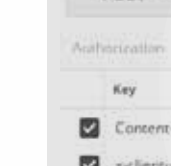




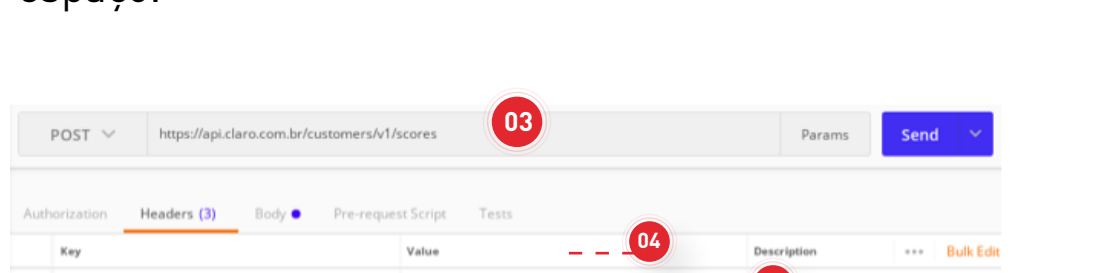
Tutorial de consumo da API Score



Para iniciar o consumo da API Score, é necessário adquirir o token de acesso, através da key e da secret, da aplicação contratada no portal.

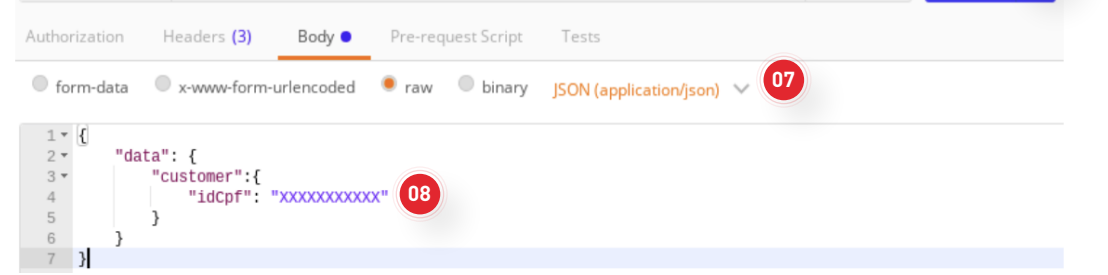
» Usar o postman para obter o token da API Score

- 01 Inicie o postman.
- 02 Selecione o método POST.
- 03 Para o URI, insira `https://api.claro.com.br/oauth2/v1/token`
- 04 Na guia Headers, adicione chave de Content-Type e para o valor `application/x-www-form-urlencoded`
- 05 Na guia Headers, adicione a chave de x-client-auth, seu valor será Basic base64(key:secret).

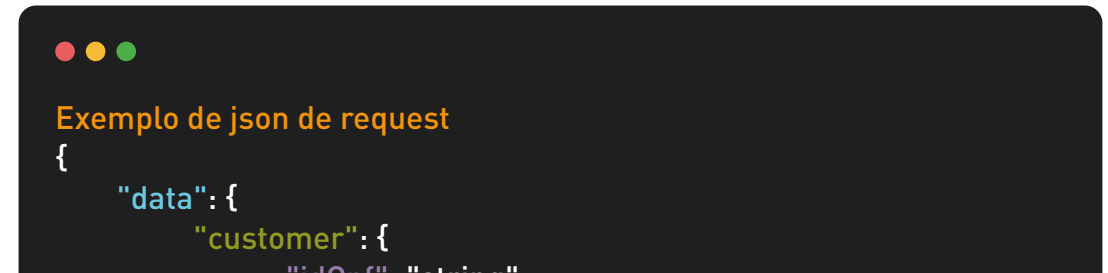


Observação, para gerar a base64 pode usar sites com essa finalidade, passando como entrada o key:secret, a saída deve ser algo como: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX=

- 06 Na guia Body, marque a caixa x-www-form-urlencoded
- 07 Na guia Body, adicione a chave de grant_type, seu valor será client_credentials
- 08 Envie a solicitação, pressionando o botão Envio

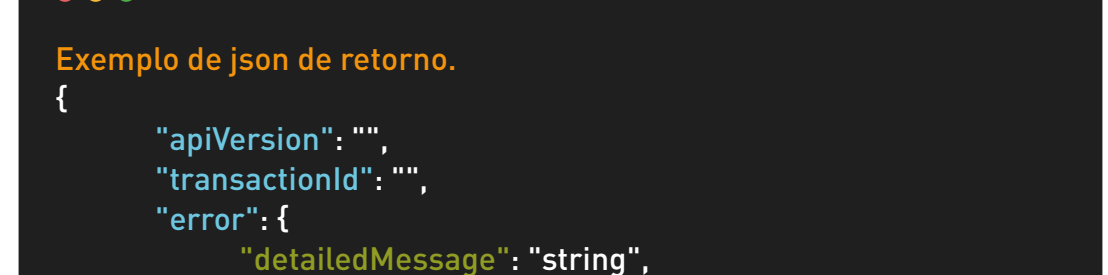


09 O token, tem 24 horas de duração, se não for solicitado novamente. Salve o access_token, para poder consumir a API.



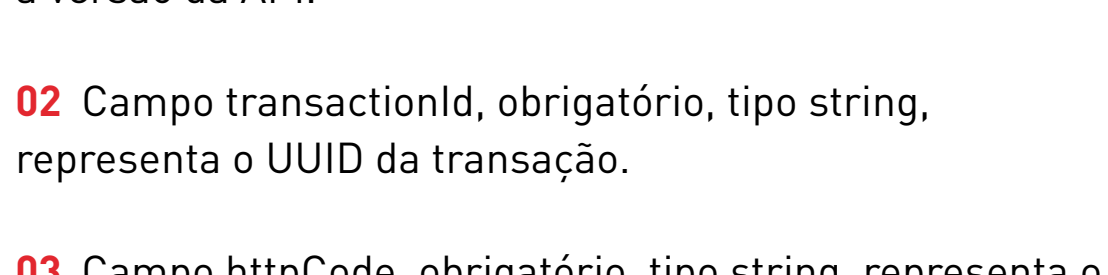
» Usar o postman para consumir API API Alert

- 01 Inicie uma nova aba no postman
- 02 Selecione o método POST
- 03 Para o URI, insira `https://api.claro.com.br/customers/v1/scores`
- 04 Na guia Headers, adicione chave de Content-Type e para o valor `application/json`.
- 05 Na guia Headers, adicione chave de x-client-auth e para o valor Bearer access_token. Observação, substitua access_token pelo valor do atributo access_token, adquirido a solicitação do token.
- 06 Na guia Headers, adicione chave de X-CustomerID e para o valor coloque a razão social da sua empresa, sem espaço.

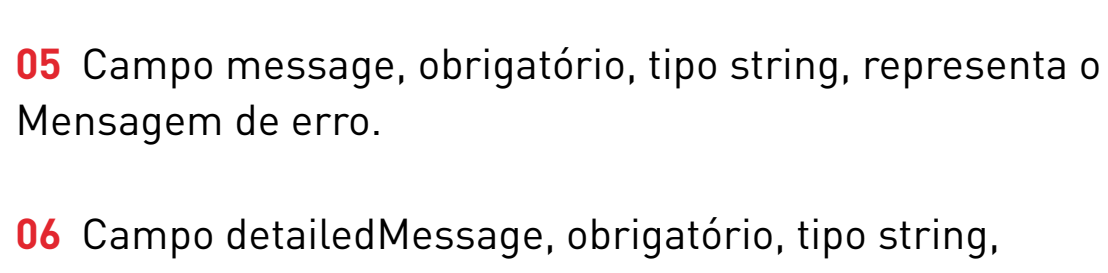


07 Na guia Body, marque a caixa raw e selecione JSON(application/json)

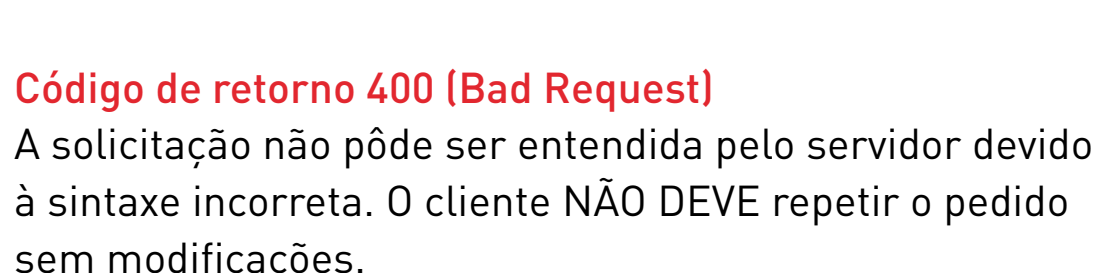
08 No corpo do Body deve ter o json com o cpf que deseja consultar.



09 Envie a solicitação, pressionando o botão Envio



» Dados de request



Exemplo de json de request

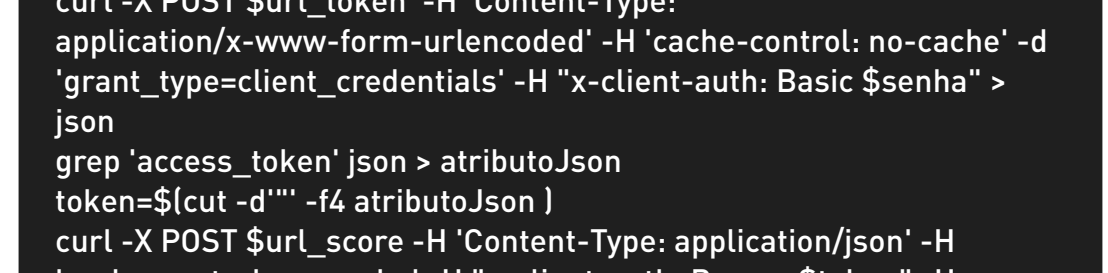
```
{
  "data": {
    "customer": {
      "idCpf": "string",
    }
  }
}
```

Explicação dos dados do json

01 Campo idCpf, obrigatório, tipo string (11 caracteres) representa o CPF do cliente

» Dados de resposta

Status com retorno com sucesso.



Exemplo de json de retorno

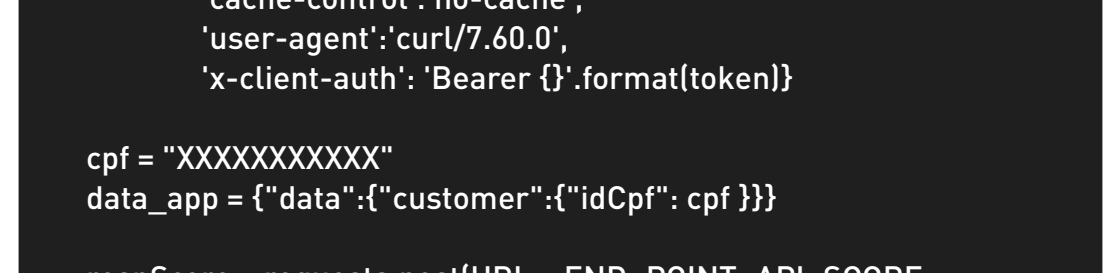
```
{
  "data": {
    "customer": {
      "idCpf": "string"
    },
    "statsResult": {
      "score": BigInteger
    }
  }
}
```

Explicação dos dados do json

01 Campo idCpf, obrigatório, tipo string (11 caracteres) representa o CPF do cliente

02 Campo score, obrigatório, tipo biginteger representa o score do CPF do cliente pesquisado.v

Status com retorno com erro.



Exemplo de json de retorno.

```
{
  "apiVersion": "string",
  "transactionId": "string",
  "error": {
    "detailedMessage": "string",
    "errorCode": "string",
    "message": "string",
    "httpCode": "string",
    "link": {
      "href": "string",
      "rel": "string"
    }
  }
}
```

01 Campo apiVersion, obrigatório, tipo string, representa a versão da API.

02 Campo transactionId, obrigatório, tipo string, representa o UUID da transação.

03 Campo httpCode, obrigatório, tipo string, representa o código de retorno HTTP.

04 Campo errorCode, obrigatório, tipo string, representa o código de erro Claro.

05 Campo message, obrigatório, tipo string, representa o Mensagem de erro.

06 Campo detailedMessage, obrigatório, tipo string, representa o Mensagem detalhada.

07 Campo rel, obrigatório, tipo string, sempre "related".

08 Campo href, obrigatório, tipo string, representa o endereço para o detalhamento dos erros.

» Status de resposta

Código de retorno 200 (Success)

Código de retorno 400 (Bad Request)

A solicitação não pôde ser entendida pelo servidor devido à sintaxe incorreta. O cliente NÃO DEVE repetir o pedido sem modificações.

Código de retorno 401 (Unauthorized)

A solicitação requer autenticação do usuário.

Código de retorno 403 (Forbidden)

O servidor entendeu a solicitação, mas está se recusando a atendê-la.

Código de retorno 404 (Not Found)

O servidor não encontrou nada que corresponda ao request-URI (CPF não encontrado na base)

Código de retorno 415 (Unsupported Media Type)

O servidor falhou ao atender uma solicitação. Por que o formato do payload não é um formato suportado.

Código de retorno 406 (Not Acceptable)

Servidor não pode produzir uma resposta que combine com a lista de valores aceitáveis definidas.

» Exemplo de código em curl



```
#!/bin/bash

url_score=https://api.claro.com.br/customers/v1/scores
url_token=https://api.claro.com.br/oauth2/v1/token
cpf="XXXXXXXXXX"
key="XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
secret="XXXXXXXXXX"

senha=$(echo -n $key:$secret | base64)

curl -X POST $url_token -H 'Content-Type: application/x-www-form-urlencoded' -H 'cache-control: no-cache' -d 'grant_type=client_credentials' -H 'x-client-auth: Basic $senha' > json

grep 'access_token' json > atributoJson
token=$(cut -d '"' -f4 atributoJson)

curl -X POST $url_score -H 'Content-Type: application/json' -H 'cache-control: no-cache' -H 'x-client-auth: Bearer $token' -H 'X-CustomerID: claro_exemplo' -d '{"data": {"customer": {"idCpf": "$cpf"}}}'
```

» Exemplo de código em python



```
import requests, json
import base64

KEY="XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SECRET="XXXXXXXXXX"

SENHA = base64.b64encode( (KEY + ':' + SECRET ).encode('ascii') ).decode('ascii')

GRANT_TYPE = "client_credentials";

URL = "https://api.claro.com.br";
END_POINT_ACCESS_TOKEN = "/oauth2/v1/token";
END_POINT_API_SCORE = "/customers/v1/scores";

headers={'Content-Type':'application/x-www-form-urlencoded','cache-control':'no-cache','x-client-auth':'Basic {}'.format(SENHA)}

data = {'grant_type':'client_credentials'}

respToken = requests.post(URL + END_POINT_ACCESS_TOKEN , headers=headers , data= data)
if respToken.status_code != 200:
    print('Error: ' + str(respToken.status_code))
else:
    token = json.loads( respToken.text)['access_token']

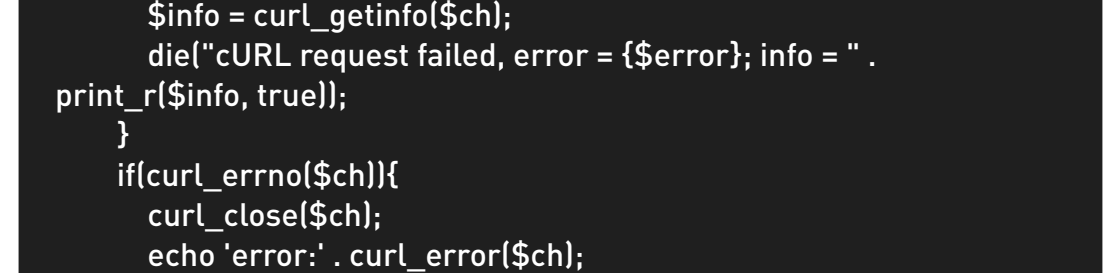
headers_app = {'Content-Type':'application/json',
'X-CustomerID': 'claro_exemplo',
'cache-control':'no-cache',
'user-agent': 'curl/7.60.0',
'x-client-auth': 'Bearer {}'.format(token)}

cpf = "XXXXXXXXXX"
data_app = {"data":{"customer":{"idCpf": cpf}}

respScore = requests.post(URL + END_POINT_API_SCORE, headers=headers_app , data= json.dumps(data_app))

if respScore.status_code != 200:
    print('Error: ' + str(respScore.status_code))
else:
    print('Score ' + str( respScore.text ))
```

» Exemplo de código em java



```
package client.claroScores;

import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.Base64;

import org.codehaus.jackson.JsonProcessingException;
import org.codehaus.jackson.map.ObjectMapper;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.util.LinkedMultiValueMap;
import org.springframework.web.client.RestClientException;
import org.springframework.web.client.RestTemplate;

public class App {

    private static final String KEY = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX";

    private static final String SECRET = "XXXXXXXXXX";

    private static final String GRANT_TYPE = "client_credentials";

    private static final String URL = "https://api.claro.com.br";

    private static final String END_POINT_ACCESS_TOKEN = "/oauth2/v1/token";

    private static final String END_POINT_API_SCORE = "/customers/v1/scores";

    public static void main( String[] args ) throws URISyntaxException,
    JsonProcessingException, IOException{

        App app = new App();

        String token = app.getAccessToken();
        String cpf = "XXXXXXXXXX";

        System.out.println("Score: " + app.getScore( token , cpf ));

    }

    public String getScore(String accessToken, String cpf) throws
    RestClientException, URISyntaxException {

        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_JSON);
        headers.add("cache-control", "no-cache");
        headers.add("x-client-auth", "Bearer " + accessToken);
        headers.add("X-CustomerID", "claro_exemplo");

        HttpEntity<Score> request = new HttpEntity<>(new Score(cpf), headers);

        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> result = restTemplate.postForEntity(new URI(URL + END_POINT_API_SCORE), request, String.class);

        return result.getBody();

    }

    public String getAccessToken() throws URISyntaxException,
    JsonProcessingException, IOException {

        String senha = Base64.getEncoder().encodeToString( App.KEY.concat(":").concat(App.SECRET).getBytes() );

        HttpHeaders headers = new HttpHeaders();

        headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);
        headers.add("cache-control", "no-cache");
        headers.add("x-client-auth", "Basic " + senha);

        MultiValueMap<String, String> parameters = new LinkedMultiValueMap<>();
        parameters.add("grant_type", GRANT_TYPE);

        HttpEntity<MultiValueMap<String, String>> request = new HttpEntity<>(parameters, headers);

        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> result = restTemplate.postForEntity(new URI(URL + END_POINT_ACCESS_TOKEN), request, String.class);

        return new ObjectMapper().readTree(result.getBody()).get("access_token").asText();

    }

    class Score{

        public Score(String cpf) {
            this.data = new Data(cpf);
        }

        private Data data;

        public Data getData() {
            return data;
        }

        public void setData(Data data) {
            this.data = data;
        }

    }

    class Data{

        public Data(String cpf) {
            this.customer = new Customer(cpf);
        }

        private Customer customer;

        public Customer getCustomer() {
            return customer;
        }

        public void setCustomer(Customer customer) {
            this.customer = customer;
        }

        class Customer{

            public Customer(String idCpf) {
                this.idCpf = idCpf;
            }

            private String idCpf;

            public String getIdCpf() {
                return idCpf;
            }

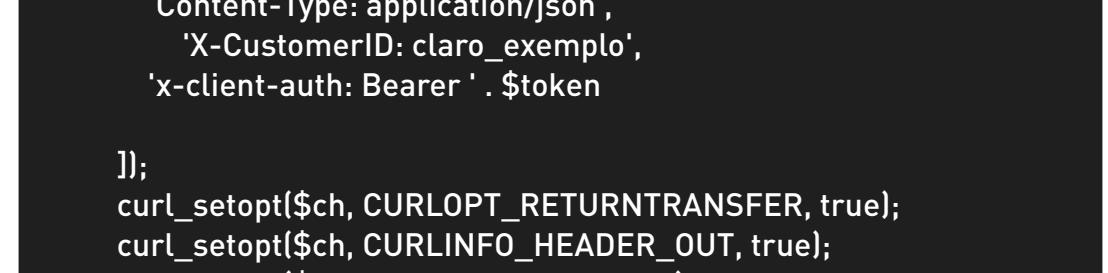
            public void setIdCpf(String idCpf) {
                this.idCpf = idCpf;
            }

        }

    }

}
```

» Exemplo de código em php



```
<?php

$key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX";
$secret = "XXXXXXXXXX";
$cpf = "XXXXXXXXXX";

$token = getAccessToken();

$score = getScore($token , $cpf);

echo $score;

function getAccessToken() {

    try{

        $senha = base64_encode($GLOBALS['key'] . ':' . $GLOBALS['secret']);

        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "https://api.claro.com.br/oauth2/v1/token");
        curl_setopt($ch, CURLOPT_HTTPHEADER, [
            'Content-Type: application/x-www-form-urlencoded',
            'X-CustomerID: claro_exemplo',
            'x-client-auth: Basic ' . $senha
        ]);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_HEADER_OUT, true);
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, 'grant_type=client_credentials');

        $response = curl_exec($ch);

        if (!$response){
            $error = curl_error($ch);
            $info = curl_getinfo($ch);
            die("cURL request failed, error = {$Error}; info = " . print_r($info, true));
        }
        if(curl_errno($ch)){
            curl_close($ch);
            echo 'error: ' . curl_error($ch);
        } else {
            curl_close($ch);
            return json_decode($response)->access_token;
        }

    } catch (Exception $e) {
        return 'Erro ' . $e;
    }

}

function getScore($token , $cpf){

    try {

        $payload = '{"data":{"customer":{"idCpf": ' . $cpf . '}}}';
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "https://api.claro.com.br/customers/v1/scores");
        curl_setopt($ch, CURLOPT_HTTPHEADER, [
            'Content-Type: application/json',
            'X-CustomerID: claro_exemplo',
            'x-client-auth: Bearer ' . $token
        ]);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_HEADER_OUT, true);
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $payload);

        $response = curl_exec($ch);

        if (!$response){
            $error = curl_error($ch);
            $info = curl_getinfo($ch);
            die("cURL request failed, error = {$Error}; info = " . print_r($info, true));
        }
        if(curl_errno($ch)){
            curl_close($ch);
            echo 'error: ' . curl_error($ch);
        } else {
            curl_close($ch);
            return $response;
        }

    } catch (Exception $e) {
        return 'Erro ' . $e;
    }

}

?>
```

