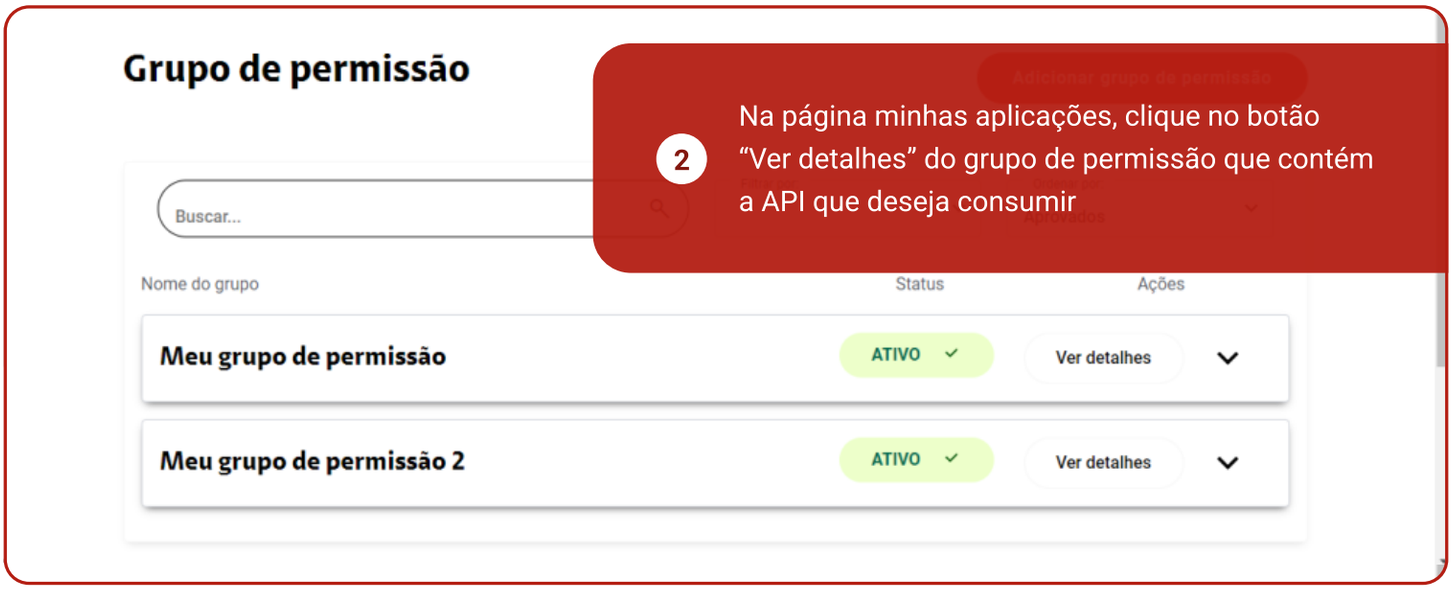
Tutorial de consumo da API Silent Mobile Verification (SMV)

Claro

Para iniciar o consumo da **API Silent Mobile Verification (SMV)**, é necessário adquirir o token de acesso através da chave pública e privada da aplicação contratada no portal.

» Como gerar um token na Base64



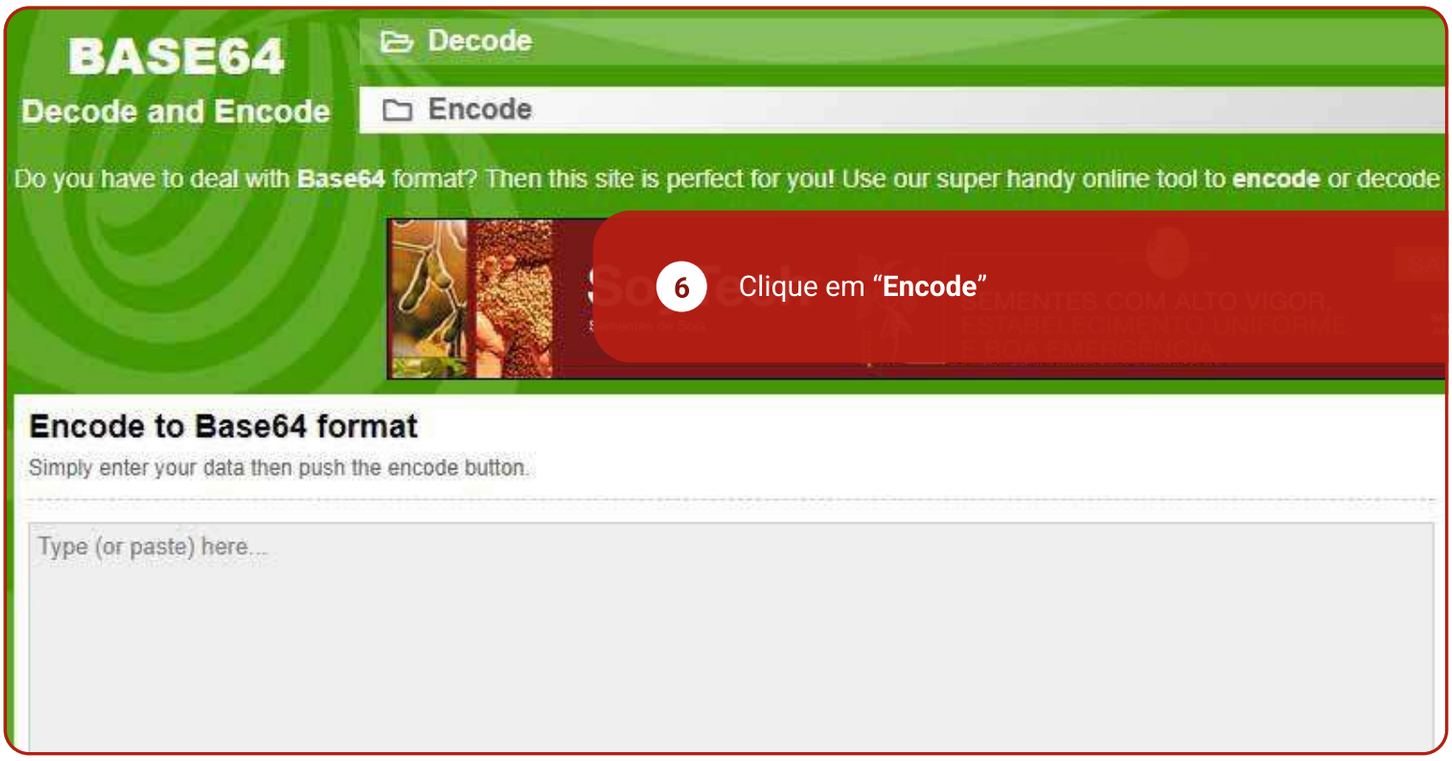
2 Na página minhas aplicações, clique no botão "Ver detalhes" do grupo de permissão que contém a API que deseja consumir

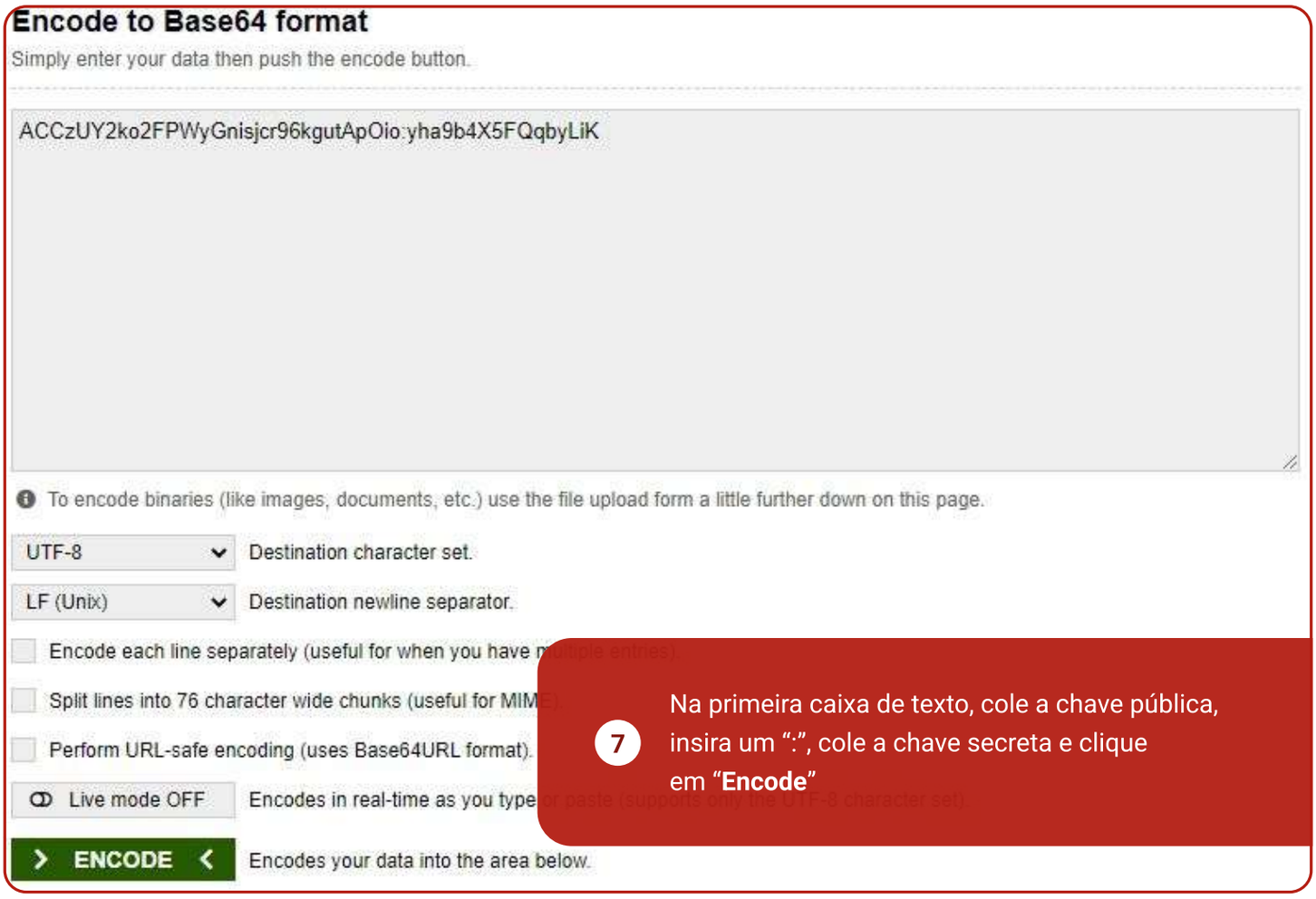
3 Clique na seta para ver os detalhes da aplicação desejada

4 Visualize as chaves pública e secreta, clique em "**Baixar arquivos**" e guarde a collection para utilizar mais tarde

Analytics

5 Acesse o site https://www.base64encode.org/.





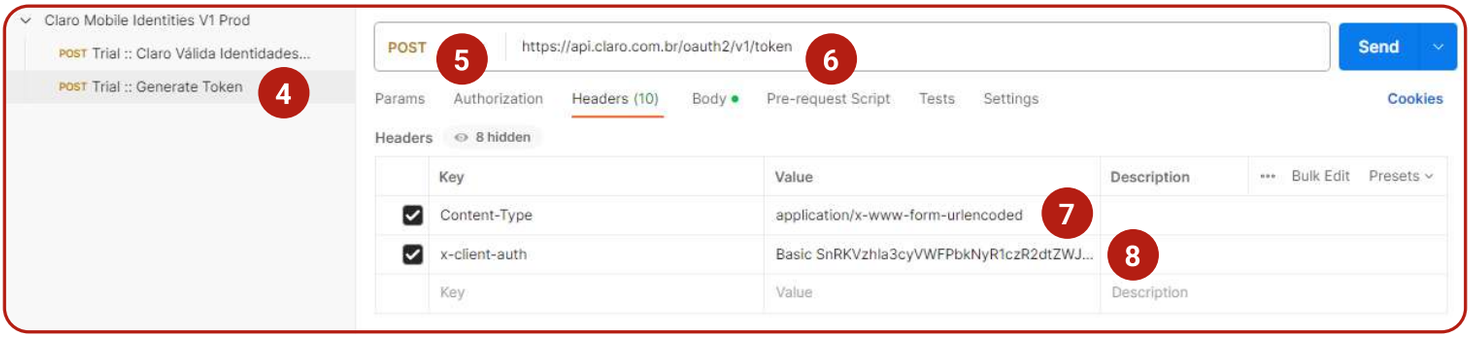
Ex: ACCzUY2ko2FPWyGnisjcr96kgutApOio : yha9b4X5FQqbyLiK

chave pública : **chave secreta**

O token não tem prazo de validade e pode ser utilizado por um período indeterminado, contanto que o status da aplicação permaneça **aprovado**

Usar o postman para obter o access token da API Silent Mobile Verification (SMV)

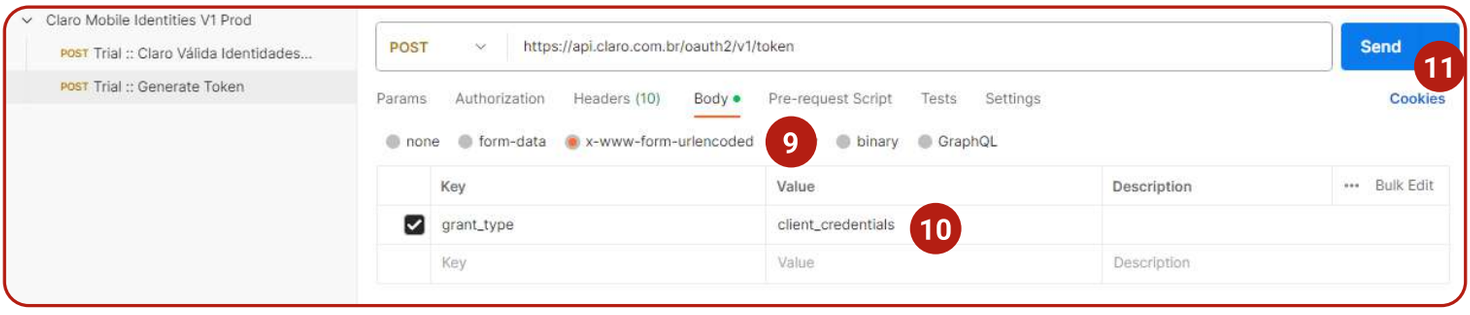
1. Inicie o postman
2. Clique no botão **Importar**
3. Selecione a **collection** baixada no passo anterior
4. Clique na opção **Generate Token**
5. Certifique-se que o método **POST** esteja selecionado
6. A URL inserida deverá ser **https://api.claro.com.br/oauth2/v1/token**
7. A guia **Headers** deve conter a chave **Content-Type** com o valor **application/x-www-form-urlencoded**
8. Troque o valor da chave **x-client-auth** colando o valor gerado na **base64** após o texto Basic



9 Na guia **Body**, a caixa **x-www-form-urlencoded** deve estar marcada

10 A guia **Body** deve conter a chave **grant-type** com o valor **client_credentials**

11 Envie a solicitação pressionando o botão **Envio**



12 Salve o **access token** para consumir a API

O **access token** gerado tem 24 horas de validade e deve ser solicitado novamente após esse período

» Usar o postman para consumir a API Silent Mobile Verification (SMV)

1 Na **collection** da API, clique na opção **Claro Valida Identidades Móveis v1**

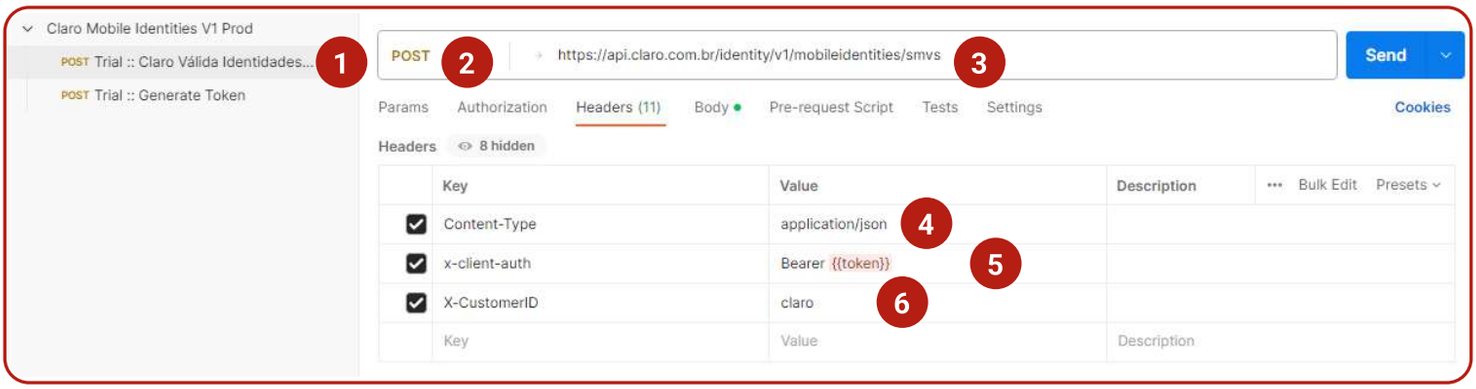
2 Certifique-se que o método **POST** esteja selecionado

3 A URL inserida deverá ser **https://api.claro.com.br/identity/v1/mobileidentities/smvs**

4 A guia **Headers** deve conter a chave **Content-Type** com o valor **application/json**

5 Troque o valor da chave **x-client-auth** colando o valor do **access token** após o texto Bearer

6 Na chave **X-CustomerID**, coloque o **nome da sua empresa** informado na criação do grupo de permissão



7 Na guia **Body**, a caixa **raw** e a seleção **JSON** devem estar marcados

8 Informe o **número de telefone**, o **IP**, a **antena da localidade** e o **horário do evento** do cliente que deseja consultar

9 Envie a solicitação pressionando o botão **Envio**

» Explicação dos dados da solicitação

```
Exemplo de json de request
{
    "data": {
        "msisdn": string,
        "ip": string,
        "port": int,
        "timestamp": int,
    }
}
```

Explicação dos dados do json

1. Campo **msisdn**, obrigatório, tipo string (13 caracteres), representa o número de telefone com dígito do país e estado do cliente
2. Campo **ip**, obrigatório, tipo string (15 caracteres), representa o ip com a separação por pontos
3. Campo **port**, opcional, tipo int (3 dígitos), representa a antena da localidade
4. Campo **timestamp**, opcional, tipo int (11 dígitos), representa o horário do evento

» Dados de retorno com sucesso e erro

Status de retorno com sucesso

```
Exemplo de json de retorno
{
    "exists": false | true
    "message": string
}
```

Explicação dos dados do json

1. Campo **exists**, obrigatório, tipo string, representa a mensagem de sucesso ou erro da consulta
2. Campo **message**, obrigatório, tipo string, representa a mensagem detalhada da mensagem

Status de retorno com erro

```
Exemplo de json de retorno:
{
    "apiVersion": “”,
    "transactionId": “”,
    "error": {
        "detailedMessage": "string",
        "errorCode": "string",
        "message": "string",
        "httpCode": "string",”link”: {
            "href": "string",
            "rel": "string"
        }
    }
}
```

Explicação dos dados do json

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

» Status de response

Código de retorno 200 (Success)

O usuário está com a conexão estabelecida na rede da operadora

Código de retorno 202 (Accepted Data)

Caso o usuário esteja desconectado da rede, ou seja, em modo avião, offline ou sem sinal, desde que esta condição tenha sido identificada pela rede da operadora

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 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 (dados não encontrados na base)

Código de retorno 418 (Out of Range)

O IP está fora do alcance da operadora

Código de retorno 500 (Internal Server Error)

Erro interno do servidor

Exemplo de código em curl

```
!/bin/bash

url_mobile=https://api.claro.com.br/identity/v1/mobileidentities/smvs
url_token=https://api.claro.com.br/oauth2/v1/token
msisdn="XXXXXXXXXXXXX"
ip="XXX.XXX.XXX.XXX"
port=XXX
timestamp=XXXXXXXXXXX
key=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
secret=XXXXXXXXXXXXX

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 )

echo $token

curl -X POST $url_mobile -H 'Content-Type:
application/json' -H 'cache-control: no-cache' -H "x-client-auth:
Bearer $token" -H "X-CustomerID: claro" -d '{"data":
{"msisdn": "'$msisdn'","ip": "'$ip'","port": '$port','timestamp": '$timestamp'}}'
```

Exemplo de código em phyton

```
import requests, json
import base64

KEY="XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SECRET="XXXXXXXXXXXX"

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_MOBILE = "/identity/v1/mobileidentities/smvs";

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


   msisdn = "XXXXXXXXXXXXX"
   ip= "XXX.XXX.XXX.XXX"
   port=XXX
   timestamp=XXXXXXXXXXX
   data_app = {"data":{"msisdn": msisdn, "ip": ip ,"port": port
"timestamp": timestamp}}

   respMobile = requests.post( URL + END_POINT_API_MOBILE,
headers=headers_app ,
data= json.dumps(data_app))

   if respMobile.status_code != 200:
      print('Erro: ' + str(respMobile.status_code))
   else:
      print('Mobile ' + str( respMobile.text ))
```

» Exemplo de código em java

```
package client.mobile;

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 =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
```

```
        private static final String SECRET = "XXXXXXXXXXXXXXXX";

        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_ALERTS =
"/identity/v1/mobileidentities/smvs";

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

        App app = new App();

        String token = app.getAccessToken();
        String msisdn = "XXXXXXXXXXXXX"
        String ip = "XXX.XXX.XXX.XXX";
        int port = XXX;
        float timestamp = XXXXXXXXXX;

        System.out.println("Mobile: " +
app.getMobile( token , msisdn , ip, port, timestamp ));
    }

    public String getMobile(String acessToken, String
msisdn, String ip, int port, float timestamp) 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", "claro_exemplo");

        HttpEntity<Request> request = new HttpEntity<>(new
Request(msisdn, ip, port, timestamp), headers);

        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> result =
restTemplate.postForEntity(new URI(URL +
END_POINT_API_ALERTS),
                            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 msisdn , String ip ,int port, float timestamp) {
                        this.data = new Data(msisdn, ip, port, timestamp);
                }

                private Data data;

                public Data getData() {
                        return data;
                }

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

    class Data{

        public Data(String msisdn, String ip, int port, float timestamp) {
                        super ();
                        this.msisdn = msisdn;
                        this.ip = ip;
                        this.port = port;
                        this.timestamp = timestamp;
                }

                private String msisdn;
                private String ip;
                private int port;
                private float timestamp;

                public String getMsisdn() {
                        return msisdn;
                }

                public void setMsisdn(String msisdn) {
                        this.msisdn = msisdn;
                }

                public String getIp () {
                        return ip;
                }

                public void setIp(String ip) {
                        this.ip = ip;
                }

                public int getPort() {
                        return port;
                }

                public void setPort(int port) {
                        this.port = port;
                }
```

```
        public float getTimestamp () {
            return timestamp;
        }

        public void setTimestamp (float timestamp) {
            this.timestamp = timestamp;
        }

    }
}
```

» Exemplo de código em php

```
<?php

$key = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
$secret = 'XXXXXXXXXXXXXX';
$msisdn = "XXXXXXXXXXXXX";
$ip = "XXX.XXX.XXX.XXX";
$port=XXX;
$timestamp=XXXXXXXXXXX;
$token = getAccessToken();
$mobile = getMobile($token , $msisdn, $ip, $port, $timestamp);
echo $mobile;
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 getMobile($token , $msisdn, $ip $port, $timestamp){

   try {

      $payload = '{"data": {"msisdn": ' . $msisdn . ' ,"ip": ' . $ip .
',"port": ' . $port . ',"timestamp": ' . $timestamp . '}}';
      $ch = curl_init();
      curl_setopt($ch, CURLOPT_URL,
"https://api.claro.com.br/identity/v1/mobileidentities/smvs");
      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, 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;
   }
}
?>
```