name",
                      "tp_capacity", "hunt_stop", "dp_name",
                      "dp_tp_priority", "dp_priority", "weight",
                      "group_name", "op_group_priority", "price_priority",
                      "profit", "client_id"]}
}
```

- Server Response

```
{ "status": 200,
  "time_elapsed": 0.028728,
  "dest_changed": false,
  "fields": ["tp_enabled", "delta_price", "dif_price", "tp_id",
            "tp_price_id", "group_id", "dialpeer_id",
            "allow_internal", "tp_without_ip", "allowed_price",
```

```

"tp_name", "tp_destination", "tp_country",
"tp_price_default", "tp_price_tarif", "tp_currency_name",
"tp_capacity", "hunt_stop",
"dp_name", "dp_tp_priority", "dp_priority", "weight",
"group_name", "op_group_priority", "price_priority",
"profit", "client_id"],
"message": "OK",
"op_data": ["H_in", 5, "Pakistan Mobile Mobilink", 13, 7, true,
"9230", "10.4000", 1, 1, 0, 1, 5, "T_H_in1",
"10.4000", "$", "USD", "QUALITY"],
"data": [[true, "0.00000", "0.00500", 8, 20, 3, 4, false, false,
true, "H_to", "9230", "Pakistan Mobile Mobilink",
"10.3500", "10.3500", "USD", 1, 1, 0, 500, false,
"H_p(CZ)", 0, 0, 0, "h_p (CZ)", 0, 0, "14.29", 6]]}

```

Request structure and fields description:

Request:

```

{
  "session_id": String(),
  "data": {
    'session_id': String(),
    'number': String(regex=B_NUMBER_REGEXP),
    'time_period': String() ,
    'op_id': Int(gt=0),
    'time_zone': String(regex=TZ_REGEXP),
    'show_disabled_pt': Bool(),
    'one_route': Bool(),
    'show_stat': Bool(),
    'fields': List(String()),
    'currency_id': Int(gt=0),
    'contexts': List(Int()),
    'limit': Int(gt=0),
    'offset': Int(gte=0),
  }
}

```

Name	Default value	Optional	Type	Description
'session_id'	session_info['id']	True	String()	user session id, need for checking user permissions. Currently not in use.
'number'	N/A	False	String(regex=B_NUMBER_REGEXP)	called number
'time_period'	N/A	False	String()	request time in format yyyy-MM-dd'T'HH:mm:ss
'op_id'	N/A	False	Int(gt=0)	id of chosen Origination Point

'time_zone'	'+00:00'	True	String(regex=TZ_REGEX)	time zone for 'time_period'
'show_disabled_pt'	False	True	Bool()	if "True" then disabled TPs also will be shown
'one_route'	False	True	Bool()	if "True" then report shows only one route per one Termination point.
'show_stat'	False	True	Bool()	if "True" then ACD,ASR for last 4 hours will returns
'fields'	default_fields	True	List(String())	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.
'currency_id'	session_info['currency_id']	True	Int(gt=0)	currency in which price will display
'contexts'	N/A	False	List(Int())	ids context, in which request will be processed
'limit'	N/A	True	Int(gt=0)	says how many rows will be in response
'offset'	N/A	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'op_name'	String
'op_capacity'	Int

'op_country'	String
'op_price_id'	Int
'op_id'	Int
'op_enabled'	Boolean
'op_destination'	String
'op_price_default'	numeric(16,5)
'op_currency_id'	Int
'op_cid'	Int - OP currency id
'op_price_tarif'	numeric(16,5) - op price in the tariff's currency
'op_currency_sign'	String
'op_currency_name'	String
'delta_price'	numeric(16,5)
'tp_name'	String
'tp_capacity'	Int
'tp_country'	String
'tp_price_id'	Int
'tp_id'	Int
'tp_enabled'	Boolean
'tp_destination'	String
'tp_price_default'	numeric(16,5)

'tp_currency_id'	Int
'dif_price'	numeric(16,5) - difference between op price and tp price in the default currency that equal op_price_default - tp_price_default
'profit'	numeric(15,2) = (dif_price/tp_price_default) * 100% if tp_price_default not equal 0, else 0
'dp_name'	String
'dp_capacity'	numeric(15,2)
'dp_tp_priority'	Int
'dp_priority'	Int
'group_name'	String
'op_group_priority'	Int
'price_priority'	Int
'hunt_stop'	Boolean
'group_id'	Int
'dialpeer_id'	Int
'weight'	Int
'client_id'	Int
'asr'	numeric or null
'acd'	Int or null
'delta_profit'	numeric(15,2)
'allow_internal'	Boolean
'tp_without_ip'	Boolean

'tp_price_tarif'	numeric(16,5) - tp price in the tariff's currency
'tp_currency_sign'	String
'tp_currency_name'	String
'allowed_price'	Bool
'op_tariff_invisible'	Bool
'tp_tariff_invisible'	Bool
'delta_price_unit'	Int
'attempts'	Int

2.353 sms_start_page_get_clients_close_to_credit_limit

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('balance_in_default_currency', 'credit_limit', 'client_name', 'percent', 'currency_name', 'client_id', 'balance')"],
  'get_cached_result': 'Bool',
  'interval': "Enum('yesterday', 'today')",
  'limit': 'Integer',
  'offset': 'Integer',
  'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['balance_in_default_curren	True	List([Enum('balance_in_default_	

	cy', 'credit_limit', 'client_name', 'percent', 'currency_name', 'client_id', 'balance']		currency', 'credit_limit', 'client_name', 'percent', 'currency_name', 'client_id', 'balance')], min_length: 0)	
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
get_cached_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
balance_in_default_currency	Numeric(23,2)
credit_limit	Numeric(23,2)
client_name	String
percent	Numeric(23,2)
currency_name	String
client_id	Int
balance	Numeric(23,2)

2.354 sms_start_page_get_closed_companies

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id')"],  
          'get_cached_result': 'Bool',  
          'interval': "Enum('yesterday', 'today')",  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'session_id': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram
get_cache_d_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id']	True	List([Enum('currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id')], min_length: 0)	
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
currency_name	String
balance_in_default_currency	Numeric(23,2)
client_name	String
balance	Numeric(23,2)
client_id	Int

2.355 sms_tariff_add**Docstring:**

Creates new SMS OP or SMS TP tariff.

Example:

- Client Request

```
{"session_id": ca84cd3eaa596f2e3f7836db065777da,
 "data": {"currency_id": 76,
          "name": "123456tar",
          "is_tp": true,
          "tariff_type_id": 2
        }
}
```

Request structure and fields description:

```
{'data': {'currency_id': 'Integer',
          'invisible': 'Bool',
          'is_tp': 'Bool',
          'name': 'String',
          'session_id': 'String',
          'tariff_type_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
currency_id	N/A	False	Integer(gt: 0)	currency of the creating tariff.
tariff_type_id	N/A	False	Integer(gt: 0, lt: 3)	defines tariff's type (1-destination, 2-MCC-MNC)
invisible	False	True	Bool	invisible or not
name	N/A	False	String(allow_blank: False)	name of the tariff. The name should unique.
session_id	N/A	True	String(allow_blank: False)	by default from user session
is_tp	N/A	False	Bool	if "true" then function creates TP tariff, if "false" - creates OP tariff.

2.356 sms_tariff_set

Docstring:

Renames chosen SMS tariff.

Example:

- Client Request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {"id": 5006,  
            "name": "1 tarif",  
            "is_tp": true}  
}
```

Request structure and fields description:

```
{'data': {'id': 'Integer',  
          'invisible': 'Bool',  
          'is_tp': 'Bool',  
          'name': 'String',  
          'session_id': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	new name of the tariff
session_id	N/A	True	String(allow_blank: False)	by default from user session
id	N/A	False	Integer(gt: 0)	OP tariff or TP tariff id. Generally OP tariff id is odd number, while TP tariff id is even number.
invisible	N/A	True	Bool	invisible or not
is_tp	N/A	False	Bool	if "true" then function searches tariff by id among TP tariffs

2.357 sms_tariffs_get

Docstring:

Returns list of SMS OP and SMS TP tariffs names, types and their ids filtered by clients locked by the current user.

Example:

- Client Request:

```
{"session_id": "0ab11223a707b59600993e99da72d167",  
  "data": {"fields": ["id", "name"],  
           "sort_by": [{"id": "desc"}]}}  
}
```

- Server Response:

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[4035, 'Tarif Blossom_Zimbabwe_in (104081)'],  
           [4034, 'Tarif Blossom_Zimbabwe_out (104081)']],  
  "time_elapsed": 0.010708  
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('type_id', 'id', 'name')"],  
          'session_id': 'String',  
          'sort_by': [{"Enum('type_id', 'id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	by default from user session
fields	['type_id', 'id', 'name']	True	List([Enum('type_id', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields".If empty - returns error.
sort_by	[]	True	List([{"Enum('type_id', 'id', 'name')": Enum('asc', 'desc', '')}], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"id": "asc"}] - sorts result ascending by field "id".

Default fields:

Name	Type & Comments
type_id	Int - tariff's type (1-destination, 2-MCC-MNC)
id	Int
name	String

2.358 sms_tariffs_op_get

Docstring:

Calls API function `_sms_tariffs_get` with option `'is_tp'=False`.

Example:

- Client Request

```
{"session_id": "0ab11223a707b59600993e99da72d167",
  "data": {"currency_id": 1,
           "sort_by": [{"name": ""}],
           "fields": ["tariff_id", "name_currency"]}}
```

- Server Response:

```
{"status": 200,
  "fields": ["tariff_id", "name_currency"],
  "message": "OK",
  "data": [[4233, "0001111 (USD)",
             [4149, "000_Zv_in (USD)",
              [4221, "011 (USD)",
               [4485, "1 (USD)"]],
            "time_elapsed": 0.010708
  ]}
```

`_sms_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of SMS OP or SMS TP tariffs existent in the database.

Request structure and fields description:

`_sms_tariffs_get`

```
{'data': {'client_or_pt_name': 'String',
          'currency_id': 'Integer',
          'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')"],
          'is_tp': 'Bool',
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'point_name': 'String',
          'session_id': 'String',
          'show_invisible': 'Bool or Null',
          'show_pts': 'Bool',
          'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer',
          'tariff_type_id': 'Integer'},
  'session_id': 'String'}
```


data validation description:

Name	Default value	Optional	Type	Description
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id'): Enum('asc', 'desc', ")}], min_length: 0)	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"name": "asc"}] - sorts result ascending by field "name".
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the request.
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
tariff_type_id	N/A	True	Integer(gt: 0, lt: 3)	filter by tariff's type (1-destination, 2-MCC-MNC)

client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or point.
session_id	N/A	True	String(allow_blank: False)	by default from user session
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
show_pts	True	True	Bool	not in use
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null

Default fields:

_sms_tariffs_get

Name	Type & Comments
dstcode_count	Int - how many actual prices for different destination codes in the tariff
name_currency	String - tariff name with currency (example - 'test_tariff(USD)')
currency_name	String(3) - name_short
invisible	Boolean
point_info	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
currency_id	Int
tariff_id	Int
name	String

counter	Int - how many points use this tariff
prices_count	Int - how many actual prices in the tariff
client_info	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id
tariff_type_id	Int - tariff's type (1-destination, 2-MCC-MNC)

2.359 sms_tariffs_tp_get

Docstring:

Calls API function `_sms_tariffs_get` with option `'is_tp'=True`.

Example:

- Client Request

```
{
  "session_id": "90b28b952c569c96a967f0501138965e",
  "data": {
    "limit": 31,
    "sort_by": [{"name": ""}],
    "offset": 0,
    "fields": ["tariff_id", "name", "currency_id",
               "dstcode_count", "prices_count",
               "name_currency", "client_point_info"]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": [
    "tariff_id", "name", "currency_id", "dstcode_count", "prices_count",
    "name_currency", "point_info", "client_info"],
  "message": "OK",
  "data": [
    [4932, "000999", 7, 65501, 65501, "000999 (UAH)", [[4918, "000999TP1"], [5076, "1234567"]],
    [[564, "12Ls"], [609, "sms1001_"]],
  ],
  "time_elapsed": 0.012435
}
```

`_sms_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of SMS OP or SMS TP tariffs existent in the database.

Request structure and fields description:

_sms_tariffs_get

```
{'data': {'client_or_pt_name': 'String',
          'currency_id': 'Integer',
          'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')"],
          'is_tp': 'Bool',
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'point_name': 'String',
          'session_id': 'String',
          'show_invisible': 'Bool or Null',
          'show_pts': 'Bool',
          'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer',
          'tariff_type_id': 'Integer'},
        'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info', 'tariff_type_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info',	defines field(s) on which result will be sorted. Result can be sorted ascending or descending. For example [{"name": "asc"}] - sorts

			'tariff_type_id'): Enum('asc', 'desc', ")}], min_length: 0)	result ascending by field "name".
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the request.
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
tariff_type_id	N/A	True	Integer(gt: 0, lt: 3)	filter by tariff's type (1-destination, 2-MCC-MNC)
client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or point.
session_id	N/A	True	String(allow_blank: False)	by default from user session
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
show_pts	True	True	Bool	not in use
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null

Default fields:

_sms_tariffs_get

Name	Type & Comments
dstcode_count	Int - how many actual prices for different destination codes in the tariff
name_currency	String - tariff name with currency (example - 'test_tariff(USD)')
currency_name	String(3) - name_short
invisible	Boolean
point_info	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
currency_id	Int
tariff_id	Int
name	String
counter	Int - how many points use this tariff
prices_count	Int - how many actual prices in the tariff
client_info	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id
tariff_type_id	Int - tariff's type (1-destination, 2-MCC-MNC)

2.360 sms_top_countries_get

Docstring:

Returns list of 10 Countries with most SMS quantity for previous day (00:00:00 - 23:59:59).

Example:

- Client Request

```
{ "session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {  
    "client_id": 11,  
    "direction": "out",  
    "fields": [ "asr",  
               "delta_price",  
               "duration",  
               "country" ]  
  }  
}
```

Request structure and fields description:

```
{ 'data': { 'client_id': 'Integer',  
           'currency_id': 'Integer',  
           'direction': 'String or String',  
           'fields': [ "Enum('asr', 'country', 'op_sum', 'tp_sum', 'delta_price', 'sa')"]  
         },  
  'session_id': 'String',  
  'session_id': 'String' }
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default from current session
client_id	N/A	False	Integer(gt: 0)	client to get countries statistic
currency_id	1	True	Integer(gt: 0)	currency ID for statistic
direction	N/A	False	Or(String(allow_blank: in), String(allow_blank: out))	filter by traffic direction relatively to the softswitch
fields	['asr', 'country', 'op_sum', 'tp_sum', 'delta_price', 'sa']	True	List([Enum('asr', 'country', 'op_sum', 'tp_sum', 'delta_price', 'sa')], min_length: 0)	

Default fields:

Name	Type & Comments
asr	String
country	String
op_sum	Numeric(20,2)
tp_sum	Numeric(20,2)
delta_price	Numeric(16,2) - difference between op_sum and tp_sum
sa	Int - success attempts

2.361 sms_tp_add_or_update

Docstring:

1) Creates new disabled termination point (using function '_sms_tp_add') if "id" not specified in request.
Function raises error if client which owns given balance is invisible.

2) Updates termination point tariff, "description", point name, flag "invisible" if "id" is present and "contexts" is absent in the request (using function '_sms_tp_update').

If flag "invisible" was set to "true" - function disables point and removes TP's relations with dialpeers and groups.

3) If "contexts" and "id" are present in request it calls function '_sms_tp_update_within_contexts' which updates point's parameters ("capacity", "max_call_time", point status, "ani_regexp" and so on) within given context(s).

Function can copy parameters of point with given id to the chosen contexts (it used for web only).

Function raises error if user tries to change invisible point. If point status ("enabled" flag) was changed manually (point wasn't closed due to achieving "maximum negative balance" value) function creates new email notification.

`_sms_tp_add` ([data_validation](#)) ([default_fields](#))

Creates new disabled termination point, add new point to the schema routing and each contexts. Allows to specify balance, tariff, "description", "name", flag "invisible" for creating TP.

Function raises error if point with such name already exists in database or if tried to assign point to client which is invisible.

Returns new point id.

Example:

- Client Request

```
{"session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {"tarif_id": 14,
           "invisible": false,
           "description": "Don't touch!",
           "name": "my_tp",
           "balance_id": 1}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 22, "time_elapsed": 0.260494}
```

`_sms_tp_update` ([data_validation](#)) ([default_fields](#))

Changes common parameters of given termination point: name, description, tariff and invisible flag.
If "invisible" flag was set as "true" function disables given TP and removes all relations of the TP with groups and dialpeers.

Example:

- Client Request

```
{"session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {"id": 4,
           "invisible": false,
           "description": "DO NOT TOUCH THIS POINT."}
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.107607}
```

`_sms_tp_update_within_contexts` ([data_validation](#)) ([default_fields](#))

Updates termination point parameters within chosen context. This function is used for copying TP to the contexts specified within list "contexts" in web panel.

Adds new record to the table api.web_mail if point status ("enabled" or "disabled") was changed manually (point wasn't closed due to achieving maximum negative balance value). The function raises error if API user is trying to change invisible point.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "id": 4,
    "contexts": [
      {
        "max_call_time": 3660,
        "ani_block": "",
        "enabled": true,
        "context_id": 1,
        "ani_regex": "",
        "capacity": 5001,
        "dnis_regex": ""
      }
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.088887
}
```

Request structure and fields description:

_sms_tp_add

```
{
  'data': {
    'balance_id': 'Integer',
    'block_spam_numbers': 'Bool',
    'client_id': 'Integer',
    'description': 'String',
    'dlr_mode': "Enum('default_dlr', 'force_dlr', 'dlr_replacement')",
    'ext_id': 'Integer',
    'invisible': 'Bool',
    'mps_limit': 'Integer',
    'name': 'String',
    'point_manager_id': 'Integer or Null',
    'tariff_id': 'Integer',
    'tariff_type_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
invisible	False	True	Bool	boolean option. If "True" - makes point invisible and disabled

ext_id	0	True	Integer(gte: 0)	TP id for external billing
block_spam_numbers	False	True	Bool	if "True" - blocks spam numbers every 90 seconds
description	""	True	String(allow_blank: True)	
tariff_type_id	1	True	Integer(gte: 1, lte: 2)	(ONLY FOR DEVELOPER USE) Type of SMS tariff: 1 - Destination, 2 - MCCMNC
name	N/A	False	String(allow_blank: False)	
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	TP manager user id for update notifications
tarif_id	0	True	Integer(gte: 0)	tariff currency has to be equal currency of chosen balance
mps_limit	100	True	Integer(gt: 0, lte: 1000)	Messages per second limit
dlr_mode	default_dlr	True	Enum('default_dlr', 'force_dlr', 'dlr_replacement')	"default_dlr" - simple DLR behaviour "force_dlr" - status "DELIVRD" will be returned in case if status has been requested by OP; "dlr_replacement" - not in use.
balance_id	N/A	True	Integer(gt: 0)	client which owns given balance should be visible.
client_id	N/A	True	Integer(gt: 0)	client with given id should be visible.

_sms_tp_update

```
{'data': {'block_spam_numbers': 'Bool',
          'description': 'String',
          'dlr_mode': "Enum('default_dlr', 'force_dlr', 'dlr_replacement')",
          'id': 'Integer',
          'invisible': 'Bool',
          'name': 'String',
          'point_manager_id': 'Integer or Null',
          'tarif_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
invisible	N/A	True	Bool	boolean option. If "True" - makes point invisible and disabled
description	N/A	True	String(allow_blank: True)	
block_spam_numbers	False	True	Bool	if "True" - blocks spam numbers every 90 seconds
id	N/A	False	Integer(gt: 0)	
tarif_id	N/A	True	Integer(gt: 0)	new tariff of given TP
name	N/A	True	String(allow_blank: False)	new point name
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	TP manager user id for update notifications. If key value is "NULL" manager user will be cleared.If key not present in function request data user id will not be updated.
dlr_mode	default_dlr	True	Enum('default_dlr', 'force_dlr', 'dlr_replacement')	"default_dlr" - simple DLR behaviour"force_dlr" - status "DELIVRD" will be returned in case if status has been requested by OP;"dlr_replacement" - not in use.

_sms_tp_update_within_contexts

```
{'data': {'contexts': [{ 'ani_block': 'String',  
                        'ani_regexp': 'String',  
                        'capacity': 'Integer',  
                        'context_id': 'Integer',  
                        'dnis_regexp': 'String',  
                        'enabled': 'Bool',  
                        'mps_limit': 'Integer'}],  
  'id': 'Integer'},  
  'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	
capacity	N/A	True	Integer(gte: 0, lte: 1000)	defines the limit on the number of simultaneous messages that can pass through this point.
ani_block	N/A	True	String(allow_blank: True, max_length: 255)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
mps_limit	N/A	True	Integer(gt: 0, lte: 1000)	Messages per second limit
enabled	N/A	True	Bool	point status. If "True" - point can participate in routing. If point status was changed creates new email notification.
ani_regexp	N/A	True	String(allow_blank: True, max_length: 64)	A number regexp
dnis_regexp	N/A	True	String(allow_blank: True, max_length: 64)	B number regexp

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([data.contexts], min_length: 0)	
id	N/A	False	Integer(gt: 0)	

2.362 sms_tp_authorization_add

Docstring:

Creates new SMS authorization account (account for authorization by

login and password) within chosen context and OP.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "tp_id": 4331,
    "context_id": 1,
    "login": "test_sip",
    "password": "121314"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'tp_id': Int(gt=0),
    'host': String(),
    'port': Int(gt=0, lte=65535),
    'enabled': Bool(),
    'protocol': Int(),
    'login': String(),
    'password': String(max_length=SMPP_PASSWORD_MAX_LENGTH),
    'context_id': Int(gt=0),
    'system_type': String(regex="^[^\\s]{0,13}$", allow_blank=True) | Null,
    'src_ton': Int(gte=0, lte=6) | Null,
    'dst_ton': Int(gte=0, lte=6) | Null,
    'src_npi': Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) | Null,
    'dst_npi': Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) | Null,
    'weight': Int(gte=0, lte=100),
  }
}
```

Name	Default value	Optional	Type	Description
'tp_id'	N/A	False	Int(gt=0)	Origination point id to which given authorization will be added
'host'	N/A	False	String()	single IP address or subnet IP address, or domain name.
'port'	N/A	False	Int(gt=0, lte=65535)	
'enabled'	False	False	Bool()	
'protocol'	2	True	Int()	
'login'	N/A	False	String()	SMS auth login. Should be unique.

'password'	N/A	False	String(max_length=SMPP_PASSWORD_MAX_LENGTH)	SMS auth password
'context_id'	N/A	False	Int(gt=0)	context to which authorization will be added
'system_type'	None	True	String(regex="^[^\s]{0,13}\$", allow_blank=True) Null	
'src_ton'	None	True	Int(gte=0, lte=6) Null	
'dst_ton'	None	True	Int(gte=0, lte=6) Null	
'src_npi'	None	True	Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) Null	
'dst_npi'	None	True	Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) Null	
'weight'	0	True	Int(gte=0, lte=100)	

2.363 sms_tp_authorization_del

Docstring:

Removes chosen SMS authorization from the given context.

Example:

```
{ "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": { "context_id": 1,
            "login": "test_sip" }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'id': Int(gt=0),
    'context_id': Int(gt=0),
    'tp_id': Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
'id'	N/A	True	Int(gt=0)	SMS account id
'context_id'	N/A	False	Int(gt=0)	context in which given SMS authorization have to be removed
'tp_id'	N/A	True	Int(gt=0)	filter by TP id

2.364 sms_tp_authorization_get

Docstring:

Function retrieves SMS accounts existent in the database.

Example:

- Client Request:

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
 "data": {"tp_id": 232,
          "context_id": 1,
          "fields": ["login"]}
}
```

- Server Response:


```
{
  "status": 200,
  "fields": ["login"],
  "message": "OK",
  "data": [{"test_sip"}],
  "time_elapsed": 0.008435
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'enabled': 'Bool',
          'fields': ['String'],
          'limit': 'Integer',
          'login': 'String',
          'offset': 'Integer',
          'sort_by': [{"Enum('protocol', 'weight', 'context_id', 'dst_npi', 'host', 'src_npi', 'system_type', 'password', 'dst_ton', 'id', 'enabled', 'port', 'tp_id', 'login', 'src_ton')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer'},
      'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
offset	N/A	True	Integer(gte: 0)	filter by Termination Point
tp_id	N/A	True	Integer(gt: 0)	filter by Termination Point
contexts	[]	False	List([Integer()], min_length: 0)	filter by context id. If absent in request - returns data from all contexts.
enabled	N/A	True	Bool	filter by is SMS host enabled
login	N/A	True	String(allow_blank: False)	filter by SMS account login
limit	N/A	True	Integer(gte: 0)	filter by Termination Point
sort_by	[]	True	List([Enum('protocol', 'weight',	

			'context_id', 'dst_npi', 'host', 'src_npi', 'system_type', 'password', 'dst_ton', 'id', 'enabled', 'port', 'tp_id', 'login', 'src_ton'): Enum('asc', 'desc', ")}], min_length: 0)	
fields	['protocol', 'weight', 'context_id', 'dst_npi', 'host', 'src_npi', 'system_type', 'password', 'dst_ton', 'id', 'enabled', 'port', 'tp_id', 'login', 'src_ton']	True	List([String(allow_blank: False)], min_length: 0)	required fields If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
protocol	Int
weight	Int
context_id	Int
dst_npi	Int
host	String
src_npi	Int
system_type	String
password	String
dst_ton	Int
id	Int
enabled	Bool
port	Int
tp_id	Int

login	String
src_ton	Int

2.365 sms_tp_authorization_set

Docstring:

Function allows to change password of chosen SMS authorization in the given context.

Example:

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "login": "test_sip",
    "context_id": 1,
    "password": "132456567"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    'id': Int(gt=0),
    'login': String() ,
    'password': String(max_length=SMPP_PASSWORD_MAX_LENGTH),
    'host': String(),
    'port': Int(gt=0, lte=65535),
    'enabled': Bool(),
    'protocol': Int(),
    'context_id': Int(gt=0),
    'system_type': String(regex="^[^\s]{0,13}$", allow_blank=True) | Null,
    'src_ton': Int(gte=0, lte=6) | Null,
    'dst_ton': Int(gte=0, lte=6) | Null,
    'src_npi': Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) | Null,
    'dst_npi': Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) | Null,
    'weight': Int(gte=0, lte=100),
  }
}
```

Name	Default value	Optional	Type	Description
'id'	N/A	False	Int(gt=0)	

'login'	N/A	True	String()	SMS account login which password have to be changed
'password'	N/A	True	String(max_length=SMPP_PASSWORD_MAX_LENGTH)	new password
'host'	N/A	True	String()	
'port'	N/A	True	Int(gt=0, lte=65535)	
'enabled'	N/A	True	Bool()	
'protocol'	N/A	True	Int()	
'context_id'	N/A	False	Int(gt=0)	context where given SMS authorization have to be changed
'system_type'	N/A	True	String(regex="^[^\s]{0,13}\$", allow_blank=True) Null	
'src_ton'	N/A	True	Int(gte=0, lte=6) Null	
'dst_ton'	N/A	True	Int(gte=0, lte=6) Null	
'src_npi'	N/A	True	Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) Null	
'dst_npi'	N/A	True	Enum(0, 1, 3, 4, 6, 8, 9, 10, 14, 18) Null	
'weight'	N/A	True	Int(gte=0, lte=100)	

2.366 sms_tp_free_dialpeers_get

Docstring:

Returns list of dialpeers that not connected with chosen TP.

Example:

- Client Request

```
{ "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": { "sort_by": [ { "dialpeer_name": "" } ],
    "contexts": [1],
    "tp_id": 4,
    "dialpeer_name": "",
    "fields": [ "dialpeer_id", "dialpeer_name" ] }
}
```

- Server Response

```
{ "status": 200,
  "fields": [ "dialpeer_id", "dialpeer_name" ],
  "message": "OK",
  "data": [ [4, "H_p (CZ)" ], [3, "tst.dialpeer2 (CZ)" ],
    [2, "tst.dialpeer (CZ)" ] ],
  "time_elapsed": 0.000866 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "dialpeer_name": String(allow_blank=True),
    "tp_id": Int(gt=0),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"dialpeer_name"	N/A	True	String(allow_blank=True)	Filter by dialpeer's name
"tp_id"	N/A	True	Int(gt=0)	termination point id

"contexts"	N/A	False	List(Int(gt=0))	required contexts ids
"fields"	default_fields	True	List(String())	requested fields, by defaults names mentioned in default fields
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field on which list will be sorted
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
"dialpeer_id"	Int
"dialpeer_name"	String

2.367 sms_tp_settings_copy

Docstring:

Copies specified SMS TP settings from given context to contexts that in given contexts list.

Request structure and fields description:

```
{'data': {'context_id_from': 'Integer',
          'context_ids_to': ['Integer'],
          'copy_ani_block': 'Bool',
          'copy_ani_regexp': 'Bool',
          'copy_auth_list': 'Bool',
          'copy_capacity': 'Bool',
          'copy_dnis_regexp': 'Bool',
          'copy_enabled': 'Bool',
          'copy_mps_limit': 'Bool',
          'id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
context_id_from	N/A	False	Integer(gt: 0)	context id from which data will be copied to another context/contexts
copy_capacity	False	True	Bool	capacity value will be copied or not
copy_ani_regexp	False	True	Bool	ani regexp will be copied or not
context_ids_to	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of context ids where data must be copied from another context
copy_dnis_regexp	False	True	Bool	dnis regexp will be copied or not
copy_mps_limit	False	True	Bool	messages per second limit will be copied or not
id	N/A	False	Integer(gt: 0)	point id
copy_enabled	False	True	Bool	enabled flag will be copied or not
copy_ani_block	False	True	Bool	ani block will be copied or not
copy_auth_list	False	True	Bool	authentication list will be copied or not

2.368 sms_tps_free_dialpeer_get

Docstring:

Returns list of termination points which aren't connected with chosen dialpeer.

Example:

- Client Request

```
{"session_id":"773b6b683024ac6b6cfd95c8b034e659",
"data":{"tp_name":"","
      "sort_by":[{"tp_name":""}],
      "contexts":[1],
      "dialpeer_id":6,
      "fields":["tp_id","tp_name"]}}
}
```

- Server Response

```
{"status": 200,
"fields": ["tp_id", "tp_name"],
"message": "OK",
"data": [[12, "H_test_tp"], [6, "ikex.TP1"], [10, "TP-14"]],
"time_elapsed": 0.008065}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tp_name": String(allow_blank=True),
    "tp_id": Int(gt=0),
    "dialpeer_id": Int(gt=0),
    "contexts": List(Int(gt=0)),
    "fields": List(String()),
    "sort_by": Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$"))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"tp_name"	N/A	True	String(allow_blank=True)	termination point name

"tp_id"	N/A	True	Int(gt=0)	id of the termination point
"dialpeer_id"	N/A	True	Int(gt=0)	dialpeer id
"contexts"	[]	False	List(Int(gt=0))	ids of required contexts
"fields"	default_fields	True	List(String())	requested fields, by defaults fields mentioned in default fields
"sort_by"	[]	True	Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$")))	field on which list will be sorted
"limit"	0	True	Int(gte=0)	rows quantity per one request
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
'tp_id'	Int
'tp_name'	String

2.369 sms_tps_free_group_get

Docstring:

Returns list of TPs aren't connected with chosen group.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "tp_name": "",
    "sort_by": [{"tp_name": ""}],
    "group_id": 8,
    "contexts": [1],
    "fields": ["tp_id", "tp_name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["tp_id", "tp_name"],
  "message": "OK",
  "data": [[8, "H_P_to"], [12, "H_test_tp"], [6, "st.TP1"],
    [14, "pablo_tp"], [10, "TP-14"]],
  "time_elapsed": 0.007114}
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tp_name": String(allow_blank=True),
    "tp_id": Int(),
    "group_id": Int(),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"tp_name"	N/A	True	String(allow_blank=True)	filter by TP name
"tp_id"	N/A	True	Int()	TP id
"group_id"	N/A	True	Int()	group id
"contexts"	[]	False	List(Int())	list of contexts id where search

"				relations
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"tp_name": "asc"}] - sorts result ascending by field "tp_name"
"limit"	0	True	Int(gte=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows..

Default fields:

Name	Type & Comments
"tp_id"	Int
"tp_name"	String

2.370 sms_tps_free_op_get

Docstring:

Returns list of TPs which aren't connected with chosen OP, i.e. all termination points that aren't belong to the same group (simple or not) as given OP within chosen context(s).

OP may be configured only in simple mode or only advanced mode at the same time.

Example:

- Client Request

```
{ "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": { "op_id": 17,
            "sort_by": [ { "name": "" } ],
            "name": "",
            "contexts": [ 1 ],
            "fields": [ "id", "name" ] }
}
```

- Server Response

```
{ "status": 200,
  "fields": [ "id", "name" ],
  "message": "OK",
  "data": [ [ 4, "Test_1_TP" ], [ 8, "ss_TP" ], [ 2, "H_out" ], [ 9, "H_to" ],
            [ 12, "H_test_tp" ], [ 6, "ss1.TP1" ], [ 3, "New_TP" ],
            [ 22, "my_tp" ], [ 10, "TP-14" ] ],
  "time_elapsed": 0.009625 }
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.371 sms_tps_from_dialpeers_del

Docstring:

Removes relations between chosen dialpeer and TP(s) or between chosen TP and dialpeer(s) within given context(s).

To remove relation created in Simple mode you should use API function rel_op_tp_del.

Example:

- Client Request

```
{ "session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": { "dialpeers_ids": [6],  
            "contexts": [1],  
            "tp_ids": ["6", "10"] }  
}
```

- Server Response

```
{ "status": 200, "message": "OK", "time_elapsed": 0.067966 }
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "tp_ids": List(Int(gt=0)),  
    "dialpeers_ids": List(Int(gt=0)),  
    "contexts": List(Int(gt=0)),  
  }  
}
```

Name	Default value	Optional	Type	Description
"tp_ids"	N/A	False	List(Int(gt=0))	termination points ids list
"dialpeers_ids"	N/A	False	List(Int(gt=0))	list of of dialpeer ids which shouldn't contain service dialpeer's id
"contexts"	N/A	False	List(Int(gt=0))	required contexts ids

2.372 sms_tps_from_groups_del

Docstring:

Removes relations between chosen group(s) and Termination Point(s) within given context(s).

Function allows to delete only relations created in advanced mode.

Example:

- Client Request

```
{"session_id": "b9fac2d963ec1967841dfe46202bfc67",  
  "data": {"contexts": [1],  
           "tp_ids": [],  
           "groups_ids": [878]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.012485}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "tp_ids": List(Int(gt=0)),  
    "groups_ids": List(Int(gt=0)),  
    "contexts": List(Int()),  
  }  
}
```

Name	Default value	Optional	Type	Description
"tp_ids"	N/A	False	List(Int(gt=0))	list of TPs id
"groups_ids"	N/A	False	List(Int(gt=0))	list of group ids
"contexts"	N/A	False	List(Int())	list of contexts id to which relation should be deleted

2.373 sms_tps_get

Docstring:

Returns list of termination points existing in database in given context(s).

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"limit":2,
        "sort_by":[{"name":""}],
        "invisible":false,
        "contexts":[1],
        "offset":0,
        "fields":["id","name","client_name","client_id","enabled",
                  "capacity","tariff_id","tariff_name",
                  "tariff_name_currency"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["id", "name", "client_name", "client_id", "enabled",
            "capacity", "tariff_id", "tariff_name",
            "tariff_name_currency"],
  "message": "OK",
  "data": [[4, "Test_1_TP", "Test_supplier", 2, true, 5001, 4,
            "Test_tariff_TP", "Test_tariff_TP (USD)"],
           [18, "ss_TP", "ss_Co", 15, true, 500, [], 22,
            "ss_TP_PrIcE", "Anna_Yess_TP_PrIcE (EUR)"]],
  "time_elapsed": 0.013895}
```

Request structure and fields description:

```
{'data': {'clients_ids': ['Integer'],
          'contexts': ['Integer'],
          'enabled': 'Bool',
          'ext_id': 'Integer',
          'fields': ["Enum('client_invisible', 'ext_id', 'context_id', 'currency_id', '
balance_name', 'balance_id', 'block_spam_numbers', 'id', 'capacity', 'ani_block', 'clie
nt_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'tariff_name', 'descri
ption', 'dlr_mode', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tar
iff_id', 'name', 'point_manager_id', 'enabled', 'dnis_regexp')"],
          'id': 'Integer',
          'invisible': 'Bool or Null',
          'limit': 'Integer',
          'manager_id': 'Integer',
          'name': 'String',
```

```

        'name_or_client_name': 'String',
        'offset': 'Integer',
        'owner_id': 'Integer',
        'sort_by': [{"Enum('client_invisible', 'ext_id', 'context_id', 'currency_id',
'balance_name', 'balance_id', 'block_spam_numbers', 'id', 'capacity', 'ani_block', 'cli
ent_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'tariff_name', 'descr
iption', 'dlr_mode', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'ta
riff_id', 'name', 'point_manager_id', 'enabled', 'dnis_regexp')": "Enum('asc', 'desc',
''")}]},
        'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['client_invisible', 'ext_id', 'context_id', 'currency_id', 'balance_name', 'balance_id', 'block_spam_numbers', 'id', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'tariff_name', 'description', 'dlr_mode', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'dnis_regexp']	True	List([Enum('client_invisible', 'ext_id', 'context_id', 'currency_id', 'balance_name', 'balance_id', 'block_spam_numbers', 'id', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'tariff_name', 'description', 'dlr_mode', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'dnis_regexp')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
name	N/A	True	String(allow_blank: True)	filter by point name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
clients_ids	N/A	True	List([Integer()], min_length: 0)	filter by clients
owner_id	N/A	True	Integer(gt: 0)	filter by owner
name_or_client_name	N/A	True	String(allow_blank: True)	filter by point or client name

offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
sort_by	[]	True	List([Enum('client_invisible', 'ext_id', 'context_id', 'currency_id', 'balance_name', 'balance_id', 'block_spam_numbers', 'id', 'capacity', 'ani_block', 'client_name', 'mps_limit', 'tariff_type_id', 'tariff_name_currency', 'tariff_name', 'description', 'dlr_mode', 'currency_name_short', 'client_id', 'invisible', 'ani_regexp', 'tariff_id', 'name', 'point_manager_id', 'enabled', 'dnis_regexp'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result sorts
manager_id	N/A	True	Integer(gt: 0)	filter by client manager
ext_id	N/A	True	Integer(gt: 0)	TP id for external billing
contexts	[]	True	List([Integer()], min_length: 0)	filter by contexts. If is absent function searches termination points in all contexts.
id	N/A	True	Integer(gt: 0)	id of Termination Point
invisible	False	True	Or(Bool, Null)	boolean option, if "True" returns only invisible TPs
enabled	N/A	True	Bool	boolean option, if equal "True" returns only

				enabled TPs
--	--	--	--	-------------

Default fields:

Name	Type & Comments
client_invisible	Boolean
ext_id	Integer
context_id	Integer
currency_id	Integer
balance_name	String
balance_id	Integer
block_spam_numbers	Boolean
id	Integer
capacity	Integer
ani_block	String
client_name	String
mps_limit	Integer
tariff_type_id	Integer
tariff_name_currency	String - DEPRECATED
tariff_name	String
description	String
dlr_mode	String

currency_name_short	String
client_id	Integer
invisible	Boolean
ani_regexp	String
tariff_id	Integer
name	String
point_manager_id	Integer - Point manager user id for update notifications
enabled	Boolean
dnis_regexp	String

2.374 sms_tps_groups_get

Docstring:

Returns Termination Points connected with chosen group or groups connected with chosen Termination Point.

Function can filter result by point or group name. You should use TP name filter in conjunction with group id and group name filter - with TP id.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "limit": 101,
    "sort_by": [{"tp_name": ""}],
    "group_id": 8,
    "contexts": [1],
    "offset": 0,
    "fields": ["tp_id", "group_id", "tp_name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["tp_id", "group_id", "tp_name"],
  "message": "OK",
}
```

```
"data": [[4, 8, "Test_1_TP"]],
"time_elapsed": 0.008679}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tp_name": String(allow_blank=True),
    "group_name": String(allow_blank=True),
    "tp_id": Int(),
    "group_id": Int(),
    "contexts": List(Int()),
    "fields": List(String()),
    "sort_by": List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc)|(desc)|()$")))),
    "limit": Int(gte=0),
    "offset": Int(gte=0),
  }
}
```

Name	Default value	Optional	Type	Description
"tp_name"	N/A	True	String(allow_blank=True)	filter by TP name or first symbols of TP name
"group_name"	N/A	True	String(allow_blank=True)	filter by group name or first symbols of group name
"tp_id"	N/A	True	Int()	termination point id
"group_id"	N/A	True	Int()	group id
"contexts"	[]	False	List(Int())	context ids where search relation
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
"sort_by"	[]	True	List(Dict(Key(field from list default_fields): VALUE(String(regex="^(asc) (desc) ()\$"))))	field on which result will be sorted. Result can be sorted ascending, descending or . For example [{"tp_name": "asc"}] - sorts result ascending by field "tp_name"
"limit"	0	True	Int(gte=0)	says how many rows will be in

				response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows..

Default fields:

Name	Type & Comments
'group_id'	Int
'tp_id'	Int
'tp_name'	String
'group_name'	String

2.375 sms_tps_op_get

Docstring:

Returns a list of TPs which are connected with given OP in simple and advanced routing modes, i.e. termination points that satisfy condition: Points and dialpeers connected with them must belong to the same group as OP in the chosen context. OP may be configured only in simple mode or only advanced mode at the same time.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "op_id": 17,
    "sort_by": [{"name": ""}],
    "contexts": [1],
  }
}
```

```
"fields":["id","name"]}]}
```

- Server Response

```
{ "status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[24, "NewTP"]],  
  "time_elapsed": 0.01015}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.376 sms_tps_to_dialpeers_add

Docstring:

Creates relations between chosen dialpeer and TP(s) or relations between chosen TP and dialpeer(s) within given context(s).

Example:

- Client Request

```
{ "session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": { "dialpeers_ids": [6],  
            "weight": 0,  
            "priority": 4,  
            "contexts": [1],  
            "tp_ids": ["6", "10"] }  
}
```

- Server Response

```
{ "status": 200, "message": "OK", "time_elapsed": 0.052635}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "tp_ids": List(Int(gt=0)),
    "dialpeers_ids": List(Int(gt=0)),
    "priority": Int(gte=-100, lte=100),
    "weight": Int(gte=0, lte=100),
    "contexts": List(Int(gt=0)),
  }
}
```

Name	Default value	Optional	Type	Description
"tp_ids"	N/A	False	List(Int(gt=0))	list of termination points ids. Points should be visible.
"dialpeers_ids"	N/A	False	List(Int(gt=0))	list of dialpeers ids
"priority"	0	True	Int(gte=-100, lte=100)	relations priority, all relations have the same priority
"weight"	0	True	Int(gte=0, lte=100)	relations weight, all relations will have the same weight.
"contexts"	N/A	False	List(Int(gt=0))	required contexts ids

2.377 sms_tps_to_groups_add

Docstring:

Adds relation between given group(s) and Termination Point(s).

Function disallows to add relation with invisible TP. Given TP(s) and group(s) should have the same services.

Example:

- Client Request

```
{"session_id":"91af15315af776928e1abfa0e88b6641",  
  "data":{"contexts":[1],  
          "tp_ids":["14"],  
          "groups_ids":[9]}  
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.240366}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
    "tp_ids": List(Int(gt=0)),  
    "groups_ids": List(Int(gt=0)),  
    "contexts": List(Int()),  
  }  
}
```

Name	Default value	Optional	Type	Description
"tp_ids"	N/A	False	List(Int(gt=0))	list of TPs id, only visible points allowed
"groups_ids"	N/A	False	List(Int(gt=0))	list of groups id
"contexts"	N/A	False	List(Int())	list of contexts id to which relation should be added

2.378 sms_trace_get

Docstring:

Returns list of possible routes (combination of IP address, prefix and port) for call from chosen OP, ANI number to the given destination number via specific RADIUS (authorization server).

API sends request to the chosen RADIUS server, which returns either currently available routes if any or RADIUS error code.

Prices are being displayed in the tariff currency.

Example:

- Client Request

```
{"session_id":"1149e8a40f1ec8812f7a4349020c7259",
"data":{"limit":101,
"op_id":7,
"a_number":"123",
"radius_id":1,
"contexts":[1],
"number":"923000000000",
"offset":0,
"op_fields":["op_id","op_price_id","op_price","op_name",
"op_destination","op_country","op_currency",
"work_time"],
"fields":["tp_name","route","tp_dest","tp_country",
"tp_price","tp_capacity","tp_currency"]}}
}
```

- Server Response

```
{"status": 200,
"time_elapsed": 0.024503,
"op_data": [7, 13, 0.04, "H_in", "9230", "Pakistan Mobile Mobilink",
"USD", "238"],
"fields": ["tp_name", "route", "tp_dest", "tp_country",
"tp_price", "tp_capacity", "tp_currency"],
"message": "OK",
"data": [{"H_to", "h323/4009230000000000@198.21.162.12:1720",
"9230", "Pakistan Mobile Mobilink", "10.3500",
500, "USD"}]}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "op_id": Int(gt=0),
```

```

    "radius_id": Int(gt=0),
    "a_number": String(allow_blank=True),
    "number": String(regex=B_NUMBER_REGEXP),
    "contexts": List(Int(gt=0)),
    "op_fields": List(String()),
    "fields": List(String()),
    "limit": Int(gt=0),
    "offset": Int(gte=0),
    "mcc": String(allow_blank=True, regex="^(\\d{3})?$"),
    "mnc": String(allow_blank=True, regex="^(\\d{1,3})?$"),
}
}

```

Name	Default value	Optional	Type	Description
"op_id"	N/A	False	Int(gt=0)	id of Origination Point
"radius_id"	N/A	False	Int(gt=0)	id of radius, which will process request. What RADIUS available depends from chosen context.
"a_number"	N/A	True	String(allow_blank=True)	caller number. May contain any symbols
"number"	N/A	False	String(regex=B_NUMBER_REGEXP)	called number. May contain: digits, "#" and "*" symbols
"contexts"	N/A	False	List(Int(gt=0))	ids context, in which request will be processed
"op_fields"	default_fields_op	True	List(String())	OP information which will be displayed in response. If absent - returns fields specified in "default_fields_op" section. If empty - returns error.
"fields"	default_fields	True	List(String())	TP information which will be displayed in response. If absent - returns fields specified in "default_fields" section. If empty - returns error.
"limit"	0	True	Int(gt=0)	says how many rows will be in response
"offset"	0	True	Int(gte=0)	Offset says to skip that many rows before beginning to return rows.
"mcc"	N/A	True	String(allow_blank=True,	

			regex="^(\\d{3})?\$")	
"mnc"	N/A	True	String(allow_blank=True, regex="^(\\d{1,3})?\$")	

Default fields:

Name	Type & Comments
"#"	deprecated
"position"	Int
"tp_name"	String
"tp_country"	String
"tp_price"	numeric(16,5)
"tp_capacity"	Int
"route"	String
"tp_dest"	String
"work_time"	String - time of request processing on Radius in microseconds
"tp_currency"	String
"tp_mcc"	String
"tp_mnc"	String

"tp_dialpeer_id"	Int
"tp_dialpeer_name"	String

Default fields OP:

Name	Type & Comments
'op_id'	Int
'op_price_id'	Int
'op_price'	numeric(16,5)
'op_name'	String
'work_time'	Float - time of request processing on Radius in microseconds
'op_destination'	String
'op_country'	String
'op_currency'	String
'op_mcc'	String
'op_mnc'	String

2.379 spam_calls_get

Docstring:

Returns cdrs of not authorized calls for given time period (calls with local disconnect code 1000). It shows calls data individually or grouped by originator IP, disconnect cause code, days or hours. If chosen "type" not "individual" or "origination_ip" - function returns "op_ip" and "nas" as "0.0.0.0".

If "type" is not "disconnect_code" or "individual" function returns

"0" instead of real "local_disconnect_cause".

Example:

- Client Request

```
{"session_id": "152bab1c9d436bad4fb64435b9398e08",
  "data": {"limit": 3,
    "time_zone": "0:00",
    "end_date": "2013-12-25T23:59:59",
    "offset": 0,
    "type": "individual",
    "start_date": "2013-12-25T00:00:00",
    "fields": ["setup_time", "op_ip", "called_number_in",
      "disconnect_cause_name", "disconnect_cause",
      "local_disconnect_cause_name",
      "local_disconnect_cause", "nas", "points"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["setup_time", "op_ip", "called_number_in",
    "disconnect_cause_name", "disconnect_cause",
    "local_disconnect_cause_name",
    "local_disconnect_cause", "nas", "points"],
  "message": "OK",
  "data": [[["2013-12-25T22:30:16.417000+00:00", "194.28.165.61",
    "3333923557300265", "No circuit/channel available", 34,
    "Point is not authorized by IP or Prefix.", 1000,
    "193.161.85.79", [[4677, "Webinar_in"]]]],
    [["2013-12-25T21:12:59.918000+00:00", "194.28.165.61",
    "3333923556502898", "No circuit/channel available", 34,
    "Point is not authorized by IP or Prefix.", 1000,
    "193.161.85.79", [[4677, "Webinar_in"]]]],
    [["2013-12-25T19:24:19.788000+00:00", "194.28.165.61",
    "3333923558206171", "No circuit/channel available", 34,
    "Point is not authorized by IP or Prefix.", 1000,
    "193.161.85.79", [[4677, "Webinar_in"]]]]],
  "time_elapsed": 0.335644}
```

Request structure and fields description:

```
{'data': {'end_date': 'String',
  'fields': ["Enum('nas', 'disconnect_cause', 'points', 'disconnect_cause_name',
    'op_ip', 'local_disconnect_cause', 'calls_count', 'local_disconnect_cause_name', 'called_number_in', 'setup_time')"],
  'hosts': ['String'],
  'incoming_ip': 'String',
  'limit': 'Integer',
  'offset': 'Integer',
  'op_destination': 'String',
  'session_id': 'String',
  'start_date': 'String',
  'termcause': 'Integer',
  'time_zone': 'String',
  'type': 'String'},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	Not in use. By defaults from user's session.
start_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})(T)(\d{2}):(\d{2}):(\d{2})\$")	start date and time for data gathering. Report returns calls which setup_time not less than given.
end_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})(T)(\d{2}):(\d{2}):(\d{2})\$")	end date and time for data gathering. Report returns calls which setup_time not more than given.
op_destination	N/A	True	String(allow_blank: False, regex: "^%?[0-9,#+]*\$")	called number or first digits of called number
incoming_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] [12]\d 3[0-2]))?(0\.{3}0(0)?)\$")	filter by IP of call originator
termcause	N/A	True	Integer(gt: 0)	filter by disconnect cause code
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?\d{1,2}):(\d{2})\$")	"start_date" and "end_date" timezone

type	N/A	False	String(allow_blank: False, regex: "(individual) (origination_ip) (disconnect_code) (hour) (day)")	method of grouping: "individual", "origination_ip", "disconnect_code", "day", "hour".
hosts	N/A	True	List([String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])?)/([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?)\$"), min_length: 0)	filter by RTP host IP address. A few addresses need separate by comma.
fields	['nas', 'disconnect_cause', 'points', 'disconnect_cause_name', 'op_ip', 'local_disconnect_cause', 'calls_count', 'local_disconnect_cause_name', 'called_number_in', 'setup_time']	True	List([Enum('nas', 'disconnect_cause', 'points', 'disconnect_cause_name', 'op_ip', 'local_disconnect_cause', 'calls_count', 'local_disconnect_cause_name', 'called_number_in', 'setup_time')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
nas	String (RTP IP address)
disconnect_cause	Int
points	List(Lists(point id, point name))
disconnect_cause_name	String
op_ip	String

local_disconnect_cause	Int
calls_count	Int
local_disconnect_cause_name	String
called_number_in	String
setup_time	String

2.380 start_page_call_charts_get

Docstring:

Returns list of points for calls in 'data' and for asr in 'asr'. null is used if data is absent.

Example:

- Client Request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {"interval": "today"}}
}
```

Request structure and fields description:

```
{'data': {'interval': "Enum('yesterday', 'today')", 'session_id': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.

interval	N/A	False	Enum('yesterday', 'today')	interval to built call chart diagram
----------	-----	-------	----------------------------	--------------------------------------

2.381 start_page_credit_limit_report_get

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency_name', 'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent')"],
            'get_cached_result': 'Bool',
            'interval': "Enum('yesterday', 'today')",
            'limit': 'Integer',
            'offset': 'Integer',
            'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.
get_cached_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
fields	['currency_name',	True	List([Enum('currency_name',	

	'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent']		'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent')], min_length: 0)	
--	--	--	---	--

Default fields:

Name	Type & Comments
currency_name	String(3) - name_short
balance_in_default_currency	Numeric(23,2)
credit_limit	Numeric(23,2)
closed	Boolean
client_id	Int
client_name	String
balance	Numeric(23,2)
percent	Numeric(23,2)

2.382 start_page_credit_limit_report_sms_get

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency_name', 'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent')"],
  'get_cached_result': 'Bool',
  'interval': "Enum('yesterday', 'today')",
  'limit': 'Integer',
  'offset': 'Integer',
  'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.
get_cached_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
fields	['currency_name', 'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent']	True	List([Enum('currency_name', 'balance_in_default_currency', 'credit_limit', 'closed', 'client_id', 'client_name', 'balance', 'percent')], min_length: 0)	

Default fields:

Name	Type & Comments
currency_name	String(3) - name_short
balance_in_default_currency	Numeric(23,2)
credit_limit	Numeric(23,2)

closed	Boolean
client_id	Int
client_name	String
balance	Numeric(23,2)
percent	Numeric(23,2)

2.383 start_page_get_clients_close_to_credit_limit

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('balance_in_default_currency', 'credit_limit', 'client_name',
'percent', 'currency_name', 'client_id', 'balance')"],
'get_cached_result': 'Bool',
'interval': "Enum('yesterday', 'today')",
'limit': 'Integer',
'offset': 'Integer',
'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram

get_cache d_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
fields	['balance_in_default_curren cy', 'credit_limit', 'client_name', 'percent', 'currency_name', 'client_id', 'balance']	True	List([Enum('balance_in_default_ currency', 'credit_limit', 'client_name', 'percent', 'currency_name', 'client_id', 'balance')], min_length: 0)	

Default fields:

Name	Type & Comments
balance_in_default_currenc y	Numeric(23,2)
credit_limit	Numeric(23,2)
client_name	String
percent	Numeric(23,2)
currency_name	String
client_id	Int
balance	Numeric(23,2)

2.384 start_page_get_closed_companies

Docstring:

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id')"],
  'get_cached_result': 'Bool',
```

```

        'interval': "Enum('yesterday', 'today')",
        'limit': 'Integer',
        'offset': 'Integer',
        'session_id': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
get_cached_result	False	False	Bool	Returns cached response for current user. Ignores limit and offset.
fields	['currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id']	True	List([Enum('currency_name', 'balance_in_default_currency', 'client_name', 'balance', 'client_id')], min_length: 0)	
interval	N/A	False	Enum('yesterday', 'today')	interval to built credit limit diagram
offset	N/A	True	Integer(gt: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
currency_name	String

balance_in_default_currency	Numeric(23,2)
client_name	String
balance	Numeric(23,2)
client_id	Int

2.385 start_page_sms_activity_get

Docstring:

Returns list of points for calls in 'data' and for asr in 'asr'. null is used if data is absent.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "interval": "today"
  }
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.386 start_page_top_countries_get

Docstring:

Returns list of 10 Countries with most calls duration or margin (delta_price).

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "acd",
      "delta_price",
      "duration",
      "country"
    ],
    "criterion": "duration",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{
  'data': {
    'criterion': "Enum('duration', 'delta_price')",
    'fields': ["Enum('asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum', 'duration')"],
    'interval': "Enum('yesterday', 'today')",
    'session_id': 'String',
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum', 'duration']	True	List([Enum('asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum', 'duration')], min_length: 0)	
criterion	N/A	False	Enum('duration', 'delta_price')	Criterion to order countries
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
interval	N/A	False	Enum('yesterday', 'today')	interval to built top countries diagram

Default fields:

Name	Type & Comments
asr	String
acd	Int
country	String
delta_price	Numeric(16,2) - difference between op_sum and tp_sum
tp_sum	Numeric(20,2)
op_sum	Numeric(20,2)
duration	Int - calls duration in minutes

2.387 start_page_top_countries_sms_get

Docstring:

Returns list of 10 Countries with most SMS attempts or margin (delta_price).

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "delta_price",
      "attempts",
      "country"
    ],
    "criterion": "attempts",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{'data': {'criterion': "Enum('attempts', 'delta_price')",
          'fields': ["Enum('asr', 'country', 'op_sum', 'attempts', 'tp_sum', 'delta_price')"],
          'interval': "Enum('yesterday', 'today')",
          'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
criterion	N/A	False	Enum('attempts', 'delta_price')	Criterion to order countries
fields	['asr', 'country', 'op_sum', 'attempts', 'tp_sum', 'delta_price']	True	List([Enum('asr', 'country', 'op_sum', 'attempts', 'tp_sum', 'delta_price')], min_length: 0)	
interval	N/A	False	Enum('yesterday', 'today')	interval to built top countries diagram
session_id	N/A	True	String(allow_blank: False)	By default get from current session.

Default fields:

Name	Type & Comments
asr	String
country	String
op_sum	Numeric(20,2)
attempts	Int
tp_sum	Numeric(20,2)
delta_price	Numeric(16,2) - difference between op_sum and tp_sum

2.388 start_page_top_offers_get

Docstring:

Returns list of 10 Country Codes with most calls duration and ASR.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "acd",
      "duration",
      "country"
    ],
    "criterion": "duration",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('asr', 'acd', 'country', 'target_price', 'attempts', 'duration')"],
          'interval': "Enum('yesterday', 'today')",
          'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
interval	N/A	False	Enum('yesterday', 'today')	interval to built top offers diagram
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
fields	['asr', 'acd', 'country', 'target_price', 'attempts', 'duration']	True	List([Enum('asr', 'acd', 'country', 'target_price', 'attempts', 'duration')], min_length: 0)	

Default fields:

Name	Type & Comments
asr	String

acd	Int
country	String
target_price	Numeric(16,2)
attempts	Int
duration	Int - calls duration in minutes

2.389 start_page_top_offers_sms_get

Docstring:

Returns list of 10 Country Codes with most SMS attempts and ASR.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "target_price",
      "attempts",
      "country"
    ],
    "criterion": "attempts",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('asr', 'attempts', 'target_price', 'country')"],
  'interval': "Enum('yesterday', 'today')",
  'session_id': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
fields	['asr', 'attempts', 'target_price', 'country']	True	List([Enum('asr', 'attempts', 'target_price', 'country')], min_length: 0)	
interval	N/A	False	Enum('yesterday', 'today')	interval to built top offers diagram

Default fields:

Name	Type & Comments
asr	String
attempts	Int
target_price	Nullable Numeric(16,2)
country	String

2.390 start_page_top_targets_get

Docstring:

Returns list of 10 Country Codes with most calls duration and ASR.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "acd",
      "duration",
      "country"
    ],
    "criterion": "duration",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('asr', 'acd', 'country', 'target_price', 'attempts', 'duration')"],  
          'interval': "Enum('yesterday', 'today')",  
          'session_id': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
interval	N/A	False	Enum('yesterday', 'today')	interval to built top targets diagram
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
fields	['asr', 'acd', 'country', 'target_price', 'attempts', 'duration']	True	List([Enum('asr', 'acd', 'country', 'target_price', 'attempts', 'duration')], min_length: 0)	

Default fields:

Name	Type & Comments
asr	String
acd	Int
country	String
target_price	Numeric(16,2)
attempts	Int
duration	Int - calls duration in

	minutes
--	---------

2.391 start_page_top_targets_sms_get

Docstring:

Returns list of 10 Country Codes with most SMS attempts and ASR.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "fields": [
      "asr",
      "attempts",
      "attempts",
      "country"
    ],
    "criterion": "attempts",
    "interval": "yesterday"
  }
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': ["Enum('asr', 'attempts', 'target_price', 'country')"],
    'interval': "Enum('yesterday', 'today')",
    'session_id': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
interval	N/A	False	Enum('yesterday', 'today')	interval to built top targets diagram
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
fields	['asr', 'attempts', 'target_price', 'country']	True	List([Enum('asr', 'attempts', 'target_price', 'country')]),	

			min_length: 0)	
--	--	--	----------------	--

Default fields:

Name	Type & Comments
asr	String
attempts	Int
target_price	Numeric(16,2)
country	String

2.392 stat_get

Docstring:

Returns summary statistic of calls by chosen RTP hosts or by all enabled ones if key "show_summary_all" is absent or False. Otherwise calls function _show_summary_all.

Function returns fields as keys values(see example below) for "show_summary_all" = False.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {"fields": ["attempts", "attempts_total", "calls", "time"]}
}
```

- Server Response

```
{
  "status": 200,
  "attempts": 33,
  "calls": 6,
  "time": 1390218756.252459,
  "message": "OK",
  "time_elapsed": 0.009557,
  "attempts_total": 3260042
}
```


`_show_summary_all` ([data_validation](#)) ([default_fields](#))

Returns calls statistic for each enabled RTP host (see example).
Response contains keys(RTP IP) and data received from RTP.

In case attempt to connect to RTP was unsuccessful - response from RTP has type str and contains error message.

Example:

- Client Request

```
{"session_id": "6726cebe6ca27ef6a8da30867d6050a8",  
  "data": {"show_summary_all": true}  
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "time_elapsed": 0.095671,  
  "192.108.128.4": {  
    'Max active calls': 393,  
    'Answered calls': 144,  
    'Answered AVG 5m': 146,  
    'Attempts 5m': 1795,  
    'Attempts AVG 5m': 1795,  
    'Normal calls': 277282,  
    'Received calls': 2031995,  
    'Active AVG 5m': 245,  
    'Active calls': 243,  
    'date_from': u'2012-12-20 09:36:56',  
    'date_to': u'2012-12-20 09:41:56',  
    'received_calls_actual': 2031995}  
}
```

`_stat_total_get` ([data_validation](#)) ([default_fields](#))

Request structure and fields description:

`_show_summary_all`

```
{'data': {'fields': ["Enum('Max active calls', 'Answered AVG 5m', 'date_to', 'Normal  
calls', 'Active calls', 'Answered calls', 'date_from', 'Attempts AVG 5m', 'Received cal  
ls', 'Active AVG 5m', 'Attempts 5m')"],  
  'hosts': ['String']},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
hosts	N/A	True	List([String(allow_blank: False)], min_length: 0)	List of hosts IP addresses. If absent returns calls statistic summary for all enabled RTP hosts
fields	['Max active calls', 'Answered AVG 5m', 'date_to', 'Normal calls', 'Active calls', 'Answered calls', 'date_from', 'Attempts AVG 5m', 'Received calls', 'Active AVG 5m', 'Attempts 5m']	True	List([Enum('Max active calls', 'Answered AVG 5m', 'date_to', 'Normal calls', 'Active calls', 'Answered calls', 'date_from', 'Attempts AVG 5m', 'Received calls', 'Active AVG 5m', 'Attempts 5m')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

_stat_total_get

```
{'data': {'fields': ["Enum('total_minutes', 'attempts', 'calls', 'total_sms', 'margin_by_currencies', 'sms_margin', 'time', 'margin', 'sms_margin_by_currencies', 'attempts_total')"],
          'hosts': ['String']},
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
hosts	N/A	True	List([String(allow_blank: False)], min_length: 0)	List of hosts IP addresses. If absent returns calls statistic summary for all enabled RTP hosts
fields	['total_minutes', 'attempts', 'calls', 'total_sms', 'margin_by_currencies', 'sms_margin', 'time', 'margin', 'sms_margin_by_currencies', 'attempts_total']	True	List([Enum('total_minutes', 'attempts', 'calls', 'total_sms', 'margin_by_currencies', 'sms_margin', 'time', 'margin', 'sms_margin_by_currencies', 'attempts_total')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

_show_summary_all

Name	Type & Comments
Max active calls	Int - max quantity simultaneous active_calls from switch start
Answered AVG 5m	Int - answered calls avg for last 5 min slice
date_to	String - start date and time of 5 min avg statistic slice
Normal calls	Int - amount of answered calls (active) after switch start
Active calls	Int - active calls (answered and not) at the moment
Answered calls	Int - quantity of answered calls at the moment
date_from	String - end date and time of 5 min avg statistic slice
Attempts AVG 5m	Int - approximate average amount of attempts during last 5 minutes slice of stat
Received calls	Int - amount of received calls (attempts) after switch start
Active AVG 5m	Int - active calls avg for last 5 min slice
Attempts 5m	Int - attempts 5 min

_stat_total_get

Name	Type & Comments
total_minutes	String - op_calc_duration for today
attempts	Int - last summary CPM for all RTPs
calls	Int - average number of calls(including calls with zero duration) for last 5 minutes for all RTPs
total_sms	String - count of successfully sent SMS for today

margin_by_currencies	Map - floored daily margin in all currencies
sms_margin	Long - floored daily SMS margin. DEPRECATED. Use 'sms_margin_by_currencies' instead.
time	float - number of seconds from the start of unix Epoch
margin	Long - floored daily margin. DEPRECATED. Use 'margin_by_currencies' instead.
sms_margin_by_currencies	Map - floored daily SMS margin in all currencies
attempts_total	Int - summary number of attempts for all RTPs

2.393 sum_active_calls_points_get

Docstring:

Returns data to draw active calls charts for all hosts for last day, week, month.

For each "period" "data" list contains 288 values which represents average active calls statistic and "connected_calls" list contains 288 values which represents average answered calls statistic for the interval.

"start_time" field contains start time of the period as POSIX timestamp (int).

If interval "day" - returns data about 2 last days (10 minutes stat resolution) and "ASR" (within key with the same name) stat for last 2 days (48 values) with 1 hour resolution from aggregation without last hour.

Example:

- Client Request

```
{
  "session_id": "64a118830aea3fad72d0df2fcc2a183a",
  "data": {
    "period": "day"
  }
}
```

- Server Response:

```
{
  "status": 200,
  "asr": [
    15.300546448087431, 6.228373702422146, 8.75912408759124, 7.42659758
    2037997, 4.732824427480916,
    ...
  ]
}
```

```

        null
    ],
    "time_elapsed": 0.226724,
    "message": "OK",
    "data": [
        1.49666666667, 1.50166666667, 0.00333333333333, 1.99666666667, 2.0, 0.0
066666666667, 0.998333333333,
        ...
        0.50166666667, 0.00333333333333, 0.498333333333, 0.0, 0.49666666667,
1.0
    ],
    "connected_calls" : [
        3.43211111111, 5.79955555556, 4.13322222222, 5.56133333333, 0.0,
        ...
        2.43444444444, 1.86633333333, 4.63333333333, 5.09944444444, 5.2668888888
9
    ],
    "start_time": 1459345800
}

```

Request structure and fields description:

```

{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')"},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.

2.394 sum_cpm_points_get

Docstring:

Returns data to draw summary attempts charts (by all hosts of the system) for the given period (288 values for each period). "start_time" field contains start time of the period as POSIX timestamp (int).

"day" period contains data 2 last days.

Example:

- Client Request

```

{
  "session_id": "1d81ff9b02a2ea9f55df0cf4327f3124",
  "data": {
    "period": "month"
  }
}

{
  "status": 200,
  "message": "OK",
  "data": [
    426.819576966, 219.767931342, 9.5516722038, 1.85424038046, 1.039668
81114, 13.9965005538,
    65.411186712, 110.981969436, 70.978141674, 70.978141674, 183.970242
378, 41.2337
    664822, 7.902392026799999, 1.8740192691000002, 7.5446593086, 33.712
0295802, 52.8460803246,
    43.88579199, 41.667179487, 113.270081532, 206.01805671, 48.20080995
9, 13.0013305092,
    7.531522794600001, 33.937035585, 80.011425102, 94.338819342, 77.992
631376, 83.337168624, ...
  ],
  "time_elapsed": 0.226724,
  "start_time": 1459345800
}

```

Request structure and fields description:

```

{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')"},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.

2.395 sum_sms_activity_points_get

Docstring:

Returns data to draw sms activity charts for all hosts for last day, week, month.

"start_time" field contains start time of the period as POSIX timestamp (int).

If interval "day" - returns data about 2 last days and "ASR" (within key with the same name) stat for last 2 days (48 values) with 1 hour resolution from aggregation without last hour.

The "data" field contains a list of 1440 samples, data[i], i=0..1439.

Here i - sequence number.

The data[i] item denotes the instantaneous SMS rate (expressed in messages per second) at the moment of time t[i]. This moment of time is computed as follows:

$t[i] = \text{start_time} + \text{delta} * (i+1)$,
where delta = 120 seconds.

The "asr" field contains a list of 48 samples, asr[i], i=0..47.

Each asr[i] item denotes the aggregated SMS ASR (Average Success Rate) for the corresponding time interval. Each interval has the length of 1 hour (3600 seconds).

The boundaries (tl[i] and tr[i]) of the asr interval can be computed as follows

:

$tl[i] = \text{start_asr_time} + d_asr * i$
 $tr[i] = \text{start_asr_time} + d_asr * (i+1)$

d_asr = 3600 seconds

Here:

start_asr_time is the maximal hour boundary which is less than or equal to the start_time.

start_asr_time can be computed as the start_time with the minutes and seconds zeroed.

Example:

- Client Request

```
{
  "session_id": "64a118830aea3fad72d0df2fcc2a183a",
  "data": {
    "period": "day"
  }
}
```

- Server Response:

```
{
  "status": 200,
  "asr": [
    15.300546448087431, 6.228373702422146, 8.75912408759124, 7.42659758
    2037997, 4.732824427480916,
    ...
    null
  ],
  "time_elapsed": 0.226724,
  "message": "OK",
  "data": [
    1.49666666667, 1.50166666667, 0.00333333333333, 1.99666666667, 2.0, 0.0
    066666666667, 0.998333333333,
    ...
    0.50166666667, 0.00333333333333, 0.498333333333, 0.0, 0.49666666667,
    1.0
  ],
  "start_time": 1459345800
}
```

Request structure and fields description:

```
{'data': {'period': "Enum('week', 'for_diagram', 'day', 'month')",  
          'rate': "Enum('per_sec', 'per_min')"},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
rate	N/A	False	Enum('per_sec', 'per_min')	Sms interest rate.
period	N/A	False	Enum('week', 'for_diagram', 'day', 'month')	Period for data gathering.

2.396 swap_deal_add

Docstring:

Adds new swap deal between chosen Customer(OP) and Supplier(TP), given destinations and time period.

Example:

- Client Request

```
{  
  "session_id":"b5a59f2f15aa82957fcf4bddc4a57362",  
  "data":{  
    "flag_minutes_out_total":false,  
    "op":[397],  
    "flag_minutes_out_day":true,  
    "flag_profit_by_total":true,  
    "flag_minutes_in_total":false,  
    "tp":[22],  
    "op_dest":"3",  
    "op_mins":1000,  
    "tp_mins":1000,  
    "tp_dest":"3",  
    "flag_profit_by_day":true,  
    "flag_minutes_in_day":true,  
    "name":"deal_100",  
    "start_time":"2014-11-01T00:00:00",  
    "swap_period":1,  
    "client_id":1  
  }  
}
```



```
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 57,
  "time_elapsed": 0.138243
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "name": Name(),
    "client_id": Int(gt=0),
    "revenue": Float(gte=0, lte=1e8),
    "start_time": String(regex=DATE_REGEX) ,
    "op": List(Int(gt=0), min_length=1),
    "op_dest": String(allow_blank=True,regex="^(\\d+(,|\\s)?)*$") ,
    "op_mins": Int(gte=0),
    "tp": List(Int(gt=0), min_length=1),
    "tp_dest": String(allow_blank=True,regex="^(\\d+(,|\\s)?)*$") ,
    "tp_mins": Int(gte=0),
    "swap_period": Int(gte=1, lte=12),
    "flag_profit_by_day": Bool(),
    "flag_profit_by_total": Bool(),
    "flag_minutes_day": Bool(),
    "flag_minutes_total": Bool(),
    "flag_revenue_day": Bool(),
    "flag_revenue_total": Bool(),
  }
}
```

Name	Default value	Optional	Type	Description
"name"	N/A	False	Name()	
"client_id"	N/A	False	Int(gt=0)	
"revenue"	.0	False	Float(gte=0, lte=1e8)	
"start_time"	N/A	False	String(regex=DATE_REGEX)	
"op"	N/A	False	List(Int(gt=0),	

			min_length=1)	
"op_dest"	N/A	False	String(allow_blank=True,regex="^(\\d+(, \\s)?)*\$")	
"op_mins"	N/A	False	Int(gte=0)	
"tp"	N/A	False	List(Int(gt=0),min_length=1)	
"tp_dest"	N/A	False	String(allow_blank=True,regex="^(\\d+(, \\s)?)*\$")	
"tp_mins"	N/A	False	Int(gte=0)	
"swap_period"	N/A	False	Int(gte=1, lte=12)	Deal length in months
"flag_profit_by_day"	False	False	Bool()	
"flag_profit_by_total"	False	False	Bool()	
"flag_minutes_day"	False	False	Bool()	
"flag_minutes_total"	False	False	Bool()	
"flag_revenue_day"	False	False	Bool()	
"flag_revenue_total"	False	False	Bool()	

2.397 swap_deal_del

Docstring:

Delete deal with given id.

Example:

- Client Request

```
{
  "session_id": "63db616315b9e7d6515e9a24aa8f0e2c",
  "data": {
    "id": 2
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 2,
  "time_elapsed": 0.051203
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
  }
}
```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	

2.398 swap_deal_fin_summary_get

Docstring:

Returns financial report for chosen time period and swap deal(s) with statistical information (common duration of calls, duration with considering prices tarification parameters, asr, acd, attempts statistic, cost in/out, profit, delta price values combining by destination codes, points or companies).

Function returns list of companies("clients" param) for which financial data is available. If there are no clients available then returns empty data and fields list.

Example:

- Client Request

```
{
  "session_id":"63db616315b9e7d6515e9a24aa8f0e2c",
  "data":{
    "flag_minutes_out_total":false,
    "op":[45],
    "flag_minutes_out_day":false,
    "flag_profit_by_total":false,
    "flag_minutes_in_total":false,
    "tp":[22],
    "op_dest":"111",
    "op_mins":1111,
    "tp_mins":1110,
    "id":32,
    "tp_dest":"100",
    "flag_profit_by_day":false,
    "flag_minutes_in_day":false,
    "name":"alex_dddd",
    "swap_period":1,
    "client_id":1
  }
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.247076,
  "agg_last_time": "2014-11-07 13:00:00",
  "fields": [
    "company", "tot", "type", "point_name", "destination", "country", "duration", "calc_duration",
    "op_price", "tp_price", "attempts", "sa", "asr", "acd", "pdd", "sc", "op_sum", "tp_sum",
```

```

        "delta_price", "delta_profit"
    ],
    "message": "OK",
    "data": [
        ["H_Company_166_new", "Total", "outbound", "", "", "", 14471, 14471, null, null, 699, 59, "8.4
4%",
        245, "1.34", "23.73%", "20.33577", "2.41184", "17.92", "743.16%"],
        ["H_Company_166_new", "Total", "inbound", "", "", "", 144885, 144885, null, null, 134172, 89
7,
        "0.67%", 161, "2.93", "35.12%", "241.47478", "40.19952", "201.28", "500.69%"]
    ]
}

```

Request structure and fields description:

Request:

```

{
    "session_id": String(),
    "data": {
        'session_id': String(),
        "context_id": Int(gt=0),
        "start_date": String() ,
        "end_date": String() ,
        "swap_deals": List(Int(gt=0), min_length=1),
        "direction": Int(gte=0),
        "show_traffic_from_other_company": Bool(),
        "time_zone": String(regex="^[+-]?d{1,2}):(\d{2})$"),
        'currency_id': Int(gt=0),
        "type": Enum(1, 5),
        "direction": Int(gte=0),
        "fields": List(String()),
    }
}

```

Name	Default value	Optional	Type	Description
'session_id'	session_info['id']	True	String()	
"context_id"	N/A	True	Int(gt=0)	
"start_date"	N/A	False	String()	
"end_date"	N/A	False	String()	
"swap_deals"	N/A	False	List(Int(gt=0), min_length=1)	filter by swap deal id
"direction"	0	False	Int(gte=0)	

"show_traffic_from_other_company"	False	False	Bool()	True if show traffic from other companies
"time_zone"	'+00:00'	True	String(regex="^[+-]?\\d{1,2}):\\d{2}\$")	requires if company works in timezone different from GMT+0
'currency_id'	session_info['currency_id']	True	Int(gt=0)	display prices and cost in chosen currency
"type"	N/A	False	Enum(1, 5)	report type id. 1 - Individual report for each destination, price, point and company, 5 - Total by company.
"direction"	0	True	Int(gte=0)	display only incoming(by OPs, 0) or outgoing(TPs, 1) traffic, if doesn't exist - both directions
"fields"	default_fields	True	List(String())	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
'company'	String
'total'	String
'direction'	Int (1 - inbound, 2 - outbound)

'point_name'	String
'destination'	String
'country'	String
'duration'	Int
'calc_duration'	Int
'op_price'	String
'tp_price'	String
'attempts'	Int
'sa'	Int - quantity of successful attempts
'asr'	String
'acd'	Int
'sc'	String
'op_sum'	String
'tp_sum'	String
'delta_price'	String
'delta_profit'	String
'pdd'	numeric - post dial delay in seconds
'is_closed'	if True - dest is closed
'deal_names'	String
'origin_country'	String

2.399 swap_deal_get

Docstring:

Returns data about all swap deals or deal with chosen id.

Example:

- Client Request

```
{
  "session_id": "63db616315b9e7d6515e9a24aa8f0e2c",
  "data": {
    "id": 32,
    "fields": [
      "id", "name", "client_id", "op", "op_dest", "op_mins", "tp", "tp_dest", "tp_mins", "start_time",
      "swap_period", "end_time", "day_of_swap", "current_swap_traffic", "traffic_forecast_in",
      "current_traffic_in", "traffic_forecast_out", "current_traffic_out", "flag_profit_by_day",
      "flag_profit_by_total", "flag_minutes_in_day", "flag_minutes_out_day", "flag_minutes_in_total",
      "flag_minutes_out_total"
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": [
    "id", "name", "client_id", "op", "op_dest", "op_mins", "tp", "tp_dest", "tp_mins", "start_time",
    "swap_period", "end_time", "day_of_swap", "current_swap_traffic", "traffic_forecast_in",
    "current_traffic_in", "traffic_forecast_out", "current_traffic_out", "flag_profit_by_day",
    "flag_profit_by_total", "flag_minutes_in_day", "flag_minutes_out_day", "flag_minutes_in_total",
    "flag_minutes_out_total"
  ],
  "message": "OK",
  "data": [
    [32, "alex_dddd", 1, [45], ["111"], 1111, [22], ["1111"], 1111, "2014-06-25T00:00:00+00:00", 1,
    "2014-07-25T23:59:59+00:00", 30, 0.0, 1111, 0, 1111, 0, false, false, false, false, false, false]
  ],
  "time_elapsed": 0.03936
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
    "id": Int(gt=0),
    "client_name": String(),
    "dest": String(),
    "name": String(),
  }
}
```



```

    "fields": List(String()) ,
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	True	Int(gt=0)	filter by id
"client_name"	N/A	True	String()	filter by client name
"dest"	N/A	True	String()	filter by swap deal destination
"name"	N/A	True	String()	filter by swap deal name
"fields"	default_fields	False	List(String())	

Default fields:

Name	Type & Comments
"id"	Integer
"name"	String
"client_id"	Integer
"revenue"	Numeric
"op"	List of Integers
"op_dest"	List of String
"op_mins"	Integer

"tp"	List of Integers
"tp_dest"	List of String
"tp_mins"	Integer
"start_time"	String
"swap_period"	Integer
"end_time"	String
"day_of_swap"	Integer
"current_swap_traffic"	Numeric
"traffic_forecast_in"	Integer
"current_traffic_in"	Integer
"traffic_forecast_out"	Integer
"current_traffic_out"	Integer
"current_revenue_in"	Float
"revenue_forecast_in"	Float
"current_revenue_out"	Float
"revenue_forecast_out"	Float
"flag_profit_by_day"	Boolean
"flag_profit_by_total"	Boolean
"flag_profit_by_day"	Boolean
"flag_minutes_day"	Boolean
"flag_minutes_total"	Boolean
"flag_revenue_day"	Boolean

"flag_revenue_total"	Boolean

2.400 swap_deal_set

Docstring:

Change deal with given id.

Example:

- Client Request

```
{
  "session_id":"63db616315b9e7d6515e9a24aa8f0e2c",
  "data":{
    "flag_minutes_out_total":false,
    "op":[45],
    "flag_minutes_out_day":false,
    "flag_profit_by_total":false,
    "flag_minutes_in_total":false,
    "tp":[22],
    "op_dest":"111",
    "op_mins":1111,
    "tp_mins":1110,
    "id":32,
    "tp_dest":"100",
    "flag_profit_by_day":false,
    "flag_minutes_in_day":false,
    "name":"alex_dddd",
    "swap_period":1,
    "client_id":1
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.073294
}
```

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
```

```

    "id": Int(gt=0),
    "name": Name(),
    "client_id": Int(gt=0),
    "revenue": Float(gte=0, lte=1e8),
    "op": List(Int(gte=0), min_length=1),
    "op_dest": String(allow_blank=True,regex="^(\\d+(,|\\s)?)*$") ,
    "op_mins": Int(gte=0),
    "tp": List(Int(gte=0), min_length=1),
    "tp_dest": String(allow_blank=True,regex="^(\\d+(,|\\s)?)*$") ,
    "tp_mins": Int(gte=0),
    "swap_period": Int(gte=1, lte=12),
    "flag_profit_by_day": Bool(),
    "flag_profit_by_total": Bool(),
    "flag_minutes_day": Bool(),
    "flag_minutes_total": Bool(),
    "flag_revenue_day": Bool(),
    "flag_revenue_total": Bool(),
  }
}

```

Name	Default value	Optional	Type	Description
"id"	N/A	False	Int(gt=0)	
"name"	N/A	True	Name()	
"client_id"	N/A	True	Int(gt=0)	
"revenue"	N/A	True	Float(gte=0, lte=1e8)	
"op"	N/A	True	List(Int(gte=0), min_length=1)	
"op_dest"	N/A	True	String(allow_blank=True, regex="^(\\d+(, \\s)?)*\$")	
"op_mins"	N/A	True	Int(gte=0)	
"tp"	N/A	True	List(Int(gte=0), min_length=1)	
"tp_dest"	N/A	True	String(allow_blank=True, regex="^(\\d+(, \\s)?)*\$")	
"tp_mins"	N/A	True	Int(gte=0)	

"swap_period"	N/A	True	Int(gte=1, lte=12)	Deal length in months
"flag_profit_by_day"	N/A	True	Bool()	
"flag_profit_by_total"	N/A	True	Bool()	
"flag_minutes_day"	N/A	True	Bool()	
"flag_minutes_total"	N/A	True	Bool()	
"flag_revenue_day"	N/A	True	Bool()	
"flag_revenue_total"	N/A	True	Bool()	

2.401 system_info_get

Docstring:

Get information about hosts listed within parameter 'hosts' via SNMP: load of CPU, memory and hdd, information about ethernet interfaces and system info.

If 'hosts' is absent in query returns information about enabled RTP host.

If no enabled RTP host then returns error: "No enabled hosts found."

Example:

- Client Request

```
{
  "session_id": "121dbdf08c0570f3b91f1072df5b6506",
  "data": {}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["host", "system_info", "cpu_info", "memory_info",
    "disk_info", "ethernet_info"],
  "message": "OK",
  "data": [
    ["193.141.80.12", "Uptime: 57 days, 17:08:47\nDate: Thu Jan 16 09:07:09 2014",
      "Load average: 0.15\nUser CPU time(%): 0\nSystem CPU time(%): 0\nIdle CPU time(%): 99",
      "Total RAM(kB): 2052340\nTotal RAM available(kB): 6838616",
      "Device: /dev/sda1\nTotal size(kB): 474802304\nAvailable space(kB): 437675488\nUsed space(kB): 13008184\nSpace used on disk(%): 3",
      "lo: exValue='7f000001/exValue='ff000000\n eth0: exValue='c1a1554f/exValue='ffffff00\n Default GW: 193.141.80.1"]],
    "time_elapsed": 6.115931
  ]
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': ["Enum('disk_info', 'system_info', 'cpu_info', 'host', 'memory_info', 'ethernet_info')"],
    'hosts': ['String']},
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
hosts	N/A	True	List([String(allow_blank: False)], min_length: 0)	list that consists of IP addresses of RTP hosts.
fields	['disk_info', 'system_info', 'cpu_info', 'host', 'memory_info', 'ethernet_info']	True	List([Enum('disk_info', 'system_info', 'cpu_info', 'host', 'memory_info', 'ethernet_info')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
------	-----------------

disk_info	String
system_info	String
cpu_info	String
host	String
memory_info	String
ethernet_info	String

2.402 target_by_prices_get

Docstring:

Returns list of tariffs (there is possible show only OP, only TP or both OP and TP prices) with allowed prices to the given destination at specific time.

Report displays price values in currency of their tariff and country names from list of countries available in database.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "limit": 101,
    "common_codes": false,
    "tariff_type": 1,
    "sort_by": [{"price": "asc"}],
    "date": "2013-10-28T00:00:00",
    "offset": 0,
    "fields": [
      "tariff_id",
      "tariff_name",
      "price",
      "destination",
      "country"
    ],
    "destination": "380"
  }
}
```

Request structure and fields description:

```
{
  'data': {
    'actual': 'Bool',
    'common_codes': 'Bool',
    'date': 'String',
    'destination': 'String',
    'fields': [
      'Enum('tariff_name', 'country', 'price', 'destination', 'price_def',
      'default_currency', 'currency_name', 'tariff_id')'
    ]
  }
}
```

```

'limit': 'Integer',
'offset': 'Integer',
'price_max': 'Float',
'price_min': 'Float',
'session_id': 'String',
'sort_by': [{"Enum('tariff_name', 'country', 'price', 'destination', 'price_d
efault_currency', 'currency_name', 'tariff_id')": "Enum('asc', 'desc', '')"}],
'tariff_type': 'Integer'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	by default session id of current user
destination	N/A	False	String(allow_blank: False)	destination code
tariff_type	0	True	Integer(gte: 0, lte: 2)	1- search prices for given destination in OP tariffs , 2- in TP tariffs , 0 - both
price_min	N/A	True	Float()	Filter by price value. Returns only rows with prices more than given
common_codes	N/A	True	Bool	If "true" also returns prices for subcodes of given "destination"
price_max	N/A	True	Float()	Filter by price value. Returns only rows with prices less than given
fields	['tariff_name', 'country', 'price', 'destination', 'price_default_currency', 'currency_name', 'tariff_id']	True	List([Enum('tariff_name', 'country', 'price', 'destination', 'price_default_currency', 'currency_name', 'tariff_id')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response

actual	True	True	Bool	If "true" returns only rows with actual(prices that satisfy 3 conditions: date_end more than or equal current, date_start less than or equal date_end and enabled is equal "true") prices. Else - returns disabled prices too.
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
sort_by	[{'tariff_name': 'desc'}]	True	List([{'tariff_name', 'country', 'price', 'destination', 'price_default_currency', 'currency_name', 'tariff_id'): Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
date	N/A	False	String(allow_blank: False)	Shows prices available in the specific date and time

Default fields:

Name	Type & Comments
tariff_name	String
country	String
price	String
destination	String
price_default_currency	numeric(16,5)
currency_name	String(3)
tariff_id	Int

2.403 tariff_add

Docstring:

Creates new OP or TP tariff.

Example:

- Client Request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {"currency_id": 76,  
           "name": "123456tar",  
           "is_tp": true,  
           "service_id": 2}  
}
```

Request structure and fields description:

```
{'data': {'currency_id': 'Integer',  
          'invisible': 'Bool',  
          'is_tp': 'Bool',  
          'name': 'String',  
          'service_id': 'Integer',  
          'session_id': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	name of the tariff. The name should be unique.
is_tp	N/A	False	Bool	if "true" then function creates TP tariff, if "false" - creates OP tariff.
currency_id	N/A	False	Integer(gt: 0)	currency of the creating tariff.
service_id	1	True	Integer(gt: 0, lt: 3)	defines tariff's service (2 for SMS service, 1 - voice service)
session_id	N/A	True	String(allow_blank: True)	by default session id of current user

			False)	
invisible	False	True	Bool	create invisible tariff or not

2.404 tariff_group_add

Docstring:

Creates new tariff group with given name and TP tariffs assigned to it. Tariff groups may be used for creation price list based on price lists of chosen TP tariffs. Then, created price list can be uploaded to a chosen OP tariff or exported into a file.

Function returns new tariff group id in the database.

Example:

- Client Request

```
{"session_id": "30aa6ac62d855c37268dcd2ed2d02226",
  "data": {"name": "test_gr"}}
}
```

- Server response

```
{"status": 200, "message": "OK", "id": 3, "time_elapsed": 0.045317}
```

Request structure and fields description:

```
{'data': {'name': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
name	N/A	False	String(allow_blank: False)	unique name for new tariff group

2.405 tariff_group_del

Docstring:

Removes given tariff group(s) from the database.

Example:

- Client Request

```
{"session_id": "30aa6ac62d855c37268dcd2ed2d02226",  
  "data": {"groups_ids": [3]}  
}
```

Request structure and fields description:

```
{'data': {'groups_ids': ['Integer']}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
groups_ids	N/A	False	List(Integer(), min_length: 0)	list of group ids to delete

2.406 tariff_group_get**Docstring:**

Shows existed in the system tariff groups for price creation.

Examples:

- Client Request

```
{"session_id": "6f4fcb60ea7d46ba8b40f800a1006261",  
  "data": {"fields": ["id", "name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[1, "first_group"]],
```

```
"time_elapsed": 2.061996}
```

Request structure and fields description:

Request:

```
{  
  "session_id": String(),  
  "data": {  
  }  
}
```

2.407 tariff_set

Docstring:

Changes chosen tariff. In case invisible=True, then all points will be disconnected from tariff.

Example:

- Client Request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",  
  "data": {"id": 5006,  
           "name": "1 tarif",  
           "is_tp": true}  
}
```

Request structure and fields description:

```
{'data': {'id': 'Integer',  
          'invisible': 'Bool',  
          'is_tp': 'Bool',  
          'name': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
is_tp	N/A	False	Bool	if "true" then function searches tariff by id among TP tariffs
id	N/A	False	Integer(gt: 0)	OP or TP tariff id. Generally OP tariff id is odd number, while TP tariff id is even number.
invisible	N/A	True	Bool	change invisible status of given tariff
name	N/A	False	String(allow_blank: False)	new name of the tariff

2.408 tariff_to_group_add

Docstring:

```
Add tariff with given id to the chosen tariff group.
```

```
Not in use.
```

Request structure and fields description:

```
{'data': {'group_id': 'Integer', 'tariff_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
group_id	N/A	False	Integer(gt: 0)	
tariff_id	N/A	False	Integer(gt: 0)	

			0)	
--	--	--	----	--

2.409 tariffs_by_group_del

Docstring:

Remove tariffs with given ids from group with given id. Function returns tariff group id.

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "id": 1,
    "tariffs_ids": [12]
  }
}
```

- Server response

```
{
  "status": 200,
  "message": "OK",
  "id": 1,
  "time_elapsed": 0.02169
}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'tariffs_ids': ['Integer']}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	id of tariff group
tariffs_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	

2.410 tariffs_by_group_get

Docstring:

Returns tariffs which assigned to the given tariff group(s).

Example:

- Client Request

```
{"session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
 "data": {"sort_by": [{"name": ""}],
          "group_id": 1,
          "fields": ["id", "name", "currency"]}
}
```

- Server Response

```
{"status": 200,
 "fields": ["id", "name", "currency"],
 "message": "OK",
 "data": [[4, "Test_tariff_TP", "USD"],
          [6, "add_tariff_validation2", "USD"]],
 "time_elapsed": 0.056964}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency', 'id', 'name')"],
            'group_id': 'Integer',
            'invisible': 'Bool',
            'limit': 'Integer',
            'name': 'String',
            'offset': 'Integer',
            'service_id': 'Integer',
            'sort_by': [{"Enum('currency', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
group_id	N/A	True	Integer(gt: 0)	filter by group
sort_by	[]	True	List([{"Enum('currency', 'id', 'name')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted

offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
name	N/A	True	String(allow_blank: False)	filter by tariff name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
invisible	N/A	True	Bool	filter tariffs by "invisible" field
service_id	1	True	Integer(gt: 0)	service type(1 - voice, 2 - sms)
fields	['currency', 'id', 'name']	True	List([Enum('currency', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
currency	String
id	Int
name	String

2.411 tariffs_by_group_set

Docstring:

Adds new tariffs to the existent tariff group. Function returns tariff group id.

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "id": 1,
    "tariffs_ids": [12]
  }
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "id": 1,  
  "time_elapsed": 0.046739}
```

Request structure and fields description:

```
{'data': {'id': 'Integer', 'tariffs_ids': ['Integer']}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	id of tariff group
tariffs_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	

2.412 tariffs_get

Docstring:

Returns list of OP and TP tariffs names and their ids filtered by clients locked by the current user.

Example:

- Client Request:

```
{"session_id": "0ab11223a707b59600993e99da72d167",  
  "data": {"fields": ["id", "name"],  
           "sort_by": [{"id": "desc"}]}}  
}
```

- Server Response:

```
{"status": 200,  
  "fields": ["id", "name"],  
  "message": "OK",  
  "data": [[4035, 'Tarif Blossom_Zimbabwe_in (104081)'],  
           [4034, 'Tarif Blossom_Zimbabwe_out (104081)']],  
  "time_elapsed": 0.010708  
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('id', 'name')"],  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['id', 'name']	True	List([Enum('id', 'name')], min_length: 0)	Fields that will be present in response. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'name': ""}]	True	List([{'Enum('id', 'name')': Enum('asc', 'desc', '')}], min_length: 0)	Field on which result will be sorted. Result can be sorted ascending or descending. Empty string is the same as asc.
service_id	1	True	Integer(gt: 0, lt: 3)	filter by tariff's service (2 for SMS service, 1 - voice service)

Default fields:

Name	Type & Comments
id	Int
name	String

2.413 tariffs_op_get

Docstring:

Calls API function `_tariffs_get` with option `'is_tp'=False`.

Example:

- Client Request

```
{"session_id": "0ab11223a707b59600993e99da72d167",  
  "data": {"currency_id": 1,  
           "sort_by": [{"name": ""}],  
           "fields": ["tariff_id", "name_currency"]}}
```

- Server Response:

```
{"status": 200,  
  "fields": ["tariff_id", "name_currency"],  
  "message": "OK",  
  "data": [[4233, "0001111 (USD)"],  
           [4149, "000_Zv_in (USD)"],  
           [4221, "011 (USD)"],  
           [4485, "1 (USD)"]],  
  "time_elapsed": 0.010708  
}
```

`_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of OP or TP tariffs existent in the database.

Request structure and fields description:

`_tariffs_get`

```
{'data': {'client_or_pt_name': 'String',  
          'currency_id': 'Integer',  
          'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')"],  
          'is_tp': 'Bool',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'point_name': 'String',  
          'service_id': 'Integer',  
          'show_invisible': 'Bool or Null',  
          'show_pts': 'Bool',  
          'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')": "Enum('asc', 'desc', '')"}],  
          'tariff_id': 'Integer'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
service_id	1	True	Integer(gt: 0, lt: 3)	filter by tariff's service id (2 for SMS service, 1 - voice service)
client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or point.
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
show_pts	True	True	Bool	not in use
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the

				request.
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

Default fields:

_tariffs_get

Name	Type & Comments
dstcode_count	Int - how many actual prices for different destination codes in the tariff
name_currency	String - tariff name with currency (example - "test_tariff(USD)")
currency_name	String
invisible	Boolean
point_info	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
currency_id	Int
tariff_id	Int

name	String
counter	Int - how many points use this tariff
prices_count	Int - how many actual prices in the tariff
client_info	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id

2.414 tariffs_tp_get

Docstring:

Calls API function `_tariffs_get` with option `'is_tp'=True`.

Example:

- Client Request

```
{
  "session_id": "90b28b952c569c96a967f0501138965e",
  "data": {
    "limit": 31,
    "sort_by": [{"name": ""}],
    "offset": 0,
    "fields": ["tariff_id", "name", "currency_id",
               "dstcode_count", "prices_count",
               "name_currency", "client_point_info"]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": [
    "tariff_id", "name", "currency_id", "dstcode_count", "prices_count",
    "name_currency", "point_info", "client_info"],
  "message": "OK",
  "data": [
    [4932, "000999", 7, 65501, 65501, "000999 (UAH)", [[4918, "000999TP1"], [5076, "1234567"]], [[564, "12Ls"], [609, "jur1001_-"]]],
    ],
  "time_elapsed": 0.012435
}
```

`_tariffs_get` ([data_validation](#)) ([default_fields](#))

Returns list of OP or TP tariffs existent in the database.

Request structure and fields description:

_tariffs_get

```
{'data': {'client_or_pt_name': 'String',
          'currency_id': 'Integer',
          'fields': ["Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')"],
          'is_tp': 'Bool',
          'limit': 'Integer',
          'name': 'String',
          'offset': 'Integer',
          'point_name': 'String',
          'service_id': 'Integer',
          'show_invisible': 'Bool or Null',
          'show_pts': 'Bool',
          'sort_by': [{"Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer'},
         'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tariff_id	N/A	True	Integer(gt: 0)	id of OP or TP tariff
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
service_id	1	True	Integer(gt: 0, lt: 3)	filter by tariff's service id (2 for SMS service, 1 - voice service)
client_or_pt_name	N/A	True	String(allow_blank: True)	Filter by company or point name. Returns tariffs assigned to the given company or

				point.
currency_id	N/A	True	Integer(gt: 0)	filter by tariff currency
show_pts	True	True	Bool	not in use
show_invisible	False	True	Or(Bool, Null)	filter by invisible/visible or both if Null
name	N/A	True	String(allow_blank: True)	tariff name or first symbols of tariff name.
point_name	N/A	True	String(allow_blank: True)	filter by point name or first symbols of point name.
is_tp	N/A	False	Bool	if "true" function returns only TP tariffs. Internal flag. This flag is not required in the request.
fields	['dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info']	True	List([Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('dstcode_count', 'name_currency', 'currency_name', 'invisible', 'point_info', 'currency_id', 'tariff_id', 'name', 'counter', 'prices_count', 'client_info'): Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

Default fields:

_tariffs_get

Name	Type & Comments
dstcode_count	Int - how many actual prices for different destination codes in the tariff
name_currency	String - tariff name with currency (example - "test_tariff(USD)")
currency_name	String
invisible	Boolean
point_info	list that contains lists with info about points bound with the tariff: [point_id(Int), point_name(String)] ordered by id
currency_id	Int
tariff_id	Int
name	String
counter	Int - how many points use this tariff
prices_count	Int - how many actual prices in the tariff
client_info	list that contains lists with info about clients bound with the tariff: [client_id(Int), client_name(String)] ordered by id

2.415 tariffs_without_group_get

Docstring:

Returns tariffs which aren't bound with chosen tariffs group.

Example:

- Client Request

```
{"session_id": "30aa6ac62d855c37268dcd2ed2d02226",
  "data": {"sort_by": [{"name": ""}],
           "group_id": 3,
           "fields": ["id", "name", "currency"]}}
```

```
}
```

- Server Response

```
{"status": 200,  
  "fields": ["id", "name", "currency"],  
  "message": "OK",  
  "data": [[4, "Test_tariff_TP", "USD"],  
           [14, "Test_tariff_TP_1", "USD"],  
           [22, "Yess_TP_PrIce", "EUR"],  
           [6, "tariff_validation2", "USD"],  
           [24, "H_out", "USD"]],  
  "time_elapsed": 0.04856}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('currency', 'id', 'name')"],  
          'group_id': 'Integer',  
          'invisible': 'Bool',  
          'limit': 'Integer',  
          'name': 'String',  
          'offset': 'Integer',  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('currency', 'id', 'name')": "Enum('asc', 'desc', '')"}]},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['currency', 'id', 'name']	True	List([Enum('currency', 'id', 'name')], min_length: 0)	required fields.If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('currency', 'id', 'name')": Enum('asc', 'desc', "")}], min_length: 0)	field on which result will be sorted
name	N/A	True	String(allow_blank: False)	filter by first symbols of tariff name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
group_id	N/A	False	Integer(gt: 0)	tariff group id
service_i	1	True	Integer(gt: 0)	service type(1 - voice, 2 - sms)

d				
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
invisible	N/A	True	Bool	filter tariffs by "invisible" field

Default fields:

Name	Type & Comments
currency	String
id	Int
name	String

2.416 tcp_dump_del

Docstring:

Removes given dump file by file name from the billing server.

Example:

- Client Request

```
{
  "session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {
    "file_name": "dump_dev_ips_by_hand_host_193.161.85.79_ip_111.222.111.222_111.222.222.222_2013.11.12_16.30.pcap"
  }
}
```

Request structure and fields description:

```
{'data': {'file_name': 'String'}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
file_name	N/A	False	String(allow_blank: False)	Dump file name

2.417 tcp_dump_file_link_get

Docstring:

Returns download link by dump file.

For dumper by all hosts (without RTP packets) file name pattern is full_

For dumper with RTP media file name pattern is following: ----.pcap where date and time are without separators ('-' or ':') and represents call set up time.

Example:

- Client Request

```
{
  "session_id": "59e93af682809527eabdf1d03b20b551",
  "data": {
    "file_name": "20140501-015945-13478855398-8801793676756-2083834381@193.104.107.7.pcap",
    "fields": ["download_link"]
  }
}
```

- Server Response

```
{
  "status": "200",
  "fields": ["download_link"],
  "message": "OK",
  "data": [
    ["http://127.0.0.1:8070/cdn/download/dumper_files/20140501-015945-13478855398-8801793676756-2083834381@193.104.107.7.pcap"]
  ],
  "time_elapsed": 1.047421
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('download_link',)"],
  'file_name': 'String',
  'host': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['download_link']	True	List([Enum('download_link'),], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
host	N/A	False	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] [12]\d 3[0-2]))?(0\.){3}0(/0)?)\$")	dumper host where to search for dumps
file_name	N/A	False	String(allow_blank: False)	file name to download

Default fields:

Name	Type & Comments
download_link	String

2.418 tcp_dump_get

Docstring:

Returns list of existing TCP dump files, download link for each file and its sizes.

Example:

- Client Request

```
{"session_id": "686c09719c468dbb8c983a99c95ba61c",  
 "data": {"fields": ["file_name",  
                    "file_link",  
                    "file_size"]}}
```

- Server Response:

```

{"status": 200,
 "fields": ["file_name", "file_link", "file_size"],
 "message": "OK",
 "data": [[{"dump_dev_Webinar_in_Webinar_out_host_193.165.85.79_ip_194.128.1
65.61_10.0.152.32_2013.12.11_15.35.pcap",
            "http://193.165.85.80:8070/cdn/download/tcpdump/
dump_dev_Webinar_in_Webinar_out_host_193.165.85.79_ip_194.128.165.61_10.0.152.32_2013.1
2.11_15.35.pcap", 0},
            {"dump_dev_Webinar_in_Webinar_out_host_193.165.85.79_ip_194.128.1
65.61_10.0.152.99_2013.12.11_15.04.pcap",
            "http://193.165.85.80:8070/cdn/download/tcpdump/
dump_dev_Webinar_in_Webinar_out_host_193.165.85.79_ip_194.128.165.61_10.0.152.99_2013.1
2.11_15.04.pcap", 24}],
 "time_elapsed": 0.000831}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('file_name', 'file_link', 'file_size')"],
          'is_sms_dump': 'Bool'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['file_name', 'file_link', 'file_size']	True	List([Enum('file_name', 'file_link', 'file_size')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
is_sms_dump	False	True	Bool	filter by SMS/Voice dumps

Default fields:

Name	Type & Comments
file_name	String
file_link	String
file_size	Int

2.419 tcp_dump_set

Docstring:

Creates new tcp dump request by specified IP addresses (limit is 20 hosts), time interval and RTP host. Size of output file has limitation - 7 MB but since size is being checked 1 time per 2 seconds it may exceed this limit.

Function allows to run only one simultaneous tcpdump process per host.

When dump process is stopped - function copies obtained file to the Billing via SFTP and removes original one from the remote host.

Example:

- Client Request

```
{"session_id": "686c09719c468dbb8c983a99c95ba61c",  
  "data": {"host": "193.161.85.79",  
           "interval": 5,  
           "ips": ["194.28.165.61\\32", "10.0.152.32\\32"],  
           "pt_ids": [4677, 5070]}  
}
```

Request structure and fields description:

```
{'data': {'host': 'String',  
          'interval': 'Integer',  
          'ips': ['String or String'],  
          'is_sms_dump': 'Bool',  
          'pt_ids': ['Integer'],  
          'user_name': 'String'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
user_name	N/A	True	String(allow_blank: False)	will be added to the dump file name, by default from user session
host	N/A	False	String(allow_blank: False)	RTP server IP address or SMS server address

				where dump have to run
is_sms_dump	False	True	Bool	true for SMS dump
interval	N/A	False	Integer(gte: 1, lte: 10)	time interval in minutes, in range [1:10]
pt_ids	N/A	True	List([Integer(gt: 0)], min_length: 0)	list of points ids. Point names will be added to name of file. If empty - 'ips_by_hand' will be added
ips	N/A	False	List([Or(String(allow_blank: False, regex: "^([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] 12)\d 3[0-2])?(0\.){3}0(0)?\$"), String(allow_blank: False, regex: "[a-zA-Z\d-]{,63}(\.[a-zA-Z\d-]{,63})*\$"))], min_length: 0)	list of IP addresses, separated by comma. Maximum quantity of IP addresses - 20.

2.420 tcp_dumper_files_get

Docstring:

Returns recorded TCP dumps which satisfies given time period, B-number and optionally op and tp IPs.

Example:

- Client Request:

```
{
  "session_id": "59e93af682809527eabdf1d03b20b551",
  "data": {
    "date_start": "2014-05-01 01:00:00",
    "date_end": "2014-05-01 01:59:59",
    "called_number": "8801793676756"
  }
}
```

- Server Response:

```
{
  "status": 200,
  "fields": ["file_name", "called_number", "setup_time", "caller_number"],
  "message": "OK",
  "data": [
    "20140501-015945-13478855398-8801793676756-2083834381@193.104.107.7.pcap", "8801793676756",
```

```

        "2014-05-01T01:59:45", "13478855398"]
    ],
    "time_elapsed": 1.047421
}

```

Request structure and fields description:

```

{'data': {'called_number': 'String',
          'date_end': 'String',
          'date_start': 'String',
          'fields': ["Enum('called_number', 'file_name', 'full_size', 'caller_number',
'setup_time')"],
          'host': 'String',
          'op_ip': 'String',
          'tp_ip': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['called_number', 'file_name', 'full_size', 'caller_number', 'setup_time']	True	List([Enum('called_number', 'file_name', 'full_size', 'caller_number', 'setup_time')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
called_number	N/A	False	String(allow_blank: False, regex: "^([0-9#+]*\$", max_length: 100)	B-number
host	N/A	False	String(allow_blank: False, regex: "^(([1-9][1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d 1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d 1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9][12]\d 3[0-2]))?(0\.){3}0(0)?)\$")	dumper host where to search for dumps
tp_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9][1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d 1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d 1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9][12]\d 3[0-2]))?(0\.){3}0(0)?)\$")	TP IP address (option for dumper where dumps for all hosts without media)
date_start	N/A	False	String(allow_blank: False)	

op_ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5]) ([1-9] 12)\d{3}[0-2]))?(0\.){3}0(0)?)\$")	OP IP address (option for dumper where dumps for all hosts without media)
date_end	N/A	False	String(allow_blank: False)	

Default fields:

Name	Type & Comments
called_number	String
file_name	String
full_size	String in kB
caller_number	String
setup_time	String

2.421 time_zones_get

Docstring:

Returns time zones that exists in the database and their description.

Example:

- Client Request

```
{"session_id":"bec5fa94e0d3278d78241d04512c154f",
  "data":{"fields":["time_zone","description"]}}
```

- Server Response

```
{"status": 200,
  "fields": ["time_zone", "description"],
  "message": "OK",
  "data": [[{"time_zone": "-12:00", "description": "-12"}, {"time_zone": "-11:00", "description": "-11"}, {"time_zone": "-10:00", "description": "-10"}, {"time_zone": "-9:00", "description": "-9"}, {"time_zone": "-8:00", "description": "-8"}, {"time_zone": "-7:00", "description": "-7"}, {"time_zone": "-6:00", "description": "-6"}, {"time_zone": "-5:00", "description": "-5"}, {"time_zone": "-4:00", "description": "-4"}, {"time_zone": "-3:00", "description": "-3"}, {"time_zone": "-2:00", "description": "-2"}, {"time_zone": "-1:00", "description": "-1"}, {"time_zone": "0:00", "description": "0"}, {"time_zone": "1:00", "description": "1"}, {"time_zone": "2:00", "description": "2"}]]
```

```

["3:00", "3"], ["4:00", "4"], ["5:00", "5"],
["5:30", "5:30"], ["6:00", "6"], ["7:00", "7"],
["8:00", "8"], ["9:00", "9"], ["10:00", "10"],
["11:00", "11"], ["12:00", "12"]],
"time_elapsed": 0.00578}

```

Request structure and fields description:

```

{'data': {'fields': ["Enum('time_zone', 'description')"],
  'limit': 'Integer',
  'offset': 'Integer',
  'sort_by': [{"Enum('time_zone', 'description')": "Enum('asc', 'desc', '')"}]}
, 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
fields	['time_zone', 'description']	True	List([Enum('time_zone', 'description')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{"Enum('time_zone', 'description'): Enum('asc', 'desc', '')"}], min_length: 0)	defines name of field on which output will be sorted and type of sorting: ascending("asc") or descending("desc"). Example: "sort_by": [{"disconnect_cause": "desc"}]
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
time_zone	String
description	String

2.422 top_countries_get

Docstring:

Returns list of 10 Countries with most calls duration for previous day (00:00:00 - 23:59:59).

Example:

- Client Request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
 "data": {"client_id": 11,
          "direction": "in",
          "fields": ["asr",
                    "acd",
                    "delta_price",
                    "duration",
                    "country"]}
}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer',
          'currency_id': 'Integer',
          'direction': 'String or String',
          'fields': ["Enum('asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum',
          'duration')"],
          'session_id': 'String'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
session_id	N/A	True	String(allow_blank: False)	By default get from current session.
client_id	N/A	False	Integer(gt: 0)	client to get countries statistic
fields	['asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum', 'duration']	True	List([Enum('asr', 'acd', 'country', 'delta_price', 'tp_sum', 'op_sum', 'duration')], min_length: 0)	
currency_i	1	True	Integer(gt: 0)	currency ID for

d				statistic
direction	N/A	False	Or(String(allow_blank: in), String(allow_blank: out))	filter by traffic direction relatively to the softswitch

Default fields:

Name	Type & Comments
asr	String
acd	Int
country	String
delta_price	Numeric(16,2) - difference between op_sum and tp_sum
tp_sum	Numeric(20,2)
op_sum	Numeric(20,2)
duration	Int - calls duration in minutes

2.423 top_duration_regions_get

Docstring:

Returns 10 (or as many as exists) top regions by yesterday calculated duration(duration considering price options mc, ci, ft) for given client.

All destinations for which there are no entry in the DB table country_codes will be grouped as 'Unknown Country' region.

Example:

- Client Request

```
{
  "session_id": "e52185466e19827155f598fcaf4faf41",
  "data": {
    "client_id": 7,
    "direction": "in"
  }
}
```

- Server Response

```
{"status": 200,  
  "message": "OK",  
  "fields": ['today_duration', 'yesterday_duration', 'region'],  
  "data": [[121263, 483136, 'BANGLADESH MOBILE']],  
  "time_elapsed": 1.41132"}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer',  
          'direction': 'String or String',  
          'fields': ["Enum('yesterday_duration', 'region', 'today_duration')"],  
          'service_id': 'Integer'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	client to get regions statistic
service_id	1	True	Integer(gt: 0, lt: 3)	1 - Voice service, 2 - SMS service. In the second case fields 'today_duration' and 'yesterday_duration' represent SMS messages quantity.
direction	N/A	False	Or(String(allow_blank: in), String(allow_blank: out))	filter by traffic direction relatively to the softswitch
fields	['yesterday_duration', 'region', 'today_duration']	True	List([Enum('yesterday_duration', 'region', 'today_duration')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
------	-----------------

yesterday_duration	Int (in minutes rounded up)
region	String
today_duration	Int (in minutes rounded up)

2.424 top_profit_regions_get

Docstring:

Returns 10 (or as many as exists) top regions by yesterday profit (difference between op cost and tp cost rounded mathamatically) of given client.

All destinations for which there are no entry in the DB table country_codes will be grouped as 'Unknown Country' region.

Example:

- Client Request

```
{
  "session_id": "e52185466e19827155f598fcaf4faf41",
  "data": {
    "client_id": 9,
    "direction": "in"
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "fields": ['today_profit', 'yesterday_profit', 'region'],
  "data": [[12126367, 8313675, 'China Proper']],
  "time_elapsed": 1.41132
}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer',
          'direction': 'String or String',
          'fields': ["Enum('yesterday_profit', 'region', 'today_profit')"],
          'service_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	client to get regions statistic
direction	N/A	False	Or(String(allow_blank: in), String(allow_blank: out))	filter by traffic direction relatively to the softswitch
service_id	1	True	Integer(gt: 0, lt: 3)	1 - Voice service, 2 - SMS service.
fields	['yesterday_profit', 'region', 'today_profit']	True	List([Enum('yesterday_profit', 'region', 'today_profit')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

Default fields:

Name	Type & Comments
yesterday_profit	Int
region	String
today_profit	Int

2.425 total_balance_get

Docstring:

Returns total balance of all visible companies in the system's default currency.

Example:

- Client request

```
{"session_id": "ca84cd3eaa596f2e3f7836db065777da",
  "data": {}}
```

- Server Response

```
{"status": 200,
  "message": "OK",
```

```
"data": [["10.0"]],
"time_elapsed": 0.008435}
```

Request structure and fields description:

```
{'data': {}, 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
		1		

2.426 tp_add_or_update

Docstring:

t 1) Creates new disabled termination point (using function '_tp_add') if "id" not specified in request.
Function raises error if client which owns given balance is invisible.

2) Updates termination point tariff, "description", point name, flag "invisible" if "id" is present and "contexts" is absent in the request (using function '_tp_update').

If flag "invisible" was set to "true" - function disables point and removes TP's relations with dialpeers and groups.

3) If "contexts" and "id" are present in request it calls function '_tp_update_within_contexts' which updates point's parameters ("capacity", "max_call_time", point status, "ani_regexp" and so on) within given context(s).

Function can copy parameters of point with given id to the chosen contexts (it used for web only).

Function raises error if user tries to change invisible point. If point status ("enabled" flag) was changed manually (point wasn't closed due to achieving "maximum negative balance" value) function creates new email notification.

`_tp_add` ([data_validation](#)) ([default_fields](#))

Creates new disabled termination point, add new point to the schema routing and each contexts. Allows to specify balance, tariff, "description", "name", flag "invisible" for creating TP.

Function raises error if point with such name already exists in database or if tried to assign point to client which is invisible.

Returns new point id.

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"tarif_id":14,
          "invisible":false,
          "description":"Don't touch!",
          "name":"my_tp",
          "balance_id":1}
}
```

- Server Response

```
{"status": 200, "message": "OK", "id": 22, "time_elapsed": 0.260494}
```

`_tp_update` ([data_validation](#)) ([default_fields](#))

Changes common parameters of given termination point: name, description, tariff and invisible flag.
If "invisible" flag was set as "true" function disables given TP and removes all relations of the TP with groups and dialpeers.

Example:

- Client Request

```
{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
  "data":{"id":4,
          "invisible":false,
          "description":"DO NOT TOUCH THIS POINT."}
}
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.107607}
```

`_tp_update_within_contexts` ([data_validation](#)) ([default_fields](#))

Updates termination point parameters within chosen context. This function is used for copying TP to the contexts specified within list "contexts" in web panel.

Adds new record to the table `api.web_mail` if point status ("enabled" or

"disabled") was changed manually (point wasn't closed due to achieving maximum negative balance value). The function raises error if API user is trying to change invisible point.

Example:

- Client Request

```
{"session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {"id": 4,
    "contexts": [{"max_call_time": 3660,
      "ani_block": "",
      "enabled": true,
      "context_id": 1,
      "ani_regex": "",
      "capacity": 5001,
      "dnis_regex": ""}]
  }
```

- Server Response

```
{"status": 200, "message": "OK", "time_elapsed": 0.088887}
```

Request structure and fields description:

_tp_add

```
{'data': {'balance_id': 'Integer',
  'block_spam_numbers': 'Bool',
  'charge_by': 'Integer',
  'client_id': 'Integer',
  'description': 'String',
  'ext_id': 'Integer',
  'gw_auth': 'Bool',
  'invisible': 'Bool',
  'jurisdict_tariff_id': 'Integer',
  'name': 'Name(at least one printable character). max_length=None',
  'north_american_routing': 'Bool',
  'point_manager_id': 'Integer or Null',
  'route_by': 'Integer',
  'rtp_timeout': 'Integer or Integer',
  'service_id': 'Integer',
  'sip_pair': "Enum('default', 'type0', 'type1', 'type2', 'type3')",
  'tarif_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description

description	""	True	String(allow_blank: True)	
sip_pai	default	True	Enum('default', 'type0', 'type1', 'type2', 'type3')	PAI for termination point
charge_by	N/A	True	Integer(gte: 1)	
name	N/A	False	Name(at least one printable character). max_length=None	
client_id	N/A	True	Integer(gt: 0)	client with given id should be visible.
balance_id	N/A	True	Integer(gt: 0)	client which owns given balance should be visible.
tarif_id	0	True	Integer(gte: 0)	tarif currency has to be equal currency of chosen balance
north_american_routing	N/A	True	Bool	
gw_auth	N/A	True	Bool	if True - GSM Gateway is enabled, False - disabled
invisible	False	True	Bool	boolean option. If True - makes point invisible and disabled
ext_id	0	True	Integer(gte: 0)	TP id for external billing
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	TP manager user id for update notifications
jurisdict_tariff_id	N/A	True	Integer(gte: 0)	
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)
block_spam_numbers	False	True	Bool	if True - blocks spam numbers every 90 seconds
route_by	N/A	True	Integer(gte: 1)	
rtp_timeout	10	True	Or(Integer(gte: 5, lte: 60),	rtp timeout in seconds

			Integer(gte: 0, lte: 0))	
--	--	--	--------------------------	--

_tp_update

```
{'data': {'block_spam_numbers': 'Bool',
          'charge_by': 'Integer',
          'description': 'String',
          'id': 'Integer',
          'invisible': 'Bool',
          'jurisdict_tariff_id': 'Integer',
          'name': 'Name(at least one printable character). max_length=None',
          'north_american_routing': 'Bool',
          'point_manager_id': 'Integer or Null',
          'route_by': 'Integer',
          'service_id': 'Integer',
          'tarif_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)
name	N/A	True	Name(at least one printable character). max_length=None	new point name
jurisdict_tariff_id	N/A	True	Integer(gte: 0)	
block_spam_numbers	False	True	Bool	if True - blocks spam numbers every 90 seconds
route_by	N/A	True	Integer(gte: 1)	
invisible	N/A	True	Bool	boolean option. If True - makes point invisible and disabled
tarif_id	N/A	True	Integer(gt: 0)	new tariff of given TP

description	N/A	True	String(allow_blank: True)	
point_manager_id	N/A	True	Or(Integer(gt: 0), Null)	TP manager user id for update notifications.If key value is "NULL" manager user will be cleared.If key not present in function request data user id will not be updated.
charge_by	N/A	True	Integer(gte: 1)	
north_american_routing	N/A	True	Bool	

_tp_update_within_contexts

```
{'data': {'contexts': [{
    'ani_block': 'String',
    'ani_regexp': 'String',
    'capacity': 'Integer',
    'charge_by': 'Integer',
    'context_id': 'Integer',
    'dnis_regexp': 'String',
    'enabled': 'Bool',
    'from_list': 'Bool',
    'gw_auth': 'Bool',
    'ip_sel_type': 'Integer',
    'jurisdict_tariff_id': 'Integer',
    'list_id': 'Integer',
    'max_call_time': 'Integer',
    'max_call_time_deviation': 'Integer',
    'north_american_routing': 'Bool',
    'route_by': 'Integer',
    'rtp_timeout': 'Integer or Integer',
    'sip_pai': "Enum('default', 'type0', 'type1', 'type2', 'type3')
"}],
    'id': 'Integer'},
    'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
ip_sel_type	N/A	True	Integer()	type of IP address selection:"0" - Sequential, "1" - Weighted, "2" - Random One
sip_pai	N/A	True	Enum('default', 'type0', 'type1',	PAI for termination point

			'type2', 'type3')	
enabled	N/A	True	Bool	point status. If True - point can participate in routing. If point status was changed creates new email notification.
capacity	N/A	True	Integer(gte: 0, lte: 1000)	defines the limit on the number of simultaneous calls that can pass through this point.
max_call_time_deviation	N/A	True	Integer(gte: 0)	Deviation for TP.
jurisdict_tariff_id	N/A	True	Integer(gte: 0)	
max_call_time	N/A	True	Integer(gte: 0, lte: 10800)	maximum call time allowed for the point in seconds
ani_regexp	N/A	True	String(allow_blank: True, max_length: 64)	A number regexp
context_id	N/A	False	Integer(gt: 0)	
charge_by	N/A	True	Integer(gte: 1)	
north_american_routing	N/A	True	Bool	
dnis_regexp	N/A	True	String(allow_blank: True, max_length: 64)	B number regexp
ani_block	N/A	True	String(allow_blank: True, max_length: 255)	exact blocked numbers or regular expressions for group of numbers separated by semicolon
gw_auth	N/A	True	Bool	if True - GSM Gateway is enabled, False - disabled
route_by	N/A	True	Integer(gte: 1)	
list_id	N/A	True	Integer(gte: 0)	ID of list containing numbers for replacement. If list_id is 0 then list will be

				removed from point
from_list	N/A	True	Bool	Flag for A-number replacement from DB. If True number will be replaced from list.
rtp_timeout	N/A	True	Or(Integer(gte: 5, lte: 60), Integer(gte: 0, lte: 0))	rtp timeout in seconds

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	
contexts	N/A	False	List([data.contexts], min_length: 0)	

2.427 tp_add_simple

Docstring:

Creates TP in "Simple routing" mode (add new entries in schema routing and all contexts), creates TP tarif with name [TP_name tarif] for the point and adds relation with service dialpeer in all contexts, if service dialpeer exists.

Function raises error:

- 1) if service dialpeer doesn't exist in context.
- 2) if user is trying to assign TP to the invisible company.

Function returns id of created point in the database.

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
          'client_id': 'Integer',
          'description': 'String',
          'invisible': 'Bool',
          'name': 'String',
          'service_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
description	""	True	String(allow_blank: True)	point description
client_id	N/A	True	Integer(gt: 0)	company owner identifier in table billing.client. Client should be visible.
name	N/A	False	String(allow_blank: False)	TP name, should be unique.
balance_id	N/A	True	Integer(gt: 0)	balance id. If it absent in request but appropriate client id is present - balance of chosen client will be selected randomly.
invisible	False	True	Bool	boolean value, if True then point is invisible and disabled
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)

2.428 tp_auth_add

Docstring:

Tp auth add.

Request structure and fields description:

```
{'data': {'contexts': [{'context_id': 'Integer',  
                        'enabled': 'Bool',  
                        'host': 'String',  
                        'login': 'String',  
                        'password': 'String',  
                        'port': 'Integer or Null',  
                        'protocol': 'Integer'}],  
         'tp_id': 'Integer'},  
'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
host	N/A	False	String(allow_blank: False)	single IP address or subnet IP address, or domain name.
protocol	N/A	True	Integer()	protocol id, 1-sip, 2-h323
port	N/A	True	Or(Integer(gt: 0, lte: 65535), Null)	termination point signalling port for outgoing calls
enabled	True	True	Bool	status, if "True" - enabled
login	N/A	False	String(allow_blank: False)	
password	N/A	False	String(allow_blank: False)	
context_id	N/A	False	Integer(gt: 0)	

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([data.contexts], min_length: 0)	list of data for each context separately
tp_id	N/A	False	Integer(gt: 0)	Termination point id

2.429 tp_auth_del

Docstring:

tp auth del.

Request structure and fields description:

```
{'data': {'contexts': ['Integer'], 'id': 'Integer', 'tp_id': 'Integer'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of contexts id where need to remove the entry.
tp_id	N/A	True	Integer(gt: 0)	Termination Point id. If present in request - deletes all IP addresses entries from chosen TP.
id	N/A	True	Integer(gt: 0)	entry id in database.

2.430 tp_auth_get

Docstring:

TP auth get.

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'fields': ['String'],  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'protocols_to_string': 'Bool',  
          'sort_by': [{"Enum('host', 'protocol', 'tp_id', 'login', 'password', 'enabled', 'id', 'port')": "Enum('asc', 'desc', '')"}]},  
  'tp_id': 'Integer'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
limit	0	True	Integer(gte: 0)	says how many rows will be in response
protocols_to_string	False	True	Bool	if "true" returns protocol's name instead of protocol id(sip - 1, h323 - 2)
contexts	[]	False	List([Integer()], min_length: 0)	filter by context id. If absent in request - returns data from all contexts.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
tp_id	N/A	True	Integer(gt: 0)	filter by tp id
fields	['host', 'protocol', 'tp_id', 'login', 'password', 'enabled', 'id', 'port']	False	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('host', 'protocol', 'tp_id', 'login', 'password', 'enabled', 'id', 'port'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.

Default fields:

Name	Type & Comments
host	String
protocol	Int (protocol id 1 - SIP, 2 - H323) or String
tp_id	Int
login	String(24)

password	String(32)
enabled	Boolean
id	Int
port	Int or null

2.431 tp_auth_set

Docstring:

```
tp auth set.
```

Request structure and fields description:

```
{'data': {'contexts': [{'context_id': 'Integer',
                        'enabled': 'Bool',
                        'host': 'String',
                        'login': 'String',
                        'password': 'String',
                        'port': 'Integer or Null',
                        'protocol': 'Integer'}],
          'id': 'Integer'},
 'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
port	N/A	True	Or(Integer(gt: 0, lte: 65535), Null)	termination point port for outgoing calls
context_id	N/A	False	Integer(gt: 0)	
enabled	N/A	True	Bool	status, if "True" - enabled
protocol	N/A	True	Integer()	

login	N/A	True	String(allow_blank: False)	
password	N/A	True	String(allow_blank: False)	
host	N/A	True	String(allow_blank: False)	single IP address or subnet IP address, or domain name.

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	id of IP address record in database
contexts	N/A	False	List([data.contexts], min_length: 0)	list of data for each context separately

2.432 tp_dest_capacity_add

Docstring:

Creates new traffic limit for given context, TP and destination.

Function raises error if context does not exist in DB or there is no TP with given id in context.

Returns new record id.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "context_id": 1,
    "tp_id": 14,
    "dest": "312,213",
    "capacity": 123
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 22,
  "time_elapsed": 0.260494
}
```

Request structure and fields description:

2.433 tp_dest_capacity_del

Docstring:

Delete tp_dest_capacity record with given id in given context.

Example:

- Client Request

```
{
  "session_id": "63db616315b9e7d6515e9a24aa8f0e2c",
  "data": {
    "id": 2,
    "context_id": 1
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 2,
  "time_elapsed": 0.051203
}
```

Request structure and fields description:

2.434 tp_dest_capacity_get

Docstring:

Returns data about tp destination limits.

Example:

- Client Request

```
{
  "session_id": "63db616315b9e7d6515e9a24aa8f0e2c",
  "data": {
    "context_id": 1,
    "id": 32,
    "fields":
  }
}
```



```

        ["id","tp_id","capacity","dest"]
    }
}
- Server Response
{
    "status": 200,
    "fields": ["id","tp_id","capacity","dest"],
    "message": "OK",
    "data": [[32,14,999,"312,213"]],
    "time_elapsed": 0.03936
}

```

Request structure and fields description:

2.435 tp_dest_capacity_set

Docstring:

Change tp_dest_capacity record with given id.

Example:

```

- Client Request
{
    "session_id":"63db616315b9e7d6515e9a24aa8f0e2c",
    "data":{
        "id":14,
        "capacity":999
    }
}
- Server Response
{
    "status": 200,
    "message": "OK",
    "time_elapsed": 0.073294
}

```

Request structure and fields description:

2.436 tp_free_dialpeers_get

Docstring:

Returns list of dialpeers that not connected with chosen TP.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"sort_by": [{"dialpeer_name": ""}],  
           "contexts": [1],  
           "tp_id": 4,  
           "dialpeer_name": "",  
           "fields": ["dialpeer_id", "dialpeer_name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["dialpeer_id", "dialpeer_name"],  
  "message": "OK",  
  "data": [[4, "H_p (CZ)", [3, "tst.dialpeer2 (CZ)",  
                        [2, "tst.dialpeer (CZ)"]],  
           "time_elapsed": 0.000866}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],  
          'dialpeer_name': 'String',  
          'fields': ["Enum('dialpeer_name', 'dialpeer_id')"],  
          'limit': 'Integer',  
          'offset': 'Integer',  
          'service_id': 'Integer',  
          'sort_by': [{"Enum('dialpeer_name', 'dialpeer_id')": "Enum('asc', 'desc', '')"}],  
          'tp_id': 'Integer'},  
  'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
dialpeer_name	N/A	True	String(allow_blank: True)	Filter by dialpeer's name
fields	['dialpeer_name', 'dialpeer_id']	True	List([Enum('dialpeer_name', 'dialpeer_id')], min_length: 0)	required fields
sort_by	[]	True	List([{"Enum('dialpeer_name', 'dialpeer_id')": Enum('asc', 'desc', '')}])	field on which list will be

			"}]], min_length: 0)	sorted
tp_id	N/A	True	Integer(gt: 0)	filter by certain TP
limit	0	True	Integer(gte: 0)	rows quantity per one request
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	required contexts ids
service_id	1	True	Integer(gt: 0)	op service type(1 - voice, 2 - sms)

Default fields:

Name	Type & Comments
dialpeer_name	String
dialpeer_id	Int

2.437 tp_ip_add

Docstring:

Add combination of IP address, prefix and protocol to chosen termination point and context(s) with specified weight and priority. Allows to add entry if there is no the same IP with the same prefix and protocol in given TP and context(s).

Example:

- Client Request

```
{
  "session_id": "57f4f9abf00013f483e2853a52d6d60b",
  "data": {
    "contexts": [
      {
        "port": 1720,
        "enabled": true,
        "protocol": 2,
        "weight": 0,
        "context_id": 1,
        "priority": 0,
        "prefix": "123",

```

- Server Response

Request structure and fields description:

data.contexts validation description:

Name	Default value	Optional	Type	Description
context_id	N/A	False	Integer(gt: 0)	
weight	0	True	Integer(gte: 0, lte: 100)	weight coefficient in range from 0 to 100
ext_id	0	True	Integer(gte: 0)	not in use
prefix	N/A	True	String(allow_blank: True, regex: "^[\\d#+]*\$")	
ip	N/A	False	Or(String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5]) ([1-9] [12]\\d 3[0-2]))?(0\\.){3}0(/0)?\$"), List([String(allow_blank: False, regex: "^(([1-9] [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.((\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.\\.){2}(\\d [1-9]\\d 1\\d{2} 2[0-4]\\d 25[0-5])\\.\\.\\.){3}0(/0)?\$")])	single IP address or subnet IP address or list of them.

			2[0-4]\d{25[0-5]}(/([1-9][12]\d{3[0-2]}))?(0\.){3}0(/0)?\$")], min_length: 0))	
enabled	N/A	True	Bool	status, if "True" - enabled
codecs	N/A	True	List([Integer(gt: 0)], min_length: 0)	codecs supported by supplier(and filtered by protocol)
protocol	N/A	False	Integer(gt: 0)	protocol id, 1-sip, 2-h323
port	N/A	True	Or(Integer(gt: 0, lte: 65535), Null)	termination point signalling port for outgoing calls
priority	0	True	Integer(gte: -100, lte: 100)	priority in range from -100 to 100

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([data.contexts], min_length: 0)	list of objects. Each object contains data for particular context.
tp_id	N/A	False	Integer(gt: 0)	Termination point id

2.438 tp_ip_del

Docstring:

Removes chosen combination of IP, prefix and protocol from chosen TP and context.
Removes all entries from given TP and context(s), if IP address id not specified in request.

Example:

- Client Request

```
{
  "session_id": "8b35fb7aa9ee0e77facc42835c51c6fc",
  "data": {
    "id": 15,
    "contexts": [1]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.065742
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'id': 'Integer',
    'tp_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of contexts id where need to remove the entry.
id	N/A	True	Integer(gt: 0)	entry id in database.
tp_id	N/A	True	Integer(gt: 0)	Termination Point id. If present in request - deletes all IP addresses entries from chosen TP.

2.439 tp_ip_set

Docstring:

Sets parameters of the chosen combination of IP address, prefix and protocol in given Termination Point and context(s). Allows to set new IP and prefix if there is no the same combination (ip+prefix+protocol) in the given TP and context(s).

Function returns ip address id in the database.

Example:

- Client Request

```
{
  "session_id": "57f4f9abf00013f483e2853a52d6d60b",
  "data": {
    "id": 14,
    "contexts": [
      {
        "port": 1720,
        "enabled": true,
        "protocol": 2,
        "weight": 0,
        "context_id": 1,
        "priority": 0,
        "prefix": "123",
        "codecs": [8, 2, 1],
        "ip": "1.2.3.4/32"
      }
    ]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 14,
  "time_elapsed": 0.112312
}
```

Request structure and fields description:

```
{'data': {'contexts': [{'codecs': ['Integer'],
                             'context_id': 'Integer',
                             'enabled': 'Bool',
                             'ip': 'String',
                             'port': 'Integer or Null',
                             'prefix': 'String',
                             'priority': 'Integer',
                             'protocol': 'Integer',
                             'weight': 'Integer'}],
          'id': 'Integer'},
 'session_id': 'String'}
```

data.contexts validation description:

Name	Default value	Optional	Type	Description
port	N/A	True	Or(Integer(gt: 0, lte: 65535), Null)	termination point port for outgoing calls
prefix	N/A	True	String(allow_blank: True, regex: "^[\\d#+]*\$")	

ip	N/A	True	String(allow_blank: False, regex: "^(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.((\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.)?){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])?([1-9] [12]\d 3[0-2]))?(0\.)?{3}0(/0)?)\$")	single IP address or subnet IP address.
enabled	N/A	True	Bool	status, if "True" - enabled
context_id	N/A	False	Integer(gt: 0)	
protocol	N/A	True	Integer(gt: 0)	protocol id, 1-sip, 2-h323
codecs	N/A	True	List([Integer(gt: 0)], min_length: 0)	codecs supported by supplier(and filtered by protocol)
priority	N/A	True	Integer(gte: -100, lte: 100)	priority in range from -100 to 100
weight	N/A	True	Integer(gte: 0, lte: 100)	weight coefficient in range from 0 to 100

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	False	Integer(gt: 0)	id of IP address record in database
contexts	N/A	False	List([data.contexts], min_length: 0)	list of data for each context separately

2.440 tp_ips_get

Docstring:

Returns IP addresses records of chosen TP and context(s) or records of all TPs filtered by given IP address(es).
This function also allows to get list of TPs to which given IP address(es) assigned.

Example:

- Client Request

```
{
  "session_id": "74c1cf47382eb1a7ffad856372e7242c",
  "data": {
    "sort_by": [{"id": "asc"}],
    "contexts": [1]
  },
  "tp_id": 164,
  "fields":
["id", "enabled", "ip", "prefix", "protocol", "port", "priority", "weight", "codecs"]
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "enabled", "ip", "prefix", "protocol", "port", "priority", "weight", "codecs"],
  "message": "OK",
  "data": [
    [254, false, "5.5.6.3/32", "090", 2, 1720, 0, 0, [8, 2, 1, 5]]
  ],
  "time_elapsed": 0.030114
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
'fields': ['String'],
'ip_list': ['String'],
'limit': 'Integer',
'offset': 'Integer',
'protocols_to_string': 'Bool',
'sort_by': [{"Enum('priority', 'prefix', 'protocol', 'weight', 'ip', 'tp_id', 'codecs', 'enabled', 'id', 'port')": "Enum('asc', 'desc', '')"}]},
'tp_id': 'Integer'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_id	N/A	True	Integer(gt: 0)	filter by tp id
protocols_to_string	False	True	Bool	if "true" returns protocol's name instead of protocol

				id(sip - 1, h323 - 2)
ip_list	N/A	True	List([String(allow_blank: False, regex: "^\(([1-9] [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])\.){2}(\d [1-9]\d 1\d{2} 2[0-4]\d 25[0-5])/([1-9] [12]\d 3[0-2]))?(0\.){3}0(0)?\$"), min_length: 0)	filter by ip address(es)
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
fields	['priority', 'prefix', 'protocol', 'weight', 'ip', 'tp_id', 'codecs', 'enabled', 'id', 'port']	False	List([String(allow_blank: False)], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{Enum('priority', 'prefix', 'protocol', 'weight', 'ip', 'tp_id', 'codecs', 'enabled', 'id', 'port'): Enum('asc', 'desc', ")}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
contexts	[]	False	List([Integer()], min_length: 0)	filter by context id. If absent in request - returns data from all contexts.

Default fields:

Name	Type & Comments
priority	Int
prefix	String(16)
protocol	Int (protocol id 1 - SIP, 2 - H323) or String

weight	Int
ip	String
tp_id	Int
codecs	List()
enabled	Boolean
id	Int
port	Int or null

2.441 tps_by_destination_get

Docstring:

This function returns currently actual tp prices which available for given destination.

Example:

- Client Request

```
{
  "data": {
    "destination": "8801",
    "context_id": 1,
    "fields": ['tp_id', 'tp_name', 'destination', 'country', 'price', 'allowed']
  }
  "session_id": "ca84cd3eaa596f2e3f7836db065777da"
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.010708,
  "message": "OK",
  "data": [
    [12, 'Arbim_H323_out', '8801', 'BANGLADESH MOBILE', '0.01380', True],
    [23, 'Cellocall_out', '8801', 'BANGLADESH MOBILE', '0.00710', True]
  ],
  "fields": ['tp_id', 'tp_name', 'destination', 'country', 'price', 'allowed']
}
```

Request structure and fields description:

```
{'data': {'context_id': 'Integer',
          'destination': 'String',
          'exact_code': 'Bool',
          'fields': ["Enum('country', 'ci', 'ft', 'mc', 'tp_name', 'tp_id', 'price', 'destination', 'allowed')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'tp_or_client_name': 'String'},
        'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
exact_code	True	True	Bool	says to filter by exact destination code or by provided and shorter(if destination code 38050 is provided, then matches 3, 38, 380, 3805 and 38050 codes)
destination	N/A	False	Destination(allow_blank: False, _dest_pattern: ^\d*\$)	exact destination code
fields	['country', 'ci', 'ft', 'mc', 'tp_name', 'tp_id', 'price', 'destination', 'allowed']	True	List([Enum('country', 'ci', 'ft', 'mc', 'tp_name', 'tp_id', 'price', 'destination', 'allowed')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
context_id	N/A	False	Integer(gt: 0)	filter by context
limit	N/A	True	Integer(gt: 0)	says how many rows will be in response
tp_or_client_name	N/A	True	String(allow_blank: False)	filter by tp name or client name
offset	N/A	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
country	String
ci	Int
ft	Int
mc	Int
tp_name	String
tp_id	Int
price	Numeric(16, 5)
destination	String
allowed	Boolean

2.442 tps_free_dialpeer_get**Docstring:**

Returns list of termination points which aren't connected with chosen dialpeer.

Example:

- Client Request

```
{"session_id": "773b6b683024ac6b6cfd95c8b034e659",  
  "data": {"tp_name": "",  
           "sort_by": [{"tp_name": ""}],  
           "contexts": [1],  
           "dialpeer_id": 6,  
           "fields": ["tp_id", "tp_name"]}  
}
```

- Server Response

```
{"status": 200,  
  "fields": ["tp_id", "tp_name"],
```

```

"message": "OK",
"data": [[12, "H_test_tp"], [6, "ikex.TP1"], [10, "TP-14"]],
"time_elapsed": 0.008065}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'dialpeer_id': 'Integer',
          'fields': ["Enum('tp_id', 'tp_name')"],
          'limit': 'Integer',
          'offset': 'Integer',
          'service_id': 'Integer',
          'sort_by': [{"Enum('tp_id', 'tp_name')": "Enum('asc', 'desc', '')"}],
          'tp_id': 'Integer',
          'tp_name': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_name	N/A	True	String(allow_blank: True)	filter by TP name
fields	['tp_id', 'tp_name']	True	List([Enum('tp_id', 'tp_name')], min_length: 0)	requested fields, by defaults fields mentioned in default fields
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
dialpeer_id	N/A	True	Integer(gt: 0)	filter by TP id
tp_id	N/A	True	Integer(gt: 0)	filter by certain TP
sort_by	[]	True	List([{"Enum('tp_id', 'tp_name')": Enum('asc', 'desc', "")}], min_length: 0)	field on which list will be sorted
limit	0	True	Integer(gte: 0)	rows quantity per one request
contexts	[]	False	List([Integer(gt: 0)], min_length: 0)	ids of required contexts
service_i	1	True	Integer(gt: 0)	service type filter(1 - voice, 2 -

d				sms)
---	--	--	--	------

Default fields:

Name	Type & Comments
tp_id	Int
tp_name	String

2.443 tps_free_group_get

Docstring:

Returns list of TPs aren't connected with chosen group.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "tp_name": "",
    "sort_by": [{"tp_name": ""}],
    "group_id": 8,
    "contexts": [1],
    "fields": ["tp_id", "tp_name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["tp_id", "tp_name"],
  "message": "OK",
  "data": [[8, "H_P_to"], [12, "H_test_tp"], [6, "st.TP1"],
    [14, "pablo_tp"], [10, "TP-14"]],
  "time_elapsed": 0.007114}
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
  'fields': ["Enum('tp_id', 'tp_name')"],
  'group_id': 'Integer',
  'limit': 'Integer',
  'offset': 'Integer',
  'service_id': 'Integer',
  'sort_by': [{"Enum('tp_id', 'tp_name')": "Enum('asc', 'desc', '')"}],
  'tp_id': 'Integer',
  'tp_name': 'String'},
}
```

```
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	[]	False	List([Integer()], min_length: 0)	list of contexts id where search relations
group_id	N/A	True	Integer()	
fields	['tp_id', 'tp_name']	True	List([Enum('tp_id', 'tp_name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([{{Enum('tp_id', 'tp_name'): Enum('asc', 'desc', "")}}, min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"tp_name": "asc"}] - sorts result ascending by field "tp_name"
tp_id	N/A	True	Integer()	
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
tp_name	N/A	True	String(allow_blank: True)	filter by TP name
service_id	1	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)

Default fields:

Name	Type & Comments
tp_id	Int

tp_name	String
---------	--------

2.444 tps_free_op_get

Docstring:

Returns list of TPs which aren't connected with chosen OP, i.e. all termination points that aren't belong to the same group (simple or not) as given OP within chosen context(s).
OP may be configured only in simple mode or only advanced mode at the same time.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "op_id": 17,
    "sort_by": [{"name": ""}],
    "name": "",
    "contexts": [1],
    "fields": ["id", "name"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [
    [4, "Test_1_TP"], [8, "ss_TP"], [2, "H_out"], [9, "H_to"],
    [12, "H_test_tp"], [6, "ss1.TP1"], [3, "New_TP"],
    [22, "my_tp"], [10, "TP-14"]
  ],
  "time_elapsed": 0.009625
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
  'fields': ["Enum('id', 'name')"],
  'limit': 'Integer',
  'name': 'String',
  'offset': 'Integer',
  'op_id': 'Integer',
  'sort_by': [{"Enum('id', 'name')": "Enum('asc', 'desc', '')"}]},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	False	Integer(gt: 0)	filter by certain OP
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
contexts	[]	False	List([Integer()], min_length: 0)	contexts ids
fields	['id', 'name']	False	List([Enum('id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'name': ''}]	True	List([{'Enum('id', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
name	N/A	True	String(allow_blank: True)	filter by TP name

Default fields:

Name	Type & Comments
id	Int
name	String - TP name

2.445 tps_from_dialpeers_del

Docstring:

Removes relations between chosen dialpeer and TP(s) or between chosen TP and dialpeer(s) within given context(s).
To remove relation created in Simple mode you should use API function rel_op_tp_del.

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "dialpeers_ids": [6],
    "contexts": [1],
    "tp_ids": ["6", "10"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.067966
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'dialpeers_ids': ['Integer'],
    'tp_ids': ['Integer']
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
tp_ids	N/A	False	List(Integer(gt: 0), min_length: 0)	termination points ids list
dialpeers_ids	N/A	False	List(Integer(gt: 0), min_length: 0)	list of of dialpeer ids which shouldn't contain service dialpeer's id
contexts	N/A	False	List(Integer(gt: 0), min_length: 0)	required contexts ids

2.446 tps_from_groups_del

Docstring:

Removes relations between chosen group(s) and Termination Point(s) within given context(s).
Function allows to delete only relations created in advanced mode.

Example:

- Client Request

```
{
  "session_id": "b9fac2d963ec1967841dfe46202bfc67",
}
```

```

    "data":{"contexts":[1],
            "tp_ids":[],
            "groups_ids":[878]}
}

```

- Server Response

```

{"status": 200, "message": "OK", "time_elapsed": 0.012485}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'groups_ids': ['Integer'],
          'tp_ids': ['Integer']},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of TPs id
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of group ids
contexts	N/A	False	List([Integer()], min_length: 0)	list of contexts id to which relation should be deleted

2.447 tps_get

Docstring:

Returns list of termination points existing in database in given context(s).

Example:

- Client Request

```

{"session_id":"8b35fb7aa9ee0e77facc42835c51c6fc",
 "data":{"limit":2,
        "sort_by":[{"name":""}]},

```

```

        "invisible":false,
        "contexts":[1],
        "offset":0,
        "fields":["id","name","client_name","client_id","enabled",
                  "capacity","tariff_id","tariff_name",
                  "tariff_name_currency"]}
    }

```

- Server Response

```

{"status": 200,
 "fields": ["id", "name", "client_name", "client_id", "enabled",
            "capacity", "tariff_id", "tariff_name",
            "tariff_name_currency"],
 "message": "OK",
 "data": [[4, "Test_1_TP", "Test_supplier", 2, true, 5001, 4,
            "Test_tariff_TP", "Test_tariff_TP (USD)"],
          [18, "ss_TP", "ss_Co", 15, true, 500, [], 22,
            "ss_TP_PrIcE", "Anna_Yess_TP_PrIcE (EUR)"]],
 "time_elapsed": 0.013895}

```

Request structure and fields description:

```

{'data': {'clients_ids': ['Integer'],
          'contexts': ['Integer'],
          'enabled': 'Bool',
          'ext_id': 'Integer',
          'fields': ["Enum('client_invisible', 'tariff_name', 'description', 'from_list',
                           'list_id', 'max_call_time_deviation', 'currency_name_short', 'route_by', 'currency_id',
                           'block_spam_numbers', 'balance_name', 'client_id', 'invisible', 'ani_regexp', 'jurisdiction_tariff_name',
                           'id', 'jurisdiction_tariff_name_currency', 'ext_id', 'gw_auth', 'tariff_id', 'capacity', 'name',
                           'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name',
                           'ip_sel_type', 'charge_by', 'tt_timeout', 'dns_regexp', 'tariff_name_currency', 'jurisdiction_tariff_id',
                           'service_id', 'balance_id', 'north_american_routing', 'sip_pai', 'max_call_time', 'rtp_timeout')"],
          'id': 'Integer',
          'invisible': 'Bool or Null',
          'limit': 'Integer',
          'manager_id': 'Integer',
          'name': 'String',
          'name_or_client_name': 'String',
          'offset': 'Integer',
          'owner_id': 'Integer',
          'points_type': "Enum('jurisdictional', 'non_jurisdictional', 'all')",
          'service_id': 'Integer',
          'sort_by': [{"Enum('client_invisible', 'tariff_name', 'description', 'from_list', 'list_id', 'max_call_time_deviation',
                              'currency_name_short', 'route_by', 'currency_id', 'block_spam_numbers', 'balance_name', 'client_id',
                              'invisible', 'ani_regexp', 'jurisdiction_tariff_name', 'id', 'jurisdiction_tariff_name_currency',
                              'ext_id', 'gw_auth', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id',
                              'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'ip_sel_type', 'charge_by', 'tt_timeout',
                              'dns_regexp', 'tariff_name_currency', 'jurisdiction_tariff_id', 'service_id', 'balance_id',
                              'north_american_routing', 'sip_pai', 'max_call_time', 'rtp_timeout')": "Enum('asc', 'desc', '')"}],
          'tariff_id': 'Integer',
          'with_enabled_ip': 'Bool'},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
ext_id	N/A	True	Integer(gt: 0)	TP id for external billing
service_id	1	True	Integer(gt: 0)	filter by service type(e.g. 1 - voice, 2 - sms)
with_enabled_ip	N/A	True	Bool	show TP with enabled IP(s) only
tariff_id	N/A	True	Integer(gt: 0)	show TP(s) locked by given tariff
fields	['client_invisible', 'tariff_name', 'description', 'from_list', 'list_id', 'max_call_time_deviation', 'currency_name_short', 'route_by', 'currency_id', 'block_spam_numbers', 'balance_name', 'client_id', 'invisible', 'ani_regexp', 'jurisdict_tariff_name', 'id', 'jurisdict_tariff_name_currency', 'ext_id', 'gw_auth', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'ip_sel_type', 'charge_by', 'tt_timeout', 'dnis_regexp',	True	List([Enum('client_invisible', 'tariff_name', 'description', 'from_list', 'list_id', 'max_call_time_deviation', 'currency_name_short', 'route_by', 'currency_id', 'block_spam_numbers', 'balance_name', 'client_id', 'invisible', 'ani_regexp', 'jurisdict_tariff_name', 'id', 'jurisdict_tariff_name_currency', 'ext_id', 'gw_auth', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'ip_sel_type', 'charge_by', 'tt_timeout',	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

	'tariff_name_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'sip_pai', 'max_call_time', 'rtp_timeout']		'dnis_regexp', 'tariff_name_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'sip_pai', 'max_call_time', 'rtp_timeout']], min_length: 0)	
contexts	[]	True	List([Integer()], min_length: 0)	filter by contexts. If is absent function searches termination points in all contexts.
invisible	False	True	Or(Bool, Null)	boolean option, if True returns only invisible TPs (flag is ignored if id is set)
clients_ids	N/A	True	List([Integer()], min_length: 0)	filter by clients
owner_id	N/A	True	Integer(gt: 0)	filter_by owner
name_or_client_name	N/A	True	String(allow_blank: True)	filter by point or client name
name	N/A	True	String(allow_blank: True)	filter by point name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
manager_id	N/A	True	Integer(gt: 0)	filter by client's manager
points_ty	all	True	Enum('jurisdictional',	filter points by

pe			'non_jurisdictional', 'all')	routing type
id	N/A	True	Integer(gt: 0)	id of Termination Point
enabled	N/A	True	Bool	boolean option, if equal True returns only enabled TPs
sort_by	[]	True	List([{Enum('client_invisible', 'tariff_name', 'description', 'from_list', 'list_id', 'max_call_time_deviation', 'currency_name_short', 'route_by', 'currency_id', 'block_spam_numbers', 'balance_name', 'client_id', 'invisible', 'ani_regex', 'jurisdict_tariff_name', 'id', 'jurisdict_tariff_name_currency', 'ext_id', 'gw_auth', 'tariff_id', 'capacity', 'name', 'point_manager_id', 'active_tariff_id', 'ani_block', 'client_name', 'enabled', 'active_tariff_name', 'ip_sel_type', 'charge_by', 'tt_timeout', 'dnis_regex', 'tariff_name_currency', 'jurisdict_tariff_id', 'service_id', 'balance_id', 'north_american_routing', 'sip_pai', 'max_call_time', 'rtp_timeout'): Enum('asc', 'desc', "")}], min_length: 0)	field on which result sorts

Default fields:

Name	Type & Comments
client_invisible	Boolean
tariff_name	String

description	String
from_list	Boolean
list_id	Integer
max_call_time_deviation	Int
currency_name_short	String
route_by	Int
currency_id	Int
block_spam_numbers	Boolean
balance_name	String
client_id	Int
invisible	Boolean
ani_regexp	String
jurisdict_tariff_name	String
id	Int
jurisdict_tariff_name_currency	String - DEPRECATED
ext_id	Int
gw_auth	Boolean
tariff_id	Int
capacity	Int
name	String
point_manager_id	Integer - Point manager user id for update

	notifications
active_tariff_id	Int
ani_block	String
client_name	String
enabled	Boolean
active_tariff_name	String
ip_sel_type	Int
charge_by	Int
tt_timeout	Int
dnis_regexp	String
tariff_name_currency	String - DEPRECATED
jurisdict_tariff_id	Int
service_id	Int
balance_id	Int
north_american_routing	Boolean
sip_pai	String
max_call_time	Int
rtp_timeout	Int

2.448 tps_groups_get

Docstring:

Returns Termination Points connected with chosen group or groups

connected with chosen Termination Point.

Function can filter result by point or group name. You should use TP name filter in conjunction with group id and group name filter - with TP id.

Example:

- Client Request

```
{
  "session_id": "a6ef0db1fee9007f8db0fc1a089d8c8f",
  "data": {
    "limit": 101,
    "sort_by": [{"tp_name": ""}],
    "group_id": 8,
    "contexts": [1],
    "offset": 0,
    "fields": ["tp_id", "group_id", "tp_name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["tp_id", "group_id", "tp_name"],
  "message": "OK",
  "data": [[4, 8, "Test_1_TP"]],
  "time_elapsed": 0.008679}
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'fields': ["Enum('group_name', 'tp_id', 'group_id', 'tp_name')"],
    'group_id': 'Integer',
    'group_name': 'String',
    'limit': 'Integer',
    'offset': 'Integer',
    'service_id': 'Integer',
    'sort_by': [{"Enum('group_name', 'tp_id', 'group_id', 'tp_name')": "Enum('asc', 'desc', '')"}],
    'tp_id': 'Integer',
    'tp_name': 'String'},
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['group_name', 'tp_id', 'group_id', 'tp_name']	True	List([Enum('group_name', 'tp_id', 'group_id', 'tp_name'), min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
tp_name	N/A	True	String(allow_blank: True)	filter by TP name or first symbols of TP

				name
group_id	N/A	True	Integer()	
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
contexts	[]	False	List([Integer()], min_length: 0)	context ids where search relation
service_id	N/A	True	Integer(gt: 0, lt: 3)	filter_by service id (1 - voice service, 2 - SMS service)
sort_by	[]	True	List([Enum('group_name', 'tp_id', 'group_id', 'tp_name'): Enum('asc', 'desc', ")]), min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending. For example [{"tp_name": "asc"}] - sorts result ascending by field "tp_name"
group_name	N/A	True	String(allow_blank: True)	filter by group name or first symbols of group name
limit	0	True	Integer(gte: 0)	says how many rows will be in response
tp_id	N/A	True	Integer()	filter by termination point

Default fields:

Name	Type & Comments
group_name	String
tp_id	Int
group_id	Int
tp_name	String

2.449 tps_op_get

Docstring:

Returns a list of TPs which are connected with given OP in simple and advanced routing modes, i.e. termination points that satisfy condition: Points and dialpeers connected with them must belong to the same group as OP in the chosen context. OP may be configured only in simple mode or only advanced mode at the same time.

Example:

- Client Request

```
{
  "session_id": "6726cebe6ca27ef6a8da30867d6050a8",
  "data": {
    "op_id": 17,
    "sort_by": [{"name": ""}],
    "contexts": [1],
    "fields": ["id", "name"]}
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name"],
  "message": "OK",
  "data": [[24, "NewTP"]],
  "time_elapsed": 0.01015
}
```

Request structure and fields description:

```
{
  'data': {
    'contexts': ['Integer'],
    'fields': ["Enum('priority', 'id', 'name')"],
    'limit': 'Integer',
    'name': 'String',
    'offset': 'Integer',
    'op_id': 'Integer',
    'sort_by': [{"Enum('priority', 'id', 'name')": "Enum('asc', 'desc', '')"}]},
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
op_id	N/A	False	Integer(gt: 0)	filter by certain OP
fields	['priority', 'id', 'name']	False	List([Enum('priority', 'id', 'name')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
contexts	[]	False	List([Integer()], min_length: 0)	contexts ids where search OP-TP relations

sort_by	[{'name': ''}]	True	List([{'Enum('priority', 'id', 'name'): Enum('asc', 'desc', '')}], min_length: 0)	field on which result will be sorted. Result can be sorted ascending or descending.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
name	N/A	True	String(allow_blank: True)	filter by TP name
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.

Default fields:

Name	Type & Comments
priority	Int (relation TP <-> Dialpeer priority)
id	Int
name	String

2.450 tps_to_dialpeers_add

Docstring:

Creates relations between chosen dialpeer and TP(s) or relations between chosen TP and dialpeer(s) within given context(s).

Example:

- Client Request

```
{
  "session_id": "773b6b683024ac6b6cfd95c8b034e659",
  "data": {
    "dialpeers_ids": [6],
    "weight": 0,
    "priority": 4,
    "contexts": [1],
    "tp_ids": ["6", "10"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "time_elapsed": 0.052635
}
```

Request structure and fields description:

```
{'data': {'contexts': ['Integer'],
          'dialpeers_ids': ['Integer'],
          'priority': 'Integer',
          'tp_ids': ['Integer'],
          'weight': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
contexts	N/A	False	List([Integer(gt: 0)], min_length: 0)	required contexts ids
weight	0	True	Integer(gte: 0, lte: 100)	relations weight, all relations will have the same weight.
tp_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of termination points ids. Points should be visible.
dialpeers_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of dialpeers ids
priority	0	True	Integer(gte: -100, lte: 100)	relations priority, all relations have the same priority

2.451 tps_to_groups_add

Docstring:

Adds relation between given group(s) and Termination Point(s). Function disallows to add relation with invisible TP. Given TP(s) and group(s) should have the same services.

Example:

- Client Request

```
{"session_id": "91af15315af776928e1abfa0e88b6641",
 "data": {"contexts": [1],
```

```

        "tp_ids":["14"],
        "groups_ids":[9]}
    }

```

- Server Response

```

{"status": 200, "message": "OK", "time_elapsed": 0.240366}

```

Request structure and fields description:

```

{'data': {'contexts': ['Integer'],
          'groups_ids': ['Integer'],
          'tp_ids': ['Integer']},
 'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
tp_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of TPs id, only visible points allowed
groups_ids	N/A	False	List([Integer(gt: 0)], min_length: 0)	list of groups id
contexts	N/A	False	List([Integer()], min_length: 0)	list of contexts id to which relation should be added

2.452 traffic_classes_get

Docstring:

Creates traffic classes report for chosen points, destinations and given time period. This report shows ASR, ACD, quantity and rate of calls with various cdr classes, total calls quantity and other statistics individually for destination of every point.

Function allows to return data only for destinations which satisfy problem types specified in the system.

There are 8 problem types in the system. Below the list of problem types and conditions appropriate to them:

- 1) Junk traffic - Non-routable; >10 and >60%

- 2) Technical problems - Routed, misconfiguration possible; >10 and >35%
- 3) Supplier overloaded - Routed, uncompleted; >10 and >70%
- 4) Low quality - Completed, unsuccessful; >10 and >30%
- 5) Terrible quality - Completed, unsuccessful; >10 and >30% and <50%
- 6) Route drop - Other successful; >10 and >20%
- 7) Incorrect DC - Other unsuccessful; >10 and >20%
- 8) Unrouted calls - Unrouted; >10 and >10%

Problem type defines by number and rate of calls with particular cdr class. For example problem type "Junk traffic" assigns to the destination if during given time period number of calls with cdr class 'Non-routable' exceeded 10 and 60% from total calls amount.

Example:

- Client Request

```
{'data': {'date_from': '2014-01-12T00:00:00',
          'date_to': '2014-01-12T23:59:59',
          'fields': ['Name',
                    'Dest',
                    'Other successful',
                    'Completed, unsuccessful',
                    'Completed, ok',
                    'calls_quantity',
                    'asr',
                    'acd',
                    'fas_possible'],
          'limit': 2,
          'offset': 0},
  'session_id': u'0043afa9c4b807619744e865fc596f48'}
```

- Server Response

```
{'data': [['H_in',
          '9230',
          '0/0.00%',
          '113/2.87%',
          '410/10.42%',
          3933,
          13.86,
          187,
          '0/0.00%'],
          ['H_in',
          '9231',
          '0/0.00%',
          '39/4.55%',
          '145/16.90%',
          858,
          21.68,
          237,
          '0/0.00%']],
  'fields': ['Name',
            'Dest',
            'Other successful',
            'Completed, unsuccessful',
            'Completed, ok',
            'calls_quantity',
            'asr',
            'acd',
            'fas_possible'],
  'message': 'OK',
  'status': 200,
```

```
'time_elapsed': 0.753967}
```

Request structure and fields description:

```
{'data': {'date_from': 'String',
          'date_to': 'String',
          'destinations': 'String',
          'fields': ["Enum('Completed, ok', 'routed_misconfiguration_possible_pct', 'fas_possible', 'Other unsuccessful', 'Name', 'completed_unsuccessful_pct', 'Routed, misconfiguration possible', 'non_routable_abs', 'Other successful', 'non_routable_pct', 'acd', 'Non-routable', 'fas_possible_abs', 'other_successful_abs', 'completed_zero_abs', 'fas_possible_pct', 'other_unsuccessful_abs', 'completed_unsuccessful_abs', 'other_unsuccessful_pct', 'completed_zero_pct', 'price_tp', 'Completed, unsuccessful', 'unrouted_pct', 'Routed, uncompleted', 'Routed wrong', 'completed_ok_pct', 'routed_wrong_pct', 'routed_uncompleted_pct', 'pt_id', 'Dest', 'Unrouted', 'calls_quantity', 'routed_misconfiguration_possible_abs', 'unrouted_abs', 'routed_wrong_abs', 'other_successful_pct', 'completed_ok_abs', 'Completed, zero', 'asr', 'routed_uncompleted_abs')"],
          'include_subcodes': 'Bool',
          'limit': 'Integer',
          'offset': 'Integer',
          'point_id': 'Integer',
          'problem_type_id': 'Integer',
          'sort_by': [{"Enum('Completed, ok', 'routed_misconfiguration_possible_pct', 'fas_possible', 'Other unsuccessful', 'Name', 'completed_unsuccessful_pct', 'Routed, misconfiguration possible', 'non_routable_abs', 'Other successful', 'non_routable_pct', 'acd', 'Non-routable', 'fas_possible_abs', 'other_successful_abs', 'completed_zero_abs', 'fas_possible_pct', 'other_unsuccessful_abs', 'completed_unsuccessful_abs', 'other_unsuccessful_pct', 'completed_zero_pct', 'price_tp', 'Completed, unsuccessful', 'unrouted_pct', 'Routed, uncompleted', 'Routed wrong', 'completed_ok_pct', 'routed_wrong_pct', 'routed_uncompleted_pct', 'pt_id', 'Dest', 'Unrouted', 'calls_quantity', 'routed_misconfiguration_possible_abs', 'unrouted_abs', 'routed_wrong_abs', 'other_successful_pct', 'completed_ok_abs', 'Completed, zero', 'asr', 'routed_uncompleted_abs')": "Enum('asc', 'desc', '')"}],
          'time_zone': 'String'},
       'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
date_from	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T ())(\\d{2}):(\\d{2}):(\\d{2})(\\.\\d{2})?\$")	Defines start date for report creation.

			d{6}}{0,1}\$")	Shouldn't be more than "date_to".
limit	0	True	Integer(gte: 0)	says how many rows will be in response
date_to	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})(T ())(\\d{2}):(?\\d{2}):(?\\d{2})(\\.\\d{6}}{0,1}\$")	Defines end date for report creation. Shouldn't be less than 'date_from'.
fields	['Completed, ok', 'routed_misconfiguration_possible_pct', 'fas_possible', 'Other unsuccessful', 'Name', 'completed_unsuccessful_pct', 'Routed, misconfiguration possible', 'non_routable_abs', 'Other successful', 'non_routable_pct', 'acd', 'Non-routable', 'fas_possible_abs', 'other_successful_abs', 'completed_zero_abs', 'fas_possible_pct', 'other_unsuccessful_abs', 'completed_unsuccessful_abs', 'other_unsuccessful_pct', 'completed_zero_pct', 'price_tp', 'Completed, unsuccessful', 'unrouted_pct', 'Routed, uncompleted', 'Routed wrong', 'completed_ok_pct', 'routed_wrong_pct', 'routed_uncompleted_pct', 'pt_id', 'Dest', 'Unrouted', 'calls_quantity', 'routed_misconfiguration_possible_abs', 'unrouted_abs', 'routed_wrong_abs', 'other_successful_pct', 'completed_ok_abs', 'Completed, zero', 'asr', 'routed_uncompleted_abs']	True	List([Enum('Completed, ok', 'routed_misconfiguration_possible_pct', 'fas_possible', 'Other unsuccessful', 'Name', 'completed_unsuccessful_pct', 'Routed, misconfiguration possible', 'non_routable_abs', 'Other successful', 'non_routable_pct', 'acd', 'Non-routable', 'fas_possible_abs', 'other_successful_abs', 'completed_zero_abs', 'fas_possible_pct', 'other_unsuccessful_abs', 'completed_unsuccessful_abs', 'other_unsuccessful_pct', 'completed_zero_pct', 'price_tp', 'Completed, unsuccessful', 'unrouted_pct', 'Routed, uncompleted', 'Routed wrong', 'completed_ok_pct', 'routed_wrong_pct', 'routed_uncompleted_pct', 'pt_id', 'Dest', 'Unrouted', 'calls_quantity', 'routed_misconfiguration_possible_abs', 'unrouted_abs', 'routed_wrong_abs', 'other_successful_pct', 'completed_ok_abs', 'Completed, zero', 'asr', 'routed_uncompleted_abs']], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
point_id	N/A	True	Integer(gt: 0)	filter by point id

include_subcodes	N/A	True	Bool	filter traffic which destinations starts with selected destinations
problem_type_id	N/A	True	Integer(gte: -1)	filter by problem type id. 1:"Junk traffic", 2:"Technical problems", 3:"Supplier overloaded", 4:"Low quality", 5:"Terrible quality", 6:"Route drop", 7:"Incorrect DC", 8:"Unrouted calls", -1: "None" all statistics not considering problem type;0, -1: without filters
sort_by	[]	True	List([Enum('Completed, ok', 'routed_misconfiguration_possible_pct', 'fas_possible', 'Other unsuccessful', 'Name', 'completed_unsuccessful_pct', 'Routed, misconfiguration possible', 'non_routable_abs', 'Other successful', 'non_routable_pct', 'acd', 'Non-routable', 'fas_possible_abs', 'other_successful_abs', 'completed_zero_abs', 'fas_possible_pct', 'other_unsuccessful_abs', 'completed_unsuccessful_abs', 'other_unsuccessful_pct', 'completed_zero_pct', 'price_tp', 'Completed, unsuccessful', 'unrouted_pct', 'Routed, uncompleted', 'Routed wrong', 'completed_ok_pct', 'routed_wrong_pct',	field on which result will be sorted. Result can be sorted ascending or descending.

			'routed_uncompleted_pct', 'pt_id', 'Dest', 'Unrouted', 'calls_quantity', 'routed_misconfiguration_possible_abs', 'unrouted_abs', 'routed_wrong_abs', 'other_successful_pct', 'completed_ok_abs', 'Completed, zero', 'asr', 'routed_uncompleted_abs': Enum('asc', 'desc', ")}], min_length: 0)	
destinations	N/A	True	String(allow_blank: True, regex: "(^\\d+(, ?\\d+)*\$)(^\$)")	destination code
time_zone	+00:00	True	String(allow_blank: False, regex: "^[+-]?\\d{1,2}):\\d{2}\$")	"start_date" and "end_date" timezone

Default fields:

Name	Type & Comments
Completed, ok	String (format: '<calls quantity>/<cdr class percentage>')
routed_misconfiguration_possible_pct	Float
fas_possible	String
Other unsuccessful	String (format: '<calls quantity>/<cdr class percentage>')
Name	String (point name)
completed_unsuccessful_pct	Float
Routed, misconfiguration possible	String (format: '<calls quantity>/<cdr class percentage>')
non_routable_abs	Int
Other successful	String (format: '<calls quantity>/<cdr class

	percentage>')
non_routable_pct	Float
acd	Int
Non-routable	String (format: '<calls quantity>/<cdr class percentage>')
fas_possible_abs	Int
other_successful_abs	Int
completed_zero_abs	Int
fas_possible_pct	String
other_unsuccessful_abs	Int
completed_unsuccessful_abs	Int
other_unsuccessful_pct	Float
completed_zero_pct	Float
price_tp	Float (0 for OP and actual value for TP)
Completed, unsuccessful	String (format: '<calls quantity>/<cdr class percentage>')
unrouted_pct	Float
Routed, uncompleted	String (format: '<calls quantity>/<cdr class percentage>')
Routed wrong	String (format: '<calls quantity>/<cdr class percentage>')
completed_ok_pct	Float
routed_wrong_pct	Float
routed_uncompleted_pct	Float

pt_id	Int
Dest	String (destination code)
Unrouted	String (format: '<calls quantity>/<cdr class percentage>')
calls_quantity	Int
routed_misconfiguration_possible_abs	Int
unrouted_abs	Int
routed_wrong_abs	Int
other_successful_pct	Float
completed_ok_abs	Int
Completed, zero	String (format: '<calls quantity>/<cdr class percentage>')
asr	Float
routed_uncompleted_abs	Int

2.453 traffic_share_get

Docstring:

Returns calls duration for client and total(in minutes for voice / in units for sms) rounded mathematically by client for following periods:

- today
- last week
- last month

Function always returns fixed set of fields: ['interval', 'start_date', 'end_date', 'client_value', 'total_value'] and actually list of fields doesn't required and will be ignored if any present in request.

Example:

- Client Request

```
{
  "session_id": "e52185466e19827155f598fc4f4f41",
  "data": {
    "client_id": 9,
    "direction": "in",
    "interval": "today"
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "fields": [
    'interval', 'start_date', 'end_date', 'client_value', 'total_value'
  ],
  "data": [
    [
      'today', '2013-12-05', '2013-12-05', 12, 13
    ]
  ],
  "time_elapsed": 1.41132
}
```

Request structure and fields description:

```
{
  'data': {
    'client_id': 'Integer',
    'direction': 'String or String',
    'interval': "Enum('yesterday', 'last_week', 'last_month')",
    'service_id': 'Integer'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
client_id	N/A	False	Integer(gt: 0)	client to get regions statistic
direction	N/A	False	Or(String(allow_blank: in), String(allow_blank: out))	filter by traffic direction relatively to the softswitch
service_id	1	True	Integer(gt: 0, lt: 3)	1 - Voice service, 2 - SMS service.
interval	N/A	True	Enum('yesterday', 'last_week', 'last_month')	interval to built traffic rate diagram

2.454 traffic_stat_get

Docstring:

Get traffic information at hosts listed within parameter 'hosts' via SNMP: ifHCInOctets and ifHCOutOctets.

If 'hosts' is absent in query returns information about enabled hosts.

If there are no enabled hosts then returns message: "No enabled hosts found."

Example:

- Client Request:

```
{
  "session_id": "121dbdf08c0570f3b91f1072df5b6506",
  "data": {
    "fields": ["traffic_info"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "time_elapsed": 0.123019,
  "errors": [],
  "fields": ["traffic_info"],
  "message": "OK",
  "data": [
    ["194.56.122.45", 1321422028395, 1332534648305]
  ]
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('traffic_info',)"], 'hosts': ['String']},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
hosts	N/A	True	List([String(allow_blank: False)], min_length: 0)	list that consists of IP addresses of RTP hosts.
fields	['traffic_info']	True	List([Enum('traffic_info',)], min_length: 0)	required fields. If absent - takes fields specified in 'default fields'. If empty - returns error.

Default fields:

Name	Type & Comments
traffic_info	String

2.455 turn_of_changes_get

Docstring:

Shows changes (in routing schemas [tables dialpeers, groups and points], in tables currencies or in prices) which haven't been applied on the Radius (or in Radiuses if there are several contexts).

Example:

- Client Request

```
{
  "session_id": "6f4fcb60ea7d46ba8b40f800a1006261",
  "data": {
    "limit": 100,
    "offset": 0,
    "fields": ["data_type", "name", "added"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["data_type", "name", "added"],
  "message": "OK",
  "data": [[1, "", "2014-01-16T13:51:10.104057+00:00"]],
  "time_elapsed": 0.012979
}
```

Request structure and fields description:

```
{
  'data': {
    'date_from': 'String',
    'date_to': 'String',
    'fields': ["Enum('added', 'name', 'data_type')"],
    'limit': 'Integer',
    'offset': 'Integer',
    'sort_by': [{"Enum('added', 'name', 'data_type')": "Enum('asc', 'desc', '')"}]
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optiona	Type	Description
------	---------------	---------	------	-------------

		1		
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
date_from	N/A	True	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	defines start date to filter result by adding date
fields	['added', 'name', 'data_type']	True	List([Enum('added', 'name', 'data_type')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[{'added': 'desc'}]	True	List([{Enum('added', 'name', 'data_type'): Enum('asc', 'desc', ")"}], min_length: 0)	field on which result will be sorted. By defaults result sorts by added time descendant.
date_to	N/A	True	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	defines end date for request to filter by adding date
limit	0	True	Integer(gte: 0)	says how many rows will be in response

Default fields:

Name	Type & Comments
added	String
name	String
data_type	Int

2.456 user_add

Docstring:

Creates new user with given permissions, accessible clients, login and password. Function checks for owners quantity limit and raises error if it has been exceeded when adding.

Password should contain lower-case and upper-case letters, digits and to be at least 8 symbols length. Password shouldn't contain login.

Returns new user id.

Example:

- Client Request

```
{
  "session_id": "376fdf46ab6197af0b379b2a1b996462",
  "data": {
    "phone": "",
    "role_id": 1,
    "show_only_my_companies": false,
    "invisible": false,
    "is_client": false,
    "selected_clients": "",
    "password": "somepassword",
    "allowed_ip": "",
    "time_zone": "0:00",
    "email": "su@test.net",
    "name": "superuser",
    "login": "superuser",
    "notes": "",
    "all_comp": true,
    "show_ip": true
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 13,
  "time_elapsed": 0.040819
}
```

Request structure and fields description:

```
{
  'data': {
    'all_comp': 'Bool',
    'allowed_ip': 'String',
    'currency_id': 'Integer',
    'email': 'String',
    'hide_owners_profit': 'Bool',
    'hide_owners_profit_conditionally': 'Bool',
    'invisible': 'Bool',
    'is_client': 'Bool',
    'login': 'String',
    'name': 'String',
    'notes': 'String',
    'password': 'String',
    'phone': 'String',
    'role_id': 'Integer',
    'selected_clients': 'String',
    'show_only_my_companies': 'Bool',
    'time_zone': 'String'
  },
  'session_id': 'String'
}
```

data validation description:

Name	Default value	Optional	Type	Description
password	N/A	False	String(allow_blank: False)	user password, must not contain user login. Recommended length of password is at least 8 symbols with upper and lower case, digits and punctuation symbols.
email	N/A	False	String(allow_blank: False, regex: "(?P^[-#\$%&'*/+=?^_`{} ~0-9A-Z]+(\\.[-#\$%&'*/+=?^_`{} ~0-9A-Z]+)* ^"([\\001-\\010\\013\\014\\016-\\037!#-\\[\\]-\\177] \\[\\001-011\\013\\014\\016-\\177])*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\\. [A-Z0-9-]{2,}\\. ?)\$)\\[(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)(\\.(25[0-5] 2[0-4]\\d [0-1]?\\d?\\d)){3}\\]" , max_length: 254)	user email
hide_owners_profit	False	True	Bool	set this flag to true to hide "Cost In/Cost Out/Margin/Profit" in incoming/outgoing calls and financial summary
phone	N/A	False	String(allow_blank: False, regex: "^\\+?\\d{1,24}\$")	user phone
time_zone	N/A	False	String(allow_blank: False, regex: "^([+-]?\\d{1,2}):\\d{2}\$")	user's time zone, default time zone for all requests such as "incoming_calls_get", "spam_calls_get" and so on from web interface.
currency_id	1	True	Integer()	ID of currency in which MC data will be shown.
notes	N/A	False	String(allow_blank: True)	notes about user
is_client	N/A	False	Bool	if "True" user can access information only about companies which assigned to him

				in traffic reports
role_id	N/A	False	Integer(gte: 1)	user profile id. If "is_client" set to True you can set "role_id" which corresponds to names "Client.", "Client. Show Ballance."
invisible	N/A	False	Bool	if "True" - makes user invisible and disabled
name	N/A	False	String(allow_blank: False, max_length: 32)	user name, no longer than 32 symbols.
login	N/A	False	String(allow_blank: False)	user login
show_only_my_companies	False	True	Bool	Show_only_my_companies flag. If "True" user can see owned clients and companies where user is manager
selected_clients	N/A	False	String(allow_blank: True)	clients ids to which user has access
all_comp	N/A	False	Bool	if "True" user has access to all clients. You can't set "all_comp"=True if "is_client"=True.
allowed_ip	N/A	True	String(allow_blank: True)	single IP address or net address from which given user allowed to get access to the web panel.
hide_owners_profit_conditions	False	True	Bool	Show cost in/out and margin in fin summary and incoming/outgoing sms by complicated condition (MCD-2741)

2.457 user_companies_get

Docstring:

Shows information about given companies. Function returns companies locked by chosen user if 'user_id' specified in request.

Example:

- Client Request

```
{"session_id": "51e6d38b18cc6a5171b7992e06ca1753",
  "data": {"id": 1,
    "fields": ["name", "description", "invisible", "email",
      "phone", "address", "email_tech", "manager_id",
      "multicurrency", "owner_id", "balance_id",
      "send_traffic_report", "service_restore_flag"]}
}
```

- Server Response

```
{"status": 200,
  "fields": ["name", "description", "invisible", "email", "phone",
    "address", "email_tech", "manager_id", "multicurrency",
    "owner_id", "balance_id", "send_traffic_report",
    "service_restore_flag"],
  "message": "OK",
  "data": [{"Test_client", "Main Test Company", false,
    "test@test.com", "test1@test.com",
    "065-82-79-481", "Ukraine.
    Kharkov city.",
    "tech_email@test.com", 1, false, 1, 1, false,
    false}],
  "time_elapsed": 0.006684}
```

Request structure and fields description:

```
{'data': {'balance_id': 'Integer',
  'currency_id': 'Integer',
  'ext_id': 'Integer',
  'fields': ["Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag',
    'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'i
nvisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currenc
y_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id',
'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_i
d', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')"],
  'id': 'Integer',
  'is_accucore': 'Bool',
  'limit': 'Integer',
  'manager_id': 'Integer',
  'name': 'String',
  'offset': 'Integer',
  'owner_id': 'Integer',
  'send_traffic_report': 'Bool',
  'services_ids': ['Integer'],
  'show_invisible': 'Bool or Null',
  'sort_by': [{"Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag'
```

```
, 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address',
'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id',
'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id',
'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id',
'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')": "Enum('asc', 'desc', '')"}]],
    'user_id': 'Integer',
    'vat_number': 'String'},
'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
sort_by	[]	True	List([Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number', 'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id'): Enum('asc', 'desc', ")}]), min_length: 0)	field on which result sorts
id	N/A	True	Integer(gt: 0)	id of company
balance_id	N/A	True	Integer(gt: 0)	balance id to get client's id if "id" not provided
fields	['email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number', 'short_currency', 'alarm_limit',	True	List([Enum('email_tech', 'owner_name', 'description', 'tp_limit_flag', 'multicurrency', 'tp_limit', 'phone', 'actual_vat_balance', 'balance_id', 'address', 'invisible', 'vat_number',	required fields. If absent - takes fields specified in "default fields". If empty - returns error.

	'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id']		'short_currency', 'alarm_limit', 'id', 'notify_flag', 'currency_id', 'send_traffic_report', 'service_restore_flag', 'name', 'services_ids', 'ext_id', 'routes_changes_report', 'limit_flag', 'alarm_flag', 'include_vat', 'limit', 'manager_id', 'alarm_limit_unit', 'balance', 'email', 'note', 'owner_id')], min_length: 0)	
currency_id	N/A	True	Integer(gt: 0)	Allows to see companies with specific currency
user_id	N/A	True	Integer(gt: 0)	this filter allows to show companies locked to the given user.
name	N/A	True	String(allow_blank: True)	company name or first chars of name
ext_id	N/A	True	Integer(gt: 0)	client id for external billing.
limit	0	True	Integer(gte: 0)	says how many rows will be in response
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
manager_id	N/A	True	Integer(gt: 0)	filter by manager
is_accucore	False	True	Bool	special parameter for accucore, to show all clients (visible and invisible)
show_invisible	False	True	Or(Bool, Null)	Allows to display only visible (if False) or only

				invisible (if True) companies in report. If this option is absent in request returns visible and invisible companies. In conjunction with company id this option doesn't have influence with response.
send_traffic_report	N/A	True	Bool	If True allows to send short traffic report email
owner_id	N/A	True	Integer(gt: 0)	filter by owner
services_ids	[1]	True	List([Integer(gt: 0)], min_length: 0)	filter by service type, by default service type is 1(voice)
vat_number	N/A	True	String(allow_blank: True)	client VAT number

Default fields:

Name	Type & Comments
email_tech	String
owner_name	String
description	String
tp_limit_flag	Boolean
multicurrency	Boolean
tp_limit	String
phone	String

actual_vat_balance	String
balance_id	Int (the id of company's balance)
address	String
invisible	Boolean
vat_number	String
short_currency	String
alarm_limit	String
id	Int (company id)
notify_flag	Boolean
currency_id	Int (id of company's currency)
send_traffic_report	Boolean
service_restore_flag	Boolean
name	String (company name)
services_ids	Array of integers
ext_id	Int (external id of company for synchronization with external billing, deprecated)
routes_changes_report	Boolean
limit_flag	Boolean
alarm_flag	Boolean
include_vat	Boolean
limit	String
manager_id	Int or null

alarm_limit_unit	Int (1-default currency, 2-%)
balance	String
email	String (billing emails of company)
note	String
owner_id	Int (id of owner assigned to the company)

2.458 user_set

Docstring:

Changes settings of chosen user. Function checks for owners quantity limit and raises error if it has been exceeded when modifying.

Function returns user id in database.

Example:

- Client Request

```
{
  "session_id": "d7dbedb0882935c82824bf868b2885c6",
  "data": {
    "phone": "",
    "role_id": 1,
    "show_only_my_companies": false,
    "invisible": false,
    "is_client": false,
    "selected_clients": "10,2,17,1,3,4,7,9,16,11,8,12,13,14,15,5,6",
    "allowed_ip": "",
    "id": 13,
    "time_zone": "0:00",
    "email": "su@gov.no",
    "name": "superuser",
    "login": "superuser",
    "notes": "",
    "all_comp": true
  }
}
```

- Server Response

```
{
  "status": 200,
  "message": "OK",
  "id": 13,
  "time_elapsed": 0.082021
}
```

Request structure and fields description:

```
{
  'data': {
    'all_comp': 'Bool',
  }
}
```

```

'allowed_ip': 'String or Null',
'currency_id': 'Integer',
'email': 'String',
'hide_owners_profit': 'Bool',
'hide_owners_profit_conditionally': 'Bool',
'id': 'Integer',
'invisible': 'Bool',
'is_banned': 'Bool',
'is_client': 'Bool',
'login': 'String',
'name': 'String',
'notes': 'String',
'password': 'String',
'phone': 'String',
'role_id': 'Integer',
'selected_clients': 'String',
'show_only_my_companies': 'Bool',
'time_zone': 'String',
'wrong_credentials_ban': 'Bool'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
login	N/A	True	String(allow_blank: False)	user login
is_banned	N/A	True	Bool	If "True" user will be banned
wrong_credentials_ban	N/A	True	Bool	flag define that user must be banned after several(by default is 5. defined in section [login] conf.ini) tries to login with wrong password
notes	N/A	True	String(allow_blank: True)	some additional information
hide_owners_profit_conditionally	N/A	True	Bool	Show cost in/out and margin in fin summary and incoming/outgoing sms by complicated condition (MCD-

				symbols.
is_client	N/A	True	Bool	if "True" user can access information only about companies which assigned to him in traffic reports.
invisible	N/A	True	Bool	if "True" - makes user invisible and banned
currency_id	N/A	True	Integer()	ID of currency in which MC data will be shown.
role_id	N/A	True	Integer(gte: 1)	new user profile id. If "is_client" is True you can assign only roles "Client.", "Client. Show Ballance."
show_only_my_companies	N/A	True	Bool	Show_only_my_companies flag. If "True" user can see owned clients and companies where user is manager
name	N/A	True	String(allow_blank: False, max_length: 32)	new user name, no longer than 32 symbols.
allowed_ip	N/A	True	Or(String(allow_blank: True), Null)	single IP address from which given user allowed to get access to the web panel.

2.459 users_get

Docstring:

Returns data and settings of mediacore users.

Example:

- Client Request

```
{
  "session_id": "fbb6a26407cad8f37fcb54ed7d94f049",
  "data": {
    "sort_by": [{"name": ""}],
    "fields": ["id", "name", "login"]
  }
}
```

- Server Response

```
{
  "status": 200,
  "fields": ["id", "name", "login"],
  "message": "OK",
  "data": [
    [4, "accucore", "accucore"],
    [1, "dev", "dev"],
    [7, "test_user", "test_user"],
    [8, "ID", "ID"],
    [11, "Irina", "irina"],
    [9, "test_1", "test_1"]
  ],
  "time_elapsed": 0.005063
}
```

Request structure and fields description:

```
{
  'data': {
    'fields': [
      "Enum('all_comp', 'is_client', 'name', 'currency_id', 'is_banned',
        'role_id', 'notes', 'show_only_my_companies', 'time_zone', 'email', 'hide_owners_profit_conditionally', 'phone', 'rl_name', 'allowed_ip', 'invisible', 'login', 'password', 'id', 'hide_owners_profit')",
      'id': 'Integer',
      'is_client': 'Bool',
      'limit': 'Integer',
      'login': 'String',
      'name': 'String',
      'offset': 'Integer',
      'role_id': 'Integer',
      'session_id': 'String',
      'show_banned': 'Bool or Null',
      'show_invisible': 'Bool',
      'sort_by': [
        [{"Enum('all_comp', 'is_client', 'name', 'currency_id', 'is_banned',
          'role_id', 'notes', 'show_only_my_companies', 'time_zone', 'email', 'hide_owners_profit_conditionally', 'phone', 'rl_name', 'allowed_ip', 'invisible', 'login', 'password', 'id', 'hide_owners_profit')": "Enum('asc', 'desc', '')"}]
      ],
      'session_id': 'String'
    ]
  }
}
```

data validation description:

Name	Default value	Optional	Type	Description
id	N/A	True	Integer(gt: 0)	user id in database
name	N/A	True	String(allow_blank: True)	first chars of user name or login
is_client	N/A	True	Bool	filter by client type
role_id	N/A	True	Integer(gte: 0)	filter by role id

show_in visible	N/A	True	Bool	allows to get invisible users, false by defaults
show_banned	None	True	Or(Bool, Null)	filter by banned users. If True - shows banned users only. If False shows not banned users only. If None - shows all users.
offset	0	True	Integer(gte: 0)	Offset says to skip that many rows before beginning to return rows.
session_id	N/A	True	String(allow_blank: False)	allows to get current user's settings according to "session_id"
login	N/A	True	String(allow_blank: False)	filter by login
fields	['all_comp', 'is_client', 'name', 'currency_id', 'is_banned', 'role_id', 'notes', 'show_only_my_companies', 'time_zone', 'email', 'hide_owners_profit_conditionally', 'phone', 'rl_name', 'allowed_ip', 'invisible', 'login', 'password', 'id', 'hide_owners_profit']	True	List([Enum('all_comp', 'is_client', 'name', 'currency_id', 'is_banned', 'role_id', 'notes', 'show_only_my_companies', 'time_zone', 'email', 'hide_owners_profit_conditionally', 'phone', 'rl_name', 'allowed_ip', 'invisible', 'login', 'password', 'id', 'hide_owners_profit')], min_length: 0)	required fields. If absent - takes fields specified in "default fields". If empty - returns error.
sort_by	[]	True	List([Enum('all_comp', 'is_client', 'name', 'currency_id', 'is_banned', 'role_id', 'notes', 'show_only_my_companies', 'time_zone', 'email', 'hide_owners_profit_conditionally', 'phone', 'rl_name', 'allowed_ip', 'invisible', 'login', 'password', 'id', 'hide_owners_profit'): Enum('asc', 'desc', "")]),	field by which users list will be sorted

			min_length: 0)	
limit	0	True	Integer(gte: 0)	rows quantity per one request

Default fields:

Name	Type & Comments
all_comp	Boolean
is_client	Boolean
name	String
currency_id	Int - ID of currency in which MC data will be shown.
is_banned	Boolean
role_id	Int, user's role id
notes	String
show_only_my_companies	Boolean
time_zone	String
email	String
hide_owners_profit_conditionally	Boolean - Show cost in/out and margin in fin summary and incoming/outgoing sms by complicated condition (MCD-2741)
phone	String
rl_name	String, user's role name
allowed_ip	String or null
invisible	Boolean
login	String

password	String. Actually function returns user login instead of password for security reasons.
id	Int
hide_owners_profit	Boolean

2.460 util_should_stop

Docstring:

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.461 util_stop_job

Docstring:

Request structure and fields description:

Request:

```
{
  "session_id": String(),
  "data": {
  }
}
```

2.462 vat_rate_add

Docstring:

Adds new entry into vat_rates table.
Note: - date for effective_date field must be in the future;
- user must be owner to change VAT;
- VAT value cannot be same as the last one.

Request structure and fields description:

```
{'data': {'effective_date': 'String', 'owner_id': 'Integer', 'vat': 'Float'},  
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
owner_id	N/A	False	Integer(gt: 0)	owner for which VAT rate will be added
effective_date	N/A	False	String(allow_blank: False)	VAT rate usage start date
vat	N/A	False	Float(gte: 0, lte: 100)	VAT rate value

2.463 vat_rates_get

Docstring:

Returns all VAT rate records.

Example:

- Client Request

```
{
  "session_id": "077d4f94e5beefa123c98fe72487d3fd",
  "data": {
    "fields": ["change_date", "effective_date", "vat", "user_name"]
  }
}
```

Request structure and fields description:

```
{'data': {'fields': ["Enum('effective_date', 'change_date', 'vat_invoice', 'user_name', 'vat')"],
          'owner_id': 'Integer'},
 'session_id': 'String'}
```

data validation description:

Name	Default value	Optional	Type	Description
fields	['effective_date', 'change_date', 'vat_invoice', 'user_name', 'vat']	True	List([Enum('effective_date', 'change_date', 'vat_invoice', 'user_name', 'vat')], min_length: 0)	fields which will be present in response, if empty - fields mentioned in "default_fields"
owner_id	N/A	False	Integer(gt: 0)	filter by owner

Default fields:

Name	Type & Comments
effective_date	String
change_date	String
vat_invoice	Float

user_name	String
vat	Float

2.464 xdr_request_add

Docstring:

Creates new XDR request for chosen company or point(s) for specific time period

Function returns id of created request in database.

Example:

- Client Request

```
{
  "session_id": "3195e5c6c7fa3490d96dbd26ae523099",
  "data": {
    "time_zone": "0:00",
    "end_date": "2013-10-30T23:59:59",
    "start_date": "2013-10-30T00:00:00",
    "point_id": 4519,
    "fields": [
      "nas_ip",
      "setup_time",
      "caller_number",
      "called_number",
      "clear_number",
      "remote_host",
      "status",
      "tp_name",
      "op_name",
      "tp_price",
      "cost_out",
      "op_price",
      "cost_in",
      "dialpeer_name"
    ]
  }
}
```

- Server Response:

```
{
  "status": 200,
  "time_elapsed": 0.047496,
  "message": "OK",
}
```

Request structure and fields description:

```
{'data': {'client_id': 'Integer',
          'destination': 'String',
```

```

'email': 'String or String',
'end_date': 'String',
'fields': ["Enum('op_name', 'tp_name', 'send_time', 'clear_number', 'called_number_out', 'country', 'op_ani', 'tp_ani', 'status', 'state', 'dlr', 'dlr_time', 'op_price', 'tp_price', 'cost_in', 'cost_out', 'op_host', 'tp_host', 'nas_ip', 'dialpeer_name', 'message_id', 'op_currency', 'tp_currency')"],
'include_subcodes': 'Bool',
'mcc': 'String',
'mnc': 'String',
'only_successful_attempts': 'Bool',
'point_id': 'Enum(-2, -1) or Integer',
'start_date': 'String',
'time_zone': 'String'},
'session_id': 'String'}

```

data validation description:

Name	Default value	Optional	Type	Description
time_zone	+00:00	True	String(allow_blank: False, regex: "^([+-]?d{1,2}):(\d{2})\$")	Time zone of the "start_date" and "end_date" of request. Specify this option if client's timezone different from GMT+0.
only_successful_attempts	N/A	True	Bool	
mnc	N/A	True	String(allow_blank: False, regex: "^(\d{1,3})?((, \s)\d{1,3})*\$")	
fields	['op_name', 'tp_name', 'send_time', 'clear_number', 'called_number_out', 'country', 'op_ani', 'tp_ani', 'status', 'state', 'dlr', 'dlr_time', 'op_price', 'tp_price', 'cost_in', 'cost_out', 'op_host', 'tp_host', 'nas_ip', 'dialpeer_name', 'message_id', 'op_currency', 'tp_currency']	False	List([Enum('op_name', 'tp_name', 'send_time', 'clear_number', 'called_number_out', 'country', 'op_ani', 'tp_ani', 'status', 'state', 'dlr', 'dlr_time', 'op_price', 'tp_price', 'cost_in', 'cost_out', 'op_host', 'tp_host', 'nas_ip', 'dialpeer_name', 'message_id', 'op_currency', 'tp_currency')], min_length: 1)	Defines fields to be exported in the cdrs report.

	'nas_ip', 'dialpeer_name', 'message_id', 'op_currency', 'tp_currency']			
mcc	N/A	True	String(allow_blank: False, regex: "^\d{3}\$")	
email	N/A	True	Or(String(allow_blank: True), String(allow_blank: False, regex: "(?P^[-#%&'+/=^_`{} ~0-9A-Z]+(\\.[-#%&'+/=^_`{} ~0-9A-Z]+)* ^"([001-010\013\014\016-037!#-\\-177])\\[\\001-011\013\014\016-177])*")@(?P(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\\.)+(?:[A-Z]{2,6}\.?[A-Z0-9-]{2,}\.?)\$) \[(25[0-5] 2[0-4]\d[0-1]? \d?\d)(\. (25[0-5] 2[0-4]\d[0-1]? \d?\d))\{3\} \\$", max_length: 254))	email for notification about cdr generation. By defaults - user email from current session. If in user profile email not specified, then owner email will be used.
destination	N/A	True	String(allow_blank: False, regex: "^(\\d+(, \\s)?)*\$")	Allows to generate CDR with single destination or destinations included only. Codes should be separated by comma.
end_date	N/A	False	String(allow_blank: False, regex: "^(\\d{4})-(\\d{2})-(\\d{2})((T) (O))(\\d{2}): (\\d{2}): (\\d{2})\$")	request end date in format "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-ddhh:mm:ss"
include_subcodes	False	False	Bool	Use this option in conjunction with "destination" key. If True CDRs will be generated for "destination" and its subcodes.
client_id	N/A	True	Integer(gt: 0)	company id. If it defined and "point_id" is not - then generates CDR's for all points of given client. For each point will be created individual file.
point_id	N/A	False	Or(Enum(-2, -1), Integer(gte: 1))	point ID. Specify here '-1' if you want to generate cdrs for all origination

				points of chosen company (if 'client_id' not specified in request then will be generated cdrs for all OPs), '-2' if you want to generate cdrs for all termination points of chosen company (if 'client_id' not specified in request then will be generated cdrs for all TPs).
start_date	N/A	False	String(allow_blank: False, regex: "^(\d{4})-(\d{2})-(\d{2})((T) (O))(\d{2}):(\d{2}):(\d{2})\$")	request start date in format "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-ddhh:mm:ss"

3. Appendix

3.1 Regular expressions abbreviations

- **IP_REGEX** - `"^([\d]{1,3}).([\d]{1,3}).([\d]{1,3}).([\d]{1,3})/([\d]{1,2}){0,1}$"`
- **DATE_REGEX** - `"^(\d{4})-(\d{2})-(\d{2})((T)|(O))(\d{2}):(\d{2}):(\d{2})$"`
- **DT_REGEX** - `"^(\d{4})-(\d{2})-(\d{2})(T|" ")(\d{2}):(\d{2}):(\d{2})$"`
- **TIME_CHECK_REGEX_PATTERN** - `"^([0-1]\d|2[0-3])(\:([0-5]\d)){2}((\-(0\d|1[0-2])|+(0\d|1[0-4]))(\:([0-5]\d))?)?$"`
- **TZ_REGEX** - `"^([+-]?\d{1,2}):(\d{2})$"`
- **B_NUMBER_REGEX** - `"^[0-9#*]+$"`
- **RELATION_REGEX** - `"(AND NOT)|(OR NOT)|(AND)|(OR)"`
- **TIME_CHECK_REGEX_PATTERN** - `"^([0-1]\d|2[0-3])(\:([0-5]\d)){2}((\-(0\d|1[0-2])|+(0\d|1[0-4]))(\:([0-5]\d))?)?$"`
- **type_regex** = `"(individual)|(origination_ip)|(disconnect_code)|(hour)|(day)"`

3.2 Validator abbreviations

- **Ani_Dnis_Regexp** - String for Ani/Dnis expression validation
- **gte** - Greater than or equal
- **lte** - Less than or equal
- **lt** - Less than
- **gt** - Greater than
- **max_length/max_len** - Maximal length of String or Array
- **min_length/min_len** - Minimal length of String or Array
- **allow_blank** - if True empty String is allowed, if absent - not allowed
- **regex** - validation by regular expression