

Tutorial de consumo da API Fraud Scores V1

Para iniciar o consumo da API Validation Address, 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 Fraud Scores V1

1 – Inicie o postman.

2 – Selecione o método POST.

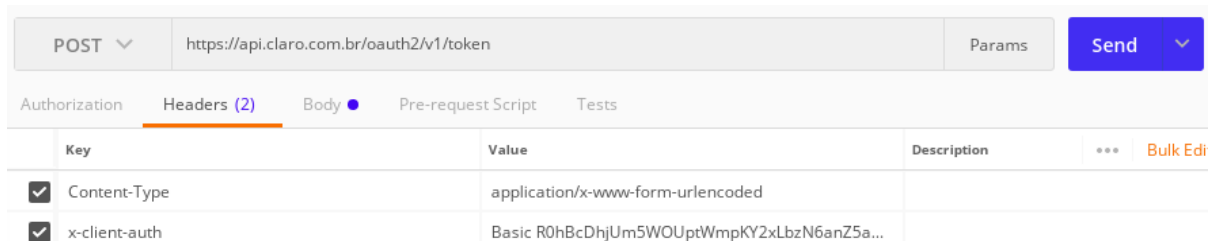
3 - Para o URI, insira <https://api.claro.com.br/oauth2/v1/token>

4 - Na guia Headers, adicione chave de Content-Type e para o valor application/x-www-form-urlencoded.

5 – Na guia Headers, adicione a chave de x-client-auth, seu valor será Basic base64(key:secret).

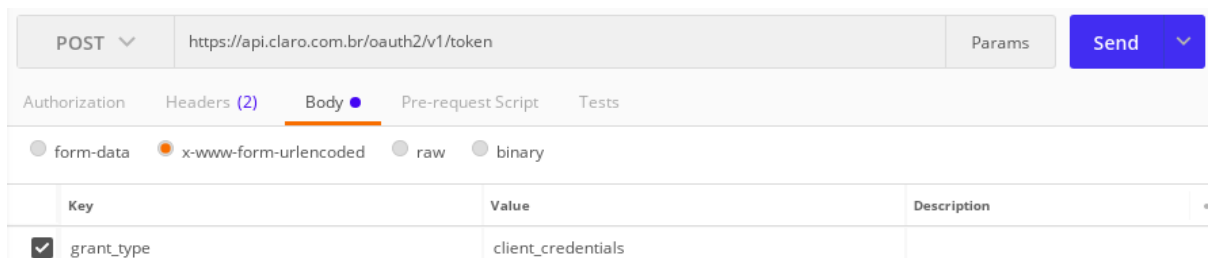
Observação, para gerar a base64 pode usar sites com essa finalidade, passando como entrada o key:secret, a saída deve ser algo como:

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX==

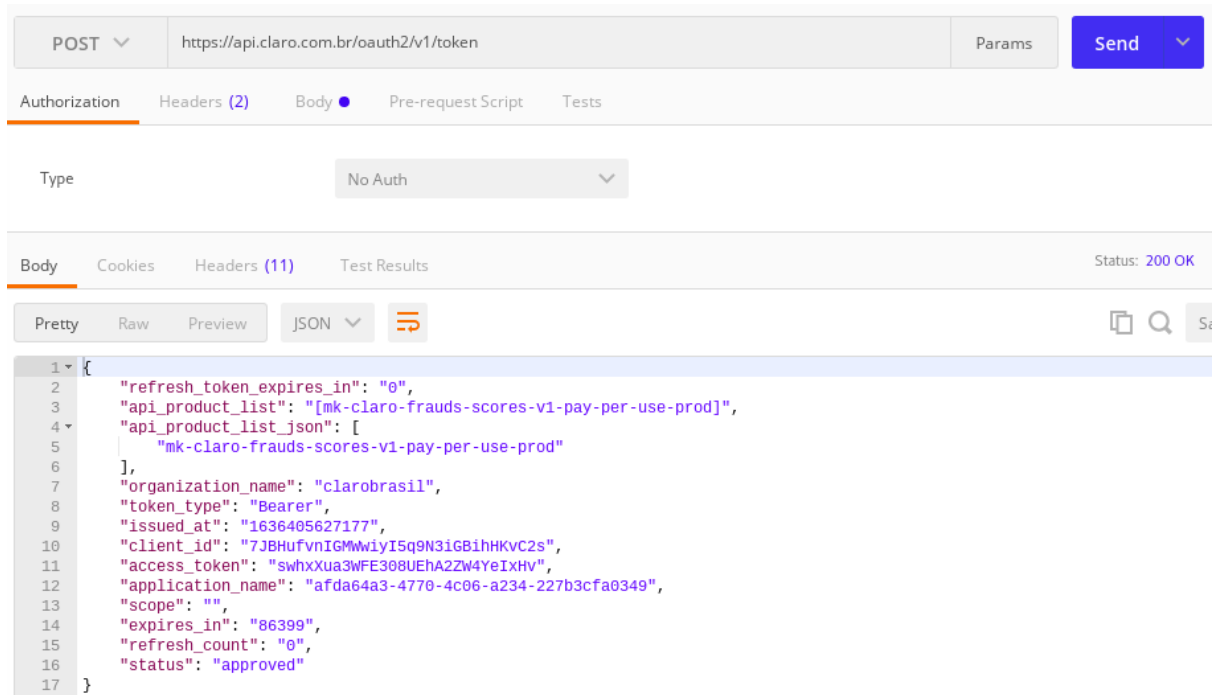


6 – Na guia Body, marque a caixa x-www-form-urlencoded

7 - Na guia Body, adicione a chave de grant_type, seu valor será client_credentials



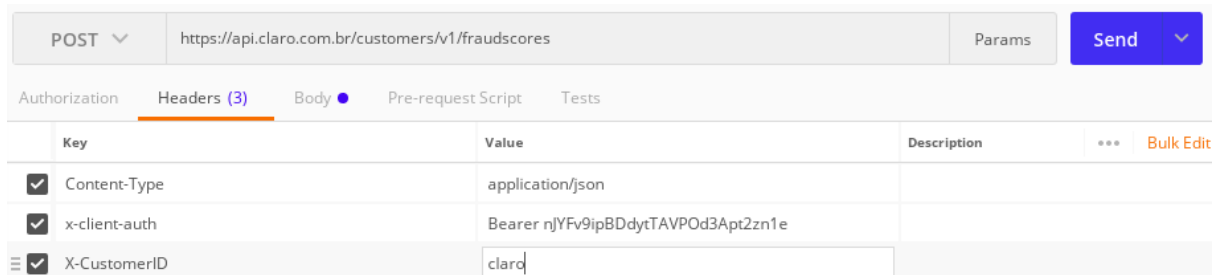
8 - Envie a solicitação, pressionando o botão Envio



9 – 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 Fraud Scores V1

- 1 – Inicie uma nova aba no postman
- 2 – Selecione o método POST
- 3 - Para o URI, insira `https://api.claro.com.br/customers/v1/fraudscores`
- 4 - Na guia Headers, adicione chave de Content-Type e para o valor `application/json`.
- 5 - 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.
- 6 - Na guia Headers, adicione chave de X-CustomerID e para o valor coloque a razão social da sua empresa, sem espaço.



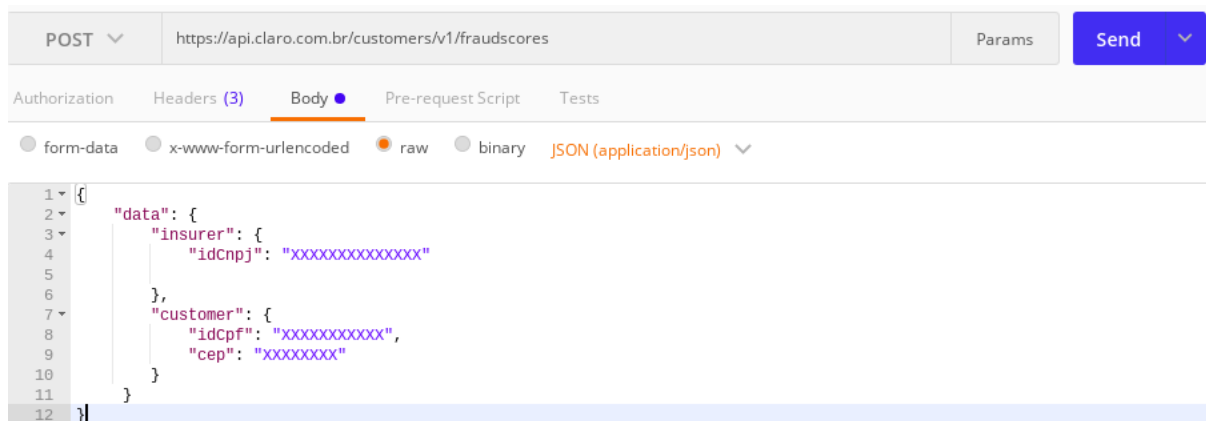
7– Na guia Body, marque a caixa raw e selecione JSON(application/json)

8 – No corpo do Body deve ter o json com o addressZipCode, addressNumber, networkMsisdn e cpf que deseja consultar.

Exemplo:

```
{
  "data": {
    "insurer": {
      "idCnpj": "XXXXXXXXXXXXXXXX"
    },
    "customer": {
      "idCpf": "XXXXXXXXXXXX",
      "cep": "XXXXXXXX"
    }
  }
}
```

9- Envie a solicitação, pressionando o botão Envio



Dados de request

Exemplo de json de request

```
{
  "data": {
    "insurer": {
      "idCnpj": "string"
    },
    "customer": {
      "idCpf": "string",
      "cep": "string"
    }
  }
}
```

Explicação dos dados do json

1. Campo Insurer, obrigatório, tipo JsonStructure, estrutura json que contém os dados da seguradora consultante.
2. Campo idCnpj, obrigatório, tipo string (14 caracteres numéricos), representa o número de CNPJ da seguradora consultante.
3. Campo Customer, obrigatório, tipo JsonStructure, estrutura json que contém os dados do CPF consultado.
4. Campo idCpf, obrigatório, tipo string (11 caracteres numéricos) representa o número do CPF do cliente consultado.
5. Campo cep, obrigatório, tipo string (8 caracteres numéricos) representa o cep do cliente consultado.

Dados de response

Status com retorno com sucesso.

Exemplo de json de retorno

```
{
  "data": {
    "insurer": {
      "idCnpj": string
    },
    "customer": {
      "idCpf": string,
      "CEP": string
    }
  },
  "statsResult": {
    "httpCode": numeric,
    "message": string,
    "flMatch_Claro": boolean,
    "Dias_Datatintc_Sum": "String",
    "Pernextelcomu_Sum": "String",
    "Pervivocomu_Sum": "String",
    "Qtcont360d": "String",
    "Max_Atraso_Aging_M0_M11": "String",
    "Atrasomaxnetcnro": "String",
    "Qtcarta2_360d": "String",
    "Qtdtpoinibnullnet": "String",
    "Flagisin": boolean,
    "Qtpaggbest_270d__Qtpaggbest_360d": "String",
    "Tpmaxbxepag_60d__Tpmaxbxepag_90d": "String",
    "Vlpag": "String",
    "Max_Recsosm0_M5": "String",
    "Maxvlfac360d": "String",
    "Max_Mbresoctothm0_M8": "String",
    "Max_Smstotentm0_M11": "String",
    "Sum_Mbresoctothm0_M5__Sum_Mbtotm0_M5": "String",
    "Tpprireprura": "String",
    "Qttiposolipontad__Qttiposoli": "String",
    "Qtvendaatp__Qtvenda": "String",
  }
}
```

```

"Qtvendas_180d__Qtvendas_270d": "String",
"Avg_Minvc1lhm0_M2__Avg_Minvc1lhm0_M8": "String",
"Avg_Minvc1mhm0_M11": "String",
"Qtdtotsaim0": "String",
"Sum_Minorioutm0_M11": "String",
"Sum_Minvc1lhm0_M11": "String",
"Sum_Minvc3lhm0_M11": "String",
"Sum_Qtdvc1mentsaim0_M11__Sum_Qtdtotoffentsaim0_M11": "String",
"Sum_Qtdvc2mentsaim0_M11": "String",
"Pernextelcomu_Sum_Area432": "Numeric",
"Qtcontrecl360d_Area432": "Numeric",
"Avg_Multas_E_Juros_M0_M11__Avg_Sldo_Aberto_Liq_M0_M11_Area432": "Numeric",
"Vlgstpddnetm1xm6_Area432": "Numeric",
"Tendvlnow9x3m_Max_Area432": "Numeric",
"Qtmodcobrboleto_360d_Area432": "Numeric",
"Sum_Temerbhorcomm0_M11__Sum_Temerbfdsm0_M11_Area432": "Numeric",
"Qtvendaatp__Qtvenda_Area432": "Numeric",
"Sum_Minvc1fhm0_M8_Area432": "Numeric",
"Sum_Minvc1fsaim0_M8__Sum_Qtdtotoffsaim0_M8_Area432": "Numeric",
"Sum_Minvc1lentm0_M11__Sum_Mintotonentm0_M11_Area432": "Numeric",
"Sum_Minvc1lsaim0_M11__Sum_Qtdtotonsaim0_M11_Area432": "Numeric",
"Sum_Minvc1lsaim0_M2__Sum_Qtdtotonsaim0_M2_Area432": "Numeric",
"Sum_Minvc2fentm0_M5_Area432": "Numeric",
"Sum_Qtdintentm0_M5__Avg_Qtdintentm0_M8_Area432": "Numeric",
}
}
}

```

Explicação dos dados do json response

A tabela a seguir possui o detalhamento dos parâmetros recebidos na resposta:

Nome do campo	Opcional ou Obrigatório	Tipo	Descrição do campo
Insurer	Obrigatório	Json Structure	Estrutura json que contém os dados da seguradora consulante

idCnpj	Obrigatório	String (14 caracteres numéricos)	CNPJ da seguradora consultante
Customer	Obrigatório	Json Structure	Estrutura json que contém os dados do CPF consultado
idCpf	Obrigatório	String (11 caracteres numéricos)	CPF do cliente consultado
CEP	Obrigatório	String (8 caracteres numéricos)	CEP do cliente consultado
httpCode	Obrigatório	Numerical	Código de retorno HTTP
message	Obrigatório	String	Representa o Mensagem de retorno
flMatch_Claro	Obrigatório	boolean	Flag que indica que o cpf é ou não cliente claro
Dias_Datatintc_Sum	Obrigatório	String	Total de dias que a pessoa é cliente claro considerando todos os telefones, em categorias.
Pernextelcomu_Sum	Obrigatório	String	A soma do percentual de assinantes NEXTEL no total de assinantes da comunidade, considerando todos os telefones.
Pervivocomu_Sum	Obrigatório	String	O mesmo que PERNEXTELCOMU_SUM, mas com clientes VIVO ao invés de NEXTEL.
Qtcont360d	Obrigatório	String	Total de contatos de (cancelamento, migração, portabilidade e

Max_Atraso_Aging_M0_M11	Obrigatório	String	reclamações claro, anatel, de cobrança ou de produtos e serviços) nos últimos 360 dias. O máximo da quantidade de dias em atraso a partir da primeira data em aberto, considerando os últimos 12 meses.
Atrasomaxnetcnro	Obrigatório	String	Máximo de dias de atraso no produto Net nos últimos 12 meses, em categorias.
Qtcarta2_360d	Obrigatório	String	Quantidade de cartas 2 nos últimos 360 dias, em categorias.
Qtdtpoinibnullnet	Obrigatório	String	Quantidade de "inibição null" em Net dentro de 12 meses, em categorias.
Flagisin	Obrigatório	boolean	Indicador se o cliente está ou não no Book NET Pagamento da Claro.
Qtpaggbest_270d__Qtpaggbest_360d	Obrigatório	String	Razão entre a quantidade de pagamentos em grandes bancos estatais nos últimos 270 dias e nos últimos 360 dias.
Tpmaxbxepag_60d__Tpmaxbxepag_90d	Obrigatório	String	Razão do tempo máximo entre baixa de pagamento e pagamento nos últimos 60 e 90 dias.
Vlpag	Obrigatório	String	Indicador do valor do último do pagamento NET, em categorias.

Max_Recsosm0_M5	Obrigat ório	String	O número máximo de recargas SOS efetuadas num único mês, considerando os últimos 6 meses.
Maxvlfac360d	Obrigat ório	String	Valor máximo de recarga realizado nos últimos 360 dias, em categorias.
Max_Mbresoctothm0_M8	Obrigat ório	String	Valor máximo usado em redes sociais no horário da madrugada, exceto fins de semana, nos últimos 9 meses.
Max_Smstotentm0_M11	Obrigat ório	String	Quantidade máxima de SMS entrante nos últimos 12 meses.
Sum_Mbresoctothm0_M5__Sum_Mbtotm0_M5	Obrigat ório	String	Soma da razão da quantidade de MB utilizados em redes sociais na madrugada nos últimos 6 meses pela quantidade de MB total nos últimos 6 meses.
Tpprirepura	Obrigat ório	String	Indicador de Tempo desde o primeiro contato representante, em categorias.
Qttiposolipontad__Qttiposoli	Obrigat ório	String	Razão da quantidade de solicitação de adesão do tipo "Ponto Adicional" pela quantidade de solicitações de: instalação de assinatura, instalação de ponto adicional, ponto adicional, mudança de pacote.

Qtvendaatp__Qtvenda	Obrigat ório	String	Razão da quantidade de vendas totais efetuada pelo canal "Loja NET" pela quantidade de vendas em todos os canais.
Qtvendas_180d__Qtvendas_270d	Obrigat ório	String	Razão da quantidade de vezes que o cliente apareceu na bases de vendas nos últimos 180 e 270 dias.
Avg_Minvc1lhm0_M2__Avg_Minvc1lhm0_M8	Obrigat ório	String	Razão entre as médias da quantidade de minutos da chamada local em minutos intra rede (Claro x Claro - VC1I) no horário das 00:00h até 06:00h e 22:00h até 23:59h, exceto sábado e domingo) nos últimos 3 meses e nos últimos 9 meses.
Avg_Minvc1mhm0_M11	Obrigat ório	String	Média da quantidade de minutos da chamada local em minutos para móvel (VC1M) no horário das 00:00h até 06:00h e 22:00h até 23:59h, exceto sábado e domingo dos últimos 12 meses.
Qtdtotsaim0	Obrigat ório	String	Quantidades de chamadas saintes do tipo local para fixo(VC1F), intra rede (VC1I), móvel (VC1M), e do tipo longa distância para fixo(INT, VC2F, VC3F), intra

Sum_Minorioutm0_M11	Obrigat ório	String	rede(VC2I, VC3I) e móvel(VC2M e VC3M) no último mês, em categorias. Quantidade de minutos das ligações realizadas com a operadora Outros como origem nos últimos 12 meses.
Sum_Minvc1lhm0_M11	Obrigat ório	String	Quantidade de minutos da chamada local em minutos intra rede (Claro x Claro - VC1I) no horário das 00:00h até 06:00h e 22:00h até 23:59h, exceto sábado e domingo, nos últimos 12 meses.
Sum_Minvc3lhm0_M11	Obrigat ório	String	Soma da quantidade de minutos da chamada longa distância em minutos intra rede (Claro x Claro - VC3I) no horário das 00:00h até 06:00h e 22:00h até 23:59h, exceto sábado e domingo, nos últimos 12 meses.
Sum_Qtdvc1mentsaim0_M11__Sum_Qtdtotoffentsaim0_M11	Obrigat ório	String	Razão da quantidade de chamadas local para móvel (VC1M) -entrante e sainte- nos últimos 12 meses e a quantidade de chamadas local para fixo (VC1F) e móvel (VC1M), e também longa distância para fixo

Sum_Qtdvc2mentsaim0_M11	Obrigatório	String	(INT, VC2F, VC3F) e móvel(VC2M, VC3M) nos últimos 12 meses.
Pernextelcomu_Sum_Area432	Obrigatório	Numeric	Quantidade de chamadas longa distância para móvel (VC2M) - entrante e sainte - nos últimos 12 meses.
Qtcontrecl360d_Area432	Obrigatório	Numeric	A média do atributo individual PERNEXTELCOMU_SUM de uma determinda região.
Avg_Multas_E_Juros_M0_M11__Avg_Sldo_Aberto_Liq_M0_M11_Area432	Obrigatório	Numeric	A média do atributo individual QTCONTRECL360D de uma região.
Vlgstpddnetm1xm6_Area432	Obrigatório	Numeric	A razão entre a (média de multas e juros nos últimos 12 meses) e (média do saldo aberto líquido nos últimos 12 meses) de uma região.
Tendvlnow9x3m_Max_Area432	Obrigatório	Numeric	A média do atributo individual (VIGstPDDNETM1xM6), no mês atual, de uma região.
Qtmodcobrboleto_360d_Area432	Obrigatório	Numeric	A média do atributo individual (TendVINow9x3m_max), no mês atual, de uma região.
Sum_Temerbhorcomm0_M11__Sum_Temerbfdsm0_M11_Area432	Obrigatório	Numeric	A média do atributo individual (QtModCobrBoleto_360d), no mês atual, de uma região.
			A razão entre as médias de (Soma de TEMERBHORCOM nos últimos 12 meses) e (Soma de

Qtvendaatp__Qtvenda_Area432	Obrigat ório	Numeri c	TEMERBFDS nos últimos 12 meses) de uma região. A média do atributo individual (QtVendaATP) por (QtVenda de uma região.
Sum_Minvc1fhm0_M8_Area432	Obrigat ório	Numeri c	A média do atributo individual (Soma de MINVC1FH nos últimos 9 meses) de uma região.
Sum_Minvc1fsaim0_M8__Sum_Qtdtotoffsaim0_M8_ Area432	Obrigat ório	Numeri c	A razão das médias de (Soma de MINVC1FSAI nos últimos 9 meses) e (Soma de QTDTOFFSAI nos últimos 9 meses) de uma região.
Sum_Minvc1lentm0_M11__Sum_Mintotonentm0_M1 1_Area432	Obrigat ório	Numeri c	A razão das médias de (Soma de MINVC1LENT nos últimos 12 meses) e (Soma de MINTOTONENT nos últimos 12 meses) de uma região.
Sum_Minvc1lsaim0_M11__Sum_Qtdtotonsaim0_M11 _Area432	Obrigat ório	Numeri c	A razão das médias de (Soma de MINVC1LSAI nos últimos 12 meses) e (Soma de QTDTONSAI nos últimos 12 meses) de uma região.
Sum_Minvc1lsaim0_M2__Sum_Qtdtotonsaim0_M2_A rea432	Obrigat ório	Numeri c	A razão das médias de (Soma de MINVC1LSAI nos últimos 3 meses) e (Soma de QTDTONSAI nos últimos 3 meses) de uma região.
Sum_Minvc2fentm0_M5_Area432	Obrigat ório	Numeri c	A média do atributo individual (Soma de MINVC2FENT nos

Sum_Qtdintentm0_M5__Avg_Qtdintentm0_M8_Area	Obrigat	Numeri	últimos 6 meses) de uma região.
432	ório	c	A razão das médias de (Soma de QTDINTENT nos últimos 5 meses) e (Média de QTDINTENT nos últimos 9 meses) de uma região.

A tabela a seguir contém a descrição dos valores de erro retornados pelos campos dos subscores:

Código do erro (httpCode)	Mensagem
-100001	Claro não possui registros para a CHAVE no book do qual o subscore faz parte
-100002	Erro no cálculo do subscore (Substitui valores NA no cálculo) - divisão por 0 ou valores Infinitos
-100003	Não há número suficiente de indivíduos na area selecionada para retornar o subscore (específico de subscore de atributos
-99999	Subscore retornou valor abaixo do range esperado
99999	Subscore retornou valor acima do range esperado

Status com retorno com erro.

Exemplo de json de retorno.

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

```
}  
}
```

1. Campo `apiVersion`, obrigatório, tipo `string`, representa a versão da API.
2. Campo `transactionId`, obrigatório, tipo `string`, representa o UUID da transação.
3. Campo `httpCode`, obrigatório, tipo `string`, representa o código de retorno HTTP.
4. Campo `errorCode`, obrigatório, tipo `string`, representa o código de erro Claro.
5. Campo `message`, obrigatório, tipo `string`, representa o Mensagem de erro.
6. Campo `detailedMessage`, obrigatório, tipo `string`, representa o Mensagem detalhada.
7. Campo `rel`, obrigatório, tipo `string`, sempre “related”.
8. Campo `href`, obrigatório, tipo `string`, representa o endereço para o detalhamento dos erros.

Status de response

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 aos dados solicitado no request-URI.

Código de retorno 405 (Not Found)

O método solicitado, não suportado pela API.

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.

Código de retorno 500 (Internal Server Error)

O servidor falhou ao atender uma solicitação aparentemente válida. Entra em contato com Claro.

Código de retorno 502 (Internal Server Error)

O servidor falhou ao atender uma solicitação aparentemente válida. Entrar em contato com a Claro.

Exemplo de código em curl

```
!/bin/bash
```

```
url_fraudscores=https://api.claro.com.br/customers/v1/fraudscores
```

```
url_token=https://api.claro.com.br/oauth2/v1/token
```

```
cnpj="XXXXXXXXXXXXXXXXXX"
```

```
cpf="XXXXXXXXXXXX"
```

```
cep="XXXXXXXXXX"
```

```
key=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
secret=XXXXXXXXXXXX
```

```
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'"' -f32 atributoJson )
```

```
curl -X POST $url_fraudscores -H 'Content-Type: application/json' -H 'cache-control: no-cache' -H "x-client-auth: Bearer $token" -H "X-CustomerID: sua_empresa" -d '{"data": {"insurer": {"idCnpj": "$cnpj"},"customer": {"idCpf": "$cpf","cep": "$cep"}}}'
```

Exemplo de código em python

```
import requests, json
```

```
import base64
```

```
KEY="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

```
SECRET="XXXXXXXXXXXXXXXXXXXX"
```

```
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_VALIDATIONSPHONES = "/customers/v1/fraudscores";
```

```
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('Erro:' + str( respToken.status_code))
```

```
else:
```

```
    token = json.loads( respToken.text)['access_token']
```

```
headers_app = {'Content-Type': 'application/json',
```

```
                'X-CustomerID': 'sua_empresa',
```

```
'cache-control': 'no-cache',  
'user-agent': 'curl/7.60.0',  
'x-client-auth': 'Bearer {}'.format(token)}
```

```
cnpj="XXXXXXXXXXXXXXXXX"  
cpf="XXXXXXXXXXXXX"  
cep="XXXXXXXXX"  
  
data_app = {"data": {"insurer": {"idCnpj": cnpj}, "customer": {"idCpf": cpf, "cep":  
cep}}}  
  
resp = requests.post( URL + END_POINT_API_VALIDATIONSPHONES,  
headers=headers_app , data= json.dumps(data_app))  
  
if resp.status_code != 200:  
    print('Erro: ' + str(resp.status_code))  
else:  
    print('Válida ' + str( resp.text ))
```

Exemplo de código em java

```
package client.fraudScores;  
  
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.util.MultiValueMap;
import org.springframework.web.client.RestClientException;
import org.springframework.web.client.RestTemplate;


public class App {

    private static final String KEY =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";

    private static final String SECRET = "XXXXXXXXXXXXXXXXXXXX";

    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_FRAUDSCORES =
"/customers/v1/fraudscores";


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

        App app = new App();

        String token = app.getAccessToken();

        String cnpj="XXXXXXXXXXXXXXXX";

```

```

String cpf="XXXXXXXXXXXX";
String cep="XXXXXXXX";

    System.out.println("FraudScore: " + app.getFraudScore( token , cnpj, cpf ,
cep));
}

    public String getFraudScore(String acessToken, String cnpj, String cpf , String
cep) throws RestClientException, URISyntaxException {

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

        HttpEntity<Request> request = new HttpEntity<>(new Request(cnpj, cpf ,
cep), headers);

        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> result = restTemplate.postForEntity(new URI(URL +
END_POINT_API_FRAUDSCORES),
            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 Request{

```

```

    public Request(String cnpj, String cpf, String cep) {
        super();

```

```

        Customer customer = new Customer(cpf, cep);
        Insurer insurer = new Insurer(cnpj);

```

```

        this.data = new Data(customer, insurer);
    }
    private Data data;

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

class Data{

    public Data(Customer customer, Insurer insurer) {
        super();
        this.customer = customer;
        this.insurer = insurer;
    }

    private Customer customer;
    private Insurer insurer;

    public Customer getCustomer() {
        return customer;
    }

    public void setCustomer(Customer customer) {

```

```
        this.customer = customer;
    }

    public Insurer getInsurer() {
        return insurer;
    }

    public void setInsurer(Insurer insurer) {
        this.insurer = insurer;
    }
}
```

```
class Insurer{

    public Insurer(String idCnpj) {
        super();
        this.idCnpj = idCnpj;
    }

    private String idCnpj;

    public String getIdCnpj() {
        return idCnpj;
    }

    public void setIdCnpj(String idCnpj) {
        this.idCnpj = idCnpj;
    }

}
```

```
class Customer{

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

    private String idCpf;
    private String cep;

    public String getIdCpf() {
        return idCpf;
    }
    public void setIdCpf(String idCpf) {
        this.idCpf = idCpf;
    }
    public String getCep() {
        return cep;
    }
    public void setCep(String cep) {
        this.cep = cep;
    }
}
}
```

Exemplo de código em php

<?php


```
$key="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
```

```
$secret="XXXXXXXXXXXX";
```

```
$cnpj="XXXXXXXXXXXXXXXX";
```

```
$cpf="XXXXXXXXXXXX";
```

```
$cep="XXXXXXX";
```

```
$token = getAccessToken();
```

```
$fraudscore = getFraudScore($token , $cnpj, $cpf, $cep);
```

```
echo $fraudscore;
```

```
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-client-auth: Basic ' . $senha
```

```
        ]);
```

```
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
        curl_setopt($ch, CURLINFO_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 getFraudScore($token , $cnpj, $cpf, $cep){

    try {

        $payload = '{"data": {"insurer": {"idCnpj":"' . $cnpj. '"}, "customer": {"idCpf": "' . $cpf. '" , "cep": "' . $cep. '"}}}';

        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "https://api.claro.com.br/customers/v1/fraudscores");
    }
}

```

```

curl_setopt( $ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
    'X-CustomerID: sua_empresa',
    'x-client-auth: Bearer ' . $token

]);

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLINFO_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;
}

}

?>

```