

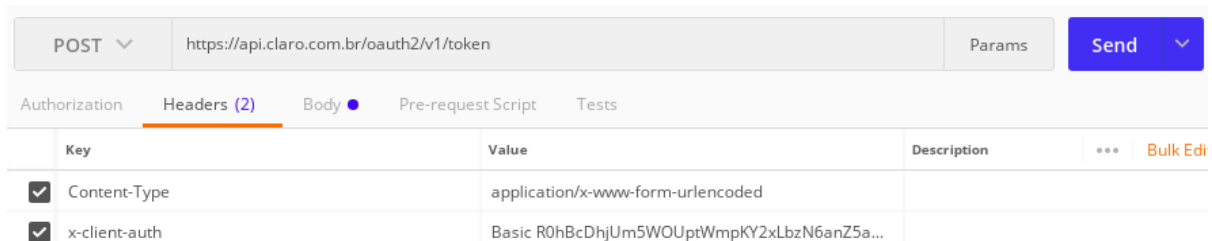
Tutorial de consumo da API Mobile Identities V1

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.

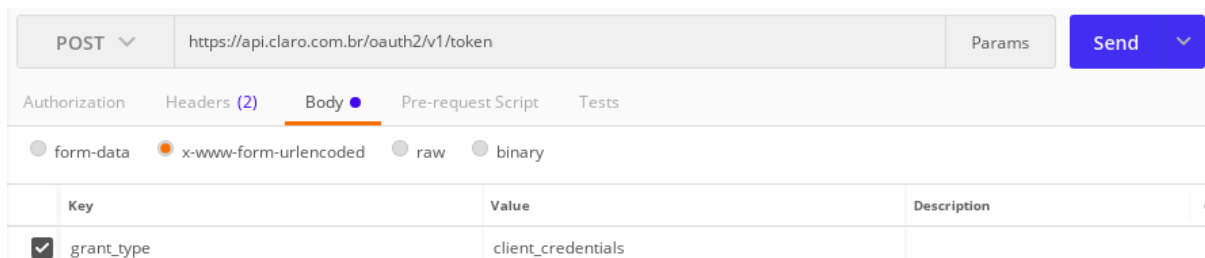
Usar o postman para obter o token da API Mobile Identities V1

- 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:
XXX==



- 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

POST <https://api.claro.com.br/oauth2/v1/token> Params Send

Authorization Headers (2) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (11) Test Results Status: 200 OK

Pretty Raw Preview JSON

```

1 {
2   "refresh_token_expires_in": "0",
3   "api_product_list": "[mk-claro-identity-mobileidentities-smvs-v1-pay-per-use-prod]",
4   "api_product_list_json": [
5     "mk-claro-identity-mobileidentities-smvs-v1-pay-per-use-prod"
6   ],
7   "organization_name": "claro-brasil",
8   "token_type": "Bearer",
9   "issued_at": "1639162677477",
10  "client_id": "JtJW8ekw2Ua0ncrGw3GgmebMiu5yxvDs",
11  "access_token": "8Fe49joEzArh69ALndpdJPiWsrMq",
12  "application_name": "4da6c721-b0fe-4894-948a-9f1f266fa14f",
13  "scope": "",
14  "expires_in": "86399",
15  "refresh_count": "0",
16  "status": "approved"
17 }

```

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 Mobile Identities V1

- 1 – Inicie uma nova aba no postman
- 2 – Selecione o método POST
- 3 - Para o URI, insira <https://api.claro.com.br/identity/v1/mobileidentities/smvs>
- 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.

POST <https://api.claro.com.br/identity/v1/mobileidentities/smvs> Params Send

Authorization Headers (3) Body Pre-request Script Tests

Key	Value	Description
<input checked="" type="checkbox"/> Content-Type	application/json	
<input checked="" type="checkbox"/> x-client-auth	Bearer WdGcAx9qVAfwPeVMADLmFv8oV6I4	
<input checked="" type="checkbox"/> X-CustomerID	claro	

... Bulk Edit

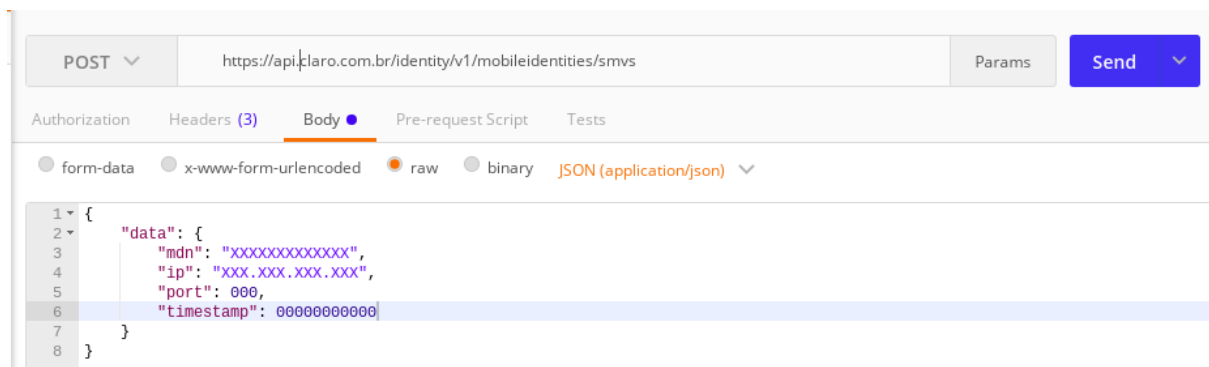
7– Na guia Body, marque a caixa raw e selecione JSON(application/json)

8 – No corpo do Body deve ter o json

Exemplo:

```
{
  "data": {
    "mdn": "XXXXXXXXXXXXXXXX",
    "ip": "XXX.XXX.XXX.XXX",
    "port": 000,
    "timestamp": 00000000000
  }
}
```

9- Envie a solicitação, pressionando o botão Envio



Dados de request

Exemplo de json de request

```
{
  "data": {
    "mdn": string,
    "ip": string,
    "port": int,
    "timestamp": int
  }
}
```

Explicação dos dados do json

1. Campo mdn, obrigatório, tipo string (13 caracteres numéricos), representa número do telefone com dígito do país e estado.
2. Campo ip, obrigatório, tipo string (15 caracteres), representa o ip, com a separação por pontos
3. Campo port, tipo int (3 dígitos), representa a antena da localidade
4. Campo timestamp, tipo int (11 dígitos) representa o horário do evento
5. Campo cep, obrigatório, tipo string (8 caracteres numéricos) representa o cep do cliente consultado.

Dados de response

Status com retorno com sucesso.

```
{
  "exists": false | true
  "message": "mensagem detalhando a existencia ou não da informação."
}
```

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_mobile=https://api.claro.com.br/identity/v1/mobileidentities/smvs
```

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

```
mdn="XXXXXXXXXXXXXXXX"
```

```
ip="XXX.XXX.XXX.XXX"
```

```
port=XXX
```

```
timestamp=XXXXXXXXXXXX
```

```
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'"' -f36 atributoJson )
```

```
echo $token
```

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

Exemplo de código em python

```
import requests, json
```

```
import base64
```

```
KEY="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

```
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)}

    mdn = "XXXXXXXXXXXXXXXX"
    ip= "XXX.XXX.XXX.XXX"
    port=XXX
    timestamp=XXXXXXXXXXXX
    data_app = {"data": {"mdn": mdn,"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


```

App app = new App();
String token = app.getAccessToken();
String mdn = "XXXXXXXXXXXXXXXX";
String ip = "XXX.XXX.XXX.XXX";
int port = XXX;
float timestamp = XXXXXXXXXXXX;
System.out.println("Mobile: " + app.getMobile( token , mdn, ip, port, timestamp
));
}

```

```

public String getMobile(String accessToken, String mdn, 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 " + accessToken);
    headers.add("X-CustomerID", "claro");

```

```

    HttpEntity<Request> request = new HttpEntity<>(new Request(mdn, 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 mdn, String ip, int port, float timestamp) {
            this.data = new Data(mdn, ip, port, timestamp);
        }
        private Data data;
        public Data getData() {
            return data;
        }
    }

```

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

```
class Data{  
    public Data(String mdn, String ip, int port, float timestamp) {  
        super();  
        this.mdn = mdn;  
        this.ip = ip;  
        this.port = port;  
        this.timestamp = timestamp;  
    }  
  
    private String mdn;  
    private String ip;  
    private int port;  
    private float timestamp;  
  
    public String getMdn() {  
        return mdn;  
    }  
  
    public void setMdn(String mdn) {  
        this.mdn = mdn;  
    }  
  
    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 = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
$secret = 'XXXXXXXXXXXXXXXXXX';
$mdn="XXXXXXXXXXXXXXXXXX";
$ip="XXX.XXX.XXX.XXX";
$port=XXX;
$timestamp=XXXXXXXXXXXX;
$token = getAccessToken();
$mobile = getMobile($token , $mdn, $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 , $mdn, $ip, $port, $timestamp){
    try {
        $payload = '{"data": {"mdn": ' . $mdn . ', "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',
            '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;
}
}
?>
```