

**PRODUCTS,
TECHNOLOGY
SERVICES AND
MARKET DATA**

**FIXED INCOME
CALCULATOR
FRONT, MIDDLE
AND OTC
PRODUCTS**



BRAZILIAN EXCHANGE AND OTC.

ÍNDICE

1 OVERVIEW.....	3
2 SSL SECURITY CERTIFICATION.....	3
3 AUTHORIZATION.....	4
4 WEB METHOD: LOGIN.....	5
5 WEB METHOD: LISTBONDCODES.....	6
6 WEB METHOD: LISTBONDCODESCSV.....	6
7 WEB METHOD: LISTBONDCODESCRA.....	6
8 WEB METHOD: LISTBONDCODESCRI.....	7
9 WEB METHOD: LISTTITULOSPUBLICOS.....	7
10 WEB METHOD: LISTTITULOSPUBLICOSCSV.....	8
11 WEB METHOD: GETBONDDETAILS.....	8
12 WEB METHOD: CALCPU.....	8
13 WEB METHOD: CALCPUCSV.....	10
14 WEB METHOD: CALCYIELD.....	11
15 WEB METHOD: CALCYIELDCSV.....	12
16 API SPECIFIC DATA TYPE.....	13
17 OPTIONAL PARAMETER FOR IDENTIFY YOUR API REQUESTS	15
18 EXTRACT	16
19 DI CALCULATION.....	18

1 OVERVIEW

This document covers all methods available on Cetip|Trader Calculator – Webservice (available to Debentures and Government Bonds).

The **Demo** URL to access the *Webservice* is:

<https://wwwhom.calculadorarendafixa.com.br/>

The **Production** URL to access the *Webservice* is:

<https://api.calculadorarendafixa.com.br/>

The Calculator RESTful API provides endpoints that allow system developers to interact with it remotely by sending and receiving JSON (JavaScript Object Notation) objects.

2 SSL SECURITY CERTIFICATION

The Webservice has the prefix “https”, which means that the connection to it will be made by the SSL Protocol.

To make the connection correctly you must use an appropriate component for this protocol.

On some cases, it is necessary to search a CA (PEM) file from a list of authorization

Certificates that will validate the certifier. One example of file that can be used can be found on the following link: <http://curl.haxx.se/docs/caextract.html>.

3 AUTHORIZATION

To call the Methods from the Webservice it is necessary user authentication to guarantee secure access to our services. The authentication can be done through the WebMethod “login”.

To receive the Authorization Code it is necessary to send:

1 – Authentication Key – Numeric: from 5 to 32 positions.

2 – Token Code – Alfa-numeric: 64 positions.

These data will not change for systems integrations.

In case of Authentication Failure (Status: 403 – Forbidden), user will receive response message explaining the reason.

On cases where authentication is well succeeded (Status: 200 – Ok), the return of WebMethod “login” is the Webservice’s Authorization Code named “authorizationID”. This

Authorization Code is Alpha-Numeric and has from 32 up to 96 positions. It should be sent every time other Methods are called using HTTP header “Authorization” as shown in the example below:

```
$ curl -H "Authorization:
```

```
5db1bfe6ef7d54b31c42ef5a493590878f276f785e2d442297514f" -X
```

```
GET <method>
```

In cases where Authorization Code is not sent or is invalid, the feedback (Status: 401 – Unauthorized) will be a text explaining why this code wasn’t accepted.

A usual scenario occurs when session expires. This happens when the Webservice doesn’t receive any call for 10 minutes. On this cases, new authentication is required.

For **Demo** access, the key and token values are always going to be '123'. This means that when logging in (Web Method below), the key value should be '123' and the token value should be '123'.

For **Production** access, you should request key and token values to B3.

4 WEB METHOD: LOGIN

Gives the Authorization Code to call the other *Webservice* Methods:

HTTP Method: POST

Arguments: Via POST method, send the data for the login on the body of the HTTP request

The data can be sent through the following formats:

Urlencoded:

Header necessary HTTP: Content-Type: application/x-www-form-urlencoded

Ex.: key=MinhaChave&token=meuToken

Json:

Header necessary HTTP: Content-Type: application/json

Ex.: {"key": "MinhaChave", "token": "MeuToken"}

Answer: (String) authorizationID – Authorization ID. This is necessary to call all other

Webservice Methods.

Method and URL: POST <webServiceUrl>/login

Exemples using cURL:

```
$ curl -H "Content-type: application/x-www-form-urlencoded" -X POST -o
```

```
'key=MinhaChave&token=MeuToken' <WebserviceUrl>/login
```

```
$ curl -H "Content-type: application/json" -X POST -d
```

```
{'key': 'MinhaChave', 'token': 'MeuToken'} <WebserviceUrl>/login
```

5 WEB METHOD: LISTBONDCODES

List all **Debentures** available to calculation.

HTTP Method: GET

Arguments: None

Answer: (JSON - Array of Strings) - Array containing all Debentures Tickers.

URL: <WebserviceUrl>/listBondCodes

6 WEB METHOD: LISTBONDCODESCSV

List all **Government Bonds** available to calculation.

HTTP Method: GET

Arguments: None

Answer: (CSV) – text file ‘;’ (semi-colon) separated

URL: <WebserviceUrl>/listBondCodesCSV

7 WEB METHOD: LISTBONDCODESCRA

List all **Agribusiness Receivables Certificate (CRA)** available to calculation.

HTTP Method: GET

Arguments: None

Answer: (JSON - Array of Strings) - Array containing all Agribusiness Receivables Certificates (CRA) Tickers.

URL: [*<WebserviceUrl>/listBondCodesCra*](#)

8 WEB METHOD: LISTBONDCODESCRI

List all Certificates of real estate receivables (CRI) available to calculation.

HTTP Method: GET

Arguments: None

Answer: (JSON - Array of Strings) - Array containing all certificates of real estate receivables (CRI) Tickers.

URL: [*<WebserviceUrl>/listBondCodesCri*](#)

9 WEB METHOD: LISTTITULOSPUBLICOS

List all Government Bonds available to calculation.

HTTP Method: GET

Arguments: None

Answer: (JSON – Array of Public Title) – Specific type described ahead.

URL: [*<WebserviceUrl>/listTitulosPublicos*](#)

10 WEB METHOD: LISTTITULOSPUBLICOSCSV

List all Government Bonds available to calculation.

HTTP Method: GET

Arguments: None

Answer: (CSV) - - text file ';' (semi-colon) separated

URL: <WebserviceUrl>/listTitulosPublicosCSV

11 WEB METHOD: GETBONDDetails

Show all data available for a specific debenture, agribusiness receivables certificate (CRA), real state receivables certificate (CRI) or regarding its cash flow, yield, maturity, etc.

HTTP Method: GET

Arguments: Debenture/CRA/CRI Ticker (bondCode - string)

Answer: (JSON - Bond) - Specific type described ahead.

URL: <WebserviceUrl>/getBondDetails/<bondCode>

Ex: <WebserviceUrl>/getBondDetails/CSMG26

Ex.: <WebserviceUrl>/getBondDetails/CRA019000XD

Obs.: For Debentures, CRI and CRA only

12 WEB METHOD: CALCPU

Calculates the Price (PU) for a specific Debenture, CRA or Government Bond given the Yield to Maturity.

HTTP Method: GET

Arguments – CRA:

CRA ticker (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – CRI:

CRI ticker (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – Debentures:

Debenture ticker (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – Government Bonds:

SELIC ID (selicid – String)

Calculation Date (calculationDate – String. Format: “YYYY-MM-DD”)

YTM (yield – double)

Answer: (JSON - CalculationResult) – Specific type described ahead.

URL: <WebserviceUrl>/calcPU/<bondCode>/<calculationDate>/<yeild>

Ex: <WebserviceUrl>/calcPU/CSMG26/2014-05-07/6.22

Ex.: <WebserviceUrl>/calcPU/CRA019000XD/2020-07-07/6.22

Obs.: The *Webmethod* automatically detects if the calculation is for a Debenture, a CRA or a Government Bond.

13 WEB METHOD: CALCPUCSV

Calculates the Price (PU) for a specific Debenture, CRI, CRA or Government Bond given the Yield to Maturity.

HTTP Method: GET

Arguments – CRA:

CRA Ticker (bondCode – String)

Calculation Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – CRI:

CRI Ticker (bondCode – String)

Calculation Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – Debentures:

Debenture Ticker (bondCode – String)

Calculation Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Arguments – Government Bonds:

SELIC ID (selicid – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

YTM (yield – double)

Answer: (CSV) - method, Yield, VNA, PUPar, PU and duration ';' (semi-colon) separated.

URL: <WebserviceUrl>/calcPUCSV/<bondCode>/<calculationDate>/<yeild>

Ex: <WebserviceUrl>/calcPUCSV/CSMG26/2014-05-07/6.22

Obs.: The *Webmethod* automatically detects if the calculation is for a Debenture, a CRA or a Government Bond

14 WEB METHOD: CALCYIELD

Calculates Yield for a specific Debenture, CRA, CRI or Government Bond given the Price.

HTTP Method: GET

Arguments – Debentures:

Debenture code (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Arguments – CRA:

CRA code (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Arguments –Government Bonds:

SELIC ID (selicid – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Answer: (JSON - CalculationResult) – Specific type described ahead.

URL: <WebServiceUrl>/calcYield/<bondCode>/<calculationDate>/<PU>

Ex: <WebServiceUrl>/calcYield/CSMG26/2014-05-07/961256.968405

Obs.: The *Webmethod* automatically detects if the calculation is for a Debenture or a Government Bond.

15 WEB METHOD: CALCYIELDCSV

Calculates Yield for a specific Debenture, CRA, CRI or Government Bond given the Price

HTTP Method: GET

Arguments – Debentures:

Debenture Code (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Arguments – CRA:

CRA Code (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Arguments – CRI:

CRI Code (bondCode – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Arguments – Government Bonds:

SELIC ID (selicid – String)

Settlement Date (calculationDate – String no formato “YYYY-MM-DD”)

Price (PU – double)

Answer: (CSV) - method, Yield, VNA, PUPar, PU and duration ';' (semi-colon) separated.

URL: <WebserviceUrl>/calcYieldCSV/<bondCode>/<calculationDate>/<PU>

Ex: <WebserviceUrl>/calcYieldCSV/CSMG26/2014-05-07/961256.968405

Obs.: The *Webmethod* automatically detects if calculation is for a Debenture or for a Government Bond.

16 API SPECIFIC DATA TYPE

The specific Data Types that are transmitted via the *Webservice* are:

Bond:

codbond (string) – Debenture, CRI or CRA Ticker

issuer (string) – Issuer’s Name

method (string) – Calculation Method for the specific Bond

vne (double) – Yield at initiation (Price at Issuance Date)

yield (double) – Yield at initiation (Yield at Issuance Date)

expiredate (string no formato “YYYY-MM-DD”) – Expiration Date

issuedate (string no formato “YYYY-MM-DD”) – Issuance Date

startingdate (string no formato “YYYY-MM-DD”) – Interest Accrual Date

events(Array de Event) – Bond Events

TituloPublico:

selicid (string) –SELIC ID

issuer (string) – Fixed: “Banco Central do Brasil”

method (string) – Calculation Method (LTN, LFT, NTN-B, etc)

expiredate (string format “YYYY-MM-DD”) – Expiration Date

CalculationResult:

codbond (string) – Debenture, CRI or CRA Ticker

method (string) - Calculation Method for the specific Bond

date (string no formato “YYYY-MM-DD”) – Settlement Date

VNA (double) – Accrued Price

interest (double) – Interest Accrued

PUPar (double) – VNA + Interest

PU (double) – Calculated Price

yield (double) – YTM

duration (double) – Duration.

cashFlowList (Array de CashFlow) – Future cash flows (from Settlement Date to Expiration Date)

CashFlow:

date (string no formato “YYYY-MM-DD”) – Cash Flow Date

eventType (string) – Type of the future event:

“J”: interest payment, “A”: Amortization e “V”: Expire

workingDays (integer) – Business days between cash flow settlement date and bond settlement Date.

yield (double) – Future Yield Value (only for DI type bonds)

finalValue (double) – Future Cash Flow Value

presentValue (double) – Present Cash Flow Value

Event:

date (string format “YYYY-MM-DD”) – Event Date

eventType (string) – Type of the future event:

“J”: interest payment, “A”: Amortization e “V”: Expire.

yield (double) – Value for that cash flow

17 OPTIONAL PARAMETER FOR IDENTIFY YOUR API REQUESTS

In order for your calculation call to be identified, an optional "Client-ID" parameter can be added to the header of each request.

Example:

▼ Request Headers view source

Accept: application/json
 Accept-Encoding: gzip, deflate, br
 Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
 Authorization: batchb3803abdbb220d2a7c656e386062e428da8d31221335932dd28c95c56973a5d0c2cf3e6ea77095745867de18
 Cache-Control: no-cache
 Client-ID: identificacao-client-1234

18 EXTRACT

This call gives you the extract of the calculations requests made by you and your dependent-users in a certain period.

HTTP Method: GET

Example: <WebserviceUrl>/consumo/{{StartDate}}/{{FinalDate}}

```
curl -X GET -H 'Authorization: conegtester79b32b914bdb19a86af73d2626d1f7f2f44fdfe77c50642e00dc868265803a01de304f7fe0a847688' -H 'Content-Type: application/json' -H 'Accept: application/json' -v -i 'https://api.hom.calculadorarendafixa.com.br/consumo/2020-07-01/2020-07-30'
```

Response:

```
{
  "quantidadeCalculos": 21,
  "quantidadeCalculosDependentes": 59,
  "detalhe": [
    {
      "quantidadeCalculos": 12,
      "login": "assinante3",
      "nome": " ASSINANTE 3 ",
      "codAtivo": "DI"
    },
    {
      "quantidadeCalculos": 6,
```



```

        "login": "assinante3",
        "nome": " ASSINANTE 3 ",
        "codAtivo": "Debenture"
    }
    {
        "quantidadeCalculos": 3,
        "login": "assinante3",
        "nome": " ASSINANTE 3 ",
        "codAtivo": "Titulo Publico"
    }
],
"detalheDependentes": [
    {
        "quantidadeCalculos": 25,
        "login": "assinante1",
        "nome": "ASSINANTE 1",
        "codAtivo": "DI"
    },
    {
        "quantidadeCalculos": 19,
        "login": "assinante1",
        "nome": "ASSINANTE 1",
        "codAtivo": "Debenture"
    },
    {
        "quantidadeCalculos": 15,
        "login": "assinante1",
        "nome": "ASSINANTE 1",
        "codAtivo": "Titulo Publico"
    }
]
}

```

19 DI CALCULATION

Example: `<WebserviceUrl /di/calculo?dataInicio=2018-09-19&dataFim=2018-09-20&percentual=100&valor=1.2`

HTTP Method: GET

```
curl -X GET -v -i 'https://api.calculadorarendafixa.com.br/di/calculo?dataInicio=2018-09-19&dataFim=2018-09-20&percentual=100&valor=1.2'
```

Response example:

```
{
  "dataInicial": "19/09/2018",
  "dataFinal": "20/09/2018",
  "percentual": "100.00",
  "fator": "1.00024583",
  "taxa": "0.02",
  "valorBase": "1.20",
  "valorCalculado": "1.20"
}
```



BRAZILIAN EXCHANGE AND OTC.

B3.COM.BR