Cbonds

CBONDS API AND DATA FEED

Developer Guide
01.06.2022

Table of contents

Introduction

About the service

Cbonds Database offers the transfer of data in a structured form through interaction with a web server for the purpose of operational processing of information by client software.

Purpose of the document

This technical documentation has been prepared as part of a project to develop and implement a system for providing operational data on issuers, parameters of issues (stocks, bonds, eurobonds), parameters of funds (mutual funds, ETFs), credit performance, indices, reports and quotes for client's internal systems and in accordance with the terms of reference for the development of the system.

Definitions

A **web service** is a software system identified by a URL string. A description of this software system can be found by other software systems that can interact with it according to this description through messages transmitted via Internet protocols. Cbonds Database web service interfaces are defined in XML and JSON.

WSDL (eng. Web Services Description Language) is a language for describing web services based on XML. The latest official specification at the time of writing the article is version 2.0 (WSDL Version 2.0 dated March 27, 2006), which has the status of recommendation candidate, and version 1.1 (WSDL Version 1.1 dated March 15, 2001), which has the status of recommendation.

SOAP (Simple Object Access Protocol) is an XML-based messaging protocol.

XML is an extensible markup language for storing and transmitting structured data.

JSON (eng. JavaScript Object Notation) is a text-based data exchange format based on JavaScript.

Web service operation, web service method (hereinafter referred to as Operation, method) is a web service function that returns data to the client in WSDL+SOAP or JSON format.

User is a client of the Cbonds Database service, having a login and password for authorization and accessing a specific data set.

Getting data from a web service

User authorization

In order to prevent unauthorized access to data for all Operations described below, you must specify the username/password assigned to the service for the user.

To get acquainted with the technical aspect of the web service, you can use the test username and password (the username/password of the test user):

Login: Test

Password: Test

The test user has access to a demo (limited) data set, and in no case claims to display the completeness of the database.

Getting the service scheme

An XSD/XML Schema or JSON schema can be obtained by a GET request with the user's username and password specified in the address:

Scheme	Request address
XSD/XML Schema	https://ws.cbonds.info/services/wsdl?login=LOGIN&password=PASSWORD
JSON-scheme	https://ws.cbonds.info/services/json?login=LOGIN&password=PASSWORD

You can also get the scheme using the POST method by specifying the username and password in the body of the POST request (not in the address).

Data request

Interaction with a web service for receiving data is carried out by sending POST queries of the HTTP protocol. Depending on your preferred response language, you should send queries to the following addresses:

Data format	Request address
WSDL+SOAP	https://ws.cbonds.info/services/wsdl/(operation name) *
JSON	https://ws.cbonds.info/services/json/(operation name) *

*for example: https://ws.cbonds.info/services/wsdl/get_emissions
For the operation list, see Operation reference

The request body contains the request parameters. Request body structure in JSON format:

```
{
   "auth":{"login":"LOGIN","password":"PASSWORD"},
   "filters":[{"field":"…","operator":"…","value":…}, {}],
   "quantity":{"limit":10,"offset":0},
   "sorting":[{"field":"…","order":"…"}],
   "fields":[{"field": "…"}, {"field": "…"}]
}
```

Where the main request parameters are:

"auth" - sends the user's login and password;

"filters" - contains the specified selection conditions, the fewer filters are used in the request, the better;

"quantity" - transfers parameters {"limit":..., "offset":...}, where **"limit"** is the number of records per page for a single request; **"offset"** is the page number. It must be a multiple of the limit, with a limit of 1000 offset can be 0, 1000, 2000, etc.

"sorting" - sends a list of fields for sorting;

"fields" - sends a list of fields to display in response;

Request body structure in WSDL+SOAP format:

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:RequestMessage_emissions xmlns:m="https://ws.cbonds.info/">
      <m:AuthData>
        <m:Login><![CDATA[LOGIN]]></m:Login>
        <m:Password><![CDATA[PASSWORD]]></m:Password>
      </m:AuthData>
      <m:Filters>
        <m:NumericFilter>
          <m:FilterField>….</m:FilterField>
          <m:FilterOperator>…</m:FilterOperator>
          <m:FilterValue><![CDATA[…]]></m:FilterValue>
        </m:NumericFilter>
      </m:Filters>
      <m:QuantityData>
        <m:Limit>10</m:Limit>
        <m:Offset>0</m:Offset>
      </m:QuantityData>
      <m:SortRules>
        <m:SortRule>
          <m:SortField></m:SortField>
          <m:SortOrder>…</m:SortOrder>
        </m:SortRule>
      </m:SortRules>
      <m:ResponseItemFields>
        <m:ResponseItemField>
          <m:ItemField>…</m:ItemField>
        </m:ResponseItemField>
        <m:ResponseItemField>
          <m:ItemField>emission_id</m:ItemField>
        </m:ResponseItemField>
      </m:ResponseItemFields>
    </m:RequestMessage_emissions>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Key response parameters

Response structure in JSON format:

```
{
   "count":4,
   "total":4,
   "limit":10,
   "offset":0,
   "exec_time":0.4443,
   "items":[...],
   "meta":{ "cms_full_gen_time":0.4444, "user_id":33 }
}
```

Where response parameters are:

"count" - the number of records found for the current page (for the last page, it may not be equal to limit);

"total" - the number of records found by limits (transferred in the filters field of the request). To build pagination (a sequential request plan for selecting all the data in parts), we focus on the given field in the response (divide it by the limit and get the last page number in the whole part, if we count from zero). For example, if the limit is 1000 and the value total=2500, you will need to perform three queries with offsets 0*1000=0, 1*1000=1000, 2*1000=2000; "limit" takes the same value as at the request;

"limit" takes the same value as at the request;

"offset" - the offset value that is not a multiple of the limit is reduced to a multiple to the lower side;

"exec_time" - this parameter is required for optimization. If there are problems with the performance of the service (and there are opportunities), you can organize logging of this parameter for each operation, in the ideal case - statistics with aggregated values for the operation, filters, offset, etc.

"items" - sends all records for the corresponding page;

"cms_full_gen_time"- is used for optimization;

"user_id" – your user code in our system, for the test user user_id = 33.

Response structure in WSDL+SOAP format:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="https://ws.cbonds.info/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<ns1:ResponseMessage_emitents>
<ns1:Count>4</ns1:Count>
<ns1:Total>4</ns1:Total>
<ns1:Limit>10</ns1:Limit>
<ns1:Offset>0</ns1:Offset>
<ns1:ExecTime>0.4139</ns1:ExecTime>
<ns1:Items><ns1:Item>
...
</ns1:Item></ns1:Items>
</ns1:ResponseMessage_emitents>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<!-- user_id: 33 -->
<!-- cms_full_gen_time: 0.4145 -->
<!-- lang: rus -->
<!-- strict_mode: 1 -->
<!-- performLogging: -1 -->
```

Data request using filters

When requesting data, you can create filters selecting only certain information that meets the specified criteria from all the data. **The list of fields for which a filter can be created is limited.**

List of possible operators for data filtering:

Operator	Decoding the operator	Operator value
'eq'	'equal'	**Equal**
'ne'	'not equal'	**Not equal**
'lt'	'less'	**Strictly less**
'le'	'less or equal'	**Less or equal**
'gt'	'greater'	**Strictly greater**
'ge'	'greater or equal'	**Greater or equal**
'in'	'is in'	**Included in the busload**
'ni'	'is not in'	**Not included in the busload**
'bw'	'begins with'	**Begins with (characters)**
'bn'	'does not begin with'	**Does not begin with (characters)**
'ew'	'ends with'	**Ends with (characters)**
'en'	'does not end with'	**Does not end with (characters)**
'cn'	'contains'	**Contains part of the string**
'nc'	'does not contain'	**Does not contain part of the string**
'nu'	'is null',	**Has a NULL value**
'nn'	'is not null'	**Not NULL**

Examples of limits that can be applied to the "get_emissions" operation (emission parameters). The request will be sent to the address https://ws.cbonds.info/services/json/get_emissions/

1. Emission selection with a specific ISIN, e.g. RU000A0AB5S7.

Request in JSON format:

```
{
"auth":{"login":"Test","password":"Test"},
"filters":[{"field":"isin_code","operator":"eq","value":"RU000A0AB5S7"}],
"quantity":{"limit":10,"offset":0},
"sorting":[{"field":"","order":"asc"}]
"fields":[{"field": "id"}, {"field": "document_rus"}]
}
```

Request in WSDL+SOAP format:

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:RequestMessage_emissions
xmlns:m="https://ws.cbonds.info/">
      <m:AuthData>
        <m:Login><![CDATA[Test]]></m:Login>
        <m:Password><![CDATA[Test]]></m:Password>
      </m:AuthData>
      <m:Filters>
        <m:NumericFilter>
          <m:FilterField>isin_code</m:FilterField>
          <m:FilterOperator>eq</m:FilterOperator>

<m:FilterValue><![CDATA[RU000A0AB5S7]]></m:FilterValue>
        </m:NumericFilter>
      </m:Filters>
      <m:QuantityData>
        <m:Limit>10</m:Limit>
        <m:Offset>0</m:Offset>
      </m:QuantityData>
      <m:SortRules>
        <m:SortRule>
          <m:SortField></m:SortField>
          <m:SortOrder>asc</m:SortOrder>
        </m:SortRule>
      </m:SortRules>
      <m:ResponseItemFields>
        <m:ResponseItemField>
          <m:ItemField>…</m:ItemField>
        </m:ResponseItemField>
        <m:ResponseItemField>
          <m:ItemField>id</m:ItemField>
          <m:ItemField>document_rus</m:ItemField>
        </m:ResponseItemField>
      </m:ResponseItemFields>
    </m:RequestMessage_emissions>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Further, for brevity, we will give examples only in JSON format.

2. Selecting multiple emissions with specific ISIN, e.g. XS0157815647 and RU000A0AB5S7.

   Request in JSON format:

```
{
"auth":{"login":"Test","password":"Test"},
"filters":[{"field":"isin_code","operator":"in","value":"XS0157815647;RU000A0AB5
S7"}],
"quantity":{"limit":10,"offset":0},
"sorting":[{"field":"","order":"asc"}],
"fields":[{"field": "id"}, {"fields": "document_rus"}]
}
```

3. Selecting emissions that were updated after October 12, 2016.

   Request in JSON format:

```
{
"auth":{"login":"Test","password":"Test"},
"filters":[{"field":"updating_date","operator":"ge","value":"2016-10-12"}],
"quantity":{"limit":10,"offset":0},
"sorting":[{"field":"","order":"asc"}]
"fields":[{"field": "id"}, {"fields": "document_rus"}]

}
```

4. Selecting emissions that were updated from October 1 to 12, 2016.

   Request in JSON format:

```
{
"auth":{"login":"Test","password":"Test"},
"filters": [ {"field":"updating_date","operator":"ge","value":"2016-10-01"},
             {"field":"updating_date","operator":"lt","value":"2016-10-12"} ],
"quantity":{"limit":10,"offset":0},
"sorting":[{"field":"","order":"asc"}]
"fields":[{"field": "id"}, {"fields": "document_rus"}]
}
```

Volume of queries

Maximum number of records in the response (items in the items field)	**1000**

To select all data, you must run multiple queries by changing the **offset.** Each response contains information about the total number of records (the total field) that are available under the selection limits transferred in the request.

Limit on the frequency of queries

All operations, except for the get_tradings_realtime operation ("Trading of bonds and eurobonds with an update every 15 minutes"), are updated 1 time per hour. At the same time, changes in the data of the main operations usually occur daily, and changes in the data of operations with directories occur less than 1 time per month.

Period	Limitation
Allowed number of user requests per operation **per minute**	No more than 30 times
Allowed number of user requests per operation **per day**	No more than 10,000 times

If the limit is exceeded, the operation is not executed, but an error is spitted out in the form:

- "Error: 900000. Max requests per minute limit (30) exceeded. Try to request the next minute.",
- "Error: 900000. Max requests per day limit (10000) exceeded. Try to request next day."

List of possible errors in the web service response

1. Errors that you can get when accessing a web service:

   a) **HTTP/1.1 301 Moved Permanently** — access via http instead of https;

   b) **HTTP/1.1 403 Forbidden** — incorrect login; Invalid password; no login, no password, no authorization block;

   c) **HTTP/1.1 500 Internal Service Error** — config with syntax error; a request for an operation to which there is no access;

   d) **HTTP/1.1 504 –** response waiting time out

2. Software errors:

   a) **https protocol required** — access via http instead of https;

   b) **Max requests per minute limit max_per_minute exceeded. Try to request next minute.** — limit on the number of requests per minute;

   c) **Max requests per day limit max_per_day exceeded. Try to request next day.** — limit on the number of requests per day;

3. Messages about intercepted errors:

   a) Undefined index: xxxx

   b) Invalid field (xxxx) for filtering

   c) The request JSON string is not valid d. The collection "%services_xxxxxx%" is not available for user id "yy

Instructions for using the demo version of the web service

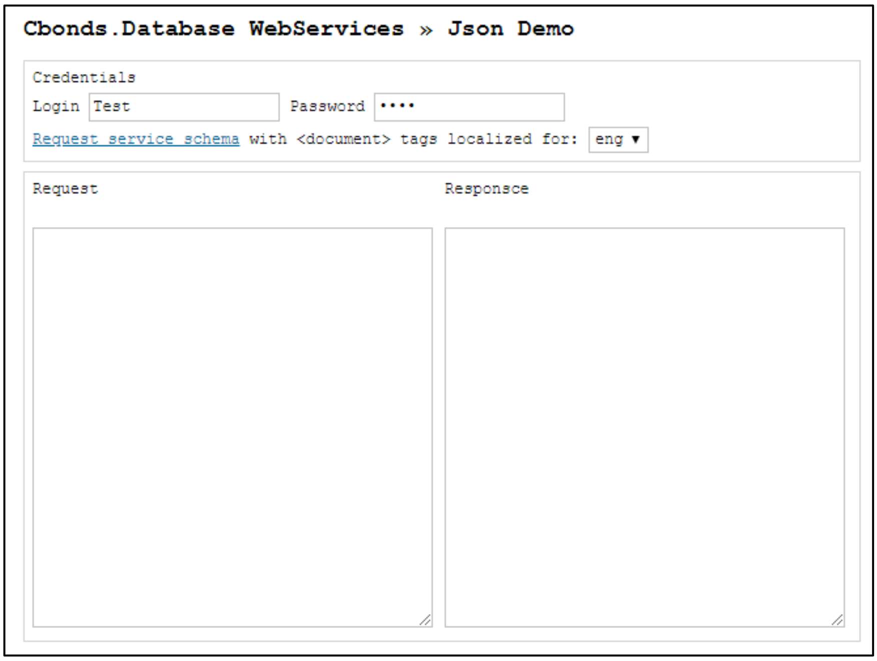
The demo version serves as an assistant for developing client applications. The tool contains the same functionality as the full version and differs only in the presence of a visual interface.

To start working with the demo version of the web service, click one of the links:

1. https://ws.cbonds.info/services/json/demo/
2. https://ws.cbonds.info/services/wsdl/demo/

Below there is a description of working only with the demo version for JSON, working with the demo version for WSDL is similar.

1. Enter username and password for test access (login: Test; password: Test) and click the "Request service schema".



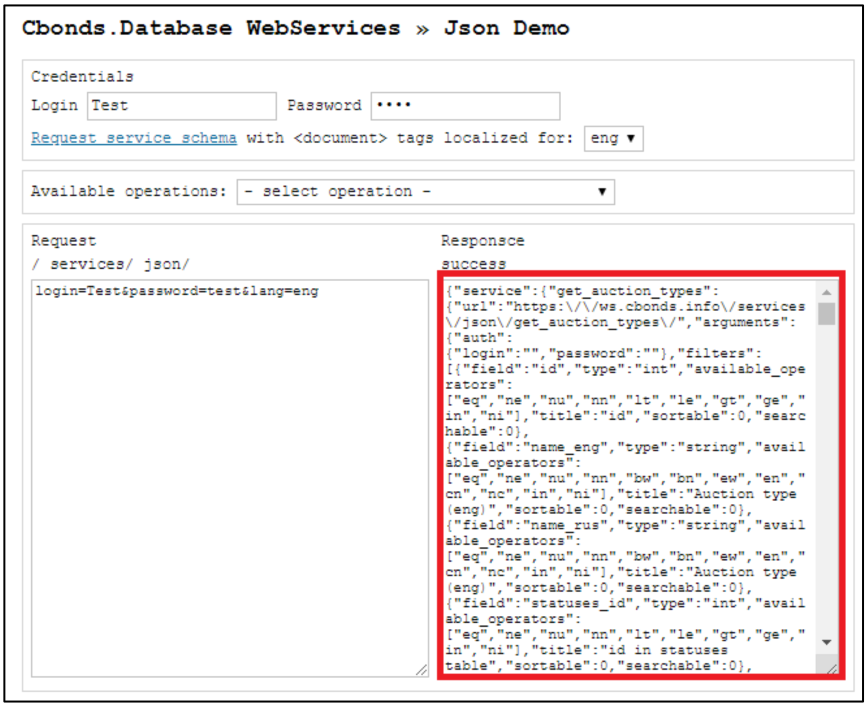
Login Test

Password ••••

Request service schema with <document> tags localized for: eng ▾

2. After requesting the service schema, the "Request" and "Response" fields will be filled in.

The "Response" field contains the service schema, in this example, a JSON schema.



Cbonds.Database WebServices » Json Demo

Credentials

Login Test Password ••••

Request service schema with <document> tags localized for: eng ▾

Available operations: - select operation - ▾

Request

/ services/ json/

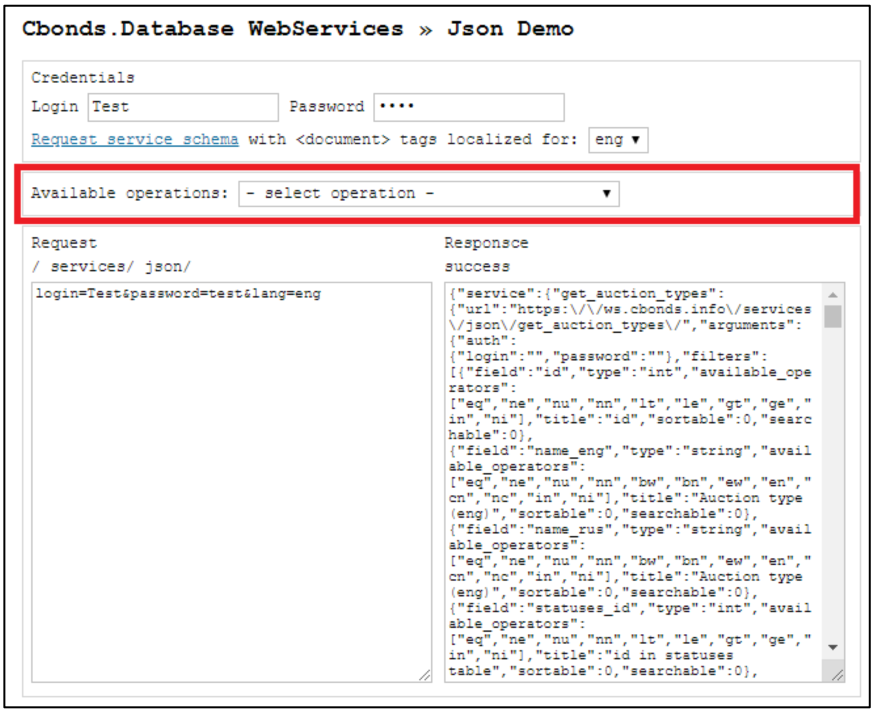
```
login=Test&password=test&lang=eng
```

Responsce

success

```
{"service":{"get_auction_types":
{"url":"https:\/\/ws.cbonds.info\/services
\/json\/get_auction_types\/","arguments":
{"auth":
{"login":"","password":""},"filters":
[{"field":"id","type":"int","available_ope
rators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id","sortable":0,"searc
hable":0},
{"field":"name_eng","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"name_rus","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"statuses_id","type":"int","avail
able_operators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id in statuses
table","sortable":0,"searchable":0},
```

3. Select one of the available operations from the "Available operation" drop-down list. For example, let's select the "get_emissions" operation.



Cbonds.Database WebServices » Json Demo

Credentials

Login Test Password ••••

Request service schema with <document> tags localized for: eng ▾

Available operations: - select operation - ▾

Request

/ services/ json/

```
login=Test&password=test&lang=eng
```

Responsce

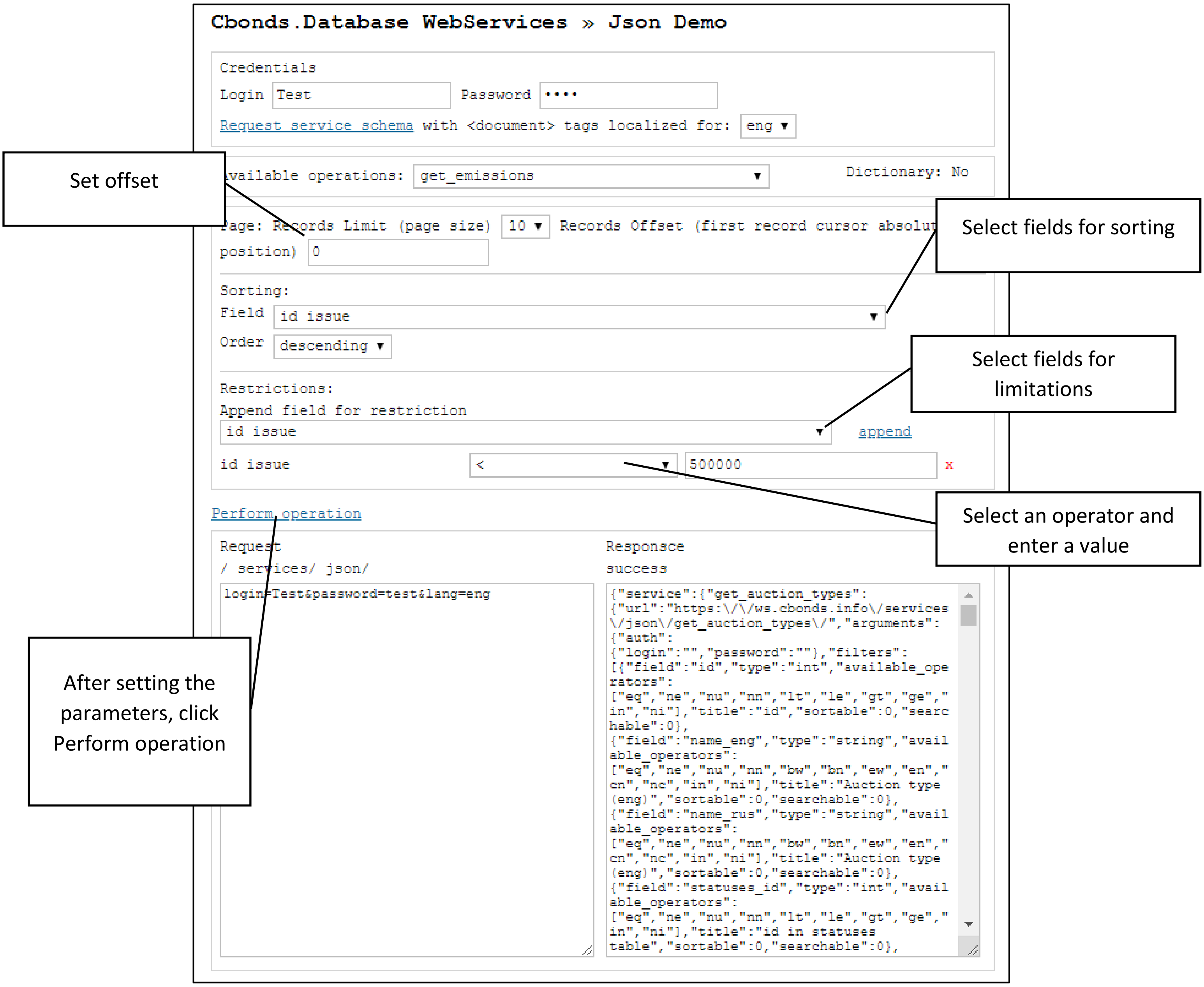
success

```
{"service":{"get_auction_types":
{"url":"https:\/\/ws.cbonds.info\/services
\/json\/get_auction_types\/","arguments":
{"auth":
{"login":"","password":""},"filters":
[{"field":"id","type":"int","available_ope
rators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id","sortable":0,"searc
hable":0},
{"field":"name_eng","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"name_rus","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"statuses_id","type":"int","avail
able_operators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id in statuses
table","sortable":0,"searchable":0},
```

4. After selecting the Operation, the following request parameter settings will be available:

   1. selecting the number of records to send;
   2. sorting by received responses;
   3. limitations (filters) on the data to be sent.

   These options can be applied to all operations except directory operations. To work with filters, select a field, click the "append" button, select an operator and a value.



Cbonds.Database WebServices » Json Demo

Credentials

Login Test Password ••••

Request service schema with <document> tags localized for: eng ▾

Available operations: get_emissions ▾ Dictionary: No

Page: Records Limit (page size) 10 ▾ Records Offset (first record cursor absolute position) 0

Sorting:

Field id issue ▾

Order descending ▾

Restrictions:

Append field for restriction

id issue ▾ append

id issue < ▾ 500000 x

Perform operation

Request

/ services/ json/

```
login=Test&password=test&lang=eng
```

Responsce

success

{"service":{"get_auction_types":
{"url":"https:\/\/ws.cbonds.info\/services
\/json\/get_auction_types\/","arguments":
{"auth":
{"login":"","password":""},"filters":
[{"field":"id","type":"int","available_ope
rators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id","sortable":0,"searc
hable":0},
{"field":"name_eng","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"name_rus","type":"string","avail
able_operators":
["eq","ne","nu","nn","bw","bn","ew","en","
cn","nc","in","ni"],"title":"Auction type
(eng)","sortable":0,"searchable":0},
{"field":"statuses_id","type":"int","avail
able_operators":
["eq","ne","nu","nn","lt","le","gt","ge","
in","ni"],"title":"id in statuses
table","sortable":0,"searchable":0},

Set offset

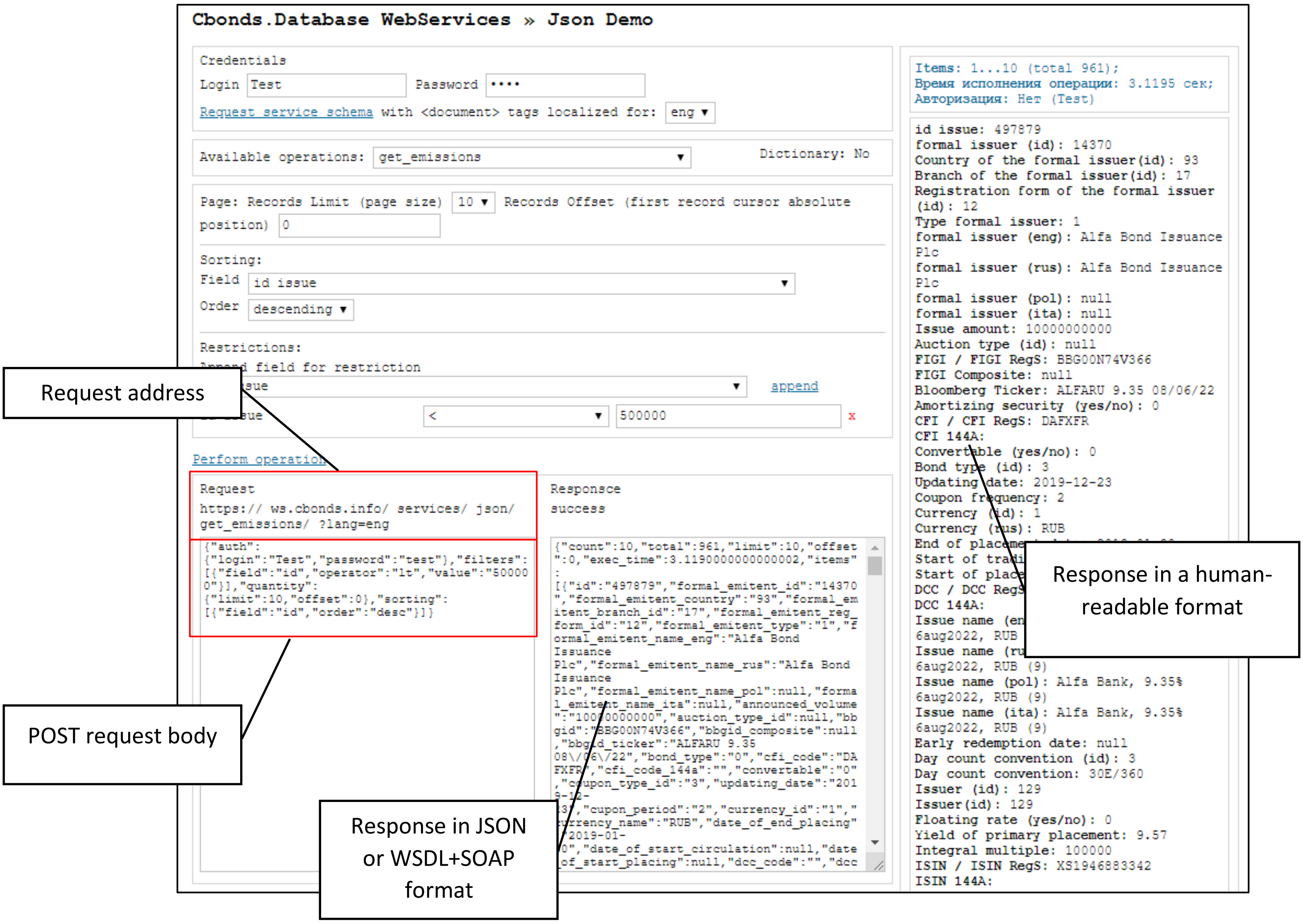
Select fields for sorting

Select fields for limitations

Select an operator and enter a value

After setting the parameters, click Perform operation

5. After the request is completed, the body of the POST request is automatically generated in the "Request" field. The "Response" field will display the response in the appropriate format.



Cbonds.Database WebServices » Json Demo

Credentials

Login: Test   Password: ••••

Request service schema with <document> tags localized for: eng ▾

Available operations: get_emissions ▾   Dictionary: No

Page: Records Limit (page size) 10 ▾ Records Offset (first record cursor absolute position) 0

Sorting:

Field: id issue ▾

Order: descending ▾

Restrictions:

field for restriction

sue ▾   append

ue   < ▾   500000   x

Perform operation

Request

https:// ws.cbonds.info/ services/ json/ get_emissions/ ?lang=eng

```
{"auth":
{"login":"Test","password":"test"},"filters":
[{"field":"id","operator":"lt","value":"50000
0"}],"quantity":
{"limit":10,"offset":0},"sorting":
[{"field":"id","order":"desc"}]}
```

Responsce

success

```
{"count":10,"total":961,"limit":10,"offset
":0,"exec_time":3.1190000000000002,"items"
:
[{"id":"497879","formal_emitent_id":"14370
","formal_emitent_country":"93","formal_em
itent_branch_id":"17","formal_emitent_reg_
form_id":"12","formal_emitent_type":"1","f
ormal_emitent_name_eng":"Alfa Bond
Issuance
Plc","formal_emitent_name_rus":"Alfa Bond
Issuance
Plc","formal_emitent_name_pol":null,"forma
l_emitent_name_ita":null,"announced_volume
":"10000000000","auction_type_id":null,"bb
gid":"BBG00N74V366","bbgid_composite":null
,"bbgid_ticker":"ALFARU 9.35
08\/06\/22","bond_type":"0","cfi_code":"DA
FXFR","cfi_code_144a":"","convertable":"0"
,"coupon_type_id":"3","updating_date":"201
9-12-
23","cupon_period":"2","currency_id":"1","
currency_name":"RUB","date_of_end_placing"
:"2019-01-
0","date_of_start_circulation":null,"date
_of_start_placing":null,"dcc_code":"","dcc
```

Items: 1...10 (total 961);
Время исполнения операции: 3.1195 сек;
Авторизация: Нет (Test)

```
id issue: 497879
formal issuer (id): 14370
Country of the formal issuer(id): 93
Branch of the formal issuer(id): 17
Registration form of the formal issuer
(id): 12
Type formal issuer: 1
formal issuer (eng): Alfa Bond Issuance
Plc
formal issuer (rus): Alfa Bond Issuance
Plc
formal issuer (pol): null
formal issuer (ita): null
Issue amount: 10000000000
Auction type (id): null
FIGI / FIGI RegS: BBG00N74V366
FIGI Composite: null
Bloomberg Ticker: ALFARU 9.35 08/06/22
Amortizing security (yes/no): 0
CFI / CFI RegS: DAFXFR
CFI 144A:
Convertable (yes/no): 0
Bond type (id): 3
Updating date: 2019-12-23
Coupon frequency: 2
Currency (id): 1
Currency (rus): RUB
End of placeme
Start of tradi
Start of place
DCC / DCC RegS
DCC 144A:
Issue name (en
6aug2022, RUB
Issue name (ru
6aug2022, RUB (9)
Issue name (pol): Alfa Bank, 9.35%
6aug2022, RUB (9)
Issue name (ita): Alfa Bank, 9.35%
6aug2022, RUB (9)
Early redemption date: null
Day count convention (id): 3
Day count convention: 30E/360
Issuer (id): 129
Issuer(id): 129
Floating rate (yes/no): 0
Yield of primary placement: 9.57
Integral multiple: 100000
ISIN / ISIN RegS: XS1946883342
ISIN 144A:
```

Request address

POST request body

Response in JSON or WSDL+SOAP format

Response in a human-readable format

Thus, when developing a product, you can consult the data returned by the web service.

Example of accessing a web service in Python 3.x

Loading and updating the emissions table

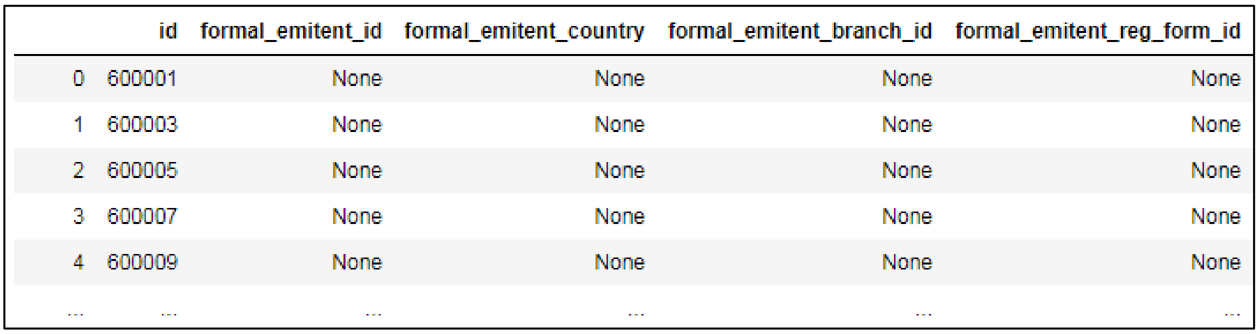
Below there is an example of Python 3 code.x performing:

1. Data request with filtering,
2. Multiple data requests to download the full table,
3. Updating records in a loaded table,
4. Recording a table in Excel.

```
# library for making requests to a web service
import requests
# library for working with data
import pandas as pd
# connect a method that converts a JSON string to a table
from pandas.io.json import json_normalize
# library for delay between requests
import time

# set request parameters
# limit - number of records in the response
# offset - response page multiple of limit, i.e. 0.1000, 2000, 3000..
limit = 1000
offset = 0

# set the address of the request where we specify the get_emissions operation
get_emissions_url = 'https://ws.cbonds.info/services/json/get_emissions/'

# define a function that returns the body of a POST request with the passed request parameters limit,
offset
def data(limit,offset):
    return({"auth":{"login":"Test","password":"Test"},
            "filters":[{"field":"id","operator":"ge","value":600000}],
            "quantity":{"limit":limit,"offset":offset},"sorting":[{"field":"id","order":"asc"}]})

# make a POST request by passing the request address and request body by the post method
response = requests.post(get_emissions_url, json=data(limit,offset)).json()

# convert the JSON response into a table
df = json_normalize(response)
```

```
# write down the total number of strings we need to request
total = df['total'][0]


# define an object of the pandas class, create the emissions table
emissions = pd.DataFrame()

# create a loop for accessing the web service many times in order to download all records
# change the offset from 0 to total by a multiple of limit
for offset in range(0,total,limit):

     # make a delay between requests since the maximum number of requests per minute is 30
    time.sleep(2.1)

    # make a request
    response = requests.post(get_emissions_url, json=data(limit,offset)).json()

    # convert the response into a table
    tmp = json_normalize(response)

   # get a JSON string with an array of data from the 'items' field and convert this array into a table
    tmp2 = json_normalize(tmp['items'][0])

    # add the received records to the emissions table
    emissions = pd.concat([emissions,tmp2], ignore_index=True)

# store the last update time
maxdate = emissions.agg({'update_time': ['max']})

# --- start of the update table download fragment ---
offset = 0
date = maxdate['update_time'][0]

# define a function that returns the body of a POST request with the passed request parameters limit,
offset, date
def bodyupdate(date,limit,offset):
    return({"auth":{"login":"Test","password":"Test"},
                "filters":[{"field":"id","operator":"ge","value":600000},{"field":"update_time","operator":"ge","value
                ":date}], "quantity":{"limit":limit,"offset":offset},"sorting":[{"field":"","order":"asc"}]})

response = requests.post(get_emissions_url, json=bodyupdate(date,limit,offset)).json()
df = json_normalize(response)
total = df['total'][0]

upddatatable = pd.DataFrame()

for offset in range(0,total,limit):
    time.sleep(2.1)
    response = requests.post(get_emissions_url, json=bodyupdate(date,limit,offset)).json()
    tmp = json_normalize(response)
    tmp2 = json_normalize(tmp['items'][0])
    upddatatable = pd.concat([upddatatable,tmp2], ignore_index=True)
# --- end of the update table download fragment ---
# delete empty records from the table
emissions = emissions.drop(emissions.loc[emissions.id.isin(upddatatable.id)].index)

# add updated records to the table
emissions = pd.concat([emissions, upddatatable], ignore_index=True)
print(emissions)
# record the result in Excel
emissions.to_excel("output.xlsx")
```

Output:

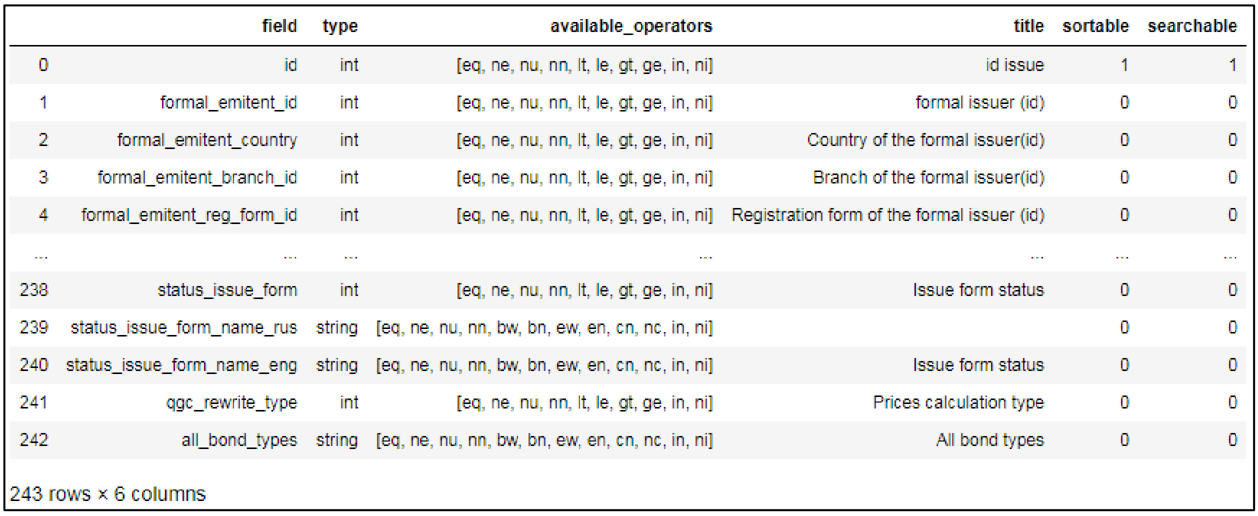


	id	formal_emitent_id	formal_emitent_country	formal_emitent_branch_id	formal_emitent_reg_form_id
0	600001	None	None	None	None
1	600003	None	None	None	None
2	600005	None	None	None	None
3	600007	None	None	None	None
4	600009	None	None	None	None
...	...	...	...	...	...

Downloading a JSON schema and renaming fields

```
url = 'https://ws.cbonds.info/services/json?login=Test&password=Test'
# url = 'https://ws.cbonds.info/services/json?login=Test&password=Test&lang=rus' - in Russian

# request a JSON schema describing all available web service operations
response = requests.get(url).json()
json_scheme = json_normalize(response)

# get a description of the get emissions operation
emissions_scheme = json_normalize(json_scheme['service.get_emissions.arguments.filters'][0])

# description output
print(emissions_scheme)
```

Output:



	field	type	available_operators	title	sortable	searchable
0	id	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	id issue	1	1
1	formal_emitent_id	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	formal issuer (id)	0	0
2	formal_emitent_country	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	Country of the formal issuer(id)	0	0
3	formal_emitent_branch_id	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	Branch of the formal issuer(id)	0	0
4	formal_emitent_reg_form_id	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	Registration form of the formal issuer (id)	0	0
...	...	...	...	...	...	...
238	status_issue_form	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	Issue form status	0	0
239	status_issue_form_name_rus	string	[eq, ne, nu, nn, bw, bn, ew, en, cn, nc, in, ni]		0	0
240	status_issue_form_name_eng	string	[eq, ne, nu, nn, bw, bn, ew, en, cn, nc, in, ni]	Issue form status	0	0
241	qgc_rewrite_type	int	[eq, ne, nu, nn, lt, le, gt, ge, in, ni]	Prices calculation type	0	0
242	all_bond_types	string	[eq, ne, nu, nn, bw, bn, ew, en, cn, nc, in, ni]	All bond types	0	0

243 rows × 6 columns

field - contains a list of fields in the get_emissions response.

type – indicates the data type.

available_operators – operators available for filtering data.

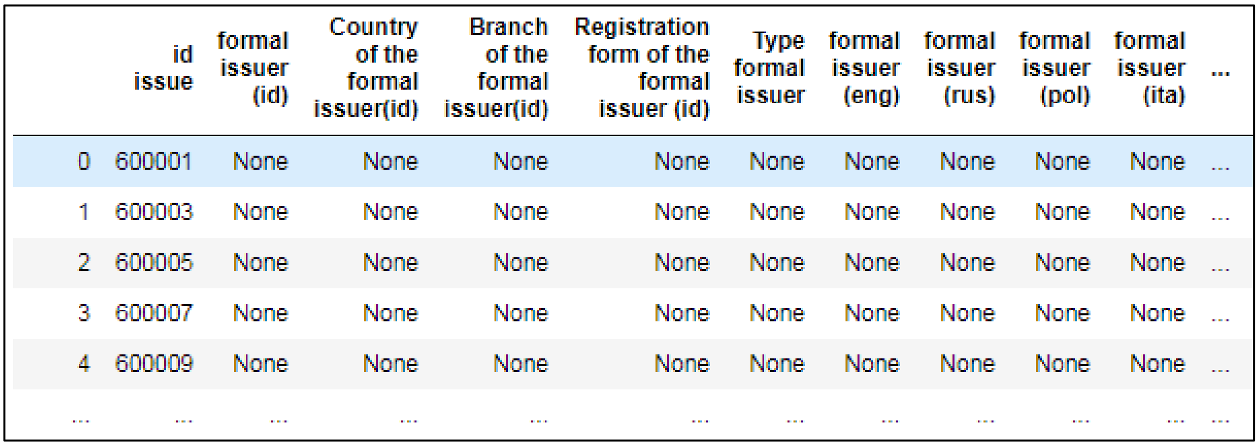
title - contains a list of field names.

sortable - indicates the possibility of sorting by field (1 - filtration is possible, 0 - not possible).

searchable – indicates the possibility of filtering by field (1 - filtration is possible, 0 - not possible).

```
# rename fields to field names
for i in range(len(emissions_scheme)):
   emissions = emissions.rename(columns={ emissions.columns[i]: emissions_scheme ['title'][i]})
print(emissions)
```

Output:



	id issue	formal issuer (id)	Country of the formal issuer(id)	Branch of the formal issuer(id)	Registration form of the formal issuer (id)	Type formal issuer	formal issuer (eng)	formal issuer (rus)	formal issuer (pol)	formal issuer (ita)	...
0	600001	None	None	None	None	None	None	None	None	None	...
1	600003	None	None	None	None	None	None	None	None	None	...
2	600005	None	None	None	None	None	None	None	None	None	...
3	600007	None	None	None	None	None	None	None	None	None	...
4	600009	None	None	None	None	None	None	None	None	None	...
...	...	...	...	...	...	...	...	...	...	...	...

List of operations

For each web service operation, the list of available, filtered, and sorted fields is limited. All available operation fields are divided into basic and additional fields. The basic operation fields are available to all users when connecting to the web service. If necessary, a number of additional fields can be added to the list of available fields for web service operations. The list of fields that can be searched or sorted can also be expanded if necessary.

Basic operations

Lists of the Central Bank of the Russian Federation - get_cbrf_listing

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_cbrf_listing/

https://ws.cbonds.info/services/wsdl/get_cbrf_listing/
https://ws.cbonds.info/services/json/get_cbrf_listing/

Lombard list of CBR bonds and eurobonds - get_cbrf_listing_lombard

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_cbrf_listing_lombard/

https://ws.cbonds.info/services/wsdl/get_cbrf_listing_lombard/
https://ws.cbonds.info/services/json/get_cbrf_listing_lombard/

Emission parameters - get emissions
https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emissions/
https://ws.cbonds.info/services/wsdl/get_emissions/
https://ws.cbonds.info/services/json/get_emissions/

Emission defaults - get_emission_default

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emission_default/

https://ws.cbonds.info/services/wsdl/get_emission_default/
https://ws.cbonds.info/services/json/get_emission_default/

Quotation lists of emissions -get_emission_quote_listings

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emission_quote_listings/

https://ws.cbonds.info/services/wsdl/get_emission_quote_listings/
https://ws.cbonds.info/services/json/get_emission_quote_listings/

Emissions ratings - get_emission_ratings

https://cbonds.com/api/catalog/API_Creditratings/get_emission_ratings/

https://ws.cbonds.info/services/wsdl/get_emission_ratings/
https://ws.cbonds.info/services/json/get_emission_ratings/

Additional emission parameters - get_emission_tap_issues

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emission_tap_issues/

https://ws.cbonds.info/services/wsdl/get_emission_tap_issues/
https://ws.cbonds.info/services/json/get_emission_tap_issues/

Emitents'parameters - get_emitents
https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_emitents/
https://ws.cbonds.info/services/wsdl/get_emitents/
https://ws.cbonds.info/services/json/get_emitents/

Emitents' ratings - get_emitent_ratings

https://cbonds.com/api/catalog/API_Creditratings/get_emitent_ratings/

https://ws.cbonds.info/services/wsdl/get_emitent_ratings/
https://ws.cbonds.info/services/json/get_emitent_ratings/

Cash flow - get_flow

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_flow/

https://ws.cbonds.info/services/wsdl/get_flow/
https://ws.cbonds.info/services/json/get_flow/

Guarantors and sureties - get_emission_guarantors

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emission_guarantors/

https://ws.cbonds.info/services/wsdl/get_ emission_guarantors/
https://ws.cbonds.info/services/json/get_ emission_guarantors/

Mutual funds and funds - get funds

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds/

https://ws.cbonds.info/services/wsdl/get_funds/
https://ws.cbonds.info/services/json/get_funds/

Mutual funds and funds auditors - get_funds_auditors

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_auditors/

https://ws.cbonds.info/services/wsdl/get_funds_auditors/
https://ws.cbonds.info/services/json/get_funds_auditors/

Managing companies - get_funds_companies

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_companies/

https://ws.cbonds.info/services/wsdl/get_funds_companies/
https://ws.cbonds.info/services/json/get_funds_companies/

Registrars of mututal funds and funds - get_funds_registrants

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_registrants/

https://ws.cbonds.info/services/wsdl/get_funds_registrants/
https://ws.cbonds.info/services/json/get_funds_registrants/

Repositories of mutual funds and funds - get_funds_repositories

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_repositories/

https://ws.cbonds.info/services/wsdl/get_funds_repositories/
https://ws.cbonds.info/services/json/get_funds_repositories/

Index portfolio structure - get_index_content

https://cbonds.com/api/catalog/API_CbondsIndex/get_index_content/

https://ws.cbonds.info/services/wsdl/get_index_content/
https://ws.cbonds.info/services/json/get_index_content/

Index values - get_index_value

https://cbonds.com/api/catalog/API_CbondsIndex/get_index_value/

https://ws.cbonds.info/services/wsdl/get_index_value/
https://ws.cbonds.info/services/json/get_index_value/

Offer parameters - get_offert

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_offert/

https://ws.cbonds.info/services/wsdl/get_offert/
https://ws.cbonds.info/services/json/get_offert/

Participation of investment banks in emissions - get_participation

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_participation/

https://ws.cbonds.info/services/wsdl/get_participation/
https://ws.cbonds.info/services/json/get_participation/

Mutual Fund quotes - get_pif_quotes

https://cbonds.com/api/catalog/API_CbondsFunds/get_pif_quotes/

https://ws.cbonds.info/services/wsdl/get_pif_quotes/
https://ws.cbonds.info/services/json/get_pif_quotes/

Stock parameters - get stocks_full

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_stocks_full/

https://ws.cbonds.info/services/wsdl/get_stocks_full/
https://ws.cbonds.info/services/json/get_stocks_full/

Shares' dividends (actual version) - get_stocks_dividends_v2

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_stocks_dividends_v2/

https://ws.cbonds.info/services/wsdl/get_stocks_dividends_v2/
https://ws.cbonds.info/services/json/get_stocks_dividends_v2/

Trading of bonds and eurobonds - get_tradings (40 days snapshot)

https://cbonds.com/api/catalog/API_CbondsQuotes/get_tradings/

https://ws.cbonds.info/services/wsdl/get_tradings/
https://ws.cbonds.info/services/json/get_tradings/

Trading of bonds and eurobonds on the Moscow exchange with an update every 15 minutes - get_tradings_realtime

https://cbonds.com/api/catalog/API_CbondsQuotes/get_tradings_realtime/

https://ws.cbonds.info/services/wsdl/get_tradings_realtime/
https://ws.cbonds.info/services/json/get_tradings_realtime/

Quotes from market participants - get_mpquotes

https://cbonds.com/api/catalog/API_CbondsQuotes/get_mpquotes/

https://ws.cbonds.info/services/wsdl/get_mpquotes/
https://ws.cbonds.info/services/json/get_mpquotes/

Shares trading - get_tradings_stocks (40 days snapshot)

https://cbonds.com/api/catalog/API_CbondsQuotes/get_tradings_stocks/

https://ws.cbonds.info/services/wsdl/get_tradings_stocks/
https://ws.cbonds.info/services/json/get_tradings_stocks/

ACI data as of the date (without trading) - get_aci

https://cbonds.ru/api/catalog/API_CbondsDescriptiveData/get_aci/

https://ws.cbonds.info/services/wsdl/get_aci/
https://ws.cbonds.info/services/json/get_aci/

Index structure - get_cbonds_indexes_portfolios

https://cbonds.com/api/catalog/API_CbondsIndex/get_cbonds_indexes_portfolios/

https://ws.cbonds.info/services/wsdl/get_cbonds_indexes_portfolios/
https://ws.cbonds.info/services/json/get_cbonds_indexes_portfolios/

Covenants of emissions - get_covenants_emissions

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_covenants_emissions/

https://ws.cbonds.info/services/wsdl/get_covenants_emissions /
https://ws.cbonds.info/services/json/get_covenants_emissions /

Emitents' capitalization - get_ emitent_stocks_capitalization

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emitent_stocks_capitalization/

https://ws.cbonds.info/services/wsdl/get_emitent_stocks_capitalization/
https://ws.cbonds.info/services/json/get_emitent_stocks_capitalization/

IFRS indicators (financial sector) - get_report_msfo_finance

https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_report_msfo_finance/

https://ws.cbonds.info/services/wsdl/get_report_msfo_finance/
https://ws.cbonds.info/services/json/get_report_msfo_finance/

IFRS indicators (real sector) - get_report_msfo_real

https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_report_msfo_real/

https://ws.cbonds.info/services/wsdl/get_report_msfo_real/
https://ws.cbonds.info/services/json/get_report_msfo_real/

RAS balance sheet - get_report_rsbu_balance

https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_report_rsbu_balance/

https://ws.cbonds.info/services/wsdl/get_report_rsbu_balance/
https://ws.cbonds.info/services/json/get_report_rsbu_balance/

RAS Report on the financial results of the company - get_report_rsbu_profit

https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_report_rsbu_profit/

https://ws.cbonds.info/services/wsdl/get_report_rsbu_profit/
https://ws.cbonds.info/services/json/get_report_rsbu_profit/

Cash flow report – get_report_cash_flow_statement_newform

https://cbonds.com/api/catalog/API_CbondsFinancialReports/get_report_cash_flow_statement_newform/

https://ws.cbonds.info/services/wsdl/get_report_cash_flow_statement_newform/
https://ws.cbonds.info/services/json/get_report_cash_flow_statement_newform/

Auctions - get auctions

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_auctions/

https://ws.cbonds.info/services/wsdl/get_auctions/
https://ws.cbonds.info/services/json/get_auctions/

Quote classes of shares - get_etf_share_classes_quotes

https://cbonds.com/api/catalog/API_CbondsFunds/get_etf_share_classes_quotes/

https://ws.cbonds.info/services/wsdl/get_etf_share_classes_quotes/
https://ws.cbonds.info/services/json/get_etf_share_classes_quotes/

Parameters of ETF funds -get_etf_funds

https://cbonds.com/api/catalog/API_CbondsFunds/get_etf_funds/

https://ws.cbonds.info/services/wsdl/get_etf_funds/
https://ws.cbonds.info/services/json/get_etf_funds/

The structure of the Fund's assets - get_funds_property_assets

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_property_assets/

https://ws.cbonds.info/services/wsdl/get_funds_property_assets/
https://ws.cbonds.info/services/json/get_funds_property_assets/

Information on basic assets for structured products - get_emission_underlying_assets

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_emission_underlying_assets/

https://ws.cbonds.info/services/wsdl/get_emission_underlying_assets/
https://ws.cbonds.info/services/json/get_emission_underlying_assets/

Current ratings for issuers - get_rating_emitent_maxdate

https://cbonds.com/api/catalog/API_Creditratings/get_rating_emitent_maxdate/

https://ws.cbonds.info/services/wsdl/get_rating_emitent_maxdate/
https://ws.cbonds.info/services/json/get_rating_emitent_maxdate/

Current ratings for emissions - get_rating_emission_maxdate

https://cbonds.com/api/catalog/API_Creditratings/get_rating_emission_maxdate/

https://ws.cbonds.info/services/wsdl/get_rating_emission_maxdate /
https://ws.cbonds.info/services/json/get_rating_emission_maxdate /

Share parameters on the trading floors - get_stocks_trading_grounds

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_stocks_trading_grounds/

https://ws.cbonds.info/services/wsdl/get_stocks_trading_grounds /
https://ws.cbonds.info/services/json/get_stocks_trading_grounds /

Share quotes from the National Settlement Depository - get_nrd_tradings_stocks

https://cbonds.com/api/catalog/API_CbondsQuotes/get_nrd_tradings_stocks/

https://ws.cbonds.info/services/wsdl/get_nrd_tradings_stocks/
https://ws.cbonds.info/services/json/get_nrd_tradings_stocks/

Capitalization of shares - get_stocks_capitalization

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_stocks_capitalization/

https://ws.cbonds.info/services/wsdl/get_stocks_capitalization/
https://ws.cbonds.info/services/json/get_stocks_capitalization/

Company programs-get_government_bond_programs

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_government_bond_programs/

https://ws.cbonds.info/services/wsdl/get_government_bond_programs /
https://ws.cbonds.info/services/json/get_government_bond_programs /

Fund commissions - get_funds_additional_costs_values

https://cbonds.com/api/catalog/API_CbondsFunds/get_funds_additional_costs_values/

https://ws.cbonds.info/services/wsdl/get_funds_additional_costs_values/
https://ws.cbonds.info/services/json/get_funds_additional_costs_values/

Syndicated credits – get loans

https://cbonds.com/api/catalog/API_CbondsDescriptiveData/get_loans/

https://ws.cbonds.info/services/wsdl/get_loans/
https://ws.cbonds.info/services/json/get_loans/

Directories
https://cbonds.com/api/catalog/dictionaries/

Auction types (directory) - get_auction_types
https://ws.cbonds.info/services/wsdl/get_auction_types/
https://ws.cbonds.info/services/json/get_auction_types/

Trading modes (directory) - get_boards
https://ws.cbonds.info/services/wsdl/get_boards/
https://ws.cbonds.info/services/json/get_boards/

Industries (directory) - get_branches
https://ws.cbonds.info/services/wsdl/get_branches/
https://ws.cbonds.info/services/json/get_branches/

Countries (directory) - get_countries
https://ws.cbonds.info/services/wsdl/get_countries/
https://ws.cbonds.info/services/json/get_countries/

Coupon types (directory) - get_coupon_types
https://ws.cbonds.info/services/wsdl/get_coupon_types/
https://ws.cbonds.info/services/json/get_coupon_types/

currencies (directory) - get_currencies
https://ws.cbonds.info/services/wsdl/get_currencies/
https://ws.cbonds.info/services/json/get_currencies/

ACI calculation base (directory) - get_day_count_conventions
https://ws.cbonds.info/services/wsdl/get_day_count_conventions/
https://ws.cbonds.info/services/json/get_day_count_conventions/

Default types (directory) - get_default_types
https://ws.cbonds.info/services/wsdl/get_default_types/
https://ws.cbonds.info/services/json/get_default_types/

Emission release form (directory) - get_emission_issue_forms
https://ws.cbonds.info/services/wsdl/get_emission_issue_forms/
https://ws.cbonds.info/services/json/get_emission_issue_forms/

Emission types (directory) - get_emission_kinds
https://ws.cbonds.info/services/wsdl/get_emission_kinds/
https://ws.cbonds.info/services/json/get_emission_kinds/

Emission status (directory) - get_emission_statuses
https://ws.cbonds.info/services/wsdl/get_emission_statuses/
https://ws.cbonds.info/services/json/get_emission_statuses/

Emission types (directory) - get_emission_subkinds
https://ws.cbonds.info/services/wsdl/get_emission_subkinds/
https://ws.cbonds.info/services/json/get_emission_subkinds/

Emitent type (directory) - get_emitent_types
https://ws.cbonds.info/services/wsdl/get_emitent_types/
https://ws.cbonds.info/services/json/get_emitent_types/

Categories of funds (directory) - get_funds_categories
https://ws.cbonds.info/services/wsdl/get_funds_categories/
https://ws.cbonds.info/services/json/get_funds_categories/

Fund specialization (directory) -get_funds_specializations
https://ws.cbonds.info/services/wsdl/get_funds_specializations/
https://ws.cbonds.info/services/json/get_funds_specializations/

Index groups (directory) - get_index_groups
https://ws.cbonds.info/services/wsdl/get_index_groups/
https://ws.cbonds.info/services/json/get_index_groups/

Index groups (directory) - get_index_types
https://ws.cbonds.info/services/wsdl/get_index_types/
https://ws.cbonds.info/services/json/get_index_types/

Offer status (directory) - get_offert_statuses
https://ws.cbonds.info/services/wsdl/get_offert_statuses/
https://ws.cbonds.info/services/json/get_offert_statuses/

Offer type (directory) - get_offert_types
https://ws.cbonds.info/services/wsdl/get_offert_types/
https://ws.cbonds.info/services/json/get_offert_types/

Type of investment Bank participation in emissions (directory) - get_participation_status

ttps://ws.cbonds.info/services/wsdl/get_participation_status/
https://ws.cbonds.info/services/json/get_participation_status/

Emission placement type (directory) - get_placing_types
https://ws.cbonds.info/services/wsdl/get_placing_types/
https://ws.cbonds.info/services/json/get_placing_types/

Emission placement methods (directory) - get_private_offerings
https://ws.cbonds.info/services/wsdl/get_private_offerings/
https://ws.cbonds.info/services/json/get_private_offerings/

Rate types (directory) - get_rates

https://ws.cbonds.info/services/wsdl/get_rates/
https://ws.cbonds.info/services/json/get_rates/

Rating agencies (directory) - get_rating_agencies
https://ws.cbonds.info/services/wsdl/get_rating_agencies/
https://ws.cbonds.info/services/json/get_rating_agencies/

Rating forecast (directory) - get_rating_forecasts
https://ws.cbonds.info/services/wsdl/get_rating_forecasts/
https://ws.cbonds.info/services/json/get_rating_forecasts/

Rating scales (directory) - get_rating_scales
https://ws.cbonds.info/services/wsdl/get_rating_scales/
https://ws.cbonds.info/services/json/get_rating_scales/

Points of the rating scale (directory) - get_rating_scale_points
https://ws.cbonds.info/services/wsdl/get_rating_scale_points/
https://ws.cbonds.info/services/json/get_rating_scale_points/

Regions (directory) - get_regions

https://ws.cbonds.info/services/wsdl/get_regions/
https://ws.cbonds.info/services/json/get_regions/

Emitent's organizational form (directory) - get_reg_forms
https://ws.cbonds.info/services/wsdl/get_reg_forms/
https://ws.cbonds.info/services/json/get_reg_forms/

The period for which the dividend was paid (directory) - get_stocks_dividends_periods
https://ws.cbonds.info/services/wsdl/get_stocks_dividends_periods/
https://ws.cbonds.info/services/json/get_stocks_dividends_periods/

Types of shares (directory) - get_stocks_full_kinds
https://ws.cbonds.info/services/wsdl/get_stocks_full_kinds/
https://ws.cbonds.info/services/json/get_stocks_full_kinds/

Subregions (directory) - get_subregions
https://ws.cbonds.info/services/wsdl/get_subregions/
https://ws.cbonds.info/services/json/get_subregions/

Exchanges (directory) - get_trading_grounds
https://ws.cbonds.info/services/wsdl/get_trading_grounds/
https://ws.cbonds.info/services/json/get_trading_grounds/

Quotation lists of exchanges (directory) - get_trading_ground_quotelists
https://ws.cbonds.info/services/wsdl/get_trading_ground_quotelists/
https://ws.cbonds.info/services/json/get_trading_ground_quotelists/

Nomenclature of financial statements (reference) - get_report_nomenclature
https://ws.cbonds.info/services/wsdl/get_report_nomenclature/
https://ws.cbonds.info/services/json/get_report_nomenclature/

Fund categories according to the Bank of Russia classification (directory) - get_cbrf_funds_categories
https://ws.cbonds.info/services/wsdl/get_cbrf_funds_categories/
https://ws.cbonds.info/services/json/get_cbrf_funds_categories/

Index categories – get_index_categories

https://ws.cbonds.info/services/wsdl/get_index_categories/
https://ws.cbonds.info/services/json/get_index_categories/

Index group 2 – get_index_groups2

https://ws.cbonds.info/services/wsdl/get_index_groups2/
https://ws.cbonds.info/services/json/get_index_groups2/