Tutorial de consumo da API Validation Adress V2

Claro

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 Validation Address V2

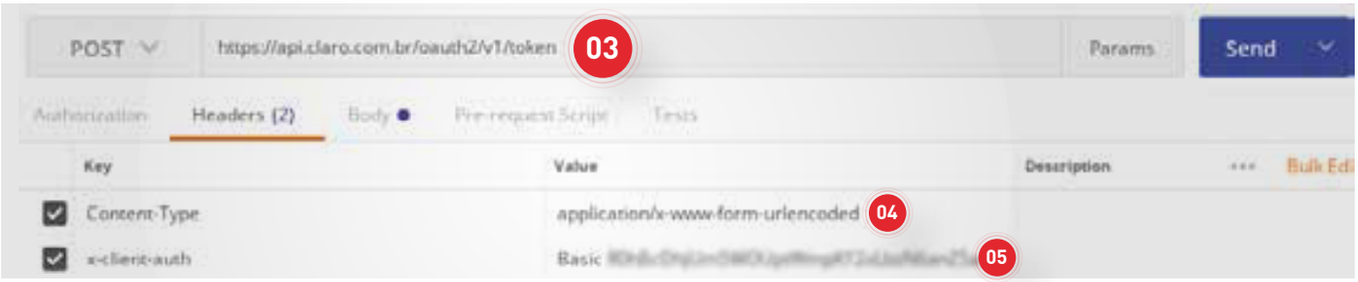
01 Inicie o postman.

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).

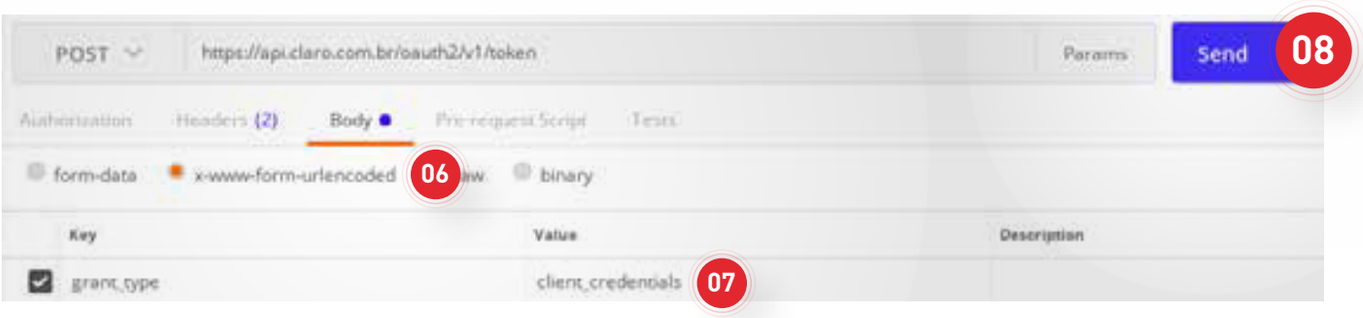


Observação, para gerar a base64 pode usar sites com essa finalidade, passando como entrada o key:secret, a saída deve ser algo como: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX==

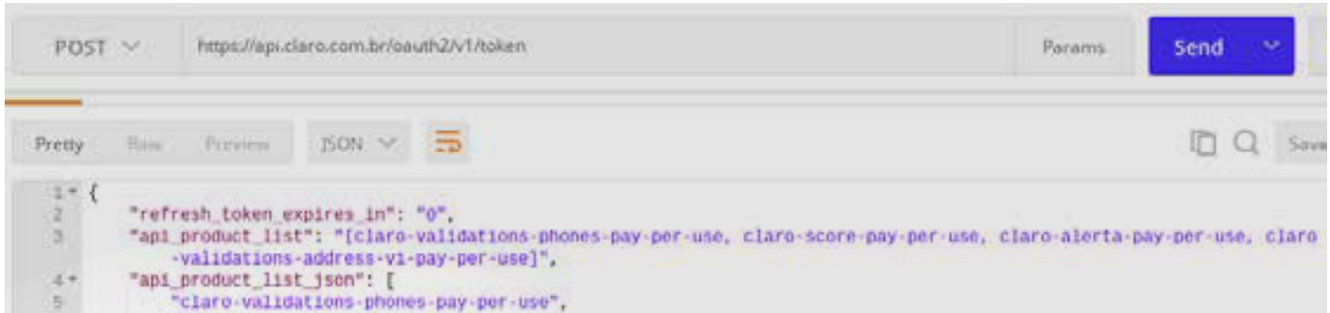
06 Na guia Body, marque a caixa x-www-form-urlencoded

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.



```
        "claro-score-pay-per-use",
        "claro-alerta-pay-per-use",
        "claro-validations-address-v1-pay-per-use"
    ],
    "organization_name": "clarobrasil",
    "developer.email": "jefferson.barbosa@claro.com.br",
    "token_type": "BearerToken",
    "issued_at": "1663208186905",
    "client_id": "GHAp8cRnv9JmZjJclXo3zjvykn6v6F3p",
    "access_token": [redacted],
    "application_name": "1b606b2e-860f-42ca-a402-7803847 5c896",
    "scope": "",
    "expires_in": "86399",
    "refresh_count": "0",
    "status": "approved"
}
```


09

» Usar o postman para consumir API Validation Address V2

01 Inicie uma nova aba no postman

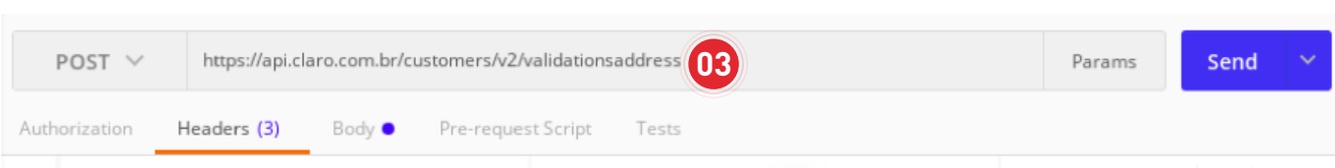
02 Selecione o método POST

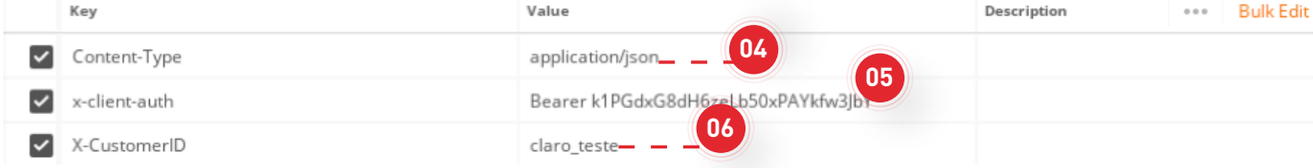
03 Para o URI, insira https://api.claro.com.br/customers/v2/validationsaddress

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, substituía 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.



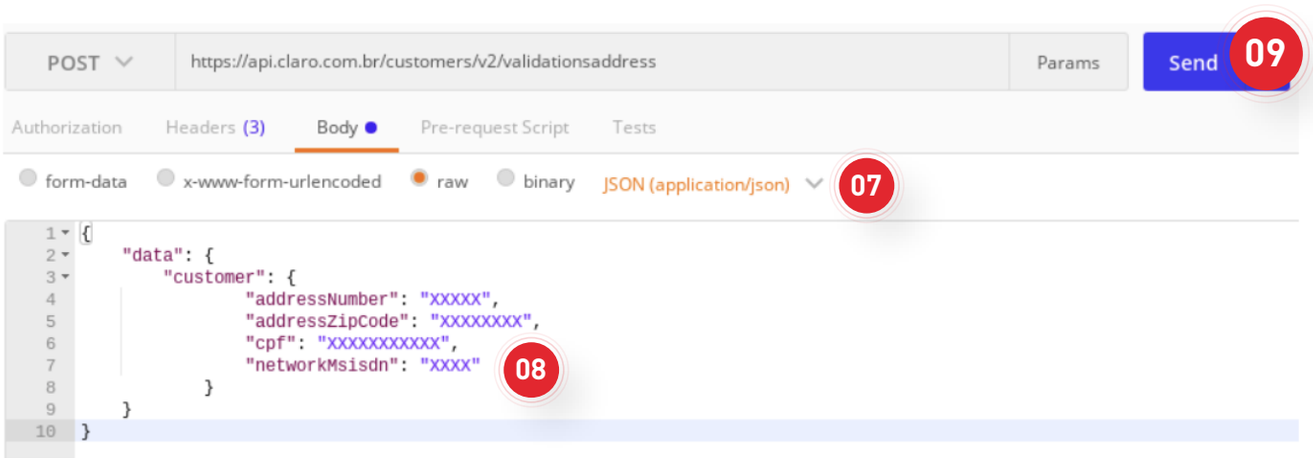


07 Na guia Body, marque a caixa raw e selecione JSON(application/json)

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

```
Exemplo:
{
    "data": {
        "customer": {
            "addressZipCode":"XXXXXXXX",
            "addressNumber": "XXX",
            "cpf": "XXXXXXXXXXX"
            "networkMsisdn": "XXXX"
        }
    }
}
```

09 Envie a solicitação, pressionando o botão Envio



» Dados de request

```
Exemplo de json de request
{
    "data": {
        "customer": {
            "addressZipCode":"string",
            "addressNumber": "string",
            "cpf": "string"
            "networkMsisdn": "string"
        }
    }
}
```

Explicação dos dados do json

01 Campo addressZipCode, obrigatório, tipo string (5-8 caracteres) representa o cep que deseja consultar.

02 Campo addressNumber, opcional, tipo string (1-6 caracteres) representa o número do endereço que deseja consultar.

03 Campo cpf, opcional, se tiver o networkMsisdn, tipo string (11 caracteres) representa o cpf do cliente.

04 Campo networkMsisdn, opcional, se tiver o cpf, tipo string (1-15 caracteres) representa o número do telefone do cliente.V

» Dados de response

Status com retorno com sucesso.

```
{
 "data": {
  "result": {
   "message": "string",
   "score": “integer”
  }
 }
}
```

Explicação dos dados do json

01 Campo message, obrigatório, tipo string, mensagem com uma interpretação descritiva do resultado obtido

02 Campo score, obrigatório, tipo int, score do CEP fornecido para o cliente encontrado

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"
        }
    }
}
```

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 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 ao 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.

» Exemplo de código em curl

```
!/bin/bash

url_validationsaddress=https://api.claro.com.br/customers/v2/valid
ationsaddress
url_token=https://api.claro.com.br/oauth2/v1/token
addressZipCode="XXXXXXXX"
addressNumber="XXXXX"
cpf="XXXXXXXXXXX"
networkMsisdn="XXXXXXXXXXXXX"
key=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
secret=XXXXXXXXXXXXXX

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_validationsaddress -H 'Content-Type:
application/json' -H 'cache-control: no-cache' -H "x-client-auth:
Bearer $token" -H "X-CustomerID: claro_exemplo"  -d '{"data":
{"customer": {"addressZipCode":
"'$addressZipCode'","addressNumber": "'$addressNumber'",
"networkMsisdn":"'$networkMsisdn'" ,"cpf": "'$cpf'"}}}'
```

Exemplo de código em python

```
import requests, json
import base64

KEY="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SECRET="XXXXXXXXXXXXXX"

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_VALIDATIONSADDRESS =
"/customers/v2/validationsaddress";

headers={'Content-Type':'application/x-www-form-urlencoded','cach
e-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_exemplo',
                   'cache-control':'no-cache',
                   'user-agent':'curl/7.60.0',
                   'x-client-auth': 'Bearer {}'.format(token)}

    addressZipCode="XXXXXXX"
    addressNumber="XXXXX"
    cpf="XXXXXXXXXXX"
    networkMsisdn="XXXXXX"

    data_app = {"data": {"customer": {"addressZipCode":
addressZipCode,"addressNumber": addressNumber ,"cpf": cpf ,
"networkMsisdn":networkMsisdn}}}

    resp = requests.post( URL +
END_POINT_API_VALIDATIONSADDRESS, headers=headers_app ,
data= json.dumps(data_app))

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

» Exemplo de código em java

```
package client.validationsAddress;

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 = "XXXXXXXXXXXXXX";

        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_VALIDATIONS_ADDRESS =
"/customers/v2/validationsaddress";


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

        App app = new App();

        String token = app.getAccessToken();
        String addressZipCode = "XXXXXXXX";
        String addressNumber = "XXXXX";
        String cpf = "XXXXXXXXXXX";
        String networkMsisdn="XXXXXXXXXXXXX";

        System.out.println("Válida Endereço: " +
app.getValidationsAddress( token , addressZipCode , addressNumber
, cpf,networkMsisdn ));
   }

   public String getValidationsAddress(String acessToken, String
```

```
addressZipCode, String addressNumber, String cpf , String
networkMsisdn) 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<NetworkMsisdn> request = new HttpEntity<>(new
NetworkMsisdn(addressZipCode , addressNumber , cpf ,
networkMsisdn), headers);

        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> result =
restTemplate.postForEntity(new URI(URL +
END_POINT_API_VALIDATIONS_ADDRESS),
                                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_URLENCOD
ED);
        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").asTex
t() ;

    }

    class NetworkMsisdn{

        public NetworkMsisdn(String addressZipCode,String
addressNumber , String cpf, String networkMsisdn) {
                        this.data = new Data(addressZipCode,
addressNumber , cpf,networkMsisdn);
                }

                private Data data;

                public Data getData() {
                        return data;
                }

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

    class Data{

        public Data(String addressZipCode,String addressNumber ,
String cpf, String networkMsisdn) {
                        this.customer = new Customer(addressZipCode ,
addressNumber , cpf, networkMsisdn);
                }

                private Customer customer;

        public Customer getCustomer() {
                        return customer;
                }

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

                class Customer{
```

```
		public Customer(String addressZipCode,String
addressNumber , String cpf, String networkMsisdn) {
			this.addressZipCode = addressZipCode;
			this.addressNumber = addressNumber;
			this.cpf = cpf;
			this.networkMsisdn = networkMsisdn;
		}

		private String cpf;

		private String networkMsisdn;

		private String addressNumber;

		private String addressZipCode;

		public String getCpf() {
			return cpf;
		}

		public void setCpf(String cpf) {
			this.cpf = cpf;
		}

		public String getAddressNumber() {
			return addressNumber;
		}

		public void setAddressNumber(String
addressNumber) {
			this.addressNumber = addressNumber;
		}

		public String getAddressZipCode() {
			return addressZipCode;
		}

		public void setAddressZipCode(String
addressZipCode) {
			this.addressZipCode = addressZipCode;
		}

	}
  }
}
```

» Exemplo de código em php

```
<?php

$key = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';

$secret = 'XXXXXXXXXXXX';

$addressZipCode = 'XXXXXXX';

$addressNumber = 'XXXX';

$cpf = 'XXXXXXXXXXX';

$networkMsisdn = 'XXXXXXXXXXXXX';

$token = getAccessToken();

$validationsaddress = getValidationsAddress($token ,
$addressZipCode, $addressNumber , $cpf, $networkMsisdn);

echo $validationsaddress;

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 getValidationsAddress($token , $addressZipCode,
$addressNumber , $cpf , $networkMsisdn){

    try {

            $payload = '{"data": {"customer": {"addressZipCode":' .
$addressZipCode . ',"addressNumber": ' . $addressNumber
.',"networkMsisdn":'.$networkMsisdn.' , "cpf": ' . $cpf . '}}}';
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL,
"https://api.claro.com.br/customers/v2/validationsaddress");
        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;
    }
}
?>
```

Claro