

Руководство разработчика. Cbonds API

| Декабрь 2025

Термины

Получение данных от веб-сервиса

Авторизация пользователя

Получение схемы сервиса

Запрос данных

Основные параметры ответа

Запрос данных с использованием фильтров

Примеры

Ограничения и ошибки

Ограничение на объем запросов

Частота обновления данных

Ограничение на частоту запросов

Ограничение на количество записей

Список возможных ошибок в ответе веб-сервиса

Демо-версия веб-сервиса

Пример обращения к веб-сервису на Python 3.x

Термины

Веб-сервис (англ. web service) — программная система, идентифицируемая строкой URI. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, передаваемых с помощью интернет-протоколов. Интерфейсы веб-сервисов Cbonds Database определены на языке XML и JSON.

WSDL (англ. Web Services Description Language) — язык описания веб-сервисов, основанный на языке XML.

SOAP (Simple Object Access Protocol) – протокол обмена сообщениями на базе XML.

XML — расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных.

JSON (англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript.

Операция веб-сервиса, метод веб-сервиса (далее – операция, метод) — функция веб-сервиса, возвращающая данные клиенту в формате WSDL+SOAP или JSON.

Список всех доступных методов веб-сервиса доступен в каталоге <https://cbonds.ru/api/catalog/folders/>

Пользователь – клиент услуги Cbonds Database, имеющий логин и пароль для авторизации, получающий доступ к определённым набору данных.

Получение данных от веб-сервиса

Авторизация пользователя

В целях предотвращения несанкционированного доступа к данным для всех операций, описанных ниже, необходимо указывать логин/пароль, закрепленные за услугой для пользователя.

Для ознакомления с технической стороной веб-сервиса можно использовать тестовые логин и пароль (далее – тестовый пользователь):

Логин: Test

Пароль: Test

Тестовый пользователь имеет доступ к демонстрационному (ограниченному) набору данных, и ни в коем случае не претендует на отображение полноты базы данных.

Получение схемы сервиса

XSD/XML схема или JSON-схема может быть получена GET запросом с указанием в адресе логина и пароля пользователя:

Схема	URL
XSD/XML	https://ws2.cbonds.info/services/wsdl?login=ЛОГИН&password=ПАРОЛЬ
JSON	https://ws2.cbonds.info/services/json?login=ЛОГИН&password=ПАРОЛЬ

Также схему можно получить методом POST, указав логин и пароль в теле POST запроса (не в адресе).

Два параметра в схеме, отражающие возможность фильтрации и сортировки:

sortable - отражает возможность сортировки по полю (1 - сортировка возможна, 0 - сортировка невозможна);

searchable – отражает возможность фильтрации по полю (1 - фильтрация возможна, 0 - фильтрация невозможна).

Запрос данных

Взаимодействие с веб-сервисом для получения данных осуществляется посредством передачи POST запросов HTTPS-протокола. В зависимости от предпочитаемого языка ответа, следует отправлять запросы на следующие адреса:

Data format	Request URL
XSD/XML	https://ws2.cbonds.info/services/wsd/ (название операции)*
JSON	https://ws2.cbonds.info/services/json/ (название операции)*

* например: https://ws2.cbonds.info/services/json/get_emissions

Список операций см. в разделе Список операций

В теле запроса передаются параметры запроса. Структура тела запроса в формате JSON:

```
{
  "auth":{"login":"ЛОГИН","password":"ПАРОЛЬ"},
  "filters":[{"field":"...", "operator":"...", "value":...}, {}],
  "quantity":{"limit":10,"offset":0},
  "sorting":[{"field":"...", "order":"..."}],
  "fields":[{"field": "..."}, {"field": "..."}]
}
```

Где основные параметры запроса:

"auth" - передает логин и пароль пользователя;

"filters" - содержит заданные условия выборки;

"quantity" - передает параметры {"limit":..., "offset":...}

"limit" - число записей на странице при одном запросе;

"offset" - номер страницы. Должен быть кратен лимиту, при лимите в 1000 offset может быть 0, 1000, 2000 и т.д.

"sorting" - передает список полей для сортировки;

"fields" - передает список полей для вывода результата запроса.

Структура тела запроса в формате WSDL+SOAP:

```
<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://ws2.cbonds.info/" >

      <m:AuthData>
        <m:Login><![CDATA[ЛОГИН]]></m:Login>
        <m>Password><![CDATA[ПАРОЛЬ]]></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:ResponseItemFields>

</m:RequestMessage_emissions>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Основные параметры ответа

Структура ответа в формате JSON:

```

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

```

Где параметры ответа:

"count" - количество записей, найденное для текущей страницы (для последней страницы может быть не равно limit);

"total" - количество записей, найденное по ограничениям (переданным в поле filters запроса). Для построения постранички (плана последовательных запросов для выборки всех данных частями) ориентируемся на данное поле в ответе (делим его на лимит, получаем в целой части номер последней страницы, если считать от нуля). Например, при лимите 1000 и значении total=2500, необходимо будет выполнить три запроса с оффсетами $0*1000=0$, $1*1000=1000$, $2*1000=2000$;

"limit" - принимает такое же значение как при запросе;

"offset" - значение offset не кратное лимиту приводится к кратному в меньшую сторону;

"exec_time" - параметр необходим для оптимизации. При наличии проблем с производительностью сервиса (и наличии возможностей), можно организовать логирование данного параметра по каждой операции, а в идеале - статистику с агрегированными значениями по операции, фильтрам, оффсету и т.д.

"items" - передает все записи для соответствующей страницы;

"cms_full_gen_time" - используется для оптимизации;

"user_id" - код вашего пользователя в нашей системе, для тестового пользователя user_id=33.

Структура ответа в формате WSDL+SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="https://ws2.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: eng →
```

```
<!-- strict_mode: 1 →  
<!-- performLogging: -1 →
```

Запрос данных с использованием фильтров

При запросе данных можно создавать фильтры, с помощью которых выбирать из всех данных только определенную информацию, соответствующую заданным критериям. **Список полей, по которым может быть создан фильтр, ограничен.**

Перечень возможных операторов для фильтрации данных:

Оператор	Расшифровка оператора	
'eq'	'equal'	Равно
'ne'	'not equal'	Не равно
'lt'	'less'	Строго меньше
'le'	'less or equal'	Меньше или равно
'gt'	'greater'	Строго больше
'ge'	'greater or equal'	Больше или равно
'in'	'is in'	Входит в множество
'ni'	'is not in'	Не входит в множество
'bw'	'begins with'	Начинается с (символов)
'bn'	'does not begin with'	Не начинается с (символов)
'ew'	'ends with'	Заканчивается (символами)
'en'	'does not end with'	Заканчивается не (символами)
'cn'	'contains'	Содержит часть строки
'nc'	'does not contain'	Не содержит часть строки
'nu'	'is null',	Имеет значение NULL
'nn'	'is not null'	Не NULL

Примеры

Примеры ограничений, которые могут быть для операции *get_emissions* (параметры эмиссий).

Запрос будет передаваться по адресу https://ws2.cbonds.info/services/json/get_emissions/

1. Выбор эмиссии с определенным ISIN, например, US037833EL06

Запрос в формате JSON:

```
{  
  "auth":{"login":"LOGIN","password":"PASSWORD"},  
  "filters":[{"field":"isin_code","operator":"eq","value":"US037833EL06"}],  
  "quantity":{"limit":10,"offset":0},  
  "sorting":[{"field":"id","order":"asc"}]
```

```
"fields":[{"field": "id"}, {"field": "maturity_date"}]
}
```

Запрос в формате WSDL+SOAP:

```
<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://ws2.cbonds.info/">
      <m:AuthData>
        <m:Login>LOGIN</m:Login>
        <m>Password>PASSWORD</m>Password>
      </m:AuthData>
      <m:Filters>
        <m:NumericFilter>
          <m:FilterField>isin_code</m:FilterField>
          <m:FilterOperator>eq</m:FilterOperator>
          <m:FilterValue><![CDATA[US037833EL06]]></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>id</m:SortField>
          <m:SortOrder>asc</m:SortOrder>
        </m:SortRule>
      </m:SortRules>
      <m:ResponseItemFields>
        <m:ResponseItemField>
          <m:ItemField>id</m:ItemField>
        </m:ResponseItemField>
        <m:ResponseItemField>
          <m:ItemField>maturity_date</m:ItemField>
        </m:ResponseItemField>
      </m:ResponseItemFields>
    </m:RequestMessage_emissions>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Далее для краткости приведем примеры только в формате JSON.

2. Выбор нескольких эмиссий с определенными ISIN, например, US037833EL06 и USU5876JAM72.

Запрос в формате JSON:

```
{
  "auth":{"login":"LOGIN","password":"PASSWORD"},
  "filters":[{"field":"isin_code","operator":"in","value":"US037833EL06;USU5876JAM72"}],
  "quantity":{"limit":10,"offset":0},
  "sorting":[{"field":"","order":"asc"}],
  "fields":[{"field":"id"}, {"fields": "maturity_date"}]
}
```

3. Выбор эмиссий, которые были обновлены после 21 августа 2025

Запрос в формате JSON:

```
{
  "auth":{"login":"LOGIN","password":"PASSWORD"},
  "filters":[{"field":"updating_date","operator":"ge","value":"2025-08-21"}],
  "quantity":{"limit":10,"offset":0},
  "sorting":[{"field":"","order":"asc"}]
  "fields":[]
}
```

4. Выбор эмиссий, которые были обновлены с 1 по 21 августа 2025

Запрос в формате JSON:

```
{
  "auth":{"login":"LOGIN","password":"PASSWORD"},
  "filters":[{"field":"updating_date","operator":"ge","value":"2025-08-01"},
            {"field":"updating_date","operator":"le","value":"2025-08-21"}],
  "quantity":{"limit":10,"offset":0},
  "sorting":[{"field":"","order":"asc"}]
  "fields":[]
}
```

Ограничения и ошибки

Ограничение на объем запросов

Максимальное количество записей в ответе (элементов поля items)

1000

Чтобы выбрать все данные, необходимо выполнить несколько запросов, меняя **offset**. В каждом ответе есть сведения об общем количестве записей (поле **total**), доступных при ограничениях выборки, переданных в запросе.

Частота обновления данных

Все операции, кроме *get_tradings_stocks_full_realtime_new* и *get_tradings_realtime* (внутридневные котировки) обновляются 1 раз в час. При этом обычно изменения в данных основных операций происходит **ежедневно**, а изменение в данных операций со справочниками происходит реже 1 раза в месяц.

Ограничение на частоту запросов

Период	Ограничение
Допустимое количество обращений пользователя к одной операции в минуту	Не больше 30 раз
Допустимое количество обращений пользователя к одной операции за сутки	Не больше 10 000 раз

Ограничение на количество записей

Для обеспечения стабильной и бесперебойной работы веб-сервиса, введено ограничение на количество записей в общем объёме выборки.

Максимальное количество записей – 1 000 000. При превышении данного значения будет транслироваться сообщение об ошибке со статусом 500100.

Список возможных ошибок в ответе веб-сервиса

1. Ошибки, которые можно получить при обращении к веб-сервису:
 - a. **HTTP/1.1 301 Moved Permanently** — доступ по http вместо https;
 - b. **HTTP/1.1 403 Forbidden** — некорректный логин; некорректный пароль; нет логина, нет пароля, нет блока с авторизацией, превышено максимально допустимое количество записей (подробнее в пункте 4 текущего раздела);
 - c. **HTTP/1.1 500 Internal Service Error** — конфиг с синтаксической ошибкой; запрос к операции, к которой нет доступа;
 - d. **HTTP/1.1 504** – истекло время ожидания ответа
2. Ошибки для ПО:
 - a. **https protocol required** — доступ по протоколу http вместо https;
 - b. **Max requests per minute limit max_per_minute exceeded. Try to request the next minute.** — ограничение на количество запросов в минуту;
 - c. **Max requests per day limit max_per_day exceeded. Try to request next day.** —ограничение на количество запросов в день;

3. Сообщения о перехваченных ошибках
 - a. Undefined index: xxxx
 - b. Invalid field (xxxx) for filtering
 - c. The request JSON string is not valid. The collection "%services_xxxxxx%" is not available for user id "yy"
4. Превышено максимально допустимое количество записей

"Превышено максимально допустимое количество записей в общем объеме выборки. Пожалуйста, используйте фильтрацию в запросе или уточните уже существующие фильтры для уменьшения объема выдачи"

Демо-версия веб-сервиса

Демонстративная версия служит помощником при разработке клиентских приложений. Инструмент содержит тот же функционал, что и полная версия, и отличается только наличием визуального интерфейса.

Для начала работы с демо-версией веб-сервиса необходимо перейти по одной из ссылок:

1. <https://ws2.cbonds.info/services/json/demo/>
2. <https://ws2.cbonds.info/services/wsdl/demo/>

Ниже приводится описание работы только с демо-версией для JSON, работа с демо-версией для WSDL аналогична.

1. Вводим логин и пароль пользователя или логин и пароль тестового пользователя и нажимаем «Request service schema»



Cbonds.Database WeServices » Json Demo

Credentials

Login Password

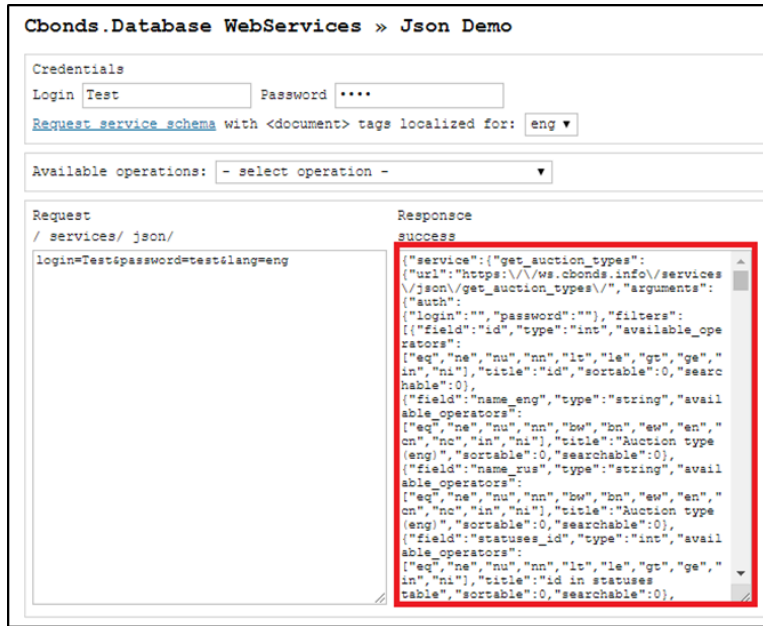
[Request service schema](#) with <document> tags localized for:

Request

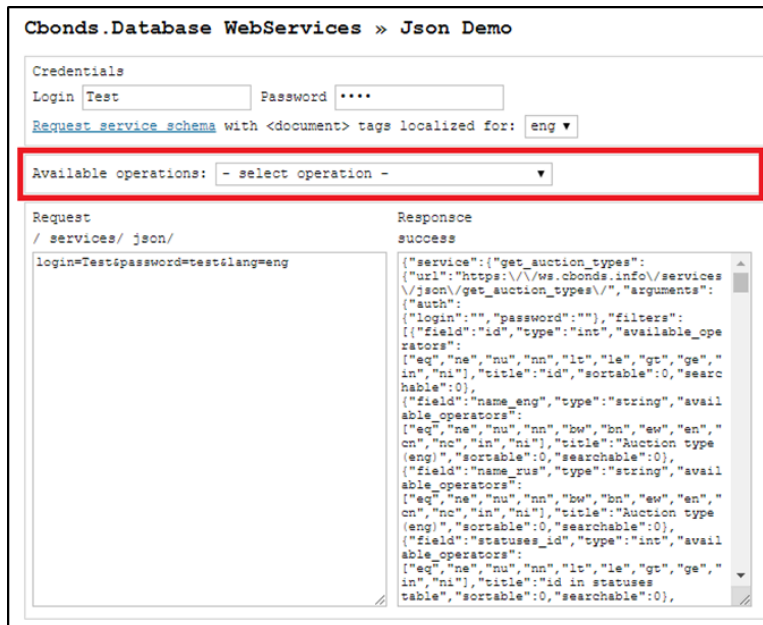
Response

2. После запроса схемы сервиса поля «Request» и «Response» станут заполненными.

В поле «Response» содержится схема сервиса, в данном примере JSON-схема.



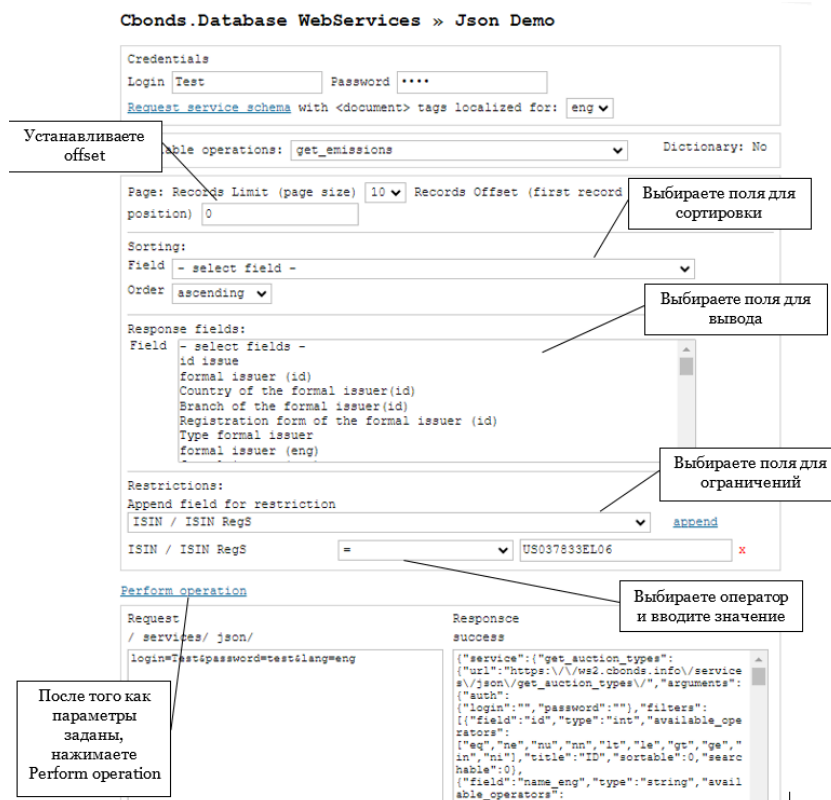
3. Выбираем одну из доступных операций из выпадающего списка «Available operation». Выберем для примера операцию «get_emissions».



4. После выбора Операции будут доступны следующие настройки параметров запроса:

- выбор количества отдаваемых записей;
- сортировка по полученным ответам;
- ограничения (фильтры) на отдаваемые данные.

Данные опции можно применить для всех операций, кроме операций-справочников. Для работы с фильтрами необходимо выбрать поле, нажать кнопку «append», выбрать оператор и значение.



5. После того, как запрос будет выполнен, в поле «Request» автоматически сгенерируется тело POST запроса. В поле «Response» появится ответ в соответствующем формате.

Таким образом, при разработке продукта можно сверяться с данными, возвращаемыми веб-сервисом.

Пример обращения к веб-сервису на Python 3.x

Ниже представлен пример кода на Python 3.x осуществляющий:

1. Запрос данных с фильтрацией;
2. Запись данных в структуру DataFrame;
3. Сохранение таблицы в csv.

```
import requests
import pandas as pd

url_emissions = 'https://ws2.cbonds.info/services/json/get_emissions/?lang=eng'

json_data_emissions = {"auth":{"login":"LOGIN","password":"PASSWORD"},
```

```
"filters":[{"field":"isin_code","operator":"in","value":"US037833EL06;USU5876JAM72"}],  
"quantity":{"limit":10,"offset":0},"sorting":[{"field":"","order":"asc"}]}
```

```
response_emissions = requests.get(url_emissions, json=json_data_emissions).json()  
print(response_emissions)
```

```
initial_dict = response_emissions['items']
```

```
df = pd.DataFrame.from_dict(initial_dict)  
df
```

```
df.to_csv(index=False)
```